

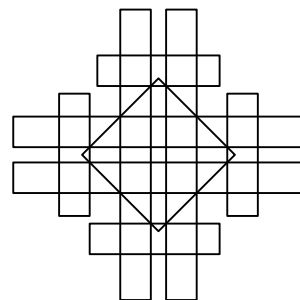
39th International Symposium on Computational Geometry

SoCG 2023, June 12–15, 2023, Dallas, Texas, USA

Edited by

Erin W. Chambers

Joachim Gudmundsson



Editors

Erin W. Chambers 

Saint Louis University, USA
erin.chambers@slu.edu

Joachim Gudmundsson 

University of Sydney, Australia
joachim.gudmundsson@sydney.edu.au

ACM Classification 2012

Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms; Mathematics of computing → Combinatorics; Mathematics of computing → Graph algorithms

ISBN 978-3-95977-273-0

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <https://www.dagstuhl.de/dagpub/978-3-95977-273-0>.

Publication date

June, 2023

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at <https://portal.dnb.de>.

License

This work is licensed under a Creative Commons Attribution 4.0 International license (CC-BY 4.0): <https://creativecommons.org/licenses/by/4.0/legalcode>.



In brief, this license authorizes each and everybody to share (to copy, distribute and transmit) the work under the following conditions, without impairing or restricting the authors' moral rights:

- Attribution: The work must be attributed to its authors.

The copyright is retained by the corresponding authors.

Digital Object Identifier: 10.4230/LIPIcs.SoCG.2023.0

ISBN 978-3-95977-273-0

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

LIPICs – Leibniz International Proceedings in Informatics

LIPICs is a series of high-quality conference proceedings across all fields in informatics. LIPICs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

Editorial Board

- Luca Aceto (*Chair*, Reykjavik University, IS and Gran Sasso Science Institute, IT)
- Christel Baier (TU Dresden, DE)
- Mikolaj Bojanczyk (University of Warsaw, PL)
- Roberto Di Cosmo (Inria and Université de Paris, FR)
- Faith Ellen (University of Toronto, CA)
- Javier Esparza (TU München, DE)
- Daniel Král' (Masaryk University - Brno, CZ)
- Meena Mahajan (Institute of Mathematical Sciences, Chennai, IN)
- Anca Muscholl (University of Bordeaux, FR)
- Chih-Hao Luke Ong (University of Oxford, GB and Nanyang Technological University, SG)
- Phillip Rogaway (University of California, Davis, US)
- Eva Rotenberg (Technical University of Denmark, Lyngby, DK)
- Raimund Seidel (Universität des Saarlandes, Saarbrücken, DE and Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Wadern, DE)

ISSN 1868-8969

<https://www.dagstuhl.de/lipics>

■ Contents

Preface	
<i>Erin W. Chambers and Joachim Gudmundsson</i>	0:xi–0:xii
Conference Organization	
.....	0:xiii–0:xv
Additional Reviewers	
.....	0:xvii–0:xix

Regular Papers

Geometric Embeddability of Complexes Is $\exists\mathbb{R}$ -Complete	
<i>Mikkel Abrahamsen, Linda Kleist, and Tillmann Miltzow</i>	1:1–1:19
Distinguishing Classes of Intersection Graphs of Homothets or Similarities of Two Convex Disks	
<i>Mikkel Abrahamsen and Bartosz Walczak</i>	2:1–2:16
Lower Bounds for Intersection Reporting Among Flat Objects	
<i>Peyman Afshani and Pingan Cheng</i>	3:1–3:16
Computing Instance-Optimal Kernels in Two Dimensions	
<i>Pankaj K. Agarwal and Sariel Har-Peled</i>	4:1–4:15
Line Intersection Searching Amid Unit Balls in 3-Space	
<i>Pankaj K. Agarwal and Esther Ezra</i>	5:1–5:14
Drawings of Complete Multipartite Graphs up to Triangle Flips	
<i>Oswin Aichholzer, Man-Kwun Chiu, Hung P. Hoang, Michael Hoffmann, Jan Kynčl, Yannic Maus, Birgit Vogtenhuber, and Alexandra Weinberger</i>	6:1–6:16
Decomposition of Zero-Dimensional Persistence Modules via Rooted Subsets	
<i>Ángel Javier Alonso and Michael Kerber</i>	7:1–7:16
On Helly Numbers of Exponential Lattices	
<i>Gergely Ambrus, Martin Balko, Nóra Frankl, Attila Jung, and Márton Naszódi</i> ...	8:1–8:16
Optimal Volume-Sensitive Bounds for Polytope Approximation	
<i>Sunil Arya and David M. Mount</i>	9:1–9:16
Coresets for Clustering in Geometric Intersection Graphs	
<i>Sayan Bandyapadhyay, Fedor V. Fomin, and Tanmay Inamdar</i>	10:1–10:16
Minimum-Membership Geometric Set Cover, Revisited	
<i>Sayan Bandyapadhyay, William Lochet, Saket Saurabh, and Jie Xue</i>	11:1–11:14
FPT Constant-Approximations for Capacitated Clustering to Minimize the Sum of Cluster Radii	
<i>Sayan Bandyapadhyay, William Lochet, and Saket Saurabh</i>	12:1–12:14



Multilevel Skeletonization Using Local Separators <i>J. Andreas Bærentzen, Rasmus Emil Christensen, Emil Toftegaard Gæde, and Eva Rotenberg</i>	13:1–13:18
Efficient Computation of Image Persistence <i>Ulrich Bauer and Maximilian Schmahl</i>	14:1–14:14
Efficient Two-Parameter Persistence Computation via Cohomology <i>Ulrich Bauer, Fabian Lenzen, and Michael Lesnick</i>	15:1–15:17
The Complexity of Geodesic Spanners <i>Sarita de Berg, Marc van Kreveld, and Frank Staals</i>	16:1–16:16
An Extension Theorem for Signotopes <i>Helena Bergold, Stefan Felsner, and Manfred Scheucher</i>	17:1–17:14
Extending Orthogonal Planar Graph Drawings Is Fixed-Parameter Tractable <i>Sujoy Bhore, Robert Ganian, Liana Khazaliya, Fabrizio Montecchiani, and Martin Nöllenburg</i>	18:1–18:16
Improved Bounds for Covering Paths and Trees in the Plane <i>Ahmad Biniiaz</i>	19:1–19:15
Sparse Higher Order Čech Filtrations <i>Mickaël Buchet, Bianca B. Dornelas, and Michael Kerber</i>	20:1–20:17
Finding Large Counterexamples by Selectively Exploring the Pachner Graph <i>Benjamin A. Burton and Alexander He</i>	21:1–21:16
Improved Algebraic Degeneracy Testing <i>Jean Cardinal and Micha Sharir</i>	22:1–22:16
Constant-Hop Spanners for More Geometric Intersection Graphs, with Even Smaller Size <i>Timothy M. Chan and Zhengcheng Huang</i>	23:1–23:16
Minimum L_∞ Hausdorff Distance of Point Sets Under Translation: Generalizing Klee’s Measure Problem <i>Timothy M. Chan</i>	24:1–24:13
Meta-Diagrams for 2-Parameter Persistence <i>Nate Clause, Tamal K. Dey, Facundo Mémoli, and Bei Wang</i>	25:1–25:16
Algorithms for Length Spectra of Combinatorial Tori <i>Vincent Delecroix, Matthijs Ebbens, Francis Lazarus, and Ivan Yakovlev</i>	26:1–26:16
Computing a Dirichlet Domain for a Hyperbolic Surface <i>Vincent Despré, Benedikt Kolbe, Hugo Partier, and Monique Teillaud</i>	27:1–27:15
The Parameterized Complexity of Coordinated Motion Planning <i>Eduard Eiben, Robert Ganian, and Iyad Kanj</i>	28:1–28:16
Non-Crossing Hamiltonian Paths and Cycles in Output-Polynomial Time <i>David Eppstein</i>	29:1–29:16
Finding a Maximum Clique in a Disk Graph <i>Jared Espenant, J. Mark Keil, and Debajyoti Mondal</i>	30:1–30:17

Linear Size Universal Point Sets for Classes of Planar Graphs <i>Stefan Felsner, Hendrik Schrezenmaier, Felix Schröder, and Raphael Steiner</i>	31:1–31:16
When Ternary Triangulated Disc Packings Are Densest: Examples, Counter-Examples and Techniques <i>Thomas Fernique and Daria Pchelina</i>	32:1–32:17
Labeled Nearest Neighbor Search and Metric Spanners via Locality Sensitive Orderings <i>Arnold Filtser</i>	33:1–33:18
Polynomial-Time Approximation Schemes for Independent Packing Problems on Fractionally Tree-Independence-Number-Fragile Graphs <i>Esther Galby, Andrea Munaro, and Shizhou Yang</i>	34:1–34:15
Voronoi Diagrams in the Hilbert Metric <i>Auguste H. Gezalyan and David M. Mount</i>	35:1–35:16
Combinatorial Designs Meet Hypercliques: Higher Lower Bounds for Klee’s Measure Problem and Related Problems in Dimensions $d \geq 4$ <i>Egor Gorbachev and Marvin Künnemann</i>	36:1–36:14
A Generalization of the Persistent Laplacian to Simplicial Maps <i>Aziz Burak Gülen, Facundo Mémoli, Zhengchao Wan, and Yusu Wang</i>	37:1–37:17
The Christoffel-Darboux Kernel for Topological Data Analysis <i>Pepijn Roos Hoefgeest and Lucas Slot</i>	38:1–38:20
The Number of Edges in Maximal 2-Planar Graphs <i>Michael Hoffmann and Meghana M. Reddy</i>	39:1–39:15
Worst-Case Deterministic Fully-Dynamic Biconnectivity in Changeable Planar Embeddings <i>Jacob Holm, Ivor van der Hoog, and Eva Rotenberg</i>	40:1–40:18
Disjoint Faces in Drawings of the Complete Graph and Topological Heilbronn Problems <i>Alfredo Hubard and Andrew Suk</i>	41:1–41:15
On the Width of Complicated JSJ Decompositions <i>Kristóf Huszár and Jonathan Spreer</i>	42:1–42:18
Reconfiguration of Colorings in Triangulations of the Sphere <i>Takehiro Ito, Yuni Iwamasa, Yusuke Kobayashi, Shun-ichi Maezawa, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki</i>	43:1–43:16
On the Geometric Thickness of 2-Degenerate Graphs <i>Rahul Jain, Marco Ricci, Jonathan Rollin, and André Schulz</i>	44:1–44:15
The Localized Union-Of-Balls Bifiltration <i>Michael Kerber and Matthias Söls</i>	45:1–45:19
Online and Dynamic Algorithms for Geometric Set Cover and Hitting Set <i>Arindam Khan, Aditya Lonkar, Saladi Rahul, Aditya Subramanian, and Andreas Wiese</i>	46:1–46:17

Sparse Euclidean Spanners with Optimal Diameter: A General and Robust Lower Bound via a Concave Inverse-Ackermann Function <i>Hung Le, Lazar Milenković, and Shay Solomon</i>	47:1–47:17
Shortest Paths in Portalgons <i>Maarten Löffler, Tim Ophelders, Rodrigo I. Silveira, and Frank Staals</i>	48:1–48:16
The Geodesic Edge Center of a Simple Polygon <i>Anna Lubiw and Anurag Murty Naredla</i>	49:1–49:15
A Structural Approach to Tree Decompositions of Knots and Spatial Graphs <i>Corentin Lunel and Arnaud de Mesmay</i>	50:1–50:16
Ephemeral Persistence Features and the Stability of Filtered Chain Complexes <i>Facundo Mémoli and Ling Zhou</i>	51:1–51:18
Abstract Voronoi-Like Graphs: Extending Delaunay’s Theorem and Applications <i>Evanthia Papadopoulou</i>	52:1–52:16
Random Projections for Curves in High Dimensions <i>Ioannis Psarros and Dennis Rohde</i>	53:1–53:15
New Approximation Algorithms for Touring Regions <i>Benjamin Qi and Richard Qi</i>	54:1–54:16
Combinatorial Depth Measures for Hyperplane Arrangements <i>Patrick Schnider and Pablo Soberón</i>	55:1–55:14
FibeRed: Fiberwise Dimensionality Reduction of Topologically Complex Data with Vector Bundles <i>Luis Scoccola and Jose A. Perea</i>	56:1–56:18
Toroidal Coordinates: Decorrelating Circular Coordinates with Lattice Reduction <i>Luis Scoccola, Hitesh Gakhar, Johnathan Bush, Nikolas Schonsheck, Tatum Rask, Ling Zhou, and Jose A. Perea</i>	57:1–57:20
Topological Universality of the Art Gallery Problem <i>Jack Stade and Jamie Tucker-Foltz</i>	58:1–58:13
On Higher Dimensional Point Sets in General Position <i>Andrew Suk and Ji Zeng</i>	59:1–59:13
Slice, Simplify and Stitch: Topology-Preserving Simplification Scheme for Massive Voxel Data <i>Hubert Wagner</i>	60:1–60:16
Maximum Overlap Area of a Convex Polyhedron and a Convex Polygon Under Translation <i>Honglin Zhu and Hyuk Jun Kweon</i>	61:1–61:16

Media Expositions

Godzilla Onions: A Skit and Applet to Explain Euclidean Half-Plane Fractional Cascading <i>Richard Berger, Vincent Ha, David Kratz, Michael Lin, Jeremy Moyer, and Christopher J. Tralie</i>	62:1–62:3
---	-----------

Interactive 2D Periodic Graphs
Alexandra Camero and Ileana Streinu 63:1–63:4

Greedy Permutations and Finite Voronoi Diagrams
Oliver A. Chubet, Paul Macnichol, Parth Parikh, Donald R. Sheehy, and Siddharth S. Sheth 64:1–64:5

The Sum of Squares in Polycubes
Donald R. Sheehy 65:1–65:6

CG Challenges

Constructing Concise Convex Covers via Clique Covers
Mikkel Abrahamsen, William Bille Meyling, and André Nusser 66:1–66:9

Shadoks Approach to Convex Covering
Guilherme D. da Fonseca 67:1–67:9

■ Preface

The 39th International Symposium on Computational Geometry (SoCG 2023) was held at the University of Texas at Dallas, June 12–15, 2023, as part of the Computational Geometry Week (CG Week 2023). The conference received 175 submissions, and after a thorough review process, in which each paper was evaluated by three or more independent reviewers, the program committee accepted 61 papers for presentation. These proceedings contain extended abstracts of the accepted papers, limited to 500 lines (excluding references). If any supporting material does not fit in the line limit, the full paper is available at a public repository and referenced in the corresponding extended abstract.

The Best Paper Award of SoCG 2023 went to the paper “Sparse higher order Čech filtrations” by Bianca Boeira Dornelas, Michael Kerber and Mickaël Buchet; this paper has been invited to submit an extended version to the Journal of the ACM. The Best Student Presentation Award was determined and announced at the symposium, based on ballots cast by the attendees. A selection of papers was invited to submit an extended version to forthcoming special issues of Discrete & Computational Geometry and the Journal of Computational Geometry dedicated to the symposium.

The SoCG Test of Time Awards of this year go to “The complexity of many faces in arrangements of lines and of segments” by Herbert Edelsbrunner, Leonidas J. Guibas, and Micha Sharir which was published in SoCG 1988, and to the Computational Geometry Algorithms Library (CGAL) project.

The scientific program of CG Week 2023 was enriched by two distinguished invited speakers. An invited talk, entitled “Redistricting as a computational geometry problem”, given by Moon Duchin from Tufts University. A second invited talk, entitled “Multi-Agent Path Finding and Its Applications”, was delivered by Sven Koenig from the University of Southern California. We thank the plenary speakers for kindly accepting our invitation.

In addition to the technical papers, there were five submissions to the multimedia exposition. Submissions were reviewed and four of them were accepted for presentation. The extended abstracts that describe these submissions are included in this proceedings volume. The multimedia content can be found at <https://www.computational-geometry.org>.

The 5th Computational Geometry Challenge was part of CG Week 2023. The challenge problem was to cover a polygon with a small number of convex polygons. This year there were 22 teams participating in the challenge, and these proceedings contain contributions by the two top-placed teams describing their winning approaches.

We thank the authors of all submitted works. We are most grateful to the members of the SoCG Program Committee, the Media Exposition Committee, and the CG Challenge Committee for their dedication, expertise, and hard work that ensured the high quality of the works in these proceedings. We are grateful for the assistance provided by the hundreds of reviewers; without their help, it would have been nearly impossible to run the selection process. Finally, we thank Irina Kostitsyna, who kindly accepted to be the Proceedings Chair and did meticulous work.

Many other people contributed to the success of SoCG 2023 and the entire CG Week. We are very grateful to the local organization committee for their work in organizing the event, and to facilitate remote participation. Finally, we thank all the members of the Test of Time Award, Workshop, and Young Researchers Forum Committees, the CG Challenge Advisory Board, and the Computational Geometry Steering Committee.



Erin W. Chambers
SoCG program committee co-chair
Saint Louis University

Joachim Gudmundsson
SoCG program committee co-chair
University of Sydney

Don Sheehy
Media Exposition chair
North Carolina State
University

Sándor Fekete
CG Challenge co-chair
TU Braunschweig

Stefan Schirra
CG Challenge co-chair
University of Magdeburg

■ Conference Organization

SoCG Program Committee

- Anastasios Sidiropoulos (University of Illinois at Chicago, US)
- André Nusser (University of Copenhagen, DK)
- Arijit Ghosh (Indian Statistical Institute, IN)
- Arindam Khan (IISc Bangalore, IN)
- Arnaud de Mesmay (CNRS, Université Gustave Eiffel, FR)
- Balázs Keszegh (Alfréd Rényi Institute of Mathematics, HU)
- Benjamin Burton (University of Queensland, AU)
- Elena Arseneva (Università della Svizzera Italiana, CH)
- Elizabeth Munch (Michigan State University, US)
- Erin Chambers (co-chair, Saint Louis University, US)
- Esther Ezra (Bar Ilan University, IL)
- Facundo Mémoli (Ohio State University, US)
- Günter Rote (Freie Universität Berlin, DE)
- Ileana Streinu (Smith College, US)
- Jack Snoeyink (University of North Carolina, US)
- Jeff Erickson (University of Illinois Urbana-Champaign, US)
- Joachim Gudmundsson (co-chair, University of Sydney, AU)
- Joseph S. B. Mitchell (Stony Brook University, US)
- Ken Clarkson (IBM Almaden Research, US)
- Kevin Verbeek (TU Eindhoven, NL)
- Konrad Swanepoel (London School of Economics, UK)
- Linda Kleist (Technische Universität Braunschweig, DE)
- Maarten Löffler (Utrecht University, NL)
- Mathijs Wintraecken (Institute of Science and Technology Austria, AT and Inria Sophia Antipolis, Université Côte d'Azur, FR)
- Matias Korman (Siemens Electronic Design Automation, US)
- Matya Katz (Ben-Gurion University, IL)
- Oswin Aichholzer (Graz University of Technology, AT)
- Pat Morin (Carleton University, CA)
- Patrizio Angelini (John Cabot University, IT)
- Saket Saurabh (IMSc Chennai, IN)
- Uli Wagner (Institute of Science and Technology, AT)
- Valentin Polishchuk (Linköping University, SE)
- Yoshio Okamoto (The University of Electro-Communications, JP)

SoCG Proceedings Chair

- Irina Kostitsyna (TU Eindhoven, NL)

39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Media Exposition Committee

- Oliver Chubet (North Carolina State University, US)
- Satyan Devadoss (University of San Diego, US)
- Kirk Gardner (Air Force Research Lab, US)
- Gregory Henselman-Petrusek (Pacific Northwest National Lab, US)
- Michael Lesnick (University at Albany, US)
- Don Sheehy (chair, North Carolina State University, US)
- Siddharth Sheth (North Carolina State University, US)

CG Challenge Committee

- Sándor Fekete (TU Braunschweig, DE)
- Phillip Keldenich (TU Braunschweig, DE)
- Dominik Krupke (TU Braunschweig, DE)
- Stefan Schirra (University of Magdeburg, DE)

CG Challenge Advisory Board

- Bill Cook (University of Waterloo, CA)
- Andreas Fabri (GeometryFactory, FR)
- Dan Halperin (Tel Aviv University, IL)
- Michael Kerber (TU Graz, AT)
- Philipp Kindermann (Universität Würzburg, DE)
- Joe Mitchell (Stony Brook University, US)
- Kevin Verbeek (TU Eindhoven, NL)

SoCG Test of Time Award Committee

- Pankaj K. Agarwal (Duke University, US)
- Siu-Wing Cheng (Hong Kong University of Science and Technology, CN)
- Raimund Seidel (Saarland University, DE)

Workshop Committee

- Anne Driemel (University of Bonn, DE)
- Sándor Fekete (TU Braunschweig, DE)
- Matya Katz (Ben-Gurion University, IL)
- Marc van Kreveld (Utrecht University, NL)
- Joe Mitchell (Stony Brook University, US)
- Yusu Wang (University of California, San Diego, US)

Young Researchers Forum Program Committee

- Hee-Kap Ahn (Pohang University of Science and Technology, KR)
- Hugo Akitaya (University of Massachusetts Lowell, US)
- Maike Buchin (chair, Ruhr University Bochum, DE)
- Ellen Gasparovic (Union College, New York, US)
- Arnaud de Mesmay (CNRS, Université Gustave Eiffel, FR)

- Tim Ophelders (Utrecht University and TU Eindhoven, NL)
- Zuzana Patáková (Charles University, CZ)
- Christiane Schmidt (Linköping University, SE)
- Alexander Wolff (University of Würzburg, DE)

Local Organizing Committee

- Emily Fox (University of Texas at Dallas, US)
- Benjamin Raichel (University of Texas at Dallas, US)

Steering Committee (2022–2024)

- Anne Driemel (co-chair, University of Bonn, DE)
- Sándor Fekete (co-chair, TU Braunschweig, DE)
- Matya Katz (Ben-Gurion University, IL)
- Marc van Kreveld (Utrecht University, NL)
- Joe Mitchell (Stony Brook University, US)
- Yusu Wang (University of California San Diego, US)

■ Additional Reviewers

Aaron Becker	Christopher Potvin	Giordano Da Lozzo
Abigail Thompson	Clément Dallard	Gopinath Mishra
Abhiruk Lahiri	Clément Levrard	Grigory Ivanov
Alena Zhukova	Csaba D. Tóth	Gugan Thoppe
Alessandra Tappini	Da Wei Zheng	Guowei Wei
Alex McCleary	Dan Feldman	Haitao Wang
Alexander Neuhaus	Dániel Gerbner	Hamish Carr
Alexandra Weinberger	Daniel Gonçalves	Hanno Lefmann
Amit Patel	David Beers	Henry Adams
Anastasios Stefanou	David Cohen-Steiner	Hiroyuki Miyata
André Lieutier	David Eppstein	Hubert Wagner
André van Renssen	David Loiseaux	Hugo Akitaya
Andreas Wiese	David Mount	Igor Pak
Andreas F. Holmsen	David Saulpic	Ioannis Psarros
Andrzej Lingas	David-Elias Künstle	Irene Parada
Anibal Medina Mardones	Dimitry Ryabogin	Irina Kostitsyna
Aniket Basu Roy	Dominik Krupke	Ishika Ghosh
Ankita Sarkar	Dominique Attali	Ivor van der Hoog
Anna Brötzner	Dömötör Pálvölgyi	József Balogh
Anna Lubiw	Donald Sheehy	Jan-Henrik Haunert
Anna Schenfisch	Edgardo Roldán-Pensado	Jayson Lynch
Anne Driemel	Elena Wang	Jean Cardinal
Anthea Monod	Emerson Escolar	Jean-Daniel Boissonnat
Antoine Commaret	Emily Fox	Jeff Phillips
Anton Ayzenberg	Erfan Hosseini Sereshgi	Jesus De Loera
Anurag Murty Naredla	Éric Colin de Verdière	Jie Gao
Aritra Banik	Eric Sedgwick	Jie Xue
Arnold Filtser	Erik Amézquita	Jisu Kim
Arturo Merino	Erik Krohn	Joachim Orthaber
Astrid Olave	Erik Waingarten	Joakim Blikstad
Bei Wang	Erik Jan van Leeuwen	Johannes Obenaus
Benjamin Raichel	Erin Taylor	Jonathan Conroy
Bettina Speckmann	Eyal Ackerman	Jonathan Spreer
Bingkai Lin	Fabian Klute	Joseph Doolittle
Bogdan Grechuk	Fabian Roll	Joseph Melby
Boris Aronov	Fabrizio Frati	Jules Wulms
Boris Zolotov	Fabrizio Montecchiani	Justin Curry
Chaya Keller	Felix Schröder	Justin Dallant
Chris Schwiigelshohn	Florian Frick	Karl Bringmann
Christian Rieck	Francis Lazarus	Kathrin Hanauer
Christian Scheffer	Frank Staals	Kevin Schewior
Christian Scheideler	Gábor Damásdi	Kolja Knauer
Christian Sohler	Gabriel Nivasch	Konstantinos Tsakalidis
Christoph Thäle	Georg Osang	Kristóf Huszár
Christopher Fillmore	Géza Tóth	Kunal Dutta

39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Ling Zhou
Lingxiao Huang
Ljubomir Perkovic
Louis Theran
Luis Scoccola
Mael Rouxel-Labbé
Maike Buchin
Majid Mirzanezhad
Man-Kwun Chiu
Marc van Kreveld
Marcos Crichigno
Mark de Berg
Mark Iwen
Mark Keil
Mart Hagedoorn
Martin Seybold
Martin Suderland
Márton Naszódi
Mathieu Carrière
Mathieu Mari
Matt Gibson
Matthias Goerner
Maximilian Pfister
Maximilian Stegemeyer
Mayank Goswami
Micha Sharir
Michael Bekos
Michael Hoffmann
Michael Kerber
Michael Lesnick
Michael Robinson
Michael Usher
Michiel Smid
Mikael Vejdemo-Johansson
Mikkel Abrahamsen
Minati De
Minki Kim
Mohammad Ali Abam
Moritz Venzin
Natan Rubin
Nate Clause
Neil Olver
Nello Blaser
Nicolas Bédaride
Nicolò Zava
Nóra Frankl
Omrit Filtser
Ondrej Draganov
Ori Parzanchevski
Owen Rouillé
Pablo Pérez-Lantero
Pankaj Agarwal
Panos Giannopoulos
Parinya Chalermsook
Pasin Manurangsi
Patrick Schnider
Paul Jungeblut
Peyman Afshani
Philipp Kindermann
Phillip Keldenich
Primož Škraba
Prosenjit Bose
Qingsong Wang
Qizheng He
Rachel Roca
Radoslav Fulek
Rafe Wenger
Rahul Gangopadhyay
Rajesh Chitnis
Randal Tuggle
Ranita Biswas
Raphaël Tinarrage
Rodrigo I. Silveira
Saladi Rahul
Salman Parsa
Samir Chowdhury
Sampson Wong
Sandhya Thekkumpadan Puthiyaveedu
Sándor Fekete
Sándor Kisfaludi-Bak
Sang Won Bae
Sarah McGuire
Sarah Percival
Sarah Tymochko
Sariel Har-Peled
Sasanka Roy
Sathish Govindarajan
Saul Schleimeris
Sebastiano Cultrera di Montesano
Sergey Barannikov
Sergey Bereg
Sergio Cabello
Shakhar Smorodinsky
Sharath Raghvendra
Shay Solomon
Shira Zerbib

Siddharth Pritam	Takeshi Tokuyama	Vincent Despré
Simon Weber	Tamal K. Dey	Wai Ming Tai
Simon Zhang	Tesshu Hanaka	Willem Sonke
Siu Wing Cheng	Théo Lacombe	William Lochet
Soeren Terziadis	Therese Biedl	Wolfgang Mulzer
Sourabh Palande	Till Miltzow	Woojin Kim
Stavros Sintos	Tim Ophelders	Xavier Goaoc
Stefan Felsner	Timothy Chan	Xuan Wu
Stephane Durocher	Torsten Mütze	Yasuaki Kobayashi
Steve Oudot	Torsten Ueckerdt	Yossi Bleile
Steve Pizer	Ulrich Bauer	Yusu Wang
Subhas Nandy	Vadim Lebovici	Yusuke Kobayashi
Sudeshna Kolay	Valeri Jean-Pierre	Yuval Peled
Sujoy Bore	Vanessa Robins	Zhengchao Wan
Sunhyuk Lim	Vida Dujmović	Ziga Virk
Sylvester Eriksson-Bique	Vincent Cohen-Addad	

Geometric Embeddability of Complexes Is $\exists\mathbb{R}$ -Complete

Mikkel Abrahamsen  

University of Copenhagen, Denmark

Linda Kleist  

Technische Universität Braunschweig, Germany

Tillmann Miltzow  

Utrecht University, The Netherlands

Abstract

We show that the decision problem of determining whether a given (abstract simplicial) k -complex has a geometric embedding in \mathbb{R}^d is complete for the Existential Theory of the Reals for all $d \geq 3$ and $k \in \{d-1, d\}$. Consequently, the problem is polynomial time equivalent to determining whether a polynomial equation system has a real solution and other important problems from various fields related to packing, Nash equilibria, minimum convex covers, the Art Gallery Problem, continuous constraint satisfaction problems, and training neural networks. Moreover, this implies NP-hardness and constitutes the first hardness result for the algorithmic problem of geometric embedding (abstract simplicial) complexes. This complements recent breakthroughs for the computational complexity of piece-wise linear embeddability.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases simplicial complex, geometric embedding, linear embedding, hypergraph, recognition, existential theory of the reals

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.1

Related Version Full details can be found in our ArXiv paper: <https://arxiv.org/abs/2108.02585>

Funding Mikkel Abrahamsen: Supported by Starting Grant 1054-00032B from the Independent Research Fund Denmark under the Sapere Aude research career programme and part of Basic Algorithms Research Copenhagen (BARC), supported by the VILLUM Foundation grant 16582.

Linda Kleist: Partially supported by a postdoc fellowship of the German Academic Exchange Service (DAAD).

Tillmann Miltzow: Generously supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 016.Veni.192.250.

Acknowledgements We thank Arkadiy Skopenkov for his kind and swift help with acquiring literature and Martin Tancer for pointing out a mistake in a previous version of this manuscript.

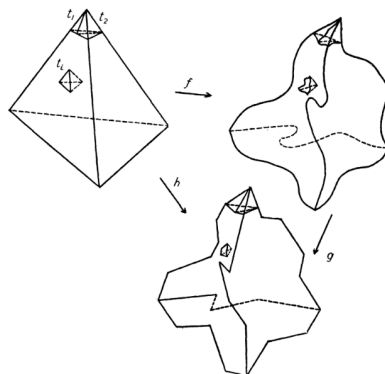


Figure 1 Illustration of different embeddings of a complex; figure taken from Bing [8, Annals of Mathematics 1959].



© Mikkel Abrahamsen, Linda Kleist, and Tillmann Miltzow;
licensed under Creative Commons License CC-BY 4.0

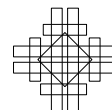
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 1; pp. 1:1–1:19

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

For now almost 100 years, much attention has been devoted to studying embeddings of complexes [8, 21, 30, 31, 42, 54, 66, 67]. Typical types of embeddings include geometric (also referred to as linear), piecewise linear (PL), and topological embeddings, see also Figure 1. For formal definitions, we refer to Section 1.2; here we give an illustrative example. Embeddings of a 1-complex in the plane correspond to crossing-free drawings of a graph in the plane. In a topological embedding, each edge is represented by a Jordan arc, in a PL embedding it is a concatenation of a finite number of segments, and in a geometric embedding each edge is represented by a segment.

We are interested in the problem of deciding whether a given k -complex has a linear/-piecewise linear/topological embedding in \mathbb{R}^d . Several necessary and sufficient conditions are easy to identify and have been known for many decades. For instance, a k -simplex requires $k + 1$ points in general position in \mathbb{R}^d and, thus, $k \leq d$ is an obvious necessary condition. Moreover, it is straight-forward to verify that every set of n points in \mathbb{R}^3 in general position allows for a geometric embedding of any 1-complex on n vertices, i.e., the points are the vertices of a straight-line drawing of a (complete) graph. Indeed, this fact generalizes to higher dimensions: every k -complex embeds (even linearly) in \mathbb{R}^{2k+1} [42]. Van Kampen and Flores [25, 57, 66] showed that this bound is tight by providing k -complexes that do not topologically embed into \mathbb{R}^{2k} . For some time, it was believed that the existence of a topological embedding also implies the existence of a geometric embedding, e.g., Grünbaum conjectured that if a k -complex topologically embeds in \mathbb{R}^{2k} , then it also geometrically embeds in \mathbb{R}^{2k} [30]. In \mathbb{R}^2 , this is in fact true: For 1-complexes this is commonly known as Fáry's theorem [35] but it also follows from Steinitz' earlier theorem [62]; for 2-complexes one needs a few additional arguments [32]. In higher dimensions, however, the conjecture was disproven. In particular, for every $k, d \geq 2$ with $k + 1 \leq d \leq 2k$, there exist k -complexes that have a PL embedding in \mathbb{R}^d , but no geometric embedding in \mathbb{R}^d [9, 10, 11]. In contrast, PL and topological embeddability coincides in many cases, e.g., if $d \leq 3$ [8, 48] or $d - k \geq 3$ [12]. Very recently, Frick, Hu, Scheel, and Simon [27] characterized when a complex on $d + 3$ vertices embeds into the d -sphere, namely, if and only if its non-faces do not form an intersecting family. Additionally, they showed that if a complex on $d + 3$ vertices embeds topologically into \mathbb{R}^d then it also embeds linearly into \mathbb{R}^d . There are many further necessary and sufficient conditions known for geometric embeddings [6, 46, 47, 57, 63, 64] and PL or/and topological embeddings [20, 26, 49, 54, 65, 61].

In recent years, the **algorithmic complexity** of deciding whether or not a given complex is embeddable gained attention. In the absence of a complete characterization, an efficient algorithm is the best tool to decide embeddability. For instance, deciding whether a 1-complex embeds in the plane corresponds to testing graph planarity and is thus polynomial time decidable [33]. Similarly, Gross and Rosen [29] present a linear time planarity algorithm for 2-complexes in the plane. On the other hand, PL embeddability is sometimes even algorithmically undecidable. To give a concrete example, let $\text{EMBED}_{k \rightarrow d}$ denote the algorithmic problem of determining whether a given k -complex has a PL embedding in \mathbb{R}^d . Because $\text{EMBED}_{4 \rightarrow 5}$ has been shown to be algorithmically undecidable [40], there is no algorithm to decide the problem (never mind an efficient one). This provides strong evidence that PL embeddability for these parameters does not allow a reasonable characterization.

More recently, there have been several breakthroughs concerning the **PL embeddability**. For an overview of the state of the art, consider Table 1. In dimensions $d \geq 4$, the decision problem $\text{EMBED}_{k \rightarrow d}$ is polynomial-time decidable for $k < \frac{2}{3} \cdot (d - 1)$ [16, 13, 15, 36] and

■ **Table 1** Overview of the complexity of $\text{EMBED}_{k \rightarrow d}$.

$d \backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	P	P	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✗	P	D	?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✗	✗	D	?	?	P	✓	✓	✓	✓	✓	✓	✓	✓
4	✗	✗	✗	?	U	?	?	P	✓	✓	✓	✓	✓	✓
5	✗	✗	✗	✗	U	U	?	?	P	P	✓	✓	✓	✓
6	✗	✗	✗	✗	✗	U	U	?	?	?	P	P	✓	✓

- ✓ always yes
- ✗ always no
- P polynomial-time
- D decidable
- U undecidable
- NP-hard

NP-hard for all remaining non-trivial cases [40], i.e., for all k with $2/3 \cdot (d - 1) \leq k \leq 2d$. For $d \geq 5$ and $k \in \{d - 1, d\}$, $\text{EMBED}_{k \rightarrow d}$ is even known to be undecidable [40]. For all other NP-hard cases and $d \geq 4$ decidability is unknown; we note that the proof for undecidability in the case of codimension > 1 in [24] has an error [58]. For the case $d = 3$, Matoušek, Sedgwick, Tancer, and Wagner proved decidability of $\text{EMBED}_{2 \rightarrow 3}$ and $\text{EMBED}_{3 \rightarrow 3}$ [39] and de Mesmay, Rieck, Sedgwick, and Tancer proved NP-hardness [43].

Building upon [40], Skopenkov and Tancer [60] proved NP-hardness for a relaxed notion called *almost (PL/topological) embeddability* where it is only required that disjoint sets are mapped to disjoint objects, i.e., two edges incident to a common vertex may cross in an interior point. More precisely, they showed that recognizing almost embeddability of k -complexes in \mathbb{R}^d is NP-hard for all $d, k \geq 2$ with $d \pmod 3 = 1$ and $2/3 \cdot (d - 1) \leq k \leq d$.

The analogous questions for **geometric embeddings** are wide open. Let $\text{GEM}_{k \rightarrow d}$ denote the algorithmic problem of determining whether a given k -complex has a geometric embedding in \mathbb{R}^d . In contrast to PL embeddability, however, it is easy to see that $\text{GEM}_{k \rightarrow d}$ is decidable for all k, d , since every instance can be expressed as a sentence in the first order theory of the reals, which is decidable; for more details see Section 1.1.

The question of whether $\text{GEM}_{k \rightarrow d}$ is complete for $\exists\mathbb{R}$ is a well-known open problem, mentioned for example by Cardinal [18, Section 4].

Our Results. In this work, we present the first results concerning open problem for any non-trivial entry with $d \geq 3$. More precisely, we establish the exact computational complexity of $\text{GEM}_{k \rightarrow d}$ for all values $d \geq 3$ and $k \in \{d - 1, d\}$. This includes a complete understanding of the most intriguing entries with $d = 3$.

► **Theorem 1.** *For every $d \geq 3$ and each $k \in \{d - 1, d\}$, the decision problem $\text{GEM}_{k \rightarrow d}$ is $\exists\mathbb{R}$ -complete. Moreover, the statement remains true even if a PL embedding is given.*

Table 2 summarizes the current knowledge on the computational complexity of $\text{GEM}_{k \rightarrow d}$. Our proof implies that distinguishing between k -complexes with PL and geometric embeddings in \mathbb{R}^d is complete for $\exists\mathbb{R}$. Because $\text{NP} \subseteq \exists\mathbb{R}$, our result yields NP-hardness for $d \geq 3$ and each $k \in \{d - 1, d\}$. This confirms the conjecture by Skopenkov that $\text{GEM}_{k \rightarrow d}$ is NP-hard for all k, d with $\max\{3, k\} \leq d \leq 3/2 \cdot k + 1$ for the corresponding values of k and d [59, Conjecture 3.2.2]. Moreover, if $\text{NP} \neq \exists\mathbb{R}$, the problem $\text{GEM}_{k \rightarrow d}$ cannot be tackled with well developed tools for NP-complete problems such as SAT and ILP solvers. For more details, we refer to Section 1.1.

■ **Table 2** Overview of the computational complexity of $\text{GEM}_{k \rightarrow d}$.

$d \backslash k$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	P	P	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
2	✗	P	$\exists\mathbb{R}c$?	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
3	✗	✗	$\exists\mathbb{R}c$	$\exists\mathbb{R}c$?	?	✓	✓	✓	✓	✓	✓	✓	✓
4	✗	✗	✗	$\exists\mathbb{R}c$	$\exists\mathbb{R}c$?	?	?	✓	✓	✓	✓	✓	✓
5	✗	✗	✗	✗	$\exists\mathbb{R}c$	$\exists\mathbb{R}c$?	?	?	?	✓	✓	✓	✓
6	✗	✗	✗	✗	✗	$\exists\mathbb{R}c$	$\exists\mathbb{R}c$?	?	?	?	?	✓	✓

✓ always yes
✗ always no
P polynomial-time
 $\exists\mathbb{R}c$ $\exists\mathbb{R}$ -complete

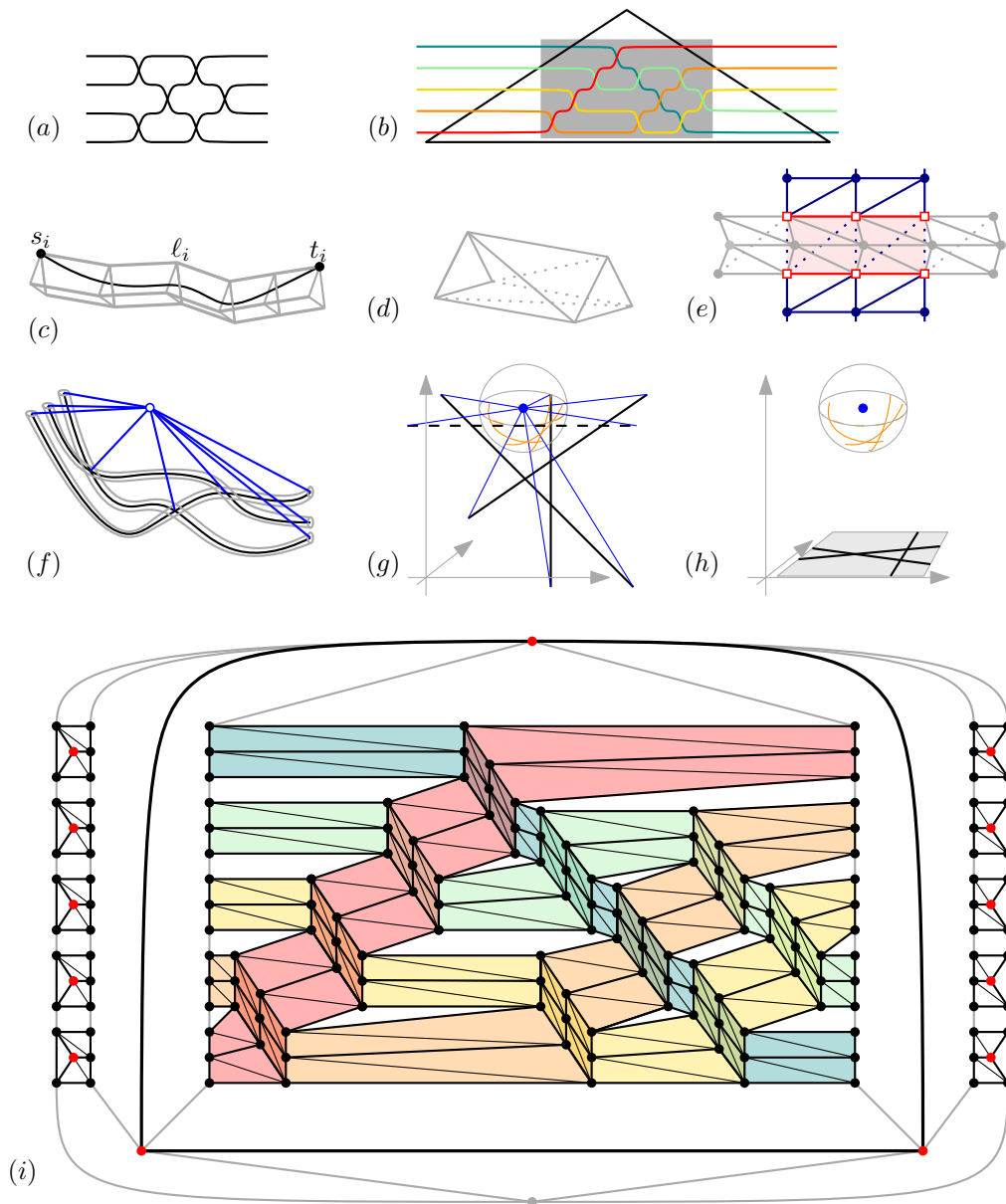
The closely related question of polyhedral complexes (generalizing simplicial complexes because each simplex is a basic polyhedron), posed in the Handbook of Discrete and Computational Geometry, reads as follows: When is a given finite poset isomorphic to the face poset of some polyhedral complex in a given space \mathbb{R}^d ? [53, Problem 20.1.1]. The recognition of polyhedral complexes (with triangles and quadrangles) in \mathbb{R}^3 has been claimed to be $\exists\mathbb{R}$ -complete [18, Theorem 5]. Focussing on convex polytopes, Richter-Gebert proved that recognizing convex polytopes in \mathbb{R}^4 is $\exists\mathbb{R}$ -complete [50, 51]. Our result settles the computational aspects of the question, even for the special case of simplicial complexes.

A geometric embedding of a complex can also be viewed as a *simplicial representation* of a hypergraph, i.e., a representation in which every hyperedge is represented by a simplex. Of particular interest is the case of uniform hypergraphs where all hyperedges have the same number of elements. Thus, in the language of hypergraphs, our result reads as follows.

► **Corollary 2.** *For all $d \geq 3$ and every $k \in \{d-1, d\}$, deciding whether a $(k+1)$ -uniform hypergraph has a simplicial representation in \mathbb{R}^d is $\exists\mathbb{R}$ -complete.*

Outline and techniques. Our proof of Theorem 1 consists of three steps: Establishing $\exists\mathbb{R}$ -membership, showing $\exists\mathbb{R}$ -hardness in \mathbb{R}^3 , i.e., of $\text{GEM}_{2 \rightarrow 3}$ and $\text{GEM}_{3 \rightarrow 3}$, and reducing $\text{GEM}_{k \rightarrow d}$ to $\text{GEM}_{k+1 \rightarrow d+1}$. The core of the proof lies in establishing hardness of $\text{GEM}_{2 \rightarrow 3}$.

The main idea to prove hardness of $\text{GEM}_{2 \rightarrow 3}$ is to reduce from the problem STRETCHABILITY . In STRETCHABILITY , we are given an arrangement of pseudolines (curves) in the plane and we are asked to decide whether there exists a set of straight lines that has the same combinatorial pattern as the pseudoline arrangement, see Figure 2(a) for an illustration and Section 1.2 for a formal definition. Given a pseudoline arrangement L , we construct a 2-complex C which has a geometric embedding in \mathbb{R}^3 if and only if L is stretchable. On a high level, our construction of C goes along the following lines: We add a helper triangle that contains all intersections of the pseudolines, see Figure 2(b). We place each pseudoline in \mathbb{R}^3 and replace it by a *special* edge of the complex C ; these will not be part of any triangle of C . We surround the special edges by so called *tunnels*, which are tubes formed by triangular sections, see Figure 2(c) and (d). One side of the tunnel defines its *bottom*, while the other two span its *roof*. For each crossing in L , we glue the corresponding tunnel sections together, see Figure 2(e). At last, we insert an apex u high above that is connected to all visible tunnel parts, see Figure 2(f) and we insert additional objects in order to ensure that the neighborhood of u is an essentially 3-connected graph, Figure 2(i). The objects incident to the apex will also ensure that the special edges actually lie inside the tunnel.



■ **Figure 2** (a) We start with a pseudoline arrangement L . (b) We add three segments forming a triangle that contains all intersections of L . (c) Each pseudoline is represented by a special edge that is surrounded by a tunnel. (d) Each tunnel consists of tunnel sections. (e) For the crossings of the special edges, we identify parts of the tunnels. (f) We add an apex u and insert triangles to the visible parts of the construction; we enhance the neighborhood of the apex to an essentially 3-connected graph depicted in (i). (g) In the correctness proof, we use a small sphere around the apex and the projection of each special edge onto the sphere. (h) We argue that the combinatorics of the projected special edges on the sphere are equivalent to L and then project the special edges onto a plane. This will yield a stretched arrangement. (i) The neighborhood graph of the apex u .

It is relatively straightforward to verify that if L is stretchable, then the complex C embeds geometrically into \mathbb{R}^3 . The other direction requires more care and work: We show that a geometric embedding of C induces a line arrangement with the same combinatorics as L . The idea of the proof is to consider a small sphere around the apex u and to project its neighborhood and the special edges onto the sphere, see Figure 2(g). Because the neighborhood graph of u is essentially 3-connected by construction, all its crossing-free drawings on the sphere are equivalent. This is a crucial property to show that each special edge lies in the projection of its tunnel roof (when restricting the attention to an interesting part within the helper triangle). We remark that our proof does not show this explicitly. Instead, we establish some even stronger properties. As a consequence, the projection of the tunnels have the intended combinatorics and thus also the special edges which represent the pseudolines. At last, we project the arcs from the sphere onto a plane, see Figure 2(h). In this way, we obtain a line arrangement with the same combinatorics as L .

In order to show hardness of $\text{GEM}_{3 \rightarrow 3}$, we use a similar construction, in which we “fatten” each triangle to a tetrahedron, by adding extra vertices.

We finally present a dimension reduction, i.e., we reduce $\text{GEM}_{k \rightarrow d}$ to $\text{GEM}_{k+1 \rightarrow d+1}$. Given a k -complex C , we create a $(k+1)$ -complex C^+ that contains C and has two additional vertices a and b . Moreover, for each subset e of C , C^+ has the additional subsets $e \cup \{a\}$ and $e \cup \{b\}$. We prove that C geometrically embeds in \mathbb{R}^d if and only if C^+ geometrically embeds in \mathbb{R}^{d+1} . In this way, we show that distinguishing PL embeddable and geometrically embeddable complexes is $\exists\mathbb{R}$ -complete.

1.1 Existential Theory of the Reals

The class of the existential theory of the reals $\exists\mathbb{R}$ (pronounced as is a complexity class which has gained a lot of interest in recent years, specifically in the computational geometry community. To define this class, we first consider the algorithmic problem *Existential Theory of the Reals* (*ETR*). An instance of this problem consists of a sentence of the form

$$\exists x_1, \dots, x_n \in \mathbb{R} : \Phi(x_1, \dots, x_n),$$

where Φ is a well-formed quantifier-free formula in the variables and the alphabet $\{0, 1, +, \cdot, \geq, >, \wedge, \vee, \neg\}$, and the goal is to check whether this sentence is true. As an example of an ETR-instance, consider $\exists x, y \in \mathbb{R} : \Phi(x, y) = (x \cdot y^2 + x \geq 0) \wedge \neg(y < x)$, for which the goal is to determine whether there exist real numbers x and y satisfying the formula $\Phi(x, y)$.

The *complexity class* $\exists\mathbb{R}$ is the family of all problems that admit a polynomial-time many-one reduction to ETR. It is known that $\text{NP} \subseteq \exists\mathbb{R} \subseteq \text{PSPACE}$. The first inclusion follows from the definition of $\exists\mathbb{R}$. Showing the second inclusion was first established by Canny in his seminal paper [17]. The complexity class $\exists\mathbb{R}$ gains its significance because a number of well-studied problems from different areas of theoretical computer science have been shown to be complete for this class.

Famous examples from discrete geometry are the recognition of geometric structures, such as unit disk graphs [41], segment intersection graphs [38], STRETCHABILITY [45, 56], and order type realizability [38]. Other $\exists\mathbb{R}$ -complete problems are related to graph drawing [37], Nash-Equilibria [7, 28], geometric packing [5], the art gallery problem [3], non-negative matrix factorization [55], polytopes [22, 51], geometric linkage constructions [1], training neural networks [4], visibility graphs [19], continuous constraint satisfaction problems [44], and convex covers [2]. The fascination for the complexity class stems not merely from the number of $\exists\mathbb{R}$ -complete problems but from the large scope of seemingly unrelated $\exists\mathbb{R}$ -complete problems. We refer the reader to the lecture notes by Matoušek [38] and surveys by Schaefer [52] and Cardinal [18] for more information on the complexity class $\exists\mathbb{R}$.

1.2 Definitions

Simplex. A k -simplex σ is a k -dimensional polytope which is the convex hull of its $k + 1$ vertices V , which are not contained in the same $(k - 1)$ -dimensional hyperplane. Hence, a 0-simplex corresponds to a point, a 1-simplex to a segment, and a 2-simplex to a triangle etc. The convex hull of any nonempty proper subset of V is called a *face* of σ . A *simplicial complex* K is a set of simplices satisfying the following two conditions: (i) Every face of a simplex from K is also in K . (ii) For any two simplices $\sigma_1, \sigma_2 \in K$ with a non-empty intersection, the intersection $\sigma_1 \cap \sigma_2$ is a face of both simplices σ_1 and σ_2 . The purely combinatorial counterpart to a simplicial complex is an abstract simplicial complex, which we refer to simply as a *complex*.

Complex. A *complex* $C = (V, E)$ is a finite set V together with a collection of subsets $E \subseteq 2^V$ which is closed under taking subsets, i.e., $e \in E$ and $e' \subseteq e$ imply that $e' \in E$. A k -*complex* is a complex where the largest subset contains exactly $k + 1$ elements. We call a complex *pure* if all (inclusion-wise) maximal elements in E have the same cardinality.

For any vertex $v \in V$ in a k -complex $C = (V, E)$, the neighbourhood of v gives rise to a lower dimensional complex $C_v := (V', E')$, where $E' := \{e \setminus \{v\} \mid v \in e \in E\}$ and $V' := N(v) = \bigcup_{e \in E'} e$ are the *neighbors* of v . Complexes are in close relation to Hypergraphs.

Hypergraphs. Hypergraphs generalize graphs by allowing edges to contain any number of vertices. Formally, a *hypergraph* H is a pair $H = (V, E)$ where V is a set of vertices, and E is a set of non-empty subsets of V called *hyperedges* (or edges). A k -uniform hypergraph is a hypergraph such that all its hyperedges contain exactly k elements. Note that the maximal sets of a pure k -complex yield a $(k + 1)$ -uniform hypergraph and vice versa. Hence, $(k + 1)$ -uniform hypergraphs and pure k -complexes are in a straightforward one-to-one correspondence. A *simplicial representation* of a $(k + 1)$ -uniform hypergraph is a geometric embedding of the corresponding complex.

Geometric embeddings. A *geometric embedding* of a complex $C = (V, E)$ in \mathbb{R}^d is a function $\varphi: V \rightarrow \mathbb{R}^d$ fulfilling the following two properties: (i) for every $e \in E$, $\overline{\varphi}(e) := \text{conv}(\{\varphi(v) : v \in e\})$ is a simplex of dimension $|e| - 1$ and (ii) for every pair $e, e' \in E$, it holds that

$$\overline{\varphi}(e) \cap \overline{\varphi}(e') = \overline{\varphi}(e \cap e').$$

Note that if φ is a geometric embedding, then $\{\overline{\varphi}(e) : e \in E\}$ is a simplicial complex. The problem $\text{GEM}_{k \rightarrow d}$ asks whether a given k -complex has a geometric embedding in \mathbb{R}^d .

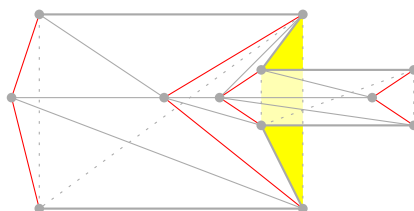
Topological and PL embeddings. Consider a complex $C = (V, E)$. In contrast to geometric embeddings, for PL or topological embeddings it is not sufficient to describe the mapping of the vertices V . Choose d' so large that C admits a geometric embedding $\varphi' : V \rightarrow \mathbb{R}^{d'}$, and define $S = \bigcup_{e \in E} \overline{\varphi'}(e)$. We then say that an injective and continuous function $\varphi : S \rightarrow \mathbb{R}^d$ is a *topological embedding* of C in \mathbb{R}^d . If furthermore for each $e \in E$, the image $\varphi(\overline{\varphi'}(e))$ is a finite union of connected subsets of $(|e| - 1)$ -dimensional hyperplanes, then φ is a *piecewise linear (PL) embedding*. The problem $\text{EMBED}_{k \rightarrow d}$ asks whether a given k -complex has a PL embedding in \mathbb{R}^d .

Graph Drawings. A graph is a 1-complex. A graph is *planar* if there exists a crossing-free drawing in the plane, i.e., a (topological) embedding in \mathbb{R}^2 . As mentioned above, a graph has a topological embedding in \mathbb{R}^2 if and only if it has a geometric embedding in \mathbb{R}^2 . A *plane* graph is a planar graph together with a *rotation system*, i.e., a cyclic ordering of the incident edges around each vertex that comes from a crossing-free drawing. By means of stereographic projection, any graph that has a crossing-free drawing in the plane also has a crossing-free drawing on the sphere and vice versa. Two crossing-free drawings of a graph (in the plane or on the sphere) are *equivalent* if they can be transformed into one another by a homeomorphism (of the plane or the sphere); note that the homeomorphism could be orientation reversing. In particular, two equivalent drawings have the same rotation system; two equivalent drawings in the plane additionally have the same outer face. When talking about an arbitrary drawing D of a plane graph G , we mean a crossing-free drawing with the same rotation system.

Stretchability. A *pseudoline arrangement* is a family of curves that apart from “straightness” share similar properties with a line arrangement. More formally, a (*Euclidean*) *pseudoline arrangement* is a set of labeled x -monotone curves in the Euclidean plane such that any two meet in exactly one point. A curve in \mathbb{R}^2 is *x -monotone* if it is the image of a continuous function $f: \mathbb{R} \rightarrow \mathbb{R}$. In fact, each pseudoline arrangement can be encoded by a *wiring diagram*; see also Figure 4. A pseudoline arrangement is *stretchable* if it is combinatorially equivalent to an arrangement of straight-lines, i.e., if the arrangements can be transformed into one another by a homeomorphism of the plane. STRETCHABILITY denotes the algorithmic problem of deciding whether a given pseudoline arrangement is stretchable. In a seminal paper, Shor [56] proved that STRETCHABILITY is NP-hard. Shor points out that Mněv’s proof implies that stretchability is complete for the existential theory of the reals. For a stream-line exposition of this result see the expository paper by Matoušek [38].

1.3 Pitfalls

While the general proof ideas are fairly straightforward, our arguments in Section 2 may at first glance appear a bit tedious. In the following, we highlight one of the appearing challenges. It is easy to see that each special edge lies inside its tunnel in any geometric embedding. It follows that the projection of the special edge lies also inside the projection of the tunnel on the sphere centered at the apex. Furthermore, we know that the roofs of the tunnels are seen by the apex. One may be tempted to (directly) conclude that the projection of the special edge is thus also contained in the projection of the roof; the underlying thought being that the projection of the tunnel bottom lies below the tunnel roof in the geometric representation and thus the projection of the tunnel bottom is contained in the projection of the tunnel roof. Yet, the latter is not true in general, as can be seen in Figure 3. In the



■ **Figure 3** From the perspective of u , the tunnel bottom is not always hidden below the tunnel roof: From the three sections displayed, the bottom (yellow) of the middle one is partially visible.

figure, the tunnel bottom is not covered by the roof. We (implicitly) show that the projection of the special edge lies inside the projection of the roof by establishing some even stronger topological and geometric properties.

2 The Proof

In this section, we prove Theorem 1. Our proof consists of the following three parts.

- a) Establishing $\exists\mathbb{R}$ -membership (Section 2.1: Lemma 3).
 - b) Showing $\exists\mathbb{R}$ -hardness in \mathbb{R}^3 , i.e., of $\text{GEM}_{2 \rightarrow 3}$ and $\text{GEM}_{3 \rightarrow 3}$ (Section 2.2: Theorem 4 and Corollary 9).
 - c) Reducing $\text{GEM}_{k \rightarrow d}$ to $\text{GEM}_{k+1 \rightarrow d+1}$ (Section 2.3: Lemma 10).
- Together Lemmas 3 and 10, Theorem 4 and Corollary 9 prove Theorem 1.

2.1 Membership

In this subsection, we show $\exists\mathbb{R}$ -membership of $\text{GEM}_{k \rightarrow d}$. Note that this is essentially folklore [14]. We present a proof for the sake of completeness.

► **Lemma 3.** *For all $k, d \in \mathbb{N}$, the decision problem $\text{GEM}_{k \rightarrow d}$ is contained in $\exists\mathbb{R}$.*

Proof. In order to show membership in $\exists\mathbb{R}$, we use the following characterization by Erickson, Hoog and Miltzow [23]: A problem P lies in $\exists\mathbb{R}$ if and only if there exists a verification algorithm A for P that runs in polynomial time on the real RAM, which we refer to as a *real verification algorithm*. In particular, for every yes-instance I of P there exists a polynomial sized witness w such that $A(I, w)$ returns yes, and for every no-instance I of P and any witness w , $A(I, w)$ returns no. In contrast to the definition of the complexity class NP, we also allow witnesses that consist of real numbers. Consequently, we execute A on the real RAM as well.

It remains to present a real verification algorithm for $\text{GEM}_{k \rightarrow d}$. While the witness describes the coordinates of the vertices, the algorithm checks for intersections between any two simplices. Note that each simplex is a convex set and the intersection of convex sets is a convex set as well. For any simplex S with n vertices, we can efficiently determine n linear inequalities and at most one linear equality that together describe S : the inequalities may describe the n facets and the equality describes the subspace in case S is not d -dimensional. Then checking for intersections can be reduced to a linear program, which is polynomial time solvable in any fixed dimension. This finishes the description of the real verification algorithm. ◀

We note that one does not need to resort to the characterization of $\exists\mathbb{R}$ with verifiers as in [23]. It is possible to directly construct a polynomial system of polynomial size (in fixed dimension) in the coordinates of the vertices of the given complex in order to encode its geometric realizability. It may appear to be overly complicated to use the tools from [23], if you do not know this tool. However, if you know this tool it appears strange not to use it.

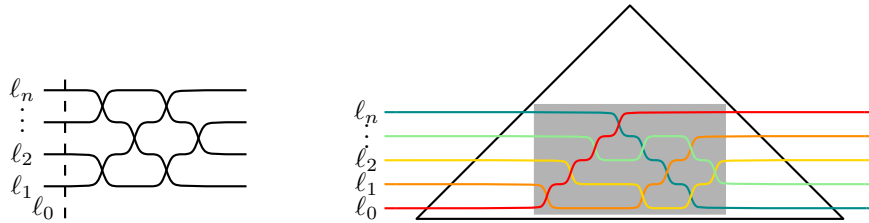
2.2 Hardness in three dimensions

This section is dedicated to proving Theorem 1 for $d = 3$ and $k \in \{2, 3\}$. The crucial part lies in the case $k = 2$.

► **Theorem 4.** *The decision problem $GEM_{2 \rightarrow 3}$ is $\exists\mathbb{R}$ -hard.*

Proof. We reduce from the $\exists\mathbb{R}$ -hard problem STRETCHABILITY, as described in Section 1.2. In particular, for each pseudoline arrangement L , we construct a 2-dimensional complex C in time polynomial in L such that C geometrically embeds in \mathbb{R}^3 if and only if L is stretchable.

Let L be an arrangement of n pseudolines in the plane. Every pseudoline arrangement has a representation as a wiring diagram in which each pseudoline is given by a monotone curve consisting of $2n - 1$ sections. For an illustration consider Figure 4; each section could be represented by a segment, however for a visual appealing display, the bend points are rounded. We add a pseudoline ℓ_0 that intersects all pseudolines in the beginning, see Figure 4, and call the resulting pseudoline arrangement L^* . Note that L^* is stretchable if and only if L is stretchable. For later reference, we endow a natural orientation upon each pseudoline from left to right. In the following, we construct a 2-complex $C = (V, E)$ that allows for a geometric realization if and only if L^* (and thus L) is stretchable. In order to define C , we add a helper triangle Δ (consisting of three segments!) to our arrangement that intersects the pseudolines of L^* as illustrated in Figure 4. In particular, the helper triangle contains all intersection points of L^* .

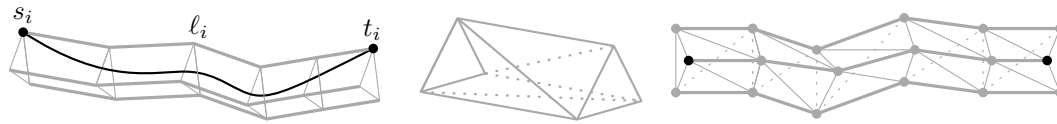


■ **Figure 4** Adding an extra pseudoline ℓ_0 and the helper triangle Δ to the construction. (left) A pseudoline arrangement L^* . (right) The crossing diagram contains an additional helper triangle.

Construction of the 2-complex. In order to define C , we associate an almost geometric embedding of already defined parts along the way; where only a set of special edges is represented in a PL fashion, all other elements are already geometrically embedded. We will refer to the subsets in C as vertices, edges, and triangles depending on whether they contain one, two or three elements. The construction has five steps.

In the first step, we place the pseudolines and the helper triangle Δ in 3-space. Each pseudoline ℓ_i lies in the plane $z = i$ such that an observer high above (at infinity) sees the wiring diagram. Similarly, we place the segments of the helper triangle Δ in 3-space such that it lies in the plane $z = n + 1$. Note that no two pseudolines intersect. Therefore, we can surround each lifted pseudoline by a triangulated sphere which we call a *tunnel*; see also Figure 5. The tunnel T_i^+ of ℓ_i is formed by $2n + 3 + i$ sections; later, we will be particularly interested in a part of a tunnel, denoted by T_i , in which the first two and last two sections are removed. Each section consists of six triangles forming a triangulated triangular prism as illustrated in Figure 5. We close the tunnel with triangles at the ends and think of the *bottom* side of the prism to lie in the plane $z = i - 1/2$ (for now). The remaining part of the tunnel, i.e., the tunnel without its bottom, constitutes the *roof*, see Figure 5. The roof

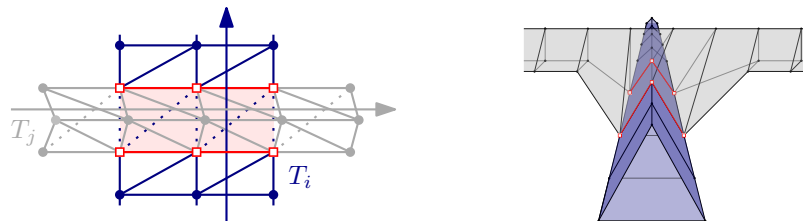
contains three disjoint paths on $2n + 4 + i$ vertices. The edges and vertices on the boundary of both the bottom and the roof form the *left* and *right roof path*; the edges of the closing triangles on either end do not belong to either path. The remaining vertices induce the *central roof path*. The three roof paths are thickened in Figure 5.



■ **Figure 5** First step in the construction of the complex C – tunnel construction. (left) A tunnel viewed from side. (middle) A section of a tunnel. (right) A tunnel viewed from above.

Note that we do not add a tunnel for the helper triangle. We distribute the sections along T_i^+ to edges and crossings of the crossing diagram as follows: Generally, we associate one section per edge and one section per crossing of two pseudolines. Moreover, we associate one extra section of T_j^+ to a crossing of ℓ_i and ℓ_j whenever $i < j$. In order to represent the pseudoline ℓ_i , we insert a *special edge* e_i between the two top vertices on either end of the tunnel; for later reference, we denote the start vertex by s_i and the end vertex by t_i . In the associated almost geometric embedding, e_i is represented inside the tunnel by a concatenation of segments, one for each tunnel section. We aim for the fact that the special edge e_i lies inside the tunnel in every geometric embedding (if one exists).

In the second step, we identify parts of the tunnels. To this end, consider the tunnel sections assigned to a crossing of a pseudoline ℓ_i with ℓ_j , $i < j$. Recall that we assigned one section of T_i^+ and two sections of T_j^+ to the crossing. We identify the four triangles in the bottom of the two sections of T_j^+ with the four triangles in the roof of one section of T_i^+ as indicated in Figure 6. Note that we hereby identify six times two vertices, four of which belong to a left or right roof path of both tunnels, T_i^+ and T_j^+ .

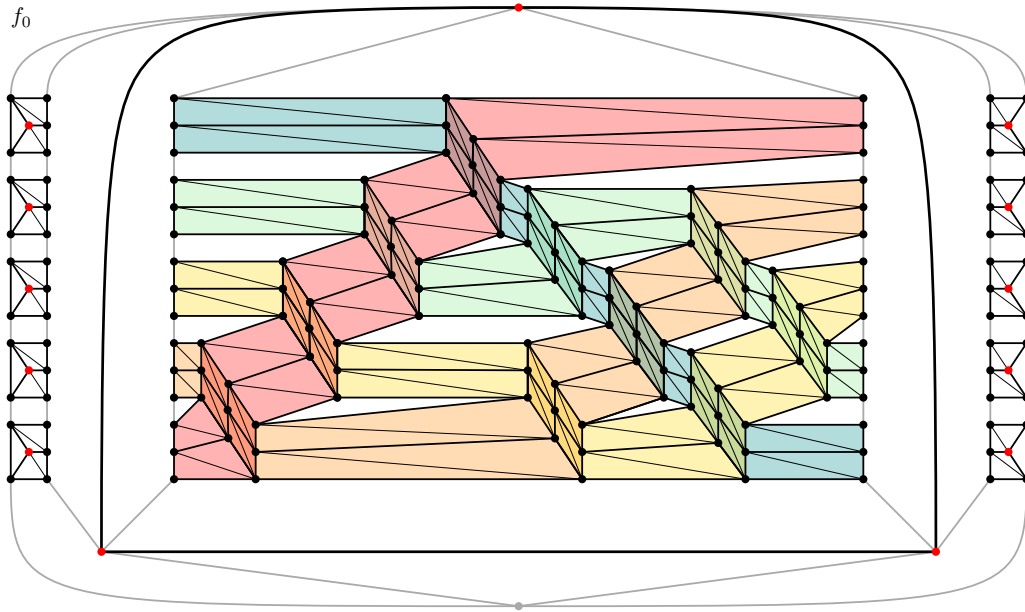


■ **Figure 6** Second step in the construction of the complex C : (left) Gluing of tunnel parts viewed from above. (right) During the identification process, the vertices of the top tunnel are moved to the vertices of the bottom tunnel.

For the associated almost geometric embedding, we shortly explain here how to geometrically embed the tunnels. To this end, we may easily distribute the sections of the tunnels such that the six vertices of both tunnels (which will be pairwise identified) have the same x, y -coordinates. Then, during the identification process, we move the vertices of the top tunnel to the vertices of the lower tunnel.

In the third step, we add a new vertex to the construction that we call the *apex* and which we denote by u . We think of u as the observer high above (at infinity) and insert a triangle defined by u and the vertices of every edge that is *visible* from u . Clearly, every edge of the helper triangle Δ is visible. Moreover, note that every roof section that is neither glued in a crossing nor hidden by the helper triangle is visible. In contrast, no bottom of any tunnel is visible in the almost geometric embedding.

In the fourth step, we enhance the 1-complex induced by the neighborhood $N(u)$ of the apex u such that it corresponds to an essentially 3-connected planar graph G^+ . We call a graph *essentially 3-connected* if it is a subdivision of a 3-connected graph. With the description so far, the 1-complex corresponds to the graph H depicted in black in Figure 7.



■ **Figure 7** Third and fourth step in the construction of the complex C : neighborhood of the apex u . The graph H after the third step is depicted in black. Together with the gray edges, the graph is a candidate for the essentially 3-connected plane graph G^+ and its subgraph G inside Δ . Each red vertex outside of Δ is a start vertex s_i or an end vertex t_i of some special edge e_i ; its black component represents the first or last section of tunnel T_i^+ , respectively.

To construct G^+ , we make use of the following fact. We define the degree of a face in a potentially disconnected plane graph as the number of edges in the face boundary (counted with multiplicity), plus 1 for each but one component incident to the face. Note that the degree of a face is thus lower bounded by the number of incident vertices and upper bounded by twice the number of incident vertices.

▷ **Claim 5.** For every plane graph $G_1 = (V_1, E_1)$, there exists an essentially 3-connected plane graph $G_2 = (V_2, E_2)$ such that G_1 is a subgraph of G_2 and any straight-line drawing D_1 of G_1 in the plane can be extended to a straight-line drawing of G_2 . Moreover, if the maximum face degree of G_1 is k , then the size of G_2 can be bounded by $|V_2| + |E_2| \leq O(k|V_1|)$.

Let $G^+ := G_2$ be an essentially 3-connected plane graph guaranteed by Claim 5 for the case that $G_1 = H$. Note that G_1 has $O(n^2)$ vertices and edges, and every face has degree $O(n)$. Hence, the size of G_2 is in $O(n^3)$. We denote the outer face of G_2 by f_0 . The reader is invited to think about the far more sparse graph depicted in Figure 7, which also serves as a candidate for G^+ . Indeed, the depicted graph also fulfills all properties necessary for our construction; however, not all properties of Claim 5. For example, the depicted graph is even 3-connected. The proof of this is straightforward, but a bit tedious. Thus, we leave it as an exercise to the interested reader to check that the graph remains connected even after the deletion of any two vertices or alternatively, that any pair of vertices is connected by three disjoint paths.

Later, the subgraph G of G^+ that is induced by all vertices of $\bigcup_i T_i$ will be of particular interest; in Figure 7, these vertices (and their convex hull) lie inside the helper triangle Δ . Recall that T_i denotes the part of the tunnel T_i^+ obtained by deleting the first two and last two sections.

It is a well-known fact that all (straight-line or topological) planar drawings of a 3-connected planar graph on the sphere are equivalent [34]; for a definition of equivalent drawings consult Section 1.2. Consequently, the result extends to *essentially* 3-connected graphs as it also holds for topological drawings. For later reference, we note the following.

▷ **Claim 6.** The planar graph G^+ is essentially 3-connected. Therefore, all crossing-free drawings of G^+ on a sphere are equivalent. Furthermore, any straight-line drawing of H in the plane can be extended to a straight-line drawing of G^+ .

We ensure that the neighborhood complex of u is the underlying planar graph of G^+ , i.e., for each edge of G^+ not present in H , we insert a triangle formed by the vertices of this edge together with u and call the resulting complex \overline{C} .

In the fifth and last step, our final complex C consist of two copies of \overline{C} in which the apex vertices are identified. We use these two copies in order to guarantee that in any geometric embedding the apex lies outside of all tunnels for one copy of \overline{C} . This finishes the construction of the abstract complex C .

It remains to show that our construction runs in polynomial time and fulfills the claimed properties.

Time Complexity. In order to verify that the construction shows $\exists\mathbb{R}$ -hardness, we argue that it has a running time that is polynomial in the size of the input. To this end, note that a pseudoline arrangement with n pseudolines can be described by the sequence of crossings along each pseudoline, i.e., by the $O(n^2)$ crossings. Thus, the input size is $N = O(n^2)$. After adding the helper triangle and ℓ_0 , the crossing diagram still has a size in $O(n^2)$. It is easy to see that our construction has a size proportional to $N^{3/2}$: For each segment and crossing of the diagram, we insert a constant number of objects. Moreover, we add a triangle for every (additional) edge in G^+ ; recall that G^+ has size $O(n^3)$. Consequently, the total construction has size $O(n^3) = O(N^{3/2})$. We remark, that a more careful choice of G^+ , as in Figure 7, yields a construction that is linear in N .

It remains to show that the pseudoline arrangement L is stretchable if and only if C has a geometric embedding in \mathbb{R}^3 .

Correctness. If L is stretchable, it is relatively straight-forward to construct a geometric embedding of C .

▷ **Claim 7.** If L is stretchable, then C has a geometric embedding.

The reverse direction is more involved and the interesting challenge.

▷ **Claim 8.** If C has a geometric embedding, then L is stretchable.

This finishes the proof of Theorem 4. ◀

Fattening the Complex. In the following, we present a simple modification for the proof of Theorem 4 to obtain hardness for pure 2- and 3-complexes.

► **Corollary 9.** *The decision problems $GEM_{2 \rightarrow 3}$ and $GEM_{3 \rightarrow 3}$ are $\exists\mathbb{R}$ -hard, even when restricting to pure complexes.*

Proof. The constructed 2-complex C in the proof of Theorem 4 was not pure because the special edges are not contained in any triangle. We obtain a pure 2-complex \hat{C} by adding one new vertex to each special edge such that it forms a special triangle. On the one hand, given a geometric embedding of C in \mathbb{R}^3 , the new vertices can easily be added close enough to their defining set in C . On the other hand, any geometric embedding of \hat{C} induces an embedding of C . Hence, C has a geometric embedding if and only if \hat{C} has a geometric embedding in \mathbb{R}^3 .

Analogously, we can add a private vertex to each triangle of \hat{C} to form a pure 3-complex which has a geometric embedding if and only if C has a geometric embedding in \mathbb{R}^3 . ◀

Alternatively for showing hardness of $GEM_{3 \rightarrow 3}$, we remark that hardness of $GEM_{k \rightarrow d}$ for $k < d$ easily implies hardness of $GEM_{\ell \rightarrow d}$ for all $k \leq \ell \leq d$ by adding a disjoint ℓ -simplex to the construction which always has a geometric embedding in \mathbb{R}^d .

2.3 Dimension Reduction

In order to show hardness for all remaining cases of Theorem 1, we establish the following dimension reduction.

► **Lemma 10.** *The decision problem $GEM_{k \rightarrow d}$ reduces to $GEM_{k+1 \rightarrow d+1}$.*

The idea is to add two apices to a k -complex C in order to obtain a $(k+1)$ -complex C^+ . We then argue that C has a geometric embedding in \mathbb{R}^d if and only if C^+ has a geometric embedding in \mathbb{R}^{d+1} . More formally, for a complex $C = (V, E)$ and a disjoint vertex set U , $C * U$ denotes the *join* complex $(V \cup U, E')$ where $E' := \{e \cup u \mid e \in E, u \in U\}$. The following claim immediately implies Lemma 10.

▷ **Claim 11.** Let $C = (V, E)$ be a complex, $a, b \notin V$ two new vertices, and $C^+ := C * \{a, b\}$ their join complex. Then C has a geometric embedding in \mathbb{R}^d if and only if C^+ has a geometric embedding in \mathbb{R}^{d+1} .

Proof. Let φ be a geometric embedding of C in \mathbb{R}^d . Then, we define for $v \in V \cup \{a, b\}$,

$$\varphi'(v) = \begin{cases} (\varphi(v), 0) & \text{if } v \in V, \\ (0, \dots, 0, +1) & \text{if } v = a, \\ (0, \dots, 0, -1) & \text{if } v = b. \end{cases}$$

It is easy to check that φ' is a geometric embedding of C^+ in \mathbb{R}^{d+1} : By definition of the last coordinate, any two simplices where one possibly contains a and the other possibly contains b can only intersect in the subspace induced by the first d coordinates. Consequently, all (interesting) potential intersections happen in the d -dimensional subspace induced by the first d coordinates. Hence φ implies the correctness of the geometric embedding.

For the reverse direction, consider a geometric embedding φ of C^+ in \mathbb{R}^{d+1} . Let $\varphi_a := \varphi(a)$ and $\varphi_b := \varphi(b)$. Without loss of generality, we assume that $\varphi_a - \varphi_b$ is orthogonal to the first d coordinates, i.e., $\varphi_a - \varphi_b$ is parallel to the $(d+1)$ -st coordinate axis. Let $\bar{\varphi}(C) := \bigcup_{e \in E} \bar{\varphi}(e)$

denote the induced geometric subrepresentation of C . We claim that the orthogonal projection $f : \overline{\varphi}(C) \rightarrow \mathbb{R}^d$ to the first d coordinates (i.e., the effect of the function is the one of restricting to the first d coordinates, \cdot) is injective. Thus $\varphi' := f \circ \varphi$ yields a representation of C in \mathbb{R}^d .

For the purpose of a contradiction, suppose that f is not injective. Then there exist two distinct points $p = (p_1, \dots, p_{d+1})$ and $q = (q_1, \dots, q_{d+1})$ with $p, q \in \overline{\varphi}(C)$ such that $(p_1, \dots, p_d) = (q_1, \dots, q_d)$ and $p_{d+1} \neq q_{d+1}$. Without loss of generality, we may assume that $p_{d+1} > q_{d+1}$. Consider the plane P spanned by φ_a, φ_b, p . Note that $q \in P$ because $\varphi_a - \varphi_b$ and $p - q$ are parallel (to the $(d + 1)$ st coordinate axis). For an illustration, see Figure 8.

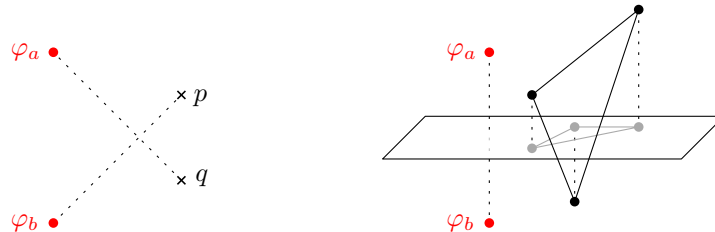


Figure 8 Illustration for the proof of Claim 11. The geometric embedding φ of C^+ gives a monotone embedding of C , otherwise we can find an intersection in C^+ .

Let us denote with $e_p \in E$ and $e_q \in E$ any choice of hyperedges such that $p \in \overline{\varphi}(e_p)$ and $q \in \overline{\varphi}(e_q)$. Consider the two open segments $\text{seg}^\circ(\varphi_a, q) \in \overline{\varphi}(e_q \cup a)$ and $\text{seg}^\circ(\varphi_b, p) \in \overline{\varphi}(e_p \cup b)$. Clearly, these open segments intersect in a point x , as illustrated in Figure 8. Because φ is a geometric embedding, it holds that

$$x \in \overline{\varphi}(e_q \cup a) \cap \overline{\varphi}(e_p \cup b) = \overline{\varphi}(e_q \cap e_p) = \overline{\varphi}(e_q) \cap \overline{\varphi}(e_p).$$

In particular, this implies that $x \in \overline{\varphi}(e_q)$ and thus that $x \in \text{seg}^\circ(\varphi_a, q) \cap \overline{\varphi}(e_q)$. However, because $\overline{\varphi}(e_q \cup a)$ is a simplex, φ_a does not lie in $\text{span}(\overline{\varphi}(e_q))$ and thus $\text{seg}^\circ(\varphi_a, q) \cap \overline{\varphi}(e_q) = \emptyset$. A contradiction. \triangleleft

3 Conclusion

We established the computational complexity of $\text{GEM}_{k \rightarrow d}$ for all $d \geq 3$ and $k \in \{d - 1, d\}$. In particular, we showed that for these values it is complete for $\exists\mathbb{R}$ to distinguish PL embeddable k -complexes in \mathbb{R}^d from geometrically embeddable ones. Arguably, $\text{GEM}_{2 \rightarrow 3}$ is the most interesting case.

Investigating the computational complexity for the remaining open entries in Table 2 remains for future work. We strengthen the conjecture of Skopenkov [59] as follows.

Conjecture. *The problem $\text{GEM}_{k \rightarrow d}$ is $\exists\mathbb{R}$ -complete for all k, d such that $\max\{3, k\} \leq d \leq 2k$.*

References

- 1 Zachary Abel, Erik Demaine, Martin Demaine, Sarah Eisenstat, Jayson Lynch, and Tao Schardl. Who needs crossings? Hardness of plane graph rigidity. In *International Symposium on Computational Geometry (SoCG)*, pages 3:1–3:15, 2016. doi:10.4230/LIPIcs.SoCG.2016.3.
- 2 Mikkel Abrahamsen. Covering polygons is even harder. In *Foundations on Computer Science (FOCS)*, 2021. doi:10.1109/FOCS52979.2021.00045.

- 3 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is $\exists\mathbb{R}$ -complete. In *Symposium on Theory of Computing (STOC)*, pages 65–73, 2018. doi:10.1145/3188745.3188868.
- 4 Mikkel Abrahamsen, Linda Kleist, and Tillmann Miltzow. Training neural networks is $\exists\mathbb{R}$ -complete. In *Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 18293–18306, 2021. URL: <https://proceedings.neurips.cc/paper/2021/file/9813b270ed0288e7c0388f0fd4ec68f5-Paper.pdf>.
- 5 Mikkel Abrahamsen, Tillmann Miltzow, and Nadja Seiferth. Framework for ER-completeness of two-dimensional packing problems. In *Foundations on Computer Science (FOCS)*, pages 1014–1021. IEEE, 2020. doi:10.1109/FOCS46700.2020.00098.
- 6 J. L. Ramírez Alfonsín. Knots and links in spatial graphs: a survey. *Discrete mathematics*, 302(1-3):225–242, 2005. doi:10.1016/j.disc.2004.07.035.
- 7 Vittorio Bilò and Marios Mavronicolas. A catalog of EXISTS-R-complete decision problems about Nash equilibria in multi-player games. In *Symposium on Theoretical Aspects of Computer Science (STACS)*, 2016. doi:10.4230/LIPIcs.STACS.2016.17.
- 8 R. H. Bing. An alternative proof that 3-manifolds can be triangulated. *Annals of Mathematics*, 69(1):37–65, 1959. doi:10.2307/1970092.
- 9 Jürgen Bokowski and A. Guedes de Oliveira. On the generation of oriented matroids. *Discrete & Computational Geometry (DCG)*, 24(2):197–208, 2000. doi:10.1007/s004540010027.
- 10 Ulrich Brehm. A nonpolyhedral triangulated Möbius strip. *Proceedings of the American Mathematical Society*, 89(3):519–522, 1983. doi:10.1090/S0002-9939-1983-0715878-1.
- 11 Ulrich Brehm and Karanbir S. Sarkaria. Linear vs. piecewise-linear embeddability of simplicial complexes. *Technical Report MPI Bonn*, pages 1–15, 1992. available at <http://kssarkaria.org/docs/Linear%20vs.%20piecewise-linear%20embeddability%20of%20simplicial%20complexes.pdf>.
- 12 J. L. Bryant. Approximating embeddings of polyhedra in codimension three. *Transactions of the American Mathematical Society*, 170:85–95, 1972. doi:10.1090/S0002-9947-1972-0307245-7.
- 13 Martin Čadek, Marek Krčál, Jiří Matoušek, Francis Sergeraert, Lukáš Vokřínek, and Uli Wagner. Computing all maps into a sphere. *Journal of the ACM (JACM)*, 61(3):1–44, 2014. doi:10.1145/2597629.
- 14 Martin Čadek, Marek Krčál, Jiří Matoušek, Lukáš Vokřínek, and Uli Wagner. Extendability of continuous maps is undecidable. *Discrete & Computational Geometry (DCG)*, 51(1):24–66, 2014. doi:10.1007/s00454-013-9551-8.
- 15 Martin Čadek, Marek Krčál, Jiří Matoušek, Lukáš Vokřínek, and Uli Wagner. Time computation of homotopy groups and Postnikov systems in fixed dimension. *SIAM Journal of Computing (SICOMP)*, 43(5):1728–1780, 2014. doi:10.1137/120899029.
- 16 Martin Čadek, Marek Krčál, and Lukáš Vokřínek. Algorithmic solvability of the lifting-extension problem. *Discrete & Computational Geometry (DCG)*, 57(4):915–965, 2017. doi:10.1007/s00454-016-9855-6.
- 17 John Canny. Some algebraic and geometric computations in PSPACE. In *Symposium on Theory of Computing (STOC)*, pages 460–467. ACM, 1988. doi:10.1145/62212.62257.
- 18 Jean Cardinal. Computational geometry column 62. *SIGACT News*, 46(4):69–78, 2015. doi:10.1145/2852040.2852053.
- 19 Jean Cardinal and Udo Hoffmann. Recognition and complexity of point visibility graphs. *Discrete & Computational Geometry (DCG)*, 57(1):164–178, 2017. doi:10.1007/s00454-016-9831-1.
- 20 Johannes Carmesin. Embedding simply connected 2-complexes in 3-space – I. A Kuratowski-type characterisation. arXiv preprint, 2019. arXiv:1709.04642.
- 21 Jean Dieudonné. *A history of algebraic and differential topology, 1900-1960*. Springer, 2009.
- 22 Michael Gene Dobbins, Andreas Holmsen, and Tillmann Miltzow. A universality theorem for nested polytopes. arXiv preprint, 2019. arXiv:1908.02213.

- 23 Jeff Erickson, Ivor van der Hoog, and Tillmann Miltzow. Smoothing the gap between NP and ER. In *Foundations on Computer Science (FOCS)*, pages 1022–1033. IEEE, 2020. doi:10.1109/FOCS46700.2020.00099.
- 24 Marek Filakovský, Uli Wagner, and Stephan Zhechev. Embeddability of simplicial complexes is undecidable. In *Symposium on Discrete Algorithms (SODA)*, pages 767–785, 2020. doi:10.1137/1.9781611975994.47.
- 25 Antonio Flores. Über n-dimensionale Komplexe, die im \mathbb{R}_{2n+1} absolut selbstverschlungen sind. In *Ergeb. Math. Kolloq*, volume 34, pages 4–6, 1933.
- 26 Michael H. Freedman, Vyacheslav S. Krushkal, and Peter Teichner. Van Kampen’s embedding obstruction is incomplete for 2-complexes in \mathbb{R}^4 . *Mathematical Research Letters*, 1(2):167–176, 1994.
- 27 Florian Frick, Mirabel Hu, Nick Scheel, and Steven Simon. Embedding dimensions of simplicial complexes on few vertices. arXiv preprint, 2021. arXiv:2109.04855.
- 28 Jugal Garg, Ruta Mehta, Vijay V. Vazirani, and Sadra Yazdanbod. $\exists\mathbb{R}$ -completeness for decision versions of multi-player (symmetric) Nash equilibria. *ACM Transactions on Economics and Computation*, 6(1):1:1–1:23, 2018. doi:10.1145/3175494.
- 29 Jonathan L. Gross and Ronald H. Rosen. A linear time planarity algorithm for 2-complexes. *Journal of the ACM (JACM)*, 26(4):611–617, 1979. doi:10.1145/322154.322156.
- 30 Branko Grünbaum. Imbeddings of simplicial complexes. *Commentarii Mathematici Helvetici*, 44(1):502–513, 1969. doi:10.5169/seals-33795.
- 31 Branko Grünbaum. Polytopes, graphs, and complexes. *Bulletin of the American Mathematical Society*, 76(6):1131–1201, 1970. doi:10.1090/S0002-9904-1970-12601-5.
- 32 Rudolf Halin and Heinz Jung. Charakterisierung der Komplexe der Ebene und der 2-Sphäre. *Archiv der Mathematik*, 15(1):466–469, 1964.
- 33 John Hopcroft and Robert Tarjan. Efficient planarity testing. *Journal of the ACM (JACM)*, 21(4):549–568, 1974. doi:10.1145/321850.321852.
- 34 Wilfried Imrich. On whitney’s theorem on the unique embeddability of 3-connected planar graphs. In *Recent advances in graph theory: Proceedings of the Symposium held in Prague*, volume 1974, pages 303–306, 1975.
- 35 Fáy István. On straight-line representation of planar graphs. *Acta scientiarum mathematicarum*, 11(229-233):2, 1948.
- 36 Marek Krčál, Jiří Matoušek, and Francis Sergeraert. Polynomial-time homology for simplicial Eilenberg–MacLane spaces. *Foundations of Computational Mathematics (FoCM)*, 13(6):935–963, 2013. doi:10.1007/s10208-013-9159-7.
- 37 Anna Lubiw, Tillmann Miltzow, and Debajyoti Mondal. The complexity of drawing a graph in a polygonal region. In *International Symposium on Graph Drawing and Network Visualization (GD)*, pages 387–401. Springer, 2018. doi:10.1007/978-3-030-04414-5_28.
- 38 Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. arXiv preprint, 2014. arXiv:1406.2636.
- 39 Jiří Matoušek, Eric Sedgwick, Martin Tancer, and Uli Wagner. Embeddability in the 3-sphere is decidable. *Journal of the ACM (JACM)*, 65(1):1–49, 2018. doi:10.1145/2582112.2582137.
- 40 Jiří Matoušek, Martin Tancer, and Uli Wagner. Hardness of embedding simplicial complexes in \mathbb{R}^d . *Journal of the European Mathematical Society (JEMS)*, 13(2):259–295, 2011. doi:10.4171/JEMS/252.
- 41 Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114–143, 2013. doi:10.1016/j.jctb.2012.09.004.
- 42 Karl Menger. *Dimensionstheorie*. Vieweg+Teubner Verlag, 1 edition, 1928. doi:10.1007/978-3-663-16056-4.
- 43 Arnaud de Mesmay, Yo’av Rieck, Eric Sedgwick, and Martin Tancer. Embeddability in \mathbb{R}^3 is NP-hard. *Journal of the ACM (JACM)*, 67(4):20:1–20:29, 2020. doi:10.1145/3396593.
- 44 Tillmann Miltzow and Reinier F. Schmiermann. On classifying continuous constraint satisfaction problems. In *Foundations of Computer Science (FOCS 2021)*, pages 781–791. IEEE, 2022. doi:10.1109/FOCS52979.2021.00081.

- 45 Nicolai Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In Oleg Y. Viro, editor, *Topology and geometry – Rohlin seminar*, pages 527–543. Springer, 1988. doi:10.1007/BFb0082792.
- 46 Isabella Novik. A note on geometric embeddings of simplicial complexes in a euclidean space. *Discrete & Computational Geometry (DCG)*, 23(2):293–302, 2000. doi:10.1007/s004549910019.
- 47 Patrice Ossona deMendez. Realization of posets. *Journal of Graph Algorithms and Applications (JGAA)*, 6(1):149–153, 2002. doi:10.7155/jgaa.00048.
- 48 Christos Papakyriakopoulos. A new proof of the invariance of the homology groups of a complex. *Bulletin of the Greek Mathematical Society*, 22:1–154, 1943.
- 49 S. Parsa and A. Skopenkov. On embeddability of joins and their ‘factors’. arXiv preprint, 2020. arXiv:2003.12285.
- 50 Jürgen Richter-Gebert. *Realization spaces of polytopes*, volume 1643 of *LNM*. Springer, 1996. doi:10.1007/BFb0093761.
- 51 Jürgen Richter-Gebert and Günter M. Ziegler. Realization spaces of 4-polytopes are universal. *Bulletin of the American Mathematical Society*, 32(4):403–412, 1995. doi:10.1090/S0273-0979-1995-00604-X.
- 52 Marcus Schaefer. Complexity of some geometric and topological problems. In *International Symposium on Graph Drawing (GD)*, LNCS, pages 334–344. Springer, 2009. doi:10.1007/978-3-642-11805-0_32.
- 53 Egon Schulte and Ulrich Brehm. Polyhedral maps. In Csaba D. Toth, Jacob E. Goodman, and Joseph O’Rourke, editors, *Handbook of Discrete and Computational Geometry, Third Edition*, pages 533–548. Chapman and Hall/CRC, 2017. doi:10.1201/9781315119601.
- 54 Arnold Shapiro. Obstructions to the imbedding of a complex in a Euclidean space.: I. the first obstruction. *Annals of Mathematics*, pages 256–269, 1957. doi:10.2307/1969998.
- 55 Yaroslav Shitov. A universality theorem for nonnegative matrix factorizations. arXiv preprint, 2016. arXiv:1606.09068.
- 56 Peter Shor. Stretchability of pseudolines is NP-hard. In Peter Gritzmann and Bernd Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, DIMACS – Series in Discrete Mathematics and Theoretical Computer Science, pages 531–554. AMS, 1991.
- 57 Arkadiy Skopenkov. Realizability of hypergraphs and ramsey link theory. arXiv preprint, 2014. arXiv:1402.0658.
- 58 Arkadiy Skopenkov. Extendability of simplicial maps is undecidable. arXiv preprint, 2020. arXiv:2008.00492.
- 59 Arkadiy Skopenkov. Invariants of graph drawings in the plane. *Arnold Mathematical Journal*, 6:21–55, 2020. doi:10.1007/s40598-019-00128-5.
- 60 Arkadiy Skopenkov and Martin Tancer. Hardness of almost embedding simplicial complexes in \mathbb{R}^d . *Discrete & Computational Geometry (DCG)*, 61(2):452–463, 2019. doi:10.1007/s00454-018-0013-1.
- 61 Mikhail Skopenkov. Embedding products of graphs into euclidean spaces. arXiv preprint, 2016. arXiv:0808.1199.
- 62 Ernst Steinitz. Polyeder und Raumeinteilungen. In *Encyclopädie der mathematischen Wissenschaften*, volume 3-1-2 (Geometrie), chapter 12, pages 1–139. B. G. Teubner, Leipzig, 1922.
- 63 Dagmar Timmreck. Necessary conditions for geometric realizability of simplicial complexes. In A.I. Bobenko, P. Schröder, J.M. Sullivan, and G.M. Ziegler, editors, *Discrete Differential Geometry*, volume 38 of *Oberwolfach Seminars*, pages 215–233. Birkhäuser Basel, 2008. doi:10.1007/978-3-7643-8621-4_11.
- 64 Dagmar Ingrid Timmreck. *Realization Problems for Point Configurations and Polyhedral Surfaces*. PhD thesis, Freie Universität Berlin, 2015. doi:10.17169/refubium-14465.
- 65 Brian R. Ummel. The product of nonplanar complexes does not imbed in 4-space. *Transactions of the American Mathematical Society*, 242:319–328, 1978. doi:10.2307/1997741.

- 66 Egbert R. Van Kampen. Komplexe in euklidischen Räumen. In *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, volume 9, pages 72–78. Springer, 1933.
- 67 Wen-tsün Wu. *A theory of imbedding, immersion, and isotopy of polytopes in a Euclidean space*. Science Press, 1965.

Distinguishing Classes of Intersection Graphs of Homothets or Similarities of Two Convex Disks

Mikkel Abrahamsen  

BARC, University of Copenhagen, Denmark

Bartosz Walczak  

Department of Theoretical Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

Abstract

For smooth convex disks A , i.e., convex compact subsets of the plane with non-empty interior, we classify the classes $G^{\text{hom}}(A)$ and $G^{\text{sim}}(A)$ of intersection graphs that can be obtained from homothets and similarities of A , respectively. Namely, we prove that $G^{\text{hom}}(A) = G^{\text{hom}}(B)$ if and only if A and B are affine equivalent, and $G^{\text{sim}}(A) = G^{\text{sim}}(B)$ if and only if A and B are similar.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases geometric intersection graph, convex disk, homothet, similarity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.2

Related Version *Full Version*: <https://arxiv.org/abs/2108.04588>

Funding *Mikkel Abrahamsen*: The author is supported by Starting Grant 1054-00032B from the Independent Research Fund Denmark under the Sapere Aude research career programme. BARC is supported by the VILLUM Foundation grant 16582.

Bartosz Walczak: The author is partially supported by the National Science Center of Poland grant 2015/17/D/ST1/00585.

1 Introduction

Disk graphs have received much attention due to their ability to model graphs appearing in practice and their interesting structural properties. In a disk graph, each vertex corresponds to a (circular) disk, and there is an edge between two vertices if and only if the two corresponding disks intersect. Disk graphs appear naturally in problems related to radio and sensor networks. For instance, the region reached by the signal from each transmitter in a radio network can be modeled as a disk, and when two disks intersect, the interference of the signals may be an issue if the transmitters use the same frequency. The problem of avoiding interference while minimizing the number of used frequencies thus corresponds to finding the chromatic number of the disk graph. Applications like these are part of the motivation for various papers on algorithms or computational hardness for problems taking disk graphs in the input [2, 3, 7, 10, 13, 14, 16, 25] as well as papers studying disk graphs from a mathematical angle [21, 22].

Combinatorial analysis of problems such as chromatic number and minimum hitting set size has often been performed in greater generality, for intersection graphs of translated copies or homothetic (i.e., translated and scaled) copies of a fixed convex shape [11, 17, 18, 23], and recently also for translated, scaled, and rotated squares [6]. Algorithmic considerations have also been generalized in a similar way – Bonnet, Grelier, and Miltzow [4] studied the maximum clique problem and extended classic algorithms for disk graphs and unit disk graphs to intersection graphs of homothetic or translated copies of a fixed convex set.



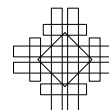
© Mikkel Abrahamsen and Bartosz Walczak;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

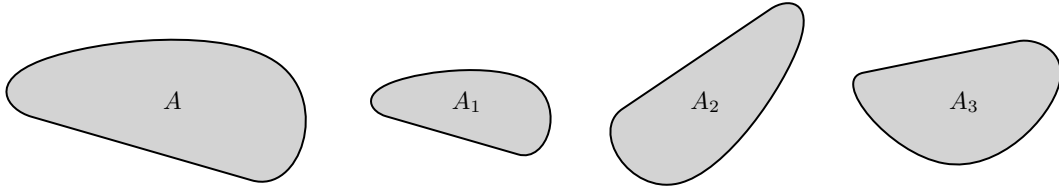
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 2; pp. 2:1–2:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Here, A_1 is a homothet of A , A_2 is a similarity but not a homothet of A , and A_3 is affine equivalent to A , but not similar to A . By Theorem 1, A and A_3 induce the same intersection graphs of homothets, but Theorem 2 implies that the intersection graphs of similarities are different.

A well-established line of research in discrete and computational geometry has been aiming at understanding the relationships between classes of geometric intersection graphs such as whether two classes are equal or whether one class is a subclass of another [5, 8, 9, 15, 20]. In view of the above-mentioned research, it is natural to investigate the relationships between the classes of intersection graphs of translated copies, homothetic copies, and copies by similarity (translation, scaling, and rotation) of a fixed convex shape.

To be precise, consider an arbitrary *convex disk* A , that is, a convex and compact set in the plane with non-empty interior. A *translate* of A is a translated copy of A (with no scaling or rotation allowed). A *homothet* of A is a positively scaled and translated copy of A (with no rotation allowed). A *similarity* is a homothet rotated by an arbitrary angle. An *affine equivalent* of A is the image of A under an invertible affine transformation. See Figure 1. The *intersection graph* of a family \mathcal{F} of sets in the plane is the graph with vertex set \mathcal{F} and edge set $\{uv: u, v \in \mathcal{F}, u \cap v \neq \emptyset\}$.

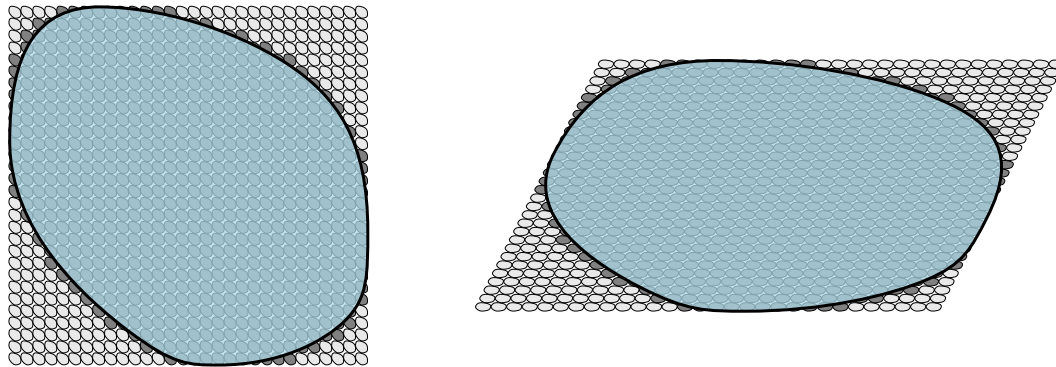
In a recent paper, Aamand, Abrahamsen, Knudsen, and Rasmussen [1] studied the question of when the translates of two convex disks induce the same intersection or contact graphs, where a *contact graph* is an intersection graph that can be realized by pairwise interior-disjoint disks. They proved for a large class of convex disks, including all strictly convex ones, that two disks A and B yield the same classes of contact and intersection graphs if and only if the central symmetrals of A and B are affine equivalent, where the *central symmetral* of a disk A is the centrally symmetric disk $\frac{1}{2}(A - A)$.

In this paper, we study the question of when the homothets or the similarities of two convex disks induce the same intersection graphs. We make the additional assumption that the convex disk A be *smooth*, that is, there is a unique tangent containing any point on the boundary of A . We let $\text{hom } A$ and $\text{sim } A$ denote the sets of homothets and similarities of A , respectively. We let $G^{\text{hom}}(A)$ and $G^{\text{sim}}(A)$ denote the classes of (finite) intersection graphs of homothets and similarities of A , respectively. For two smooth convex disks A and B , we are able to say exactly when $G^{\text{hom}}(A) = G^{\text{hom}}(B)$ and $G^{\text{sim}}(A) = G^{\text{sim}}(B)$, as follows.

► **Theorem 1.** *Let A and B be smooth convex disks. Then $G^{\text{hom}}(A) = G^{\text{hom}}(B)$ if and only if A and B are affine equivalent. Moreover, if A and B are not affine equivalent, then neither $G^{\text{hom}}(A) \subseteq G^{\text{hom}}(B)$ nor $G^{\text{hom}}(B) \subseteq G^{\text{hom}}(A)$.*

► **Theorem 2.** *Let A and B be smooth convex disks. Then $G^{\text{sim}}(A) = G^{\text{sim}}(B)$ if and only if B is similar to A or to the reflection $A^r = \{(-x, y): (x, y) \in A\}$.*

If A and B are affine equivalent, then $G^{\text{hom}}(A) = G^{\text{hom}}(B)$, because the affine transformation that maps A to B transforms every realization in $\text{hom } A$ to a realization of the same graph in $\text{hom } B$, and vice versa. Likewise, if B is similar to A or to A^r , then $G^{\text{sim}}(A) = G^{\text{sim}}(B)$, because the similarity transformation (possibly with reflection) that maps A to B transforms every realization in $\text{sim } A$ to a realization of the same graph in $\text{sim } B$, and vice versa. The difficult part is the necessity of these conditions.



■ **Figure 2** To the left is shown the grid of small copies of A and one large copy of A on top. The disks in the grid that are intersected (dark gray) define the shape of A to an arbitrarily high precision, if we make the grid sufficiently fine. To the right is shown the same graph realized by another disk B . As we will show, the arrangement must again form a grid of small disks with one large copy of B on top. An affine map that makes the two grids coincide then also maps B to A to within a small error, since the two disks intersect the same “pixels” in the grids.

When A and B are not affine equivalent, we point out graphs $G_A \in G^{\text{hom}}(A)$ and $G_B \in G^{\text{hom}}(B)$ such that $G_A \notin G^{\text{hom}}(B)$ and $G_B \notin G^{\text{hom}}(A)$, which yields the second part of Theorem 1. By contrast, when B is dissimilar to both A and A^r , then $G^{\text{sim}}(A)$ and $G^{\text{sim}}(B)$ may be properly nested. Indeed, if A is a circular disk and B is a non-circular filled ellipse, then $G^{\text{sim}}(A) \subset G^{\text{sim}}(B)$, because the affine stretch that maps A to B transforms every realization in $\text{hom } A = \text{sim } A$ to a realization of the same graph in $\text{hom } B \subseteq \text{sim } B$, while in the proof of Theorem 2, we construct a graph in $G^{\text{sim}}(B)$ that is not in $G^{\text{sim}}(A)$.

One may or may not allow scaling by negative numbers when defining the homothets of A , which corresponds to rotating A by 180° . We remark that Theorem 1 holds in either case (with the same proof). Likewise, one may or may not allow reflection along the y -axis when defining the similarities of A , and Theorem 2 holds in either case (with the same proof).

We note that although we establish results for more general families of graphs, our results are not generalizations of the ones in [1]. We also remark that the contact graphs of homothets or similarities of a smooth convex disk have already been characterized. The Koebe-Andreev-Thurston Circle Packing Theorem, first proved by Koebe in 1936 [19], asserts that every planar graph is the contact graph of some set of pairwise interior-disjoint circular disks. Since every contact graph is planar, the contact graphs are exactly the planar graphs. The Monster Packing Theorem by Schramm [24] generalizes the result in the following way. Suppose that a planar graph is given, together with a correspondence which assigns to each vertex of the graph a smooth convex disk. Then there exists a contact representation of the graph where each vertex is represented by a homothet of the associated disk. Hence the contact graphs of homothets or similarities of any smooth convex disk are the planar graphs.

Outline of the paper

In Section 2, we set our notation and define the central concepts. In Section 3, we introduce a notion of convergence of sequences of compact subsets of \mathbb{R}^2 . The usual definition of convergence based on the Hausdorff distance between sets only allows us to talk about convergence towards a compact set, but in our case, we also need to be able to express, for instance, that a sequence of (growing) convex disks converges to a half-plane.

In Sections 4 and 5, we introduce the constructions that enable us to distinguish the graph classes. At an overall level, the idea behind our constructions is to define a graph G such that however G is realized as an intersection graph of homothets or similarities of a smooth convex disk A , then a subset of the disks in the realization will form a large and almost regular grid of small copies of A ; see Figure 2. We use this grid in a somewhat similar manner as the grid of pixels in television: We put one large disk A on top of the grid. The disks in the grid that intersect A will then with high precision define the shape of A . If now another disk B is able to realize the same graph, then we can consider an affine transformation that makes the two grids “match”, and it follows that A and B must be nearly identical under this transformation, since the same “pixels” in the two grids are intersected by the large disks on top. If B can realize the graph for an arbitrarily fine resolution of the grid, then we get in the limit a transformation f^* that maps A to B .

In the case of homothets (Section 6), the transformation f^* is an arbitrary affine transformation, which leads to Theorem 1. In the case of similarities (Section 7), we can further prove that the grid must be square-shaped. It then follows that the limit transformation f^* is angle preserving, so B must be similar to A or A^c .

The construction of this grid is rather delicate and relies on a careful analysis of various building blocks described in Section 4. Our first basic tool (Lemma 9) is that if the complete bipartite graph $K_{2,n}$ is realized as an intersection graph of similarities of a convex disk A , then the distance between the two disks U_1 and U_2 in the first vertex class can be made arbitrarily much smaller than the size of U_1 and U_2 by choosing n large enough. In other words, in the limit where $n \rightarrow \infty$, the two disks U_1 and U_2 behave as if they were in contact.

We are then able to define a larger graph L_n where a realization has two disks U_1, U_2 and n disks V_1, \dots, V_n , such that by choosing n large enough, we know that all of the latter disks are arbitrarily small compared to both of U_1 and U_2 (Lemma 11), and they must furthermore be “squeezed in” between these disks. The disks in each row and each column of the aforementioned grid in the final construction will be a subset of the disks V_1, \dots, V_n in a realization of this graph L_n . Here, it is necessary to place chains of overlapping disks on top of each row and each column of the grid to ensure that when the grid becomes arbitrarily fine, it does not degenerate into a segment.

In the case of similarities, we introduce the concept of the *stretch* of a convex disk A , denoted ρ_A . We consider two parallel lines of distance 1 and a chain of n consecutively overlapping similarities of A , contained in the strip bounded by these lines. The stretch is the ratio between the (geometric) length of a longest such chain and n , as $n \rightarrow \infty$. Now if $\rho_B < \rho_A$, then it will be impossible for similarities of B to realize the graph that we construct for A , as there is no chain of similarities of B that can “reach far enough”. If $\rho_B = \rho_A$, then for both A and B the graph can be realized only so that the grid is square-shaped, since otherwise some chains in the realizations will not be able to reach far enough.

We conclude the paper in Section 8 by mentioning some open questions.

2 Preliminaries

Let $\text{int } X$ and ∂X denote the interior and the boundary of a set $X \subseteq \mathbb{R}^2$, respectively. A *convex disk* is a convex compact subset of \mathbb{R}^2 with non-empty interior. Every convex disk is the closure of its interior. Two non-empty subsets of \mathbb{R}^2 *touch* if they intersect but the interior of either one is disjoint from the other. A *tangent* to a convex disk A is a line that touches A (whence it follows that A lies in one of the two half-planes bounded by the line). For every convex disk A and every point $p \in \partial A$, there is at least one tangent to A containing p . A convex disk A is *smooth* if for every point $p \in \partial A$, there is exactly one tangent to A containing p . All convex disks that we consider are implicitly assumed to be smooth.

A *similarity* of a convex disk A is a rotated, scaled, and translated copy of A , that is, a set of the form

$$A' = \left\{ r \cdot \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} a + z : a \in A \right\},$$

where $r > 0$, $z \in \mathbb{R}^2$, and $\theta \in [0, 2\pi)$. We call r the *radius* of A' and denote it by $r_A(A')$. When A is clear from the context, we simplify the notation to $r(A')$. A similarity A' is a *homothet* of A if $\theta = 0$, that is, A' is a scaled and translated copy of A . We let $\text{sim } A$ and $\text{hom } A$ denote the set of similarities and the set of homothets of A , and we let $\text{sim}^r A = \text{sim } A \cup \text{sim } A^r$, where A^r is the reflection of A about the y axis: $A^r = \{(-x, y) : (x, y) \in A\}$.

A *realization* of a graph $G = (V, E)$ in a family \mathcal{F} of subsets of \mathbb{R}^2 is a mapping $R: V \rightarrow \mathcal{F}$ such that $R(u) \cap R(v) \neq \emptyset$ if and only if $uv \in E$. We consider only finite graphs and their realizations with $\mathcal{F} = \text{sim } A$ or $\mathcal{F} = \text{hom } A$ for some convex disk A .

The Euclidean norm of a vector $a \in \mathbb{R}^2$ is denoted by $\|a\|$. The Euclidean distance between points $p, q \in \mathbb{R}^2$ is denoted by $\text{dist}(p, q)$. This notation extends to the distance between a point $p \in \mathbb{R}^2$ and a set $X \subseteq \mathbb{R}^2$ or between two sets $X, Y \subseteq \mathbb{R}^2$:

$$\text{dist}(p, X) = \inf_{x \in X} \text{dist}(p, x), \quad \text{dist}(X, Y) = \inf_{x \in X} \inf_{y \in Y} \text{dist}(x, y).$$

For a point $q \in \mathbb{R}^2$ and $\delta > 0$, let $\text{ball}(q, \delta) = \{p \in \mathbb{R}^2 : \text{dist}(p, q) \leq \delta\}$. For a compact set $X \subseteq \mathbb{R}^2$ and $\delta > 0$, let $\text{ball}(X, \delta) = \{p \in \mathbb{R}^2 : \text{dist}(p, X) \leq \delta\}$. The diameter of a set $X \subseteq \mathbb{R}^2$, which is $\sup_{x, y \in X} \text{dist}(x, y)$, is denoted by $\text{diam } X$. The *bounding box* of a compact set $X \subset \mathbb{R}^2$ is the unique minimal box of the form $[x_1, x_2] \times [y_1, y_2]$ containing X . Let $\mathbb{N} = \{1, 2, \dots\}$ and $[n] = \{1, \dots, n\}$ for $n \in \mathbb{N}$.

3 Convergence and limits

Recall the notion of Hausdorff distance between non-empty subsets X and Y of a metric space:

$$d_H(X, Y) = \max \left\{ \sup_{x \in X} \text{dist}(x, Y), \sup_{y \in Y} \text{dist}(y, X) \right\}.$$

It is well known that the family of non-empty compact subsets of a (compact) metric space equipped with this notion of distance forms a (compact) metric space. This leads to a notion of *convergence* of a sequence of non-empty compact subsets of \mathbb{R}^2 to a non-empty compact subset of \mathbb{R}^2 in *Hausdorff distance*. If a sequence of non-empty compact convex subsets of \mathbb{R}^2 converges in Hausdorff distance, then its limit is also convex. We need to extend the notion of convergence in Hausdorff distance by allowing the limit object to be an unbounded closed subset of \mathbb{R}^2 while assuming convexity of the members of the sequence.

A pair $(p, r) \in \mathbb{R}^2 \times \mathbb{R}_+$ is an *anchor* for a sequence $(X^n)_{n=1}^\infty$ of non-empty compact convex subsets of \mathbb{R}^2 if $\text{dist}(p, X^n) \leq r$ for every $n \in \mathbb{N}$. A sequence of non-empty compact convex subsets of \mathbb{R}^2 is *anchored* if it has an anchor. We say that an anchored sequence $(X^n)_{n=1}^\infty$ of non-empty compact convex subsets of \mathbb{R}^2 *converges* to a set $X^* \subseteq \mathbb{R}^2$ (and write $X^n \rightarrow X^*$), and we call X^* the *limit* of $(X^n)_{n=1}^\infty$, if for every anchor (p, r) for it, the sequence $(X^n \cap \text{ball}(p, r))_{n=1}^\infty$ converges to $X^* \cap \text{ball}(p, r)$ in Hausdorff distance. Since the latter limit is unique, so is the limit $X^* = \bigcup_{(p,r)} (X^* \cap \text{ball}(p, r))$, where the union is taken over all anchors (p, r) for $(X^n)_{n=1}^\infty$. It is easy to see that the limit X^* is a closed convex set.

The following lemmas assert basic properties of this extended notion of convergence. See the full version of the paper for the proofs that are missing from the current version.

► **Lemma 3.** *If $(X^n)_{n=1}^\infty$ is a sequence of non-empty compact convex subsets of \mathbb{R}^2 with anchor (p, r) that converges to a set $X^* \subseteq \mathbb{R}^2$ in Hausdorff distance, then the sequence $(X^n \cap \text{ball}(p, r))_{n=1}^\infty$ converges to $X^* \cap \text{ball}(p, r)$ in Hausdorff distance.*

► **Lemma 4.** *Every anchored sequence of non-empty compact convex subsets of \mathbb{R}^2 has a convergent subsequence.*

► **Lemma 5.** *Let A be a convex disk and $\mathcal{F} = \text{hom } A$ or $\mathcal{F} = \text{sim } A$. Let $(X^n)_{n=1}^\infty$ be a sequence of members of \mathcal{F} that converges to a set $X^* \subseteq \mathbb{R}^2$. Then the sequence $(r(X^n))_{n=1}^\infty$ converges or diverges to ∞ . Furthermore,*

- *if $r(X^n) \rightarrow r^* \in \mathbb{R}$, where $r^* > 0$, then $X^* \in \mathcal{F}$,*
- *if $r(X^n) \rightarrow 0$, then $X^* = \{z^*\}$ for some point $z^* \in \mathbb{R}^2$,*
- *if $r(X^n) \rightarrow \infty$, then X^* is a half-plane or $X^* = \mathbb{R}^2$.*

► **Lemma 6.** *Let A be a convex disk and $\mathcal{F} = \text{hom } A$ or $\mathcal{F} = \text{sim } A$. For every set X^* that is a member of \mathcal{F} or a half-plane, there is a sequence $(X^n)_{n=1}^\infty$ of members of \mathcal{F} that converges to X^* and satisfies $X^n \subset \text{int } X^*$ for every $n \in \mathbb{N}$.*

An interior-realization of a graph $G = (V, E)$ in a family $\bar{\mathcal{F}}$ of subsets of \mathbb{R}^2 is a mapping $\bar{R}: V \rightarrow \bar{\mathcal{F}}$ such that $\text{int } \bar{R}(u) \cap \text{int } \bar{R}(v) \neq \emptyset$ if and only if $uv \in E$. Our main construction in Section 5 is easier to present in terms of interior-realizations rather than realizations, and the following lemma turns an interior-realization into a realization.

► **Lemma 7.** *Let A be a convex disk, $\mathcal{F} = \text{hom } A$ or $\mathcal{F} = \text{sim } A$, and \mathcal{H} be the family of all half-planes. If a graph G has an interior-realization in $\mathcal{F} \cup \mathcal{H}$, then G has a realization in \mathcal{F} .*

Proof. Let $G = (V, E)$, and let \bar{R} be an interior-realization of G in $\mathcal{F} \cup \mathcal{H}$. Let $p_{uv} \in \text{int } \bar{R}(u) \cap \text{int } \bar{R}(v)$ for every edge $uv \in E$. Let mappings $R^n: V \rightarrow \mathcal{F}$ for $n \in \mathbb{N}$ be such that the sequence $(R^n(v))_{n=1}^\infty$ converges to $\bar{R}(v)$ for every $v \in V$ and $R^n(v) \subset \text{int } \bar{R}(v)$ for all $v \in V$ and $n \in \mathbb{N}$; they exist by Lemma 6. It follows that $R^n(u) \cap R^n(v) \neq \emptyset$ implies $\text{int } \bar{R}(u) \cap \text{int } \bar{R}(v) \neq \emptyset$ and thus $uv \in E$, for all $n \in \mathbb{N}$. If $n \in \mathbb{N}$ is sufficiently large that $p_{uv} \in R^n(u) \cap R^n(v)$ for every edge $uv \in E$, then R^n is a realization of G in \mathcal{F} . ◀

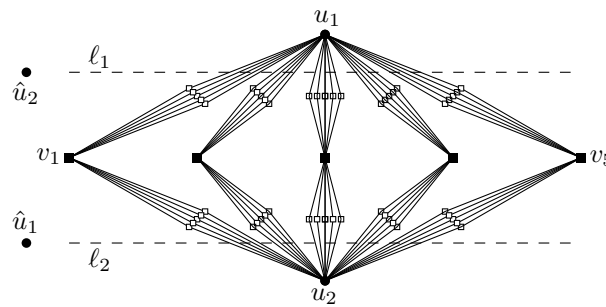
4 Basic configurations

Let $K_{m,n}$ denote the complete bipartite graph with vertices u_1, \dots, u_m on one side and v_1, \dots, v_n on the other side, so that $u_i v_j$ is an edge of $K_{m,n}$ for all $i \in [m]$ and $j \in [n]$. The following lemma is proved by a simple area argument.

► **Lemma 8.** *For every convex disk A and every $\varepsilon > 0$, if n is sufficiently large, then every realization R of $K_{1,n}$ in $\text{sim } A$ satisfies $\min_{i \in [n]} r(R(v_i)) < \varepsilon r(R(u_1))$.*

► **Lemma 9.** *Let A be a convex disk and N be an infinite subset of \mathbb{N} . For every sequence $(R^n)_{n \in N}$ such that R^n is a realization of $K_{2,n}$ in $\text{sim } A$ and $R^n(u_1)$ converges to a convex disk or singleton set U_1^* , the sequence $(R^n(u_2))_{n \in N}$ is anchored and for all of its convergent subsequences, the limit touches U_1^* .*

Proof. When $n \rightarrow \infty$, since $r(R^n(u_1)) \rightarrow r(U_1^*)$, Lemma 8 yields $\text{dist}(R^n(u_1), R^n(u_2)) \leq \min_{i \in [n]} \text{diam } R^n(v_i) = \min_{i \in [n]} r(R^n(v_i)) \cdot \text{diam } A \rightarrow 0$, which implies $\text{dist}(U_1^*, R^n(u_2)) \leq \text{dist}(R^n(u_1), R^n(u_2)) + d_H(R^n(u_1), U_1^*) \rightarrow 0$, and the lemma follows. ◀



■ **Figure 3** The graph L_5 . Here, \hat{u}_1 has an edge to all vertices above the line ℓ_2 , and \hat{u}_2 has an edge to all vertices below ℓ_1 .

► **Construction 10** (the graph L_n). The graph L_n has vertices $u_1, u_2, v_1, \dots, v_n$, vertices w_{ijk} and edges $u_i w_{ijk}, w_{ijk} v_j$ for all $i \in [2]$ and $j, k \in [n]$ (so that $u_i, v_j, w_{ij1}, \dots, w_{ijn}$ form a copy of $K_{2,n}$), and two additional vertices \hat{u}_1, \hat{u}_2 such that \hat{u}_1 has an edge to every vertex except u_2 and \hat{u}_2 has an edge to every vertex except u_1 . See Figure 3.

When considering a specific realization R of L_n (possibly with a superscript), we write V_i, U_i , and \hat{U}_i (with the same superscript) as shorthand for $R(v_i), R(u_i)$, and $R(\hat{u}_i)$, respectively. The following lemma makes essential use of the assumption that A is smooth.

► **Lemma 11.** *For every convex disk A and every $\varepsilon > 0$, if n is sufficiently large, then every realization of L_n in $\text{sim } A$ satisfies $\max_{j \in [n]} r(V_j) \leq \varepsilon \min\{r(U_1), r(U_2)\}$.*

Proof. Suppose for the sake of contradiction that there is $\varepsilon > 0$ such that for every n , there is a realization R^n of L_n in $\text{sim } A$ such that $\max_{j \in [n]} r(V_j^n) > \varepsilon \min\{r(U_1^n), r(U_2^n)\}$. Assume without loss of generality that $r(U_1^n) \leq r(U_2^n)$ for all n . Furthermore, assume that U_1^n is constant (equal to U_1) while the other disks may change size and placement as a function of n .

Suppose there is $\rho > 0$ such that $\min_{i \in [n]} r(V_i^n) \geq \rho$ for every n . Let $k \in \mathbb{N}$. By Lemma 9, we can pass to a subsequence of $(R^n)_{n=k}^\infty$ in which $V_i^n \rightarrow V_i^*$ and V_i^* touches U_1 for every $i \in [k]$. At least $k - 2$ of these limits, say V_1^*, \dots, V_{k-2}^* , are not half-planes. Along with U_1 , they form a realization of $K_{1,k-2}$ in $\text{sim } A$. When k is sufficiently large, Lemma 8 yields $\min_{i \in [k-2]} r(V_i^*) < \rho$. This contradiction shows that $\min_{i \in [n]} r(V_i^n) \rightarrow 0$ as $n \rightarrow \infty$.

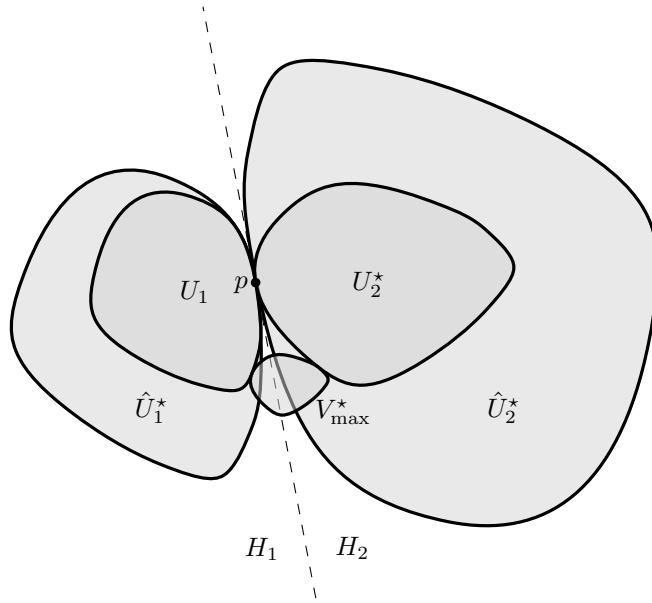
For each n , let V_{\min}^n and V_{\max}^n be disks among V_1^n, \dots, V_n^n with minimum and maximum radii, respectively, so that $r(V_{\max}^n) > \varepsilon r(U_1)$ and $r(V_{\min}^n) \rightarrow 0$ as $n \rightarrow \infty$. See Figure 4. Considering $n \rightarrow \infty$ and passing to a subsequence, by Lemmas 5 and 9, we can assume that

- V_{\min}^n converges to a singleton set $\{p\}$, where $p \in \partial U_1$,
- U_2^n converges to a member of $\text{sim } A$ or half-plane U_2^* that touches U_1 at p ,
- \hat{U}_1^n converges to a limit \hat{U}_1^* that touches U_2^* at p , as $p \in \hat{U}_1^*$ and $\text{int}(\hat{U}_1^* \cap U_2^*) = \emptyset$,
- \hat{U}_2^n converges to a limit \hat{U}_2^* that touches U_1 at p , as $p \in \hat{U}_2^*$ and $\text{int}(U_1 \cap \hat{U}_2^*) = \emptyset$,
- V_{\max}^n converges to a member of $\text{sim } A$ or half-plane V_{\max}^* that touches both U_1 and U_2^* .

It follows that the unique line tangent to both U_1 and U_2^* at p splits the plane into two half-planes H_1 and H_2 such that $U_1, \hat{U}_1^* \subseteq H_1$ and $U_2^*, \hat{U}_2^* \subseteq H_2$.

Suppose that at least one of U_2^*, V_{\max}^* is a member of $\text{sim } A$. By Lemma 8, there are disks W_1^n and W_2^n (members of $\text{sim } A$) such that

- W_1^n intersects V_{\max}^n, U_1 , and \hat{U}_2^n ,
- W_2^n intersects V_{\max}^n, U_2^n , and \hat{U}_1^n ,
- $r(W_1^n) \rightarrow 0$ and $r(W_2^n) \rightarrow 0$ as $n \rightarrow \infty$.



■ **Figure 4** Situation from the proof of Lemma 11.

Considering $n \rightarrow \infty$ and passing to a subsequence, we can assume that $W_1^n \rightarrow \{q_1\}$ and $W_2^n \rightarrow \{q_2\}$, where $q_1 \in V_{\max}^* \cap U_1 \cap \hat{U}_2^*$ and $q_2 \in V_{\max}^* \cap \hat{U}_1^* \cap U_2^*$. It follows that V_{\max}^* touches U_1 at q_1 and U_2^* at q_2 , whereas both q_1 and q_2 lie on the boundary line between H_1 and H_2 . This is possible only when $V_{\max}^* = \{q_1\} = \{q_2\}$, which is a contradiction.

Now, suppose that both U_2^* and V_{\max}^* are half-planes (in particular $U_2^* = H_2$). It follows that they are disjoint half-planes (as they must have disjoint interiors), while $\hat{U}_2^* \subseteq H_2 = U_2^*$, so V_{\max}^* and \hat{U}_2^* are disjoint, which is again a contradiction. ◀

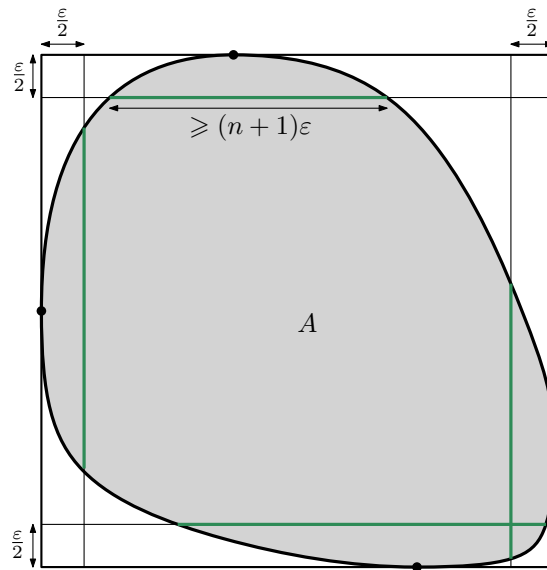
► **Lemma 12.** *Let A be a convex disk and N be an infinite subset of \mathbb{N} . For each $n \in N$, let L'_n be a graph which contains, as induced subgraphs, L_n and a fixed connected graph H containing v_1 such that u_1 and u_2 have no edges to any vertex of H . Let $(R^n)_{n \in N}$ be a sequence such that R^n is a realization of L'_n in $\text{sim } A$ for $n \in N$ and V_1^n converges to a convex disk V_1^* . Then $(R^n)_{n \in N}$ has a subsequence in which*

- U_1^n and U_2^n converge to disjoint half-planes U_1^* and U_2^* ,
- \hat{U}_1^n and \hat{U}_2^n converge to limits that touch U_2^* and U_1^* , respectively,
- for every vertex w of H , $R^n(w)$ converges to a convex disk or singleton set.

Proof sketch. By Lemma 9, the sequences $(U_1^n)_{n \in N}$ and $(U_2^n)_{n \in N}$ are anchored, and so are the sequences $(\hat{U}_1^n)_{n \in N}$ and $(\hat{U}_2^n)_{n \in N}$, so we can pass to a subsequence in which they converge to limits U_1^* , U_2^* , \hat{U}_1^* , and \hat{U}_2^* , respectively. Moreover, by Lemma 9, U_1^* touches V_1^* and \hat{U}_2^* at a common point, and U_2^* touches V_1^* and \hat{U}_1^* at a common point. By Lemma 11, $r(U_1^n) \rightarrow \infty$ and $r(U_2^n) \rightarrow \infty$, so U_1^* and U_2^* are disjoint half-planes. Simple induction shows that we can further pass to a subsequence in which $R^n(w)$ converges to a convex disk or singleton set for every vertex w of H . ◀

5 Main construction

An n -chain aligned to parallel lines ℓ_1, ℓ_2 is an n -tuple A_1, \dots, A_n of convex disks all touching ℓ_1 and ℓ_2 and such that $A_i \cap A_{i+1} \neq \emptyset$ for all $i \in [n-1]$. The *length* of such an n -chain is the length of the orthogonal projection of $A_1 \cup \dots \cup A_n$ on ℓ_1 (or ℓ_2) divided by $\text{dist}(\ell_1, \ell_2)$.



■ **Figure 5** Lemma 13 asserts that for every $n \in \mathbb{N}$, if $\varepsilon > 0$ is sufficiently small, then the lengths of the four green segments are at least $(n + 1)\varepsilon$.

Such an n -chain is *strict* if $\text{int}(A_i \cap A_{i+1}) \neq \emptyset$ for all $i \in [n - 1]$. A *horizontal* or *vertical* n -chain is an n -chain aligned to horizontal or vertical lines, respectively. Before using the n -chains to construct the key graph of our proof, we need the following lemma, which relies on the assumption that A is smooth; see Figure 5 for an illustration.

► **Lemma 13.** *For every convex disk A with bounding box $[0, 1]^2$ and every $n \in \mathbb{N}$, there is $\varepsilon_0 > 0$ such that for every $\varepsilon \in (0, \varepsilon_0)$, the lengths of the four segments $A \cap (\mathbb{R} \times \{\frac{\varepsilon}{2}\})$, $A \cap (\mathbb{R} \times \{1 - \frac{\varepsilon}{2}\})$, $A \cap (\{\frac{\varepsilon}{2}\} \times \mathbb{R})$, and $A \cap (\{1 - \frac{\varepsilon}{2}\} \times \mathbb{R})$ are at least $(n + 1)\varepsilon$.*

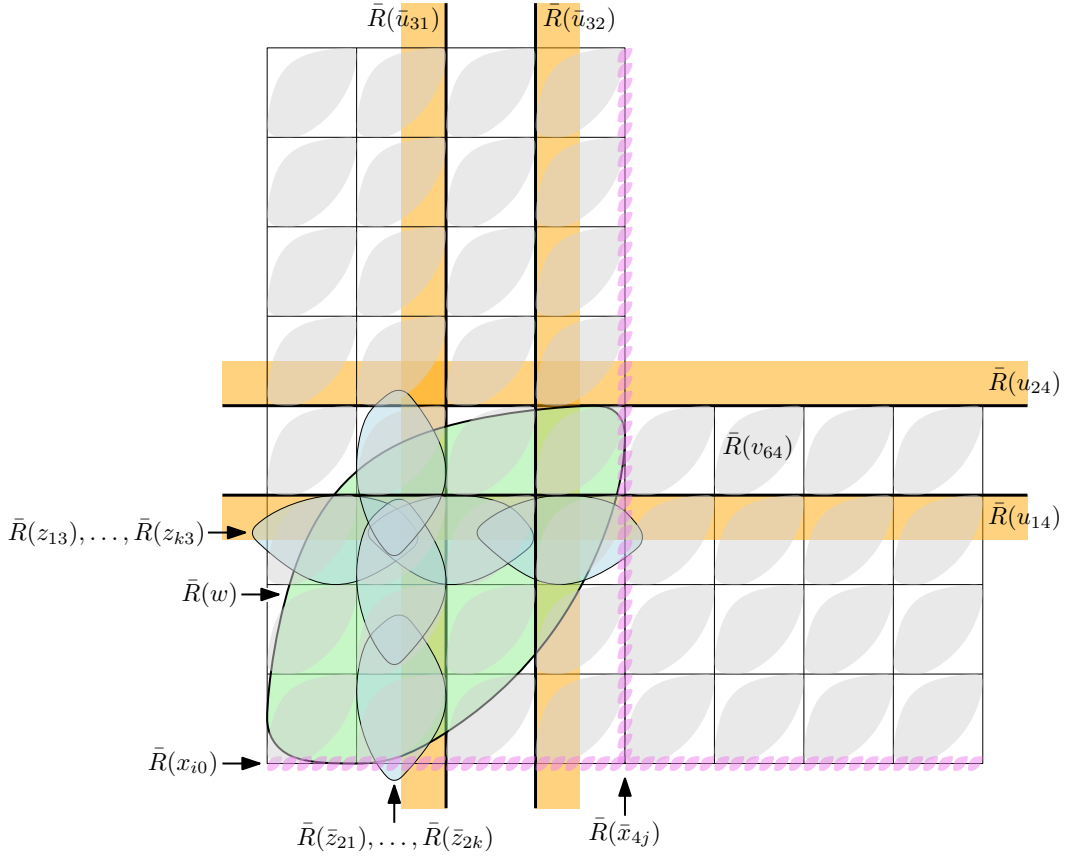
For an illustration of the following construction, see Figure 6.

► **Construction 14** (the graph $G_{mn}^{A, \mathcal{F}}$). Let A be a convex disk with bounding box $[0, 1]^2$. Let $\mathcal{F} = \text{hom } A$ or $\mathcal{F} = \text{sim } A$. Let $m, n \in \mathbb{N}$ with $m \leq n$. Let $k \in \mathbb{N}$ be minimal such that there exist a strict horizontal k -chain and a strict vertical k -chain in \mathcal{F} of length greater than m . Let $\varepsilon > 0$ be as in Lemma 13 for A and n . The graph $G_{mn}^{A, \mathcal{F}}$ has the following vertices and the following interior-realization \bar{R} by members of \mathcal{F} and half-planes:

- $\bar{R}(v_{ij}) = \frac{1}{m}A + (\frac{i-1}{m}, \frac{j-1}{m})$ for $(i, j) \in ([n] \times [m]) \cup ([m] \times [n])$,
- $\bar{R}(u_{1j}) = \mathbb{R} \times (-\infty, \frac{j-1}{m}]$ for $j = 1, \dots, m+1$ and $\bar{R}(u_{2j}) = \mathbb{R} \times [\frac{j}{m}, +\infty)$ for $j = 0, \dots, m$,
- $\bar{R}(\bar{u}_{i1}) = (-\infty, \frac{i-1}{m}] \times \mathbb{R}$ for $i = 1, \dots, m+1$ and $\bar{R}(\bar{u}_{i2}) = [\frac{i}{m}, +\infty) \times \mathbb{R}$ for $i = 0, \dots, m$,
- $\bar{R}(z_{1j}), \dots, \bar{R}(z_{kj})$ that form a strict horizontal k -chain in \mathcal{F} with bounding box $[-\delta, 1 + \delta] \times [\frac{j-1}{m}, \frac{j}{m}]$ for $j = 1, \dots, m$ and some sufficiently small $\delta > 0$,
- $\bar{R}(\bar{z}_{i1}), \dots, \bar{R}(\bar{z}_{ik})$ that form a strict vertical k -chain in \mathcal{F} with bounding box $[\frac{i-1}{m}, \frac{i}{m}] \times [-\delta, 1 + \delta]$ for $i = 1, \dots, m$ and some sufficiently small $\delta > 0$,
- $\bar{R}(w) = A$,
- $\bar{R}(x_{ij}) = \frac{\varepsilon}{m}A + (\frac{\varepsilon i}{m}, \frac{j}{m} - \frac{\varepsilon}{2m})$ for $i = 0, \dots, \lceil \frac{n}{\varepsilon} \rceil - 1$ and $j = 0, \dots, m$,
- $\bar{R}(\bar{x}_{ij}) = \frac{\varepsilon}{m}A + (\frac{\varepsilon}{m} - \frac{\varepsilon j}{2m}, \frac{\varepsilon j}{m})$ for $i = 0, \dots, m$ and $j = 0, \dots, \lceil \frac{n}{\varepsilon} \rceil - 1$.

By Lemma 7, $G_{mn}^{A, \mathcal{F}}$ has a realization in \mathcal{F} . When considering a specific realization R of $G_{mn}^{A, \mathcal{F}}$ (possibly with a superscript), we write V_{ij} , U_{ij} , \bar{U}_{ij} , Z_{ij} , \bar{Z}_{ij} , and W (with the same superscript) as shorthand for $R(v_{ij})$, $R(u_{ij})$, $R(\bar{u}_{ij})$, $R(z_{ij})$, $R(\bar{z}_{ij})$, and $R(w)$, respectively.

For $m \in \mathbb{N}$ and $i, j \in [m]$, let $S_{ij}^m = [\frac{i-1}{m}, \frac{i}{m}] \times [\frac{j-1}{m}, \frac{j}{m}]$. The following lemma asserts basic properties of Construction 14.

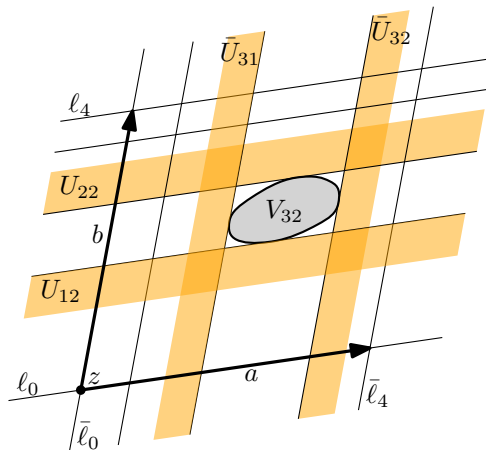


■ **Figure 6** The interior-realization of the graph $G_{48}^{A, \text{sim}}$. The figure is not to scale; in reality, the pink disks $\bar{R}(x_{ij})$ and $\bar{R}(\bar{x}_{ij})$ would be much smaller (and thus more numerous).

► **Lemma 15.** Let A, \mathcal{F}, m, n, k be as in Construction 14. The following hold for $G_{mn}^{A, \mathcal{F}}$:

1. For every $j \in [m]$, there is an induced subgraph isomorphic to L_n in which the vertices $u_{1j}, u_{2j}, u_{1(j+1)}, u_{2(j-1)}$, and v_{1j}, \dots, v_{nj} play the roles of $u_1, u_2, \hat{u}_1, \hat{u}_2$, and v_1, \dots, v_n , respectively; for every $i \in [m]$, there is an induced subgraph isomorphic to L_n in which the vertices $\bar{u}_{i1}, \bar{u}_{i2}, \bar{u}_{(i+1)1}, \bar{u}_{(i-1)2}$, and v_{i1}, \dots, v_{in} play the roles of $u_1, u_2, \hat{u}_1, \hat{u}_2$, and v_1, \dots, v_n , respectively.
2. For every $j \in [m]$, the subgraph induced on $v_{1j}, \dots, v_{mj}, z_{1j}, \dots, z_{kj}$ is connected and contains a path $z_{1j} \cdots z_{kj}$; for every $i \in [m]$, the subgraph induced on $v_{i1}, \dots, v_{im}, \bar{z}_{i1}, \dots, \bar{z}_{ik}$ is connected and contains a path $\bar{z}_{i1} \cdots \bar{z}_{ik}$.
3. The vertices z_{11}, \dots, z_{1m} are adjacent to \bar{u}_{11} , the vertices z_{k1}, \dots, z_{km} are adjacent to \bar{u}_{m2} , the vertices $\bar{z}_{11}, \dots, \bar{z}_{m1}$ are adjacent to u_{11} , and the vertices $\bar{z}_{1k}, \dots, \bar{z}_{mk}$ are adjacent to u_{2m} .
4. The vertex w is adjacent to at least one of z_{1j}, \dots, z_{kj} for every $j \in [m]$ and at least one of $\bar{z}_{i1}, \dots, \bar{z}_{ik}$ for every $i \in [m]$; for every $u \in \{u_{11}, u_{2j}, \bar{u}_{11}, \bar{u}_{i2}\}$, there is an induced subgraph isomorphic to $K_{2,n}$ in which the vertices u and w form one of the parts of the bipartition.
5. For all $i, j \in [m]$, if $S_{ij}^m \subseteq A$, then $v_{ij}w$ is an edge, and if $v_{ij}w$ is an edge, then $S_{ij}^m \cap A \neq \emptyset$.

An m -grid is a collection of two $(m+1)$ -tuples of parallel lines $\ell_0, \ell_1, \dots, \ell_m$ and $\bar{\ell}_0, \bar{\ell}_1, \dots, \bar{\ell}_m$ that are images of horizontal lines at coordinates $0 = y_0 < y_1 < \dots < y_m = 1$ and $m+1$ vertical lines at coordinates $0 = x_0 < x_1 < \dots < x_m = 1$, respectively, under an



■ **Figure 7** An example of a 4-grid with aligned disks and half-planes.

affine transformation $f: \mathbb{R}^2 \ni (x, y) \mapsto z + xa + yb \in \mathbb{R}^2$ for some point $z \in \mathbb{R}^2$ called the *origin* of the m -grid and some linearly independent vectors $a, b \in \mathbb{R}^2$ that form the *basis* of the m -grid; see Figure 7. The differences $x_1 - x_0, \dots, x_m - x_{m-1}$ and $y_1 - y_0, \dots, y_m - y_{m-1}$ are the *horizontal* and *vertical distances* of the m -grid, respectively. A configuration of convex disks V_{ij} with $i, j \in [m]$ and half-planes $U_{11}, U_{21}, \dots, U_{1m}, U_{2m}, \bar{U}_{11}, \bar{U}_{12}, \dots, \bar{U}_{m1}, \bar{U}_{m2}$ is *aligned* to such an m -grid if the following holds:

- $U_{1j} = f(\mathbb{R} \times (-\infty, y_{j-1}])$ and $U_{2j} = f(\mathbb{R} \times [y_j, +\infty))$ for $j \in [m]$,
- $\bar{U}_{i1} = f((-\infty, x_{i-1}] \times \mathbb{R})$ and $\bar{U}_{i2} = f([x_i, +\infty) \times \mathbb{R})$ for $i \in [m]$,
- V_{ij} touches the four half-planes $U_{1j}, U_{2j}, \bar{U}_{i1}, \bar{U}_{i2}$ for $i, j \in [m]$.

The following lemma is at the heart of our argument. Among other things, it asserts that in realizations of the graph $G_{mn}^{A, \mathcal{F}}$, the disks V_{ij}^n , for $i, j \in [m]$, are indeed forced to form an aligned m -grid as $n \rightarrow \infty$. This will be the foundation for the proofs of Theorems 1 and 2.

► **Lemma 16.** *Let A and B be convex disks such that A has bounding box $[0, 1]^2$. Let $\mathcal{F} = \text{hom } A$ or $\mathcal{F} = \text{sim } A$. Let $m \in \mathbb{N}$. Let $k \in \mathbb{N}$ be minimal such that there exist a strict horizontal k -chain and a strict vertical k -chain in \mathcal{F} of length greater than m . Every sequence $(R^n)_{n=m}^\infty$ such that R^n is a realization of $G_{mn}^{A, \mathcal{F}}$ in $\text{sim } B$ and V_{11}^n is constant has a subsequence in which the disks V_{ij}^n with $i, j \in [m]$, U_{1j}^n, U_{2j}^n with $j \in [m]$, and $\bar{U}_{i1}^n, \bar{U}_{i2}^n$ with $i \in [m]$ converge to convex disks V_{ij}^* and half-planes U_{1j}^*, U_{2j}^* and $\bar{U}_{i1}^*, \bar{U}_{i2}^*$, respectively, that are aligned to an m -grid, and the disks $Z_{1j}^n, \dots, Z_{kj}^n$ with $j \in [m]$, $\bar{Z}_{i1}^n, \dots, \bar{Z}_{ik}^n$ with $i \in [m]$, and W^n converge to convex disks $Z_{1j}^*, \dots, Z_{kj}^*$, $\bar{Z}_{i1}^*, \dots, \bar{Z}_{ik}^*$, and W^* , respectively, where W^* touches $U_{11}^*, U_{2m}^*, \bar{U}_{11}^*, \bar{U}_{m2}^*$.*

Proof. Let $(R^n)_{n=m}^\infty$ be a sequence of realizations R^n of $G_{mn}^{A, \mathcal{F}}$ in $\text{sim } B$ such that V_{11}^n is constant. By Lemma 15 (1 and 2), we can apply Lemma 12 repeatedly as follows, in order:

- with vertices $u_{11}, u_{21}, u_{12}, u_{20}$, and v_{11}, \dots, v_{n1} playing the roles of $u_1, u_2, \hat{u}_1, \hat{u}_2$, and v_1, \dots, v_n (respectively) in L_n , and with the graph H formed by $v_{11}, \dots, v_{m1}, z_{11}, \dots, z_{k1}$,
- for each $i \in [m]$, with vertices $\bar{u}_{i1}, \bar{u}_{i2}, \bar{u}_{(i+1)1}, \bar{u}_{(i-1)2}$, and v_{i1}, \dots, v_{in} playing the roles of $u_1, u_2, \hat{u}_1, \hat{u}_2$, and v_1, \dots, v_n (respectively) in L_n , and with the graph H formed by $v_{i1}, \dots, v_{im}, \bar{z}_{i1}, \dots, \bar{z}_{ik}$,
- for each $j \in [m] \setminus \{1\}$, with vertices $u_{1j}, u_{2j}, u_{1(j+1)}, u_{2(j-1)}$, and v_{1j}, \dots, v_{nj} playing the roles of $u_1, u_2, \hat{u}_1, \hat{u}_2$, and v_1, \dots, v_n (respectively) in L_n , and with the graph H formed by $v_{1j}, \dots, v_{mj}, z_{1j}, \dots, z_{kj}$.

This yields a subsequence in which the disks V_{ij}^n with $i, j \in [m]$, $U_{1j}^n, U_{2j}^n, Z_{1j}^n, \dots, Z_{kj}^n$ with $j \in [m]$, and $\bar{U}_{i1}^n, \bar{U}_{i2}^n, \bar{Z}_{i1}^n, \dots, \bar{Z}_{ik}^n$ with $i \in [m]$ converge to limits V_{ij}^* , U_{1j}^* , U_{2j}^* , Z_{1j}^* , \dots , Z_{kj}^* , and \bar{U}_{i1}^* , \bar{U}_{i2}^* , \bar{Z}_{i1}^* , \dots , \bar{Z}_{ik}^* , respectively, where

- V_{ij}^* is a convex disk for $i, j \in [m]$,
- U_{1j}^* and U_{2j}^* are disjoint half-planes for $j \in [m]$,
- $U_{1(j+1)}^*$ and U_{2j}^* touch and therefore share the boundary line, for $j \in [m-1]$,
- \bar{U}_{i1}^* and \bar{U}_{i2}^* are disjoint half-planes for $i \in [m]$,
- $\bar{U}_{(i+1)1}^*$ and \bar{U}_{i2}^* touch and therefore share the boundary line, for $i \in [m-1]$.

Let

- $\ell_0 = \partial U_{11}^*$, $\ell_j = \partial U_{1(j+1)}^* = \partial U_{2j}^*$ for $j \in [m-1]$, and $\ell_m = \partial U_{2m}^*$,
- $\bar{\ell}_0 = \partial \bar{U}_{11}^*$, $\bar{\ell}_i = \partial \bar{U}_{(i+1)1}^* = \partial \bar{U}_{i2}^*$ for $i \in [m-1]$, and $\bar{\ell}_m = \partial \bar{U}_{m2}^*$.

It follows that the lines ℓ_0, \dots, ℓ_m are parallel and occur in this order, and so do the lines $\bar{\ell}_0, \dots, \bar{\ell}_m$. Consequently, they form an m -grid, the origin of which is the intersection point of ℓ_0 and $\bar{\ell}_0$, and the basis vectors of which are the vectors from the origin to the intersection point of ℓ_0 and $\bar{\ell}_m$ and from the origin to the intersection point of ℓ_m and $\bar{\ell}_0$. Furthermore, Lemma 9 implies that V_{ij}^* touches U_{1j}^* , U_{2j}^* , \bar{U}_{i1}^* , \bar{U}_{i2}^* for $i, j \in [m]$. This shows that the disks V_{ij}^* with $i, j \in [m]$, U_{1j}^* , U_{2j}^* with $j \in [m]$, and \bar{U}_{i1}^* , \bar{U}_{i2}^* with $i \in [m]$ are aligned to the m -grid.

By Lemma 15 (4), for every n , the vertex w has an edge to at least one of the vertices z_{ij} in $G_{mn}^{A, \mathcal{F}}$ and therefore $W^n \cap Z_{ij}^n \neq \emptyset$. It follows that the sequence $(W^n)_{n \in \mathbb{N}}$ (where N comprises the indices of the considered subsequence) is anchored and therefore, passing yet to a subsequence, W^n converges to a limit W^* . Moreover, by Lemma 15 (4) and Lemma 9, W^* touches U_{11}^* , U_{2m}^* , \bar{U}_{11}^* , \bar{U}_{m2}^* ; in particular, it is a convex disk. ◀

6 Classifying intersection graphs of homothets

The proof of Theorem 1 is based on the following lemma.

► **Lemma 17.** *Let A and B be convex disks such that A has bounding box $[0, 1]^2$. If for all $m, n \in \mathbb{N}$ with $m \leq n$, there is a realization of $G_{mn}^{A, \text{hom } A}$ in $\text{hom } B$, then there is an affine transformation that maps A to B .*

Before proving Lemma 17, let us see how Theorem 1 follows.

Proof of Theorem 1. Let A and B be convex disks. As we already observed, if A and B are affine equivalent, then $G^{\text{hom}}(A) = G^{\text{hom}}(B)$, because the affine transformation that maps A to B transforms every realization in $\text{hom } A$ to a realization of the same graph in $\text{hom } B$, and vice versa. Now, suppose $G^{\text{hom}}(A) = G^{\text{hom}}(B)$. We can assume without loss of generality that the bounding box of A is $[0, 1]^2$, otherwise we can apply an affine transformation to A to obtain a convex disk with that bounding box; as observed before, such a transformation does not change the intersection graphs realized in $\text{hom } A$. Now, since $G_{mn}^{A, \text{hom } A} \in G^{\text{hom}}(B)$ for all $m, n \in \mathbb{N}$ with $m \leq n$, the lemma asserts that A and B are affine equivalent.

The last statement of the theorem asserts that when A and B are not affine equivalent, then the classes of intersection graphs are not nested. Under this assumption, the lemma yields $G_{mn}^{A, \text{hom } A} \notin G^{\text{hom}}(B)$ for some m and n . Using the lemma with A and B interchanged, we also have $G_{mn}^{B, \text{hom } B} \notin G^{\text{hom}}(A)$ for m and n . Therefore, the graph classes are not nested. ◀

Proof of Lemma 17. For all $m, n \in \mathbb{N}$ with $m \leq n$, let R^{mn} be a realization of $G_{mn}^{A, \text{hom } A}$ in $\text{hom } B$. We first fix m and consider the sequence of realizations $(R^{mn})_{n=m}^{\infty}$. Without loss of generality, V_{11}^{mn} is constant in this sequence. By Lemma 16, we can pass to a subsequence such that the disks V_{ij}^{mn} with $i, j \in [m]$, U_{1j}^{mn}, U_{2j}^{mn} with $j \in [m]$, and $\bar{U}_{i1}^{mn}, \bar{U}_{i2}^{mn}$ with $i \in [m]$

converge to disks $V_{ij}^{m^*} \in \text{hom } B$ and half-planes $U_{1j}^{m^*}, U_{21}^{m^*}$ and $\bar{U}_{i1}^{m^*}, \bar{U}_{i2}^{m^*}$, respectively, that are aligned to an m -grid, and the disks W^{mn} converge to a disk $W^{m^*} \in \text{hom } B$. It follows that all $V_{ij}^{m^*}$ with $i, j \in [m]$ have the same radius, so the horizontal and vertical distances of the m -grid are all equal to $\frac{1}{m}$. Without loss of generality, the origin of the m -grid is $(0, 0)$ and $r(W^{m^*}) = 1$. Let $a^m, b^m \in \mathbb{R}^2$ be the basis vectors of the m -grid, and let $f^m: \mathbb{R}^2 \ni (x, y) \mapsto xa^m + yb^m \in \mathbb{R}^2$. It follows that $V_{ij}^{m^*} \subseteq f^m(S_{ij}^m)$ for $i, j \in [m]$ and $W^{m^*} \subseteq f^m([0, 1]^2)$.

Recall that in Construction 14, the edges between w and the vertices v_{ij} with $i, j \in [m]$ are meant to “encode” the shape of A . The following two claims are implied by the existence and non-existence of these edges in the realizations R^{mn} .

▷ **Claim 17.1.** There is a constant $\eta > 0$ such that $\|a^m\| + \|b^m\| \leq \eta$ for all m .

▷ **Claim 17.2.** For every $\varepsilon > 0$, if m is sufficiently large, then $d_H(W^{m^*}, f^m(A)) \leq \varepsilon$, where d_H denotes the Hausdorff distance.

Since $\|a^m\| + \|b^m\| \leq \eta$ (by Claim 17.1), we can find an infinite set of indices m such that a^m and b^m converge to vectors $a^*, b^* \in \mathbb{R}^2$, respectively, as $m \rightarrow \infty$ over that set of indices. Let $f^*: \mathbb{R}^2 \ni (x, y) \mapsto xa^* + yb^* \in \mathbb{R}^2$. We show that $W^{m^*} \rightarrow f^*(A)$ in Hausdorff distance. To this end, let $\varepsilon > 0$, and let m be sufficiently large that $d_H(W^{m^*}, f^m(A)) \leq \frac{\varepsilon}{2}$ (by Claim 17.2) and $\|a^m - a^*\| + \|b^m - b^*\| \leq \frac{\varepsilon}{2}$. Since $A \subseteq [0, 1]^2$, we have $\text{dist}(f^m((x, y)), f^*((x, y))) = \|(a^m - a^*)x + (b^m - b^*)y\| \leq \|a^m - a^*\| + \|b^m - b^*\| \leq \frac{\varepsilon}{2}$ for every point $(x, y) \in A$, whence it follows that $d_H(f^m(A), f^*(A)) \leq \frac{\varepsilon}{2}$. This yields $d_H(W^{m^*}, f^*(A)) \leq d_H(W^{m^*}, f^m(A)) + d_H(f^m(A), f^*(A)) \leq \varepsilon$. Since $W^{m^*} \rightarrow f^*(A)$, Lemma 5 yields $f^*(A) \in \text{hom } B$, that is, there is a homothetic transformation $h: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ that maps B to $f^*(A)$. We conclude that $h^{-1} \circ f^*$ is an affine transformation that maps A to B . ◀

7 Classifying intersection graphs of similarities

For a convex disk A and $n \in \mathbb{N}$, we define $\sigma_A(n)$ as the maximum length of an n -chain in $\text{sim } A$. The sequence $(\sigma_A(n))_{n=1}^\infty$ is subadditive, that is, $\sigma_A(n_1 + n_2) \leq \sigma_A(n_1) + \sigma_A(n_2)$ for all $n_1, n_2 \in \mathbb{N}$. Indeed, in an $(n_1 + n_2)$ -chain realizing the value $\sigma_A(n_1 + n_2)$, the first n_1 disks form an n_1 -chain of length $x_1 \leq \sigma_A(n_1)$, and the last n_2 disks form an n_2 -chain of length $x_2 \leq \sigma_A(n_2)$, whence it follows that $\sigma_A(n_1 + n_2) \leq x_1 + x_2 \leq \sigma_A(n_1) + \sigma_A(n_2)$. By Fekete’s Subadditive Lemma [12], the limit $\lim_{n \rightarrow \infty} \sigma_A(n)/n$ exists and is equal to $\inf_{n \in \mathbb{N}} \sigma_A(n)/n$. We call this limit the *stretch* of A and denote it by ρ_A .

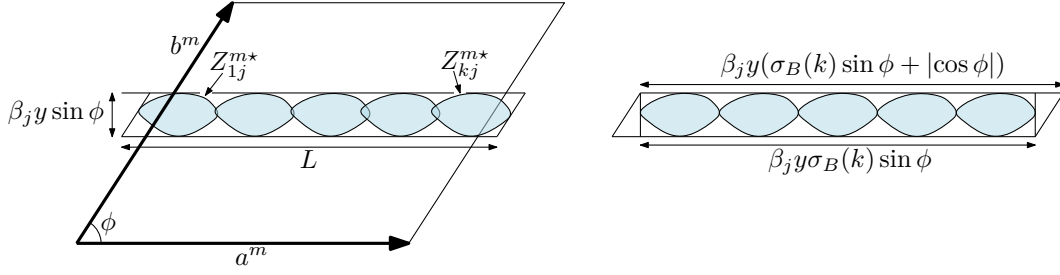
▶ **Lemma 18.** For every $k \in \mathbb{N}$, $\rho_A k \leq \sigma_A(k) \leq \rho_A k + \sigma_A(1)$.

The proof of Theorem 2 is based on the following lemma.

▶ **Lemma 19.** Let A and B be convex disks such that A has bounding box $[0, 1]^2$ and $\rho_A \geq \rho_B$. If for all $m, n \in \mathbb{N}$ with $m \leq n$, there is a realization of $G_{mn}^{A, \text{sim } A}$ in $\text{sim } B$, then $B \in \text{sim}^r A$.

Before proving the lemma, let us see how Theorem 2 follows.

Proof of Theorem 2. Let A and B be convex disks. As we have already observed, if B is similar to A or to A^r , then $G^{\text{sim}}(A) = G^{\text{sim}}(B)$, because the similarity transformation (possibly with reflection) that maps A to B transforms every realization in $\text{sim } A$ to a realization of the same graph in $\text{sim } B$, and vice versa. Now, suppose $G^{\text{sim}}(A) = G^{\text{sim}}(B)$. We can assume without loss of generality that $\rho_A \geq \rho_B$. We can further assume that the bounding box of A is $[0, 1]^2$, otherwise we can rotate, scale, and translate A to obtain a disk



■ **Figure 8** To the left is shown the definition of the length L . To the right is shown a maximum k -chain between two lines of distance $\beta_j y \sin \phi$. It holds that $x = \|a^m\| \leq L \leq \beta_j y (\sigma_B(k) \sin \phi + |\cos \phi|)$.

with this bounding box, and that transformation does not change the intersection graphs realized in $\text{sim } A$. Since $G_{mn}^{A, \text{sim } A} \in G^{\text{sim}}(B)$ for all $m, n \in \mathbb{N}$ with $m \leq n$, we get from the lemma that $B \in \text{sim}^r A$, as claimed. ◀

Proof of Lemma 19. For all $m, n \in \mathbb{N}$ with $m \leq n$, let R^{mn} be a realization of $G_{mn}^{A, \text{sim } A}$ in $\text{sim } B$. We first fix m and consider the sequence of realizations $(R^{mn})_{n=m}^{\infty}$. Without loss of generality, V_{11}^{mn} is constant in this sequence. By Lemma 16, we can pass to a subsequence such that the disks V_{ij}^{mn} with $i, j \in [m]$, U_{1j}^{mn}, U_{2j}^{mn} with $j \in [m]$, and $\bar{U}_{i1}^{mn}, \bar{U}_{i2}^{mn}$ with $i \in [m]$ converge to disks $V_{ij}^{m*} \in \text{hom } B$ and half-planes U_{1j}^{m*}, U_{2j}^{m*} and $\bar{U}_{i1}^{m*}, \bar{U}_{i2}^{m*}$, respectively, that are aligned to an m -grid, the disks Z_{ij}^{mn} and \bar{Z}_{ij}^{mn} converge to disks $Z_{ij}^{m*} \in \text{sim } B$ and $\bar{Z}_{ij}^{m*} \in \text{sim } B$, respectively, and the disks W^{mn} converge to a disk $W^{m*} \in \text{sim } B$ that touches $U_{11}^*, U_{2m}^*, \bar{U}_{11}^*, \bar{U}_{m2}^*$. Without loss of generality, the origin of the m -grid is $(0, 0)$ and $r(W^{m*}) = 1$. Let $a^m, b^m \in \mathbb{R}^2$ be the basis vectors of the m -grid, and let $f^m: \mathbb{R}^2 \ni (x, y) \mapsto xa^m + yb^m \in \mathbb{R}^2$. Let $\alpha_1^m, \dots, \alpha_m^m$ and $\beta_1^m, \dots, \beta_m^m$ be the horizontal and vertical distances of the m -grid, respectively, where $\sum_{i=1}^m \alpha_i = \sum_{j=1}^m \beta_j = 1$.

▷ **Claim 19.1.** There is a constant $c > 0$ (which depends only on B) such that for every m , if $x = \|a^m\|$, $y = \|b^m\|$, and $\phi \in (0, \pi)$ is the angle between a^m and b^m , then

$$\begin{aligned} \frac{x}{y} &\leq 1 + \frac{c}{m}, & \frac{y}{x} &\leq 1 + \frac{c}{m}, & \sin \phi &\leq 1 - \frac{c}{m}, \\ \frac{i}{m} - \frac{2c}{m} &< \alpha_1 + \dots + \alpha_i < \frac{i}{m} + \frac{2c}{m} & \text{for every } i &\in [m-1], \\ \frac{j}{m} - \frac{2c}{m} &< \beta_1 + \dots + \beta_j < \frac{j}{m} + \frac{2c}{m} & \text{for every } j &\in [m-1]. \end{aligned}$$

The following claims are analogous to Claims 17.1 and 17.2.

▷ **Claim 19.2.** There is a constant $\eta > 0$ such that $\|a^m\| + \|b^m\| \leq \eta$ for all m .

▷ **Claim 19.3.** For every $\varepsilon > 0$, if m is sufficiently large, then $d_H(W^{m*}, f^m(A)) \leq \varepsilon$.

Since $\|a^m\| + \|b^m\| \leq \eta$ (by Claim 19.2), we can find an infinite set of indices m such that a^m and b^m converge to vectors $a^*, b^* \in \mathbb{R}^2$, respectively, as $m \rightarrow \infty$ over that set of indices. Let $f^*: \mathbb{R}^2 \ni (x, y) \mapsto xa^* + yb^* \in \mathbb{R}^2$. It follows from Claim 19.1 that $\|a^*\| = \|b^*\|$ and the vectors a^* and b^* are orthogonal, so f^* is a similarity transformation or similarity transformation with reflection. The same argument as in the proof of Lemma 17, using Claim 19.3, shows that $W^{m*} \rightarrow f^*(A)$ in Hausdorff distance. Since $W^{m*} \rightarrow f^*(A)$, Lemma 5 yields $f^*(A) \in \text{sim } B$, and we have $f^*(A) \in \text{sim}^r A$, so $B \in \text{sim}^r A$. ◀

8 Open problems

For our row construction to work, we need the disks to be smooth. In particular, Lemmas 5, 11, and 13 do not hold if A is not smooth. Distinguishing the classes of intersection graphs for non-smooth convex disks remains an interesting question.

One may also consider the even larger class $G^{\text{aff}}(A)$ of intersection graphs of disks that are affine equivalent to a convex disk A and ask when $G^{\text{aff}}(A) = G^{\text{aff}}(B)$ for two convex disks A and B . Other classes that have so far not been investigated are the contact and intersection graphs that can be obtained from rotated translations of a disk A , i.e., with no scaling allowed.

References

- 1 Anders Aamand, Mikkel Abrahamsen, Jakob Bæk Tejs Knudsen, and Peter Michael Reichstein Rasmussen. Classifying convex bodies by their contact and intersection graphs. In *37th International Symposium on Computational Geometry (SoCG 2021)*, pages 3:1–3:16, 2021. doi:10.4230/LIPIcs.SocG.2021.3.
- 2 Jochen Alber and Jiří Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *Journal of Algorithms*, 52(2):134–151, 2004. doi:10.1016/j.jalgor.2003.10.001.
- 3 Marthe Bonamy, Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Panos Giannopoulos, Eun Jung Kim, Paweł Rzażewski, Florian Sikora, and Stéphan Thomassé. EPTAS and subexponential algorithm for maximum clique on disk and unit ball graphs. *Journal of the ACM*, 68(2):9:1–9:38, 2021. doi:10.1145/3433160.
- 4 Édouard Bonnet, Nicolas Grelier, and Nicolas Miltzow. Maximum clique in disk-like intersection graphs. In *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, pages 17:1–17:18, 2020. doi:10.4230/LIPIcs.FSTTCS.2020.17.
- 5 Sergio Cabello and Miha Ježič. Refining the hierarchies of classes of geometric intersection graphs. *Electronic Journal of Combinatorics*, 24(1):P1.33, 19 pp., 2017. doi:10.37236/6040.
- 6 Marco Caoduro and András Sebő. Packing, hitting and coloring squares, 2022. arXiv:2206.02185.
- 7 Ioannis Caragiannis, Aleksei V. Fishkin, Christos Kaklamanis, and Evi Papaioannou. A tight bound for online colouring of disk graphs. *Theoretical Computer Science*, 384(2–3):152–160, 2007. doi:10.1016/j.tcs.2007.04.025.
- 8 Jean Cardinal, Stefan Felsner, Tillmann Miltzow, Casey Tompkins, and Birgit Vogtenhuber. Intersection graphs of rays and grounded segments. *Journal of Graph Algorithms and Applications*, 22(2):273–295, 2018. doi:10.7155/jgaa.00470.
- 9 Steven Chaplick, Stefan Felsner, Udo Hoffmann, and Veit Wiechert. Grid intersection graphs and order dimension. *Order*, 35(2):363–391, 2018. doi:10.1007/s11083-017-9437-0.
- 10 Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Mathematics*, 86(1–3):165–177, 1990. doi:10.1016/0012-365X(90)90358-0.
- 11 Adrian Dumitrescu and Minghui Jiang. Piercing translates and homothets of a convex body. *Algorithmica*, 61:94–115, 2011. doi:10.1007/s00453-010-9410-4.
- 12 Mihály Fekete. Über die Verteilung der Wurzeln bei gewissen algebraischen Gleichungen mit ganzzahligen Koeffizienten. *Mathematische Zeitschrift*, 17:228–249, 1923. doi:10.1007/BF01504345.
- 13 Matt Gibson and Imran A. Pirwani. Algorithms for dominating set in disk graphs: breaking the $\log n$ barrier. In *18th Annual European Symposium on Algorithms (ESA 2010)*, pages 243–254, 2010. doi:10.1007/978-3-642-15775-2_21.
- 14 Albert Gräf, Martin Stumpf, and Gerhard Weißensfeld. On coloring unit disk graphs. *Algorithmica*, 20:277–293, 1998. doi:10.1007/PL00009196.

- 15 Svante Janson and Jan Kratochvíl. Thresholds for classes of intersection graphs. *Discrete Mathematics*, 108(1–3):307–326, 1992. doi:10.1016/0012-365X(92)90684-8.
- 16 Haim Kaplan, Alexander Kauer, Katharina Klost, Kristin Knorr, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth. Dynamic connectivity in disk graphs, 2021. arXiv:2106.14935.
- 17 Seog-Jin Kim, Alexandr Kostochka, and Kittikorn Nakprasit. On the chromatic number of intersection graphs of convex sets in the plane. *Electronic Journal of Combinatorics*, 11:R52, 12 pp., 2004. doi:10.37236/1805.
- 18 Seog-Jin Kim, Kittikorn Nakprasit, Michael J. Pelsmajer, and Jozef Skokan. Transversal numbers of translates of a convex body. *Discrete Mathematics*, 306(18):2166–2173, 2006. doi:10.1016/j.disc.2006.05.014.
- 19 Paul Koebe. Kontaktprobleme der konformen Abbildung. *Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse*, 88:141–164, 1936.
- 20 Jan Kratochvíl and Jiří Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory, Series B*, 62(2):289–315, 1994. doi:10.1006/jctb.1994.1071.
- 21 Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114–143, 2013. doi:10.1016/j.jctb.2012.09.004.
- 22 Colin McDiarmid and Tobias Müller. The number of disk graphs. *European Journal of Combinatorics*, 35:413–431, 2014. doi:10.1016/j.ejc.2013.06.037.
- 23 Irina G. Perepelitsa. Bounds on the chromatic number of intersection graphs of sets in the plane. *Discrete Mathematics*, 262(1–3):221–227, 2003. doi:10.1016/S0012-365X(02)00501-0.
- 24 Oded Schramm. Combinatorially prescribed packings and applications to conformal and quasiconformal maps, 2007. arXiv:0709.0710.
- 25 My T. Thai, Ning Zhang, Ravi Tiwari, and Xiaochun Xu. On approximation algorithms of k -connected m -dominating sets in disk graphs. *Theoretical Computer Science*, 385(1–3):49–59, 2007. doi:10.1016/j.tcs.2007.05.025.

Lower Bounds for Intersection Reporting Among Flat Objects

Peyman Afshani 
Aarhus University, Denmark

Pingan Cheng 
Aarhus University, Denmark

Abstract

Recently, Ezra and Sharir [20] showed an $O(n^{3/2+\sigma})$ space and $O(n^{1/2+\sigma})$ query time data structure for ray shooting among triangles in \mathbb{R}^3 . This improves the upper bound given by the classical $S(n)Q(n)^4 = O(n^{4+\sigma})$ space-time tradeoff for the first time in almost 25 years and in fact lies on the tradeoff curve of $S(n)Q(n)^3 = O(n^{3+\sigma})$. However, it seems difficult to apply their techniques beyond this specific space and time combination. This phenomenon appears persistently in almost all recent advances of flat object intersection searching, e.g., line-tetrahedron intersection in \mathbb{R}^4 [19], triangle-triangle intersection in \mathbb{R}^4 [19], or even among flat semialgebraic objects [5].

We give a timely explanation to this phenomenon from a lower bound perspective. We prove that given a set \mathcal{S} of $(d-1)$ -dimensional simplices in \mathbb{R}^d , any data structure that can report all intersections with a query line in small $(n^{o(1)})$ query time must use $\Omega(n^{2(d-1)-o(1)})$ space. This dashes the hope of any significant improvement to the tradeoff curves for small query time and almost matches the classical upper bound. We also obtain an almost matching space lower bound of $\Omega(n^{6-o(1)})$ for triangle-triangle intersection reporting in \mathbb{R}^4 when the query time is small. Along the way, we further develop the previous lower bound techniques by Afshani and Cheng [2, 3].

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Computational Geometry, Intersection Searching, Data Structure Lower Bounds

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.3

Related Version *Full Version:* <https://arxiv.org/abs/2302.11433>

Funding Supported by DFF (Det Frie Forskningsråd) of Danish Council for Independent Research under grant ID DFF-7014-00404.

1 Introduction

Given a set \mathcal{S} of triangles in \mathbb{R}^3 , how to preprocess \mathcal{S} such that given any query ray γ , we can efficiently determine the first triangle intersecting γ or report no such triangle exists? This problem, known as ray shooting, is one of the most important problems in computational geometry with countless papers published over the last three decades [27, 8, 25, 26, 17, 10, 29, 30, 11, 18, 20, 5]. For a comprehensive overview of this problem, we refer the readers to an excellent recent survey [28].

Recently, there have been considerable and significant advances on ray shooting and a number of problems related to intersection searching on the upper bound side. We complement these attempts by giving lower bounds for a number of intersection searching problems; these also settle a recent open question asked by Ezra and Sharir [20].



© Peyman Afshani and Pingan Cheng;

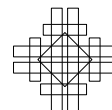
licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 3; pp. 3:1–3:16

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1.1 Background and Previous Results

In geometric intersection searching, the input is a set \mathcal{S} of geometric objects and the goal is to preprocess \mathcal{S} into a data structure such that given a geometric object γ at the query time, one can find all the objects in \mathcal{S} that intersect γ . In the reporting variant of such a query, the output should be the list of all the intersecting objects in \mathcal{S} . Intersection searching is a generalization of range searching, a fundamental and core area of computational geometry [4]. This captures many natural classic problems e.g., simplex range reporting where the inputs are points (0-flats) and the queries are simplices (subsets of d -flats), ray shooting reporting among triangles in \mathbb{R}^3 where the inputs are triangles (subsets of 2-flats) and the queries are rays (subsets of 1-flats) and so on. See [4, 28] for more information.

Without going too much in-depth, it suffices to say that by now, the simplex range searching problem is more or less well-understood. There are classical solutions that offer the space and query time trade-off of $S(n)Q(n) = \tilde{O}(n^d)$ where $S(n)$ and $Q(n)$ are the space and query time of the data structure [15, 23, 12] and there are a number of almost matching lower bounds that show these are essentially tight [1, 13, 16].

However, intersection searching in higher dimensions is less well-understood. The classical technique is to lift the problem to the parametric space of the input or the query, reducing the problem to semialgebraic range searching, a generalized version of simplex range searching, where queries are semialgebraic sets of constant description complexity. In mid-1990s, semialgebraic range searching could only be solved efficiently in four and lower dimensions by classical tools developed for simplex range searching [7], resulting in a space-time trade-off bound of $S(n)Q(n)^4 = O(n^{4+\sigma})$ for line-triangle intersection searching in \mathbb{R}^3 , where $\sigma > 0$ can be any small constant.

Recently, using polynomial techniques [22, 21], several major advances have been made on semialgebraic range searching. For example, near optimal small linear space and fast query data structure were developed [9, 24, 6]. These almost match the newly discovered lower bound bounds [2, 3]. However, these polynomial techniques also have led to significant advances in intersection searching. For ray-triangle intersection reporting in \mathbb{R}^3 , Ezra and Sharir [20] showed that using algebraic techniques, it is possible to build a data structure of space $S(n) = O(n^{3/2+\sigma})$ and query time $Q(n) = O(n^{1/2+\sigma})$ for ray shooting among triangles. The significance of this result is that it improves the upper bound given by the trade-off curve of $S(n)Q(n)^4 = O(n^{4+\sigma})$ for the first time in almost 25 years and in fact it lies on the trade-off curve of $S(n)Q(n)^3 = O(n^{3+\sigma})$. This leads to the following very interesting question asked by Ezra and Sharir. To quote them directly: *“There are several open questions that our work raises. First, can we improve our trade-off for all values of storage, beyond the special values of $O(n^{3/2+\varepsilon})$ storage and $O(n^{1/2+\varepsilon})$ query time? Ideally, can we obtain query time of $O(n^{1+\varepsilon}/s^{1/3})$, with s storage, as in the case of ray shooting amid planes? Alternatively, can one establish a lower-bound argument that shows the limitations of our technique?”*

Inspired by [20], additional results for flat intersection searching were discovered during the last two years, e.g., triangle-triangle intersection searching in \mathbb{R}^4 [19], line-tetrahedron intersection searching in \mathbb{R}^4 [19], curve-disk intersection searching in \mathbb{R}^3 [5], and even more general semialgebraic flat intersection searching [5]. Similar to the result in [20], the improved results are only observed for a special space-time combination and the improvement to the entire trade-off curve is limited. This once again raises the question of whether it is possible to obtain the trade-off curve of $S(n)Q(n)^d = O(n^{d+\sigma})$ for intersection searching in \mathbb{R}^d .

1.2 Our Results

We give a negative answer to this question. We show that answering intersection searching queries in polylogarithmic time when the queries are lines in \mathbb{R}^d and input objects are subsets of $(d-1)$ -flats (that we call hyperslabs) requires $\mathring{\Omega}(n^{2(d-1)})$ space¹. Our lower bound in fact applies to “thin” $(d-1)$ -dimensional slabs (e.g., in 3D, that would be the intersection of the region between two parallel hyperplanes with another hyperplane). This almost matches the current upper bound for the problem and shows that the improvement in [20] cannot significantly improve the trade-off curve when the query time is small. To be specific, we obtain a lower bound of

$$S(n) = \mathring{\Omega}\left(\frac{n^{2(d-1)}}{Q(n)^{4(3d-1)(d-1)-1}}\right)$$

for line-hyperslab intersection reporting in \mathbb{R}^d and a lower bound of

$$S(n) = \mathring{\Omega}\left(\frac{n^6}{Q(n)^{125}}\right)$$

for triangle-triangle intersection reporting in \mathbb{R}^4 . Here, $S(n)$ and $Q(n)$ are the space and query time of the data structure. Similar to the other semialgebraic range reporting lower bounds [2, 3], these lower bounds have a much larger exponent on $Q(n)$ than on n which does allow for substantial improvements when $Q(n)$ is no longer too small; we have not opted for optimizing the exponent of $Q(n)$ in our bounds and using tighter arguments, these exponents can be improved but they cannot match the exponent of n .

We believe our results are timely as flat intersection searching is a hotly investigated field recently, and as mentioned, with many open questions that need to be answered from a lower bound point of view.

1.3 Technical Contributions

From a technical point of view, our results require going beyond the previous attempts [2, 3]. To elaborate, the previous general technique assumed a particular form for the polynomials involved in defining the query semialgebraic ranges, namely, of the form $X_1 = X_2^\Delta + P(X_1, \dots, X_d)$ where the coefficients of P had to be independent and thus could be set arbitrarily small. Unfortunately, the problems in intersection searching cannot fit this framework and there seems to be no easy fix for the following reason. The previous technique relies heavily on the fact that if the coefficients of P is small enough, then one can approximate X_1 with X_2^Δ and for the technique to work both conditions must hold (i.e., small coefficients for P and having degree Δ on X_2).

Generally speaking, the previous techniques do not say anything about problems in which the polynomials involved have a specific form; the only exception is the lower bound for annuli [2] where specific approaches had to be created that could only be applied to the specific algebraic form of circles.

The issue is very prominent in intersection searching where we are dealing with polynomials where the coefficients of the monomials are no longer independent and the polynomials involved have specific forms; for instance, the coefficient of X_2^Δ is zero. We introduce techniques that allows us circumvent these limitations and obtain lower bounds for some broader class of problems that involve polynomials with some specific forms.

¹ In this paper, $\mathring{\Omega}(\cdot)$, $\mathring{\Theta}(\cdot)$, $\mathring{O}(\cdot)$ hides $n^{o(1)}$ factors; $\tilde{\Omega}(\cdot)$, $\tilde{\Theta}(\cdot)$, $\tilde{O}(\cdot)$ hides $\log^{O(1)} n$ factors.

2 Preliminaries

2.1 The Geometric Range Reporting Lower Bound Framework in the Pointer Machine

We use the pointer machine lower bound framework that was also used in the latest proofs [3]. This is a streamlined version of the one originally proposed by Chazelle [14] and Chazelle and Rosenberg [16]. In the pointer machine model, the memory is represented as a directed graph where each node stores one point as well as two pointers pointing to two other nodes in the graph. Given a query, the algorithm starts from a special “root” node, and then explores a subgraph which contains all the input points to report. The size of the directed graph is then a lower bound for the space usage and then minimum subgraph needed to explore to answer any query is a lower bound for the query time.

Intuitively, to answer a range reporting query efficiently, we need to store the output points to the query close to each other. If the answer to any query contains many points and two queries share very few points in common, many points must be stored multiple times, leading to a big space usage.

The streamlined version of the framework is the following [3].

► **Theorem 1.** *Suppose a d -dimensional geometric range reporting problem admits an $S(n)$ space and $Q(n) + O(k)$ query time data structure, where n is the input size and k is the output size. Let $\text{Vol}(\cdot)$ denote the d -dimensional Lebesgue measure. Assume we can find $m = n^c$, for a positive constant c , ranges $\mathcal{R}_1, \mathcal{R}_2, \dots, \mathcal{R}_m$ in a d -dimensional hyperrectangle \mathbf{R} such that*

1. $\forall i = 1, 2, \dots, m, \text{Vol}(\mathcal{R}_i \cap \mathbf{R}) \geq 4c \text{Vol}(\mathbf{R})Q(n)/n;$
2. $\text{Vol}(\mathcal{R}_i \cap \mathcal{R}_j) = O(\text{Vol}(\mathbf{R})/(n2^{\sqrt{\log n}}))$ for all $i \neq j$.

Then, we have $S(n) = \tilde{\Omega}(mQ(n))$.

2.2 Notations and Definitions for Polynomials

In this paper, we only consider polynomials on the reals. Let $P(X_1, \dots, X_d)$ be a polynomial on d indeterminates of degree Δ . Sometimes we will use the notation X to denote the set of d indeterminates X_1, \dots, X_d and so we can write P as $P(X)$. We denote by $I_{d,\Delta}$ a set of d -tuples of non-negative integers (i_1, \dots, i_d) whose sum is at most Δ . We might omit the subscripts d and Δ if they are clear from the context. For an $\mathbf{i} \in I$, we use the notation $X^{\mathbf{i}}$ to represent the monomial $\prod_{j=1}^d X_j^{i_j}$ where $\mathbf{i} = (i_1, \dots, i_d)$. Thus, given real coefficients $A_{\mathbf{i}}$, for $\mathbf{i} \in I$, we can write P as $\sum_{\mathbf{i} \in I} A_{\mathbf{i}} X^{\mathbf{i}}$.

2.3 Geometric Lemmas

We introduce and generalize some geometric lemmas about the intersection of polynomials used in [2]. We first generalize the core Lemma in [2] for univariate polynomials, using a proof similar to [3]. We refer the readers to the full version of the paper for the proof.

► **Lemma 2.** *Let $P(x) = \sum_{i=0}^{\Delta} a_i x^i$ and $Q(x) = \sum_{i=0}^{\Delta} b_i x^i$ be two univariate (constant) degree- Δ polynomials in $\mathbb{R}[x]$ and $|a_i - b_i| \geq \eta$ for some $0 \leq i \leq \Delta$.*

Suppose there is an interval \mathcal{I} of x such that for every $x_0 \in \mathcal{I}$ we have $|P(x_0) - Q(x_0)| \leq w$, then the length of \mathcal{I} is upper bounded by $O((w/\eta)^{1/U})$, where $U = \binom{\Delta+1}{2}$ and the $O(\cdot)$ notation hides constant factors that depend on Δ .

Using Lemma 2, we can show the following. The proof is given in the full version.

► **Lemma 3.** Let $P_1(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} A_{\mathbf{i}} X^{\mathbf{i}}$ and $P_2(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} B_{\mathbf{i}} X^{\mathbf{i}}$ be two d -variate degree- Δ polynomials in $\mathbb{R}[X]$ and $|A_{\mathbf{i}} - B_{\mathbf{i}}| \geq \eta_d$ for some $\mathbf{i} \in I_{d,\Delta}$.

Suppose for each assignment $X_d \in \mathcal{I}_d$ to P_1, P_2 , where \mathcal{I}_d is an interval for X_d , all the coefficients of the resulting $(d-1)$ -variate polynomial $Q_1(X_1, \dots, X_{d-1})$ and $Q_2(X_1, \dots, X_{d-1})$ differ by at most η_{d-1} , then $|\mathcal{I}_d| = O((\eta_{d-1}/\eta_d)^{1/\mu})$.

We can use Lemma 3 $d - 2$ times, and obtain the following corollary.

► **Corollary 4.** Let $P_1(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} A_{\mathbf{i}} X^{\mathbf{i}}$ and $P_2(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} B_{\mathbf{i}} X^{\mathbf{i}}$ be two d -variate degree- Δ polynomials in $\mathbb{R}[X]$ and $|A_{\mathbf{i}} - B_{\mathbf{i}}| \geq \eta_d$ for some $\mathbf{i} \in I_{d,\Delta}$ for $d \geq 3$.

Suppose for each assignment $X_i \in \mathcal{I}_i$ to P_1, P_2 , where \mathcal{I}_i is an interval for X_i , for $i = 3, 4, \dots, d$, all the coefficients of the resulting bivariate polynomial $Q_1(X_1, X_2)$ and $Q_2(X_1, X_2)$ differ by at most η_2 , then $|\mathcal{I}_i| = O((\eta_{i-1}/\eta_i)^{1/\mu})$ for all $i = 3, 4, \dots, d$.

To get the final corollary, we would like the set each η_i such that the length of all each interval \mathcal{I}_i is bounded by some parameter ϑ for $i = 3, \dots, d$. We thus set $\eta_{d-i} = \eta_{d-i+1} \vartheta^{\mu}$.

► **Corollary 5.** Let $P_1(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} A_{\mathbf{i}} X^{\mathbf{i}}$ and $P_2(X) = \sum_{\mathbf{i} \in I_{d,\Delta}} B_{\mathbf{i}} X^{\mathbf{i}}$ be two d -variate degree- Δ polynomials in $\mathbb{R}[X]$ and $|A_{\mathbf{i}} - B_{\mathbf{i}}| \geq \eta_d$ for some $\mathbf{i} \in I_{d,\Delta}$ for $d \geq 3$.

Suppose for each assignment $X_i \in \mathcal{I}_i$ to P_1, P_2 , where \mathcal{I}_i is an interval for X_i , for $i = 3, 4, \dots, d$, all the coefficients of the resulting bivariate polynomial $Q_1(X_1, X_2)$ and $Q_2(X_1, X_2)$ differ by at most $\eta_d \vartheta^{\mu(d-2)}$, then $|\mathcal{I}_i| = O(\vartheta)$ for all $i = 3, 4, \dots, d$.

2.4 Algebra Preliminaries

In this section, we review some tools from algebra. The first tool we will use is the linearity of determinants from linear algebra.

► **Theorem 6 (Linearity of Determinants).** Let $A = [\mathbf{a}_1 \ \dots \ \mathbf{a}_n]$ be an $n \times n$ matrix where each $\mathbf{a}_i \in \mathbb{R}^n$ is a vector. Suppose $\mathbf{a}_j = r \cdot \mathbf{w} + \mathbf{v}$ for some $r \in \mathbb{R}$ and $\mathbf{w}, \mathbf{v} \in \mathbb{R}^n$, then the determinant of A , denoted by $\det(A)$, is

$$\begin{aligned} \det(A) &= \det([\mathbf{a}_1 \ \dots \ \mathbf{a}_{j-1} \ \mathbf{a}_j \ \mathbf{a}_{j+1} \ \dots \ \mathbf{a}_n]) \\ &= r \cdot \det([\mathbf{a}_1 \ \dots \ \mathbf{a}_{j-1} \ \mathbf{w} \ \mathbf{a}_{j+1} \ \dots \ \mathbf{a}_n]) + \det([\mathbf{a}_1 \ \dots \ \mathbf{a}_{j-1} \ \mathbf{v} \ \mathbf{a}_{j+1} \ \dots \ \mathbf{a}_n]). \end{aligned}$$

We will use two types of special matrices in the paper. The first is Vandermonde matrices.

► **Definition 7 (Vandermonde Matrices).** An $n \times n$ Vandermonde matrix is defined by n values x_1, \dots, x_n such that each entry $e_{ij} = x_i^{j-1}$ for $1 \leq i, j \leq n$.

We can compute the determinant of Vandermonde matrices easily.

► **Theorem 8 (Determinant of Vandermonde Matrices).** Let V be a Vandermonde matrix defined by parameters x_1, \dots, x_n . Then $\det(V) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$.

We also need Sylvester matrices.

► **Definition 9 (Sylvester Matrices).** Let $P = \sum_{i=0}^{\Delta_1} a_i x^i$ and $Q = \sum_{i=0}^{\Delta_2} b_i x^i$ be two univariate polynomials over $\mathbb{R}[x]$ of degrees Δ_1, Δ_2 respectively. Then the Sylvester matrix of P and Q , denoted by $Syl(P, Q)$, is a $(\Delta_1 + \Delta_2) \times (\Delta_1 + \Delta_2)$ matrix of the following form

$$\begin{bmatrix} a_{\Delta_1} & a_{\Delta_1-1} & \cdots & a_0 & 0 & \cdots & 0 & 0 \\ 0 & a_{\Delta_1} & a_{\Delta_1-1} & \cdots & a_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{\Delta_1} & a_{\Delta_1-1} & \cdots & a_1 & a_0 \\ b_{\Delta_2} & b_{\Delta_2-1} & \cdots & b_0 & 0 & \cdots & 0 & 0 \\ 0 & b_{\Delta_2} & b_{\Delta_2-1} & \cdots & b_0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & b_{\Delta_2} & b_{\Delta_2-1} & \cdots & b_1 & b_0 \end{bmatrix}.$$

The Sylvester matrix has Δ_2 rows with entries from P and Δ_1 rows with entries from Q . For example, the Sylvester matrix of two polynomials $P = p_1x + p_2$ and $Q = q_1x + q_2$ is

$$\text{Syl}(P, Q) = \begin{bmatrix} p_1 & p_2 \\ q_1 & q_2 \end{bmatrix}$$

One application of Sylvester matrices is to compute the resultant, which is one of the important tools in algebraic geometry. One significance of the resultant is that it equals zero if and only if P and Q have a common factor.

► **Definition 10.** Let P, Q be two univariate polynomials over \mathbb{R} . The resultant of P and Q , denoted by $\text{Res}(P, Q)$, is defined to be the determinant of the Sylvester matrix of P and Q , i.e., $\text{Res}(P, Q) = \det(\text{Syl}(P, Q))$.

3 An Algebraic Geometry Lemma

In this section, we prove an important algebraic geometry lemma that will later be used in our lower bound proof.

► **Lemma 11.** Let F and G be two univariate polynomials on x of degree Δ_F and Δ_G respectively and the leading coefficient of G is 1. Let $P(x, y) \equiv yG(x) - F(x)$.

Let L be a set of $\ell = \Delta_1 + \Delta_G + 1$ points (x_k, y_k) where $\Delta_1 \geq \Delta_F - 1$ and each $x_k = \Theta(1)$ such that $|P(x_k, y_k)| \leq \varepsilon < 1$ for a parameter ε , and $G(x_k) = \Theta(1)$.

Let V be a vector of ℓ monomials consisting of monomials x^i for $0 \leq i \leq \Delta_1$ and monomials yx^i for $0 \leq i \leq \Delta_G - 1$.

If A is an $\ell \times \ell$ matrix where the k -th row of A is the evaluation of the vector V on point (x_k, y_k) , then $|\det(A)| \geq \Omega(\text{Res}(G, F)\lambda^{\ell^2}) - O(\varepsilon)$ where $\lambda = \min_{1 \leq k_1 < k_2 \leq \ell} |x_{k_1} - x_{k_2}|$.

Proof. Note that if $\text{Res}(G, F) = 0$, then there is nothing to prove and thus we can assume this is not the case. Now observe that since $G(x_k) = \Theta(1)$, we can write $y_k = \frac{F(x_k)}{G(x_k)} + \gamma_k$ where $|\gamma_k| = O(\varepsilon)$.

Now consider the matrix A and plug in this value of y_k . An entry of A is in the form of a monomial yx^i being evaluated on a point (x_k, y_k) and thus we have:

$$y_k x_k^i = \left(\frac{F(x_k)}{G(x_k)} + \gamma_k \right) x_k^i = \frac{F(x_k)}{G(x_k)} x_k^i + \gamma_{i,k} \tag{1}$$

where $|\gamma_{i,k}| = O(\varepsilon)$. We use the linearity of determinants (see Theorem 6) in a similar fashion that was also used in [3]. In particular, consider a column of the matrix A ; it consists of the evaluations of a monomial yx^i on all the points $(x_1, y_1), \dots, (x_\ell, y_\ell)$. Using Eq. (1), we

can write this column as the addition of a column C_i that consists of the evaluation of the rational function $\frac{F(x)}{G(x)}x^i$ on the points x_1, \dots, x_ℓ and a column Γ_i that consists of all the values $\gamma_{i,k}$ for $1 \leq k \leq \ell$. By the linearity of determinants, we can write the determinant of A as the sum of determinants of two matrices where one matrix includes the column C_i and the other has Γ_i ; observe that the magnitude of the determinant of the latter matrix can be upper bounded by $O(\varepsilon)$, with hidden constants that depend on Δ . By performing this operation on all the columns, we can separate all the entries involving $\gamma_{i,k}$ into separate matrices and the magnitude of sum of the determinants can be bounded by $O(\varepsilon)$.

Let B be the matrix that remains after removing all the $\gamma_{i,k}$ terms. We bound $|\det(B)|$. Note that B consists of row vectors

$$U = (1 \quad x \quad \dots \quad x^{\Delta_1} \quad y \quad yx \quad \dots \quad yx^{\Delta_G-1}).$$

evaluated at some value $x = x_k$ and $y = \frac{F(x_k)}{G(x_k)}$ at its k -th row. This is equivalent to the evaluation of the following vector:

$$(1 \quad x \quad \dots \quad x^{\Delta_1} \quad \frac{F}{G} \quad \frac{F}{G}x \quad \dots \quad \frac{F}{G}x^{\Delta_G-1}).$$

Observe that row k of matrix B will be evaluating U on the point x_k . Since $G(x_k) = \Theta(1) \neq 0$, we can multiply row k by $G(x_k)$ and this will only change the determinant by a constant factor. With a slight abuse of the notation, let B denote the matrix after this multiplication step. Thus, the columns of B now correspond to the evaluation of the following vector.

$$(G \quad Gx \quad \dots \quad Gx^{\Delta_1} \quad F \quad Fx \quad \dots \quad Fx^{\Delta_G-1}).$$

Note that we can exchange columns and it will only flip the signs of the determinant of a matrix. We will focus on bounding the determinant of

$$(Gx^{\Delta_1} \quad Gx^{\Delta_1-2} \quad \dots \quad G \quad Fx^{\Delta_G-1} \quad Fx^{\Delta_G-2} \quad \dots \quad F).$$

The key observation is that there is a strong connection between the Sylvester matrix of G, F and matrix B . Recall that the Sylvester matrix of G and F is of the form

$$\text{Syl}(G, F) = \begin{bmatrix} G_{\Delta_G} & G_{\Delta_G-1} & \dots & G_0 & 0 & \dots & 0 & 0 \\ 0 & G_{\Delta_G} & G_{\Delta_G-1} & \dots & G_0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & G_{\Delta_G} & G_{\Delta_G-1} & \dots & G_1 & G_0 \\ F_{\Delta_F} & F_{\Delta_F-1} & \dots & F_0 & 0 & \dots & 0 & 0 \\ 0 & F_{\Delta_F} & F_{\Delta_F-1} & \dots & F_0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & F_{\Delta_F} & F_{\Delta_F-1} & \dots & F_1 & F_0 \end{bmatrix},$$

where G_i (resp. F_i) is the coefficient of x^i in G (resp. F). Observe that

$$(Gx^{\Delta_F-1} \quad Gx^{\Delta_F-2} \quad \dots \quad G \quad Fx^{\Delta_G-1} \quad Fx^{\Delta_G-2} \quad \dots \quad F) = \text{Syl}(G, F) \cdot (x^{\Delta_F+\Delta_G-1} \quad x^{\Delta_F+\Delta_G-2} \quad \dots \quad x \quad 1)^T,$$

which means that by the linear transformation described by $\text{Syl}(G, F)^{-1}$, which exists as $\text{Res}(G, F) = \det(\text{Syl}(G, F)) \neq 0$, we can turn the last $\Delta_F + \Delta_G$ columns in B to

$$(x^{\Delta_F+\Delta_G-1} \quad x^{\Delta_F+\Delta_G-2} \quad \dots \quad x \quad 1).$$

Since the remaining columns are all polynomials in x and the highest degree in column i is $\Delta_G + \Delta_1 - i$ for $i = 0, 1, \dots, \Delta_F$, by using column operations, we can eliminate all lower degree terms for each column and the only term left for column i is $G_{\Delta_G} x^{\Delta_G + \Delta_1 - i}$. Note that column operations do not change the determinant.

By assumption, the leading coefficients of G is 1, i.e., $G_{\Delta_G} = 1$. Thus, this transforms B into a Vandermonde matrix V_B of size $\ell \times \ell$. By Theorem 8, $|\det(V_B)| = \Omega(\lambda^{\ell^2})$. Since multiplying the inverse of $\text{Syl}(G, F)$ scales $\det(B)$ by a factor of $\Theta(|\det(\text{Syl}(G, F)^{-1})|) = \Theta(|\text{Res}(G, F)^{-1}|)$, we bound $|\det(B)| = |\det(V_B)| / (1/|\text{Res}(G, F)|) = \Omega(|\text{Res}(G, F)| \lambda^{\ell^2})$. The claim then follows from this. \blacktriangleleft

4 Lower Bounds for Flat Intersection Reporting

We are now ready to show lower bounds for flat intersection reporting. We first establish a reduction from special polynomial slab reporting problems to flat intersection reporting.

4.1 A Reduction from Polynomial Slab Range Reporting to Flat-hyperslab Intersection Reporting

We study the following flat intersection reporting problem.

► **Definition 12** (Flat-hyperslab Intersection Reporting). *In the t -flat-hyperslab intersection reporting problem, we are given a set \mathcal{S} of n $(d-t)$ -dimensional hyperslabs in \mathbb{R}^d , i.e., regions created by a linear translation of $(d-t-1)$ -flats, where $0 \leq t < d$, as the input, and the goal is to preprocess \mathcal{S} into a data structure such that given any query t -flat γ , we can output $\mathcal{S} \cap \gamma$, i.e., the set of $(d-t)$ -hyperslabs intersecting the query t -flat, efficiently.*

First, observe that any t -flat that is not parallel to any of the axes can be formulated as

$$\begin{bmatrix} a_{0,1} & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ a_{1,1} & a_{1,2} & \cdots & a_{1,t} & a_{1,t+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{d-t,1} & a_{d-t,2} & \cdots & a_{d-t,t} & a_{d-t,t+1} \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_t \\ 1 \end{bmatrix} = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix},$$

where $a_{i,j}$'s are the parameters defining the t -flat, and τ_1, \dots, τ_t are the free variables that generate points in the t -flat. Note that we only need $(d-t)(t+1)$ independent $a_{i,j}$'s to define a t -flat.

On the other hand, we consider $(d-t)$ -hyperslabs of form

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & b_{1,1} & b_{1,2} & \cdots & b_{1,d-t} \\ 0 & 1 & \cdots & 0 & b_{2,1} & b_{2,2} & \cdots & b_{2,d-t} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & b_{t,1} & b_{t,2} & \cdots & b_{t,d-t} \\ 0 & 0 & \cdots & 0 & b_{t+1,1} & b_{t+1,2} & \cdots & b_{t+1,d-t} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{d-1} \\ x_d \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 + w \end{bmatrix},$$

where $b_{i,j}$'s are the parameters defining a $(d-t-1)$ -flat, and parameter $w \in [0, w_0]$ adds one extra dimension to the flat to make it $(d-t)$ -dimensional; in essence, we will be considering all the $(d-t-1)$ -flats for all $w \in [0, w_0]$ which will turn it into a $(d-t)$ -hyperslab.

Therefore, the intersection of a t -flat and a $(d - t)$ -hyperslab must be a solution to

$$\begin{bmatrix} 1 & 0 & \cdots & 0 & b_{1,1} & b_{1,2} & \cdots & b_{1,d-t} \\ 0 & 1 & \cdots & 0 & b_{2,1} & b_{2,2} & \cdots & b_{2,d-t} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & b_{t,1} & b_{t,2} & \cdots & b_{t,d-t} \\ 0 & 0 & \cdots & 0 & b_{t+1,1} & b_{t+1,2} & \cdots & b_{t+1,d-t} \end{bmatrix} \cdot \begin{bmatrix} a_{0,1} & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ a_{1,1} & a_{1,2} & \cdots & a_{1,t} & a_{1,t+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{d-t,1} & a_{d-t,2} & \cdots & a_{d-t,t} & a_{d-t,t+1} \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \vdots \\ \tau_t \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 + w \end{bmatrix}.$$

Multiplying the two matrices, we obtain the following system

$$\begin{bmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{t+1,i} \end{bmatrix} \cdot \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_t \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ -1 + w \end{bmatrix}.$$

We denote this linear system by $A\tau = \mathbf{s}$ and assume

$$\det(A) \neq 0 \tag{2}$$

which is the case when the t -flat and the $(d - t)$ -hyperslab properly intersect, and this system has a solution iff the last entry of the solution vector is 1. So by Cramer's rule, we have

$$1 = \frac{\begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & 0 \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & -1 + w \end{vmatrix}}{\begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{t+1,i} \end{vmatrix}}.$$

By the linearity of determinants, we have

$$0 = \begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1}b_{1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1}b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2}b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t+1}b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1}b_{t+1,i} & \sum_{i=1}^{d-t} a_{i,2}b_{t+1,i} & \cdots & 1 + \sum_{i=1}^{d-t} a_{i,t+1}b_{t+1,i} - w \end{vmatrix}. \tag{3}$$

Consider the value of the above determinant using Leibniz formula for determinants, which is the sum of $(t + 1)!$ terms. Consider the terms that have at most 1 factor of $b_{i,j}$; these can only come from the diagonals. Thus, any t -flat parameterized by $\mathbf{a} = (a_{i,j})$ intersects a query $(d - t)$ -hyperslab parameterized by $\mathbf{b} = (b_{i,j})$ if and only if

$$0 = a_{0,1} + a_{0,1} \sum_{j=2}^{t+1} \sum_{i=1}^{d-1} a_{i,j}b_{j,i} + \sum_{i=1}^{d-1} a_{i,1}b_{1,i} + E(\mathbf{a}, \mathbf{b}) + f(\mathbf{a}, \mathbf{b}, w) = P(\mathbf{a}, \mathbf{b}) + f(\mathbf{a}, \mathbf{b}, w),$$

where $E(\mathbf{a}, \mathbf{b})$ contains the sum of products of at least two distinct $a_{i_1,i_2}b_{i_3,i_1}$ and $f(\mathbf{a}, \mathbf{b}, w)$ is a polynomial with factor w .

3:10 Lower Bounds for Intersection Reporting Among Flat Objects

Note that after fixing \mathbf{a}, \mathbf{b} , $f(\mathbf{a}, \mathbf{b}, w)$ is a polynomial in w and we assume that

$$\frac{\partial f(\mathbf{a}, \mathbf{b}, w)}{\partial w} = - \begin{vmatrix} a_{0,1} + \sum_{i=1}^{d-t} a_{i,1} b_{1,i} & \sum_{i=1}^{d-t} a_{i,2} b_{1,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t} b_{1,i} \\ \sum_{i=1}^{d-t} a_{i,1} b_{2,i} & 1 + \sum_{i=1}^{d-t} a_{i,2} b_{2,i} & \cdots & \sum_{i=1}^{d-t} a_{i,t} b_{2,i} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{d-t} a_{i,1} b_{t,i} & \sum_{i=1}^{d-t} a_{i,2} b_{t,i} & \cdots & 1 + \sum_{i=1}^{d-t} a_{i,t} b_{t,i} \end{vmatrix} < 0. \quad (4)$$

This implies the following lemma.

► **Lemma 13.** *Assuming \mathbf{a}, \mathbf{b} satisfying Assumptions (2) and (4), for any fixed \mathbf{a} , there is a \mathbf{b} such that $0 \leq P(\mathbf{a}, \mathbf{b}) \leq -f(\mathbf{a}, \mathbf{b}, w_0)$ if and only if there is some $w \in [0, w_0]$ such that $P(\mathbf{a}, \mathbf{b}) + f(\mathbf{a}, \mathbf{b}, w) = 0$.*

Proof. Since $f(\mathbf{a}, \mathbf{b}, w)$ is a polynomial in w and $\frac{\partial f}{\partial w} < 0$, $f(\mathbf{a}, \mathbf{b}, w)$ is continuous and decreasing in $[0, w_0]$. Furthermore $f(\mathbf{a}, \mathbf{b}, 0) = 0$ as w is a factor of f . The lemma follows. ◀

Fixing \mathbf{a} in $P(\mathbf{a}, \mathbf{b})$, we obtain a polynomial in \mathbf{b} . Let $(P(\mathbf{a}, \mathbf{b}), f(\mathbf{a}, \mathbf{b}, w_0)) = \{\mathbf{b} : 0 \leq P(\mathbf{a}, \mathbf{b}) \leq -f(\mathbf{a}, \mathbf{b}, w_0)\}$ be a polynomial slab. This essentially establishes a reduction between polynomial slab reporting and flat intersection reporting.

► **Corollary 14.** *Assuming \mathbf{a}, \mathbf{b} satisfying Assumptions (2) and (4), for any fixed \mathbf{a} , there is a \mathbf{b} such that $\mathbf{b} \in (P(\mathbf{a}, \mathbf{b}), f(\mathbf{a}, \mathbf{b}, w_0))$ if and only if a t -flat parameterized by \mathbf{a} intersects a $(d-t)$ -hyperslab of width w_0 parameterized by \mathbf{b} .*

4.2 Lower Bounds for Flat-hyperslab Intersection Reporting

We are now ready to prove the lower bounds. We show lower bounds for 1-flat-hyperslab intersection reporting in \mathbb{R}^d and 2-flat-hyperslab intersection reporting in \mathbb{R}^4 .

First observe that by setting $t = 1$ in Eq. (3) and using Corollary 14 a polynomial slab reporting problem with polynomial

$$\begin{aligned} P_1(\mathbf{a}, \mathbf{b}) &= a_{0,1} + a_{0,1} \sum_{i=1}^{d-1} a_{i,2} b_{2,i} + \sum_{i=1}^{d-1} a_{i,1} b_{1,i} + \sum_{i,j=1 \wedge i \neq j}^{d-1} (a_{i,1} a_{j,2} - a_{j,1} a_{i,2}) b_{1,i} b_{2,j} \\ &= b_{1,1} G_1(b_{2,2}) + F_1(b_{2,2}), \end{aligned} \quad (5)$$

reduces to a line-hyperslab intersection reporting problem, where to get G_1 , we have collected all the monomials that have $b_{1,1}$ in them and then we have factored $b_{1,1}$ out and we are considering it as a polynomial of $b_{2,2}$ (all the other variables are considered “constant”). F_1 is defined similarly by considering the remaining terms as a function of $b_{2,2}$. Observe that the polynomial does not have any term with degree 3. Let $G_1 = g_{1,1} b_{2,2} + g_{1,0}$ and $F_1 = f_{1,1} b_{2,2} + f_{1,0}$.

Similarly, polynomial slab reporting with

$$\begin{aligned} P_2(\mathbf{a}, \mathbf{b}) &= a_{0,1} + a_{0,1} \sum_{j=1}^2 \sum_{i=2}^3 a_{j,i} b_{i,j} + \sum_{j=1}^2 a_{j,1} b_{1,j} \\ &\quad + a_{0,1} \sum_{j,l=1 \wedge j \neq l}^2 (a_{j,2} a_{l,3} - a_{j,3} a_{l,2}) b_{2,j} b_{3,l} + \sum_{j,l=1 \wedge j \neq l}^2 \sum_{k=2}^3 (a_{j,1} a_{l,k} - a_{j,k} a_{l,1}) b_{1,j} b_{k,l} \\ &= b_{1,1} G_2(b_{2,2}) + F_2(b_{2,2}) \end{aligned} \quad (6)$$

reduces to 2-flat-hyperslab intersection reporting in \mathbb{R}^4 where G_2, F_2 are defined similarly as G_1, F_1 .

For the moment, we focus on the case of line-hyperslab intersection reporting but the same applies also to 2-flat-hyperslab intersection reporting in \mathbb{R}^4 since the polynomials F_2 and G_2 involved in the definition of Eq. (6) are quite similar to Eq. (5).

Here, we will use our techniques from Section 3. The general idea is that we will use Corollary 5, to reduce the $2(d - 1)$ -variate polynomials P_1 and P_2 into bivariate polynomials on $b_{1,1}$ and $b_{2,2}$. Then, the variable $b_{1,1}$ will be our y variable and $b_{2,2}$ will be the x variable in Section 3, and G_1 and F_1 here will play the same role as in that section. We will set

$$a_{1,1} = \frac{1 + a_{1,2}a_{2,1}}{a_{2,2}} \tag{7}$$

which will ensure that the leading coefficient of G_1 is 1. This is our normalization step, since we can divide the equations defining the intersection (and thus polynomials P_1 and P_2) by any constant. Eventually, the resultant of the polynomials F_1 and G_1 will play an important role. Observe that the resultant is

$$\text{Res}(G_1, F_1) = \begin{vmatrix} 1 & g_0 \\ f_1 & f_0 \end{vmatrix} = f_0 - g_0f_1. \tag{8}$$

4.3 Construction of Input Points and Queries

Now we are ready to describe our input and query construction. Assume we have a data structure that uses $S(n)$ space and has the query time $Q(n) + O(k)$ where k is the output size; for brevity we use $Q = Q(n)$.

We will start with a fixed line and a fixed hyperslab and then build the queries and inputs very close to these two fixed objects. However, we require a certain “general position” property with respect to these two fixed objects.

Recall that Eq. (5) refers to the condition of whether a (query) line described by \mathbf{a} variables intersects a $(d - 2)$ -dimensional flat described by the \mathbf{b} variables (which corresponds to setting the variable w to zero). Consider a fixed flat and a fixed line. To avoid future confusion, let \mathbf{A} and \mathbf{B} refer to this fixed line and flat. We require the following.

- \mathbf{A} and \mathbf{B} must intersect properly (i.e., the line is not contained in the flat). Observe that it implies that when we consider $P_1(\mathbf{A}, \mathbf{b})$ as a polynomial in \mathbf{b} variables, \mathbf{B} does not belong to the zero set of $P_1(\mathbf{A}, \mathbf{b})$. Note that this satisfies Assumption (2).
- The polynomial $P_1(\mathbf{A}, \mathbf{b})$ (as a polynomial in \mathbf{b}) is irreducible. This is true as long as \mathbf{A} is chosen so that no coefficient in P_1 is zero. To see this, note that P_1 is a polynomial in \mathbf{b} and any variable $b_{i,j}$ has degree 1. Suppose for the sake of contradiction that P_1 is reducible, then the factorization must be of the form

$$P_1(\mathbf{A}, \mathbf{b}) = \left(c_{10} + \sum_{i=1}^{d-1} c_{1i}b_{1i} \right) \cdot \left(c_{20} + \sum_{i=1}^{d-1} c_{2i}b_{2i} \right),$$

for nonzero coefficients $c_{10}, c_{20}, c_{1i}, c_{2i}$. Then by Eq. (5),

1. $a_{0,1} = c_{10}c_{20}$,
2. $\forall i = 1, \dots, d - 1 : a_{0,1}a_{i,2} = c_{10}c_{2i}$,
3. $\forall i = 1, \dots, d - 1 : a_{i,1} = c_{1i}c_{20}$,
4. $\forall i, j = 1, 2, \dots, d - 1 : a_{i,1}a_{j,2} - a_{j,1}a_{i,2} = c_{1i}c_{2j}$.

However, for these conditions to hold, all coefficients of P_1 must be zero, a contradiction.

- Observe that the irreducibility of $P_1(\mathbf{A}, \mathbf{b})$ as a polynomial in \mathbf{b} implies that it has only finitely many points where the tangent hyperplane at those points is parallel to some axis. We assume \mathbf{B} is not one of those points.

3:12 Lower Bounds for Intersection Reporting Among Flat Objects

- The irreducibility of $P_1(\mathbf{A}, \mathbf{b})$ as a polynomial in \mathbf{b} can be used to satisfy Assumption (4) since the corresponding polynomial of the determinant involved in Assumption (4) can only have $\Theta(1)$ many common roots with $P_1(\mathbf{A}, \mathbf{b})$.
- Finally, since the polynomial $P_1(\mathbf{A}, \mathbf{b})$ is irreducible and since $\text{Res}(G_1, F_1)$ is also of degree 2 in \mathbf{b} variables, it follows that $\text{Res}(G_1, F_1)$ is algebraically independent of $P_1(\mathbf{A}, \mathbf{b})$. This means that there are only finitely many places where both polynomials are zero, meaning, we can additionally assume that Eq. (8) is non-zero (when evaluated at \mathbf{B}).

Consider two parameters ε_p and $\varepsilon_q = \varepsilon_p/C$ where C is a large enough constant and ε_p is a parameter to be set later. Consider the parametric space of the input objects, where the variable \mathbf{b} defines a single point. In such a space, \mathbf{B} defines a single point. Place an axis-aligned cube \mathbf{R} of side-length ε_p centered around \mathbf{B} . The input slabs are defined by placing a set of n random points inside \mathbf{R} . Each point in \mathbf{R} defines a $(d-2)$ -dimensional flat. We set $w = \Theta(\frac{Q}{n})$ which in turn defines a “narrow $(d-1)$ -hyperslab”.

We now define the set of queries. Notice that P_1 has exactly $2(d-1)$ algebraically independent coefficients; these are the coefficients of linear terms involved plus $a_{0,1}$; recall that by Eq. (7), $a_{1,1}$ was fixed as a function of $a_{1,2}a_{2,1}$ and $a_{2,2}$ but we still have $a_{0,1}$ as a free parameter. These $2(d-1)$ coefficients define another parametric space, where \mathbf{A} denotes a single point. Place a $2(d-1)$ -dimensional hypercube of side length ε_q and then subdivide it into a grid where the side-length of every cell is τ . Every grid point now defines a different query. Let \mathcal{Q} be the set of all the queries we have constructed.

Notice that a query defined by a point $\mathbf{a} \in \mathcal{Q}$ defines a line in the primal space, but when considered in the parametric space \mathbf{R} , it corresponds to a manifold (zeroes of a degree two multilinear polynomial) that includes the set of points that correspond to $(d-2)$ -dimensional flats that pass through the line in the primal space. The variable w allows us to turn it to a range reporting problem where we need to output any $(d-2)$ -dimensional flat that passes within w vertical distance of the query line. The following observations and lemmas are the important geometric properties that we require out of our construction.

► **Observation 15.** *For two different queries \mathbf{a}_1 and \mathbf{a}_2 , the polynomials $P_1(\mathbf{a}_1, \mathbf{b})$ and $P_1(\mathbf{a}_2, \mathbf{b})$ differ by at least τ in at least one of their coefficients.*

► **Observation 16.** *Consider a line f parallel to an axis. For small enough ε_p , and any $\mathbf{a} \in \mathcal{Q}$, the function $P_1(\mathbf{a}, \mathbf{b})$ evaluated on the line f is such that the magnitude of its derivative is bounded by $\Omega(1)$.*

Proof. Recall that \mathbf{B} was chosen such that the manifold corresponding to \mathbf{A} does not have a tangent parallel to any of the axes at point \mathbf{B} and thus the derivative of the function $P_1(\mathbf{A}, \mathbf{B})$ is non-zero at \mathbf{B} . The lemma then follows since ε_p and ε_q are small enough and $P_1(\mathbf{A}, \mathbf{B})$ is a continuous function w.r.t any of its variables. ◀

Let $\text{Vol}'(\mathbf{R})$ be the $(d-1)$ -dimensional volume of \mathbf{R} , i.e., the volume of the projection of \mathbf{R} to any of its $(d-1)$ -dimensional subspace.

► **Observation 17.** *The intersection volume of the range defined by a query \mathbf{a} and \mathbf{R} is $\Theta(w \text{Vol}'(\mathbf{R}))$ if C in the definition of ε_q is large enough, for $w \leq \varepsilon_p$.*

Proof. Observe that the query manifold defined by \mathbf{A} passes through the center, \mathbf{B} , of \mathbf{R} by construction. Since each coordinate of \mathbf{a} differs from \mathbf{A} by at most ε_q , it thus follows that by setting C large enough, we can ensure that the distance between \mathbf{B} and \mathbf{a} is less than $\varepsilon_p/2$. Also observe that the width of the range along any axis will be $\Theta(w)$. The claim now follows by integrating the volume over vertical lines using Observation 16. ◀

► **Lemma 18.** Consider a query $\mathbf{a} \in \mathcal{Q}$ and let \mathbf{r} be the range that represents \mathbf{a} in the parametric space defined by \mathbf{R} . Consider an interval \mathcal{I} on the i -th side of \mathbf{R} , for some i . Let $\mathbf{r}_{\mathcal{I}}$ be the subset of \mathbf{r} whose projection on the i -th side of \mathbf{R} falls inside \mathcal{I} . Then, the volume of $\mathbf{r}_{\mathcal{I}}$ is $O(\text{Vol}(\mathbf{R})w|\mathcal{I}|/\varepsilon_p)$.

Proof. Both claims follow through Observation 16 by integrating the corresponding volumes over lines parallel to axes. ◀

4.4 Using the Framework

Observe that by the above Observation 17, setting $w = \Theta(\frac{Q}{n}\varepsilon_p)$ satisfies Condition 1 of the lower bound framework in Theorem 1.

Satisfying Condition 2 requires a bit more work however. To do that, consider two queries defined by points \mathbf{a}_1 and \mathbf{a}_2 . Let \mathbf{r}_1 and \mathbf{r}_2 be the two corresponding ranges in the parametric space of \mathbf{R} .

To satisfy Condition 2, assume for contradiction that the volume of $\mathbf{r}_1 \cap \mathbf{r}_2$ is large, i.e., $\omega(\text{Vol}(\mathbf{R})/(n\psi))$ where $\psi = 2^{\sqrt{\log n}}$. We now combine Observation 15, and Corollary 5 with parameter ϑ set to $\varepsilon_0 \frac{\varepsilon_p}{Q\psi}$ where ε_0 is a small enough constant and where X_1 represents $b_{1,1}$, X_2 represents $b_{2,2}$ and the remaining indeterminates represent the rest of variables in \mathbf{b} ; note that the value of d in Corollary 5 is $\beta = 2(d - 1)$ and $\mathcal{U} = \binom{2+1}{2} = 3$. Observe that each interval \mathcal{I}_i determined by Corollary 5 defines a slab parallel to the i -th axis in \mathbf{R} ; let \mathbf{R}_{bad} be the union of these slabs. By Lemma 18, and choice of small enough ε_0 , a positive fraction of the intersection volume of \mathbf{r}_1 and \mathbf{r}_2 must lie outside \mathbf{R}_{bad} . In addition, Corollary 5 allows us to pick some fixed values for all variables in \mathbf{b} , except for $b_{1,1}$ and $b_{2,2}$ with the property the final polynomials H_1 and H_2 (on indeterminates $b_{1,1}$ and $b_{2,2}$) that we obtain have the property that they have at least one coefficient which differs by

$$\Omega \left(\tau \left(\varepsilon_0 \frac{\varepsilon_p}{Q\psi} \right)^{3(\beta-2)} \right) \tag{9}$$

between them; we call this operation of plugging values for all \mathbf{b} except for $b_{1,1}$ and $b_{2,2}$ *slicing*. After slicing, we are reduced to the bivariate case; consider the set of points on which both H_1 and H_2 have value $O(w)$. If the 1D interval length of such points is $O(\varepsilon_p/(Q\psi))$, we call this a *good slice*, otherwise a *bad slice*. By Lemma 18, there must be bad slices since if all the slices are good, by integration of the intersection area of \mathbf{r}_1 and \mathbf{r}_2 over all the remaining variables in \mathbf{b} , \mathbf{r}_1 and \mathbf{r}_2 intersect with volume $O(\text{Vol}(\mathbf{R})/(n\psi))$, a contradiction.

We now show that we can arrive at a contradiction, assuming the existence of a bad slice. Given a bad slice, and any constant ℓ , we can find ℓ points $(x_1, y_1), \dots, (x_\ell, y_\ell)$ such that $|x_{k_1} - x_{k_2}| = \omega(\varepsilon_p/(Q\psi))$ for all $1 \leq k_1 < k_2 \leq \ell$ and that $H_1(x_k, y_k), H_2(x_k, y_k) = O(w)$ for all $k \in \{1, 2, \dots, \ell\}$. Observe that $H_i(x, y)$ has only monomials y, x, xy and a constant term. The critical observation here is that the coefficient of the monomial xy is always 1 since the coefficient of the monomial $b_{1,1}b_{2,2}$ was 1 and there was no monomial of degree three in P_1 , meaning, after slicing this coefficient will not change. We pick $\ell = 3$ and thus we tweak all the three other coefficients of H_1 . Tweaking H_1 such that $\tilde{H}_1(x_k, y_k) = H_2(x_k, y_k)$ corresponds to solving a linear system of equations that come from evaluations of monomials X, Y , and a constant term at points (x_k, y_k) . We can thus use Lemma 11 with $\Delta_1 = \Delta_F = 1$, $\lambda = \omega(\varepsilon_p/(Q\psi))$. Observe that $\text{Res}(G, F)$ here is a constant by the properties of our construction. Also observe that by Lemma 11, the magnitude of the determinant of matrix A defined in Lemma 11 is

$$\omega \left((\varepsilon_p/(Q\psi))^9 \right).$$

By the same argument in [3], this means that the tweaking operation can be done such that each coefficient of H_1 is changed by

$$o\left(\left(\varepsilon_p/(Q\psi)\right)^{-9} w\right). \quad (10)$$

We observe that after tweaking, \tilde{H}_1 and H_2 must coincide since by Lemma 11, the determinant of the relevant monomials is non-zero and thus there's a unique polynomial that passes through points $(x_1, y_1), \dots, (x_\ell, y_\ell)$. Finally, to get a contradiction, we simply need to ensure that Eq. (10) is asymptotically smaller than Eq. (9). This yields a bound for the value of τ ,

$$\tau = \Theta\left(w(Q\psi)^{3(\beta-2)+9}\right) = \Theta\left(w(Q\psi)^{3\beta+3}\right) \quad (11)$$

where we have assumed that ε_p , and ε_0 are small enough constants that have been absorbed in the $\Theta(\cdot)$ notation. Thus, this choice of τ will make sure that Condition 2 of the framework is also satisfied. It remains to calculate the number of queries that have been generated. Observe that τ was the side-length of a small enough grid around the point \mathbf{A} in a β -dimensional space. Thus, the number of queries we generated is

$$m = \overset{\circ}{\Omega}\left(\left(\frac{1}{\tau}\right)^\beta\right) = \overset{\circ}{\Omega}\left(\frac{n^\beta}{Q^{\beta(3\beta+4)}}\right). \quad (12)$$

Applying Theorem 1 yields a space lower bound of

$$S(n) = \overset{\circ}{\Omega}(mQ) = \overset{\circ}{\Omega}\left(\frac{n^{2(d-1)}}{Q^{4(3d-1)(d-1)-1}}\right) \quad (13)$$

for line-hyperslab intersection reporting since $\beta = 2(d-1)$. One can verify that the same argument works for triangle-triangle intersection reporting in \mathbb{R}^4 , since P_2 is also a multilinear polynomial of degree two. In this case, $\beta = 6$ which yields a space lower bound of

$$S(n) = \overset{\circ}{\Omega}\left(\frac{n^6}{Q^{125}}\right). \quad (14)$$

To sum up, we obtain the following results:

► **Theorem 19.** *Any data structure that solves line-hyperslab intersection reporting in \mathbb{R}^d must satisfy a space-time tradeoff of $S(n) = \overset{\circ}{\Omega}\left(\frac{n^{2(d-1)}}{Q(n)^{4(3d-1)(d-1)-1}}\right)$.*

► **Theorem 20.** *Any data structure that solves triangle-triangle intersection reporting in \mathbb{R}^4 must satisfy a space-time tradeoff of $S(n) = \overset{\circ}{\Omega}\left(\frac{n^6}{Q(n)^{125}}\right)$.*

5 Conclusion and Open Problems

We study line-hyperslab intersecting reporting in \mathbb{R}^d and triangle-triangle intersecting reporting in \mathbb{R}^4 . We show that any data structure with $n^{o(1)} + O(k)$ query time must use space $\overset{\circ}{\Omega}(n^{2(d-1)})$ and $\overset{\circ}{\Omega}(n^6)$ for the two problems respectively. This matches the classical upper bounds for the small $n^{o(1)}$ query time case for the two problems and answer an open problem for lower bounds asked by Ezra and Sharir [20]. Along the way, we generalize and develop the lower bound technique used in [2, 3].

The major open problem is how to show a lower bound for general intersection reporting between objects of t and $(d-t)$ dimensions or for flat semialgebraic objects as studied recently in [5]. Many of our techniques work, however, one big challenge is that after applying Corollary 5, the leading coefficient changes and thus we can no longer guarantee big gaps between coefficients.

References



- 1 Peyman Afshani. Improved pointer machine and I/O lower bounds for simplex range reporting and related problems. In *Proceedings of the Twenty-Eighth Annual Symposium on Computational Geometry*, SoCG '12, pages 339–346, New York, NY, USA, 2012. Association for Computing Machinery. doi:10.1145/2261250.2261301.
- 2 Peyman Afshani and Pingan Cheng. Lower bounds for semialgebraic range searching and stabbing problems. In *37th International Symposium on Computational Geometry*, volume 189 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 8, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2021.
- 3 Peyman Afshani and Pingan Cheng. On semialgebraic range reporting. In *38th International Symposium on Computational Geometry*, volume 224 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Paper No. 3, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022. doi:10.4230/lipics.socg.2022.3.
- 4 Pankaj K. Agarwal. Simplex range searching and its variants: a review. In *A journey through discrete mathematics*, pages 1–30. Springer, Cham, 2017.
- 5 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir. Intersection queries for flat semi-algebraic objects in three dimensions and related problems. In *38th International Symposium on Computational Geometry*, volume 224 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Paper No. 4, 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2022. doi:10.4230/lipics.socg.2022.4.
- 6 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, and Joshua Zahl. Efficient algorithm for generalized polynomial partitioning and its applications. *SIAM J. Comput.*, 50(2):760–787, 2021. doi:10.1137/19M1268550.
- 7 Pankaj K. Agarwal and Jirí Matousek. On range searching with semialgebraic sets. *Discret. Comput. Geom.*, 11:393–418, 1994. doi:10.1007/BF02574015.
- 8 Pankaj K. Agarwal and Jirí Matoušek. Ray shooting and parametric search. *SIAM J. Comput.*, 22(4):794–806, 1993. doi:10.1137/0222051.
- 9 Pankaj K. Agarwal, Jirí Matoušek, and Micha Sharir. On range searching with semialgebraic sets. II. *SIAM J. Comput.*, 42(6):2039–2062, 2013. doi:10.1137/120890855.
- 10 Pankaj K. Agarwal and Micha Sharir. Ray shooting amidst convex polyhedra and polyhedral terrains in three dimensions. *SIAM J. Comput.*, 25(1):100–116, 1996. doi:10.1137/S0097539793244368.
- 11 Boris Aronov, Mark de Berg, and Chris Gray. Ray shooting and intersection searching amidst fat convex polyhedra in 3-space. *Comput. Geom.*, 41(1-2):68–76, 2008. doi:10.1016/j.comgeo.2007.10.006.
- 12 Timothy M. Chan. Optimal partition trees. *Discrete Comput. Geom.*, 47(4):661–690, 2012. doi:10.1007/s00454-012-9410-z.
- 13 Bernard Chazelle. Lower bounds on the complexity of polytope range searching. *J. Amer. Math. Soc.*, 2(4):637–666, 1989. doi:10.2307/1990891.
- 14 Bernard Chazelle. Lower bounds for orthogonal range searching. I. The reporting case. *J. Assoc. Comput. Mach.*, 37(2):200–212, 1990. doi:10.1145/77600.77614.
- 15 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, December 1993. doi:10.1007/BF02189314.
- 16 Bernard Chazelle and Burton Rosenberg. Simplex range reporting on a pointer machine. *Comput. Geom.*, 5(5):237–247, 1996. doi:10.1016/0925-7721(95)00002-X.
- 17 M. de Berg, D. Halperin, M. Overmars, J. Snoeyink, and M. van Kreveld. Efficient ray shooting and hidden surface removal. *Algorithmica*, 12(1):30–53, 1994. doi:10.1007/BF01377182.
- 18 Mark de Berg and Chris Gray. Vertical ray shooting and computing depth orders for fat objects. *SIAM J. Comput.*, 38(1):257–275, 2008. doi:10.1137/060672261.
- 19 Esther Ezra and Micha Sharir. Intersection searching amid tetrahedra in four dimensions. *CoRR*, abs/2208.06703, 2022. doi:10.48550/arXiv.2208.06703.

- 20 Esther Ezra and Micha Sharir. On ray shooting for triangles in 3-space and related problems. *SIAM J. Comput.*, 51(4):1065–1095, 2022. doi:10.1137/21M1408245.
- 21 Larry Guth. Polynomial partitioning for a set of varieties. *Math. Proc. Cambridge Philos. Soc.*, 159(3):459–469, 2015. doi:10.1017/S0305004115000468.
- 22 Larry Guth and Nets Hawk Katz. On the Erdős distinct distances problem in the plane. *Ann. of Math. (2)*, 181(1):155–190, 2015. doi:10.4007/annals.2015.181.1.2.
- 23 Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993. doi:10.1007/BF02573972.
- 24 Jiří Matoušek and Zuzana Patáková. Multilevel polynomial partitions and simplified range searching. *Discrete Comput. Geom.*, 54(1):22–41, 2015. doi:10.1007/s00454-015-9701-2.
- 25 Jiří Matoušek and Otfried Schwarzkopf. On ray shooting in convex polytopes. *Discrete Comput. Geom.*, 10(2):215–232, 1993. doi:10.1007/BF02573975.
- 26 M. Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9(5):471–494, 1993. doi:10.1007/BF01187036.
- 27 Marco Pellegrini. Stabbing and ray shooting in 3 dimensional space. In Raimund Seidel, editor, *Proceedings of the Sixth Annual Symposium on Computational Geometry, Berkeley, CA, USA, June 6-8, 1990*, pages 177–186. ACM, 1990. doi:10.1145/98524.98563.
- 28 Marco Pellegrini. Ray shooting and lines in space. In *Handbook of discrete and computational geometry (3rd Edition)*, CRC Press Ser. Discrete Math. Appl., pages 1093–1112. CRC, Boca Raton, FL, 2017.
- 29 Edgar A. Ramos. On range reporting, ray shooting and k -level construction. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry (Miami Beach, FL, 1999)*, pages 390–399. ACM, New York, 1999. doi:10.1145/304893.304993.
- 30 Micha Sharir and Hayim Shaul. Ray shooting and stone throwing with near-linear storage. *Comput. Geom.*, 30(3):239–252, 2005. doi:10.1016/j.comgeo.2004.10.001.

Computing Instance-Optimal Kernels in Two Dimensions

Pankaj K. Agarwal  

Department of Computer Science, Duke University, Durham, NC, USA

Sariel Har-Peled  

Department of Computer Science, University of Illinois, Urbana, IL, USA

Abstract

Let P be a set of n points in \mathbb{R}^2 . For a parameter $\varepsilon \in (0, 1)$, a subset $C \subseteq P$ is an ε -kernel of P if the projection of the convex hull of C approximates that of P within $(1 - \varepsilon)$ -factor in every direction. The set C is a *weak* ε -kernel of P if its directional width approximates that of P in every direction. Let $k_\varepsilon(P)$ (resp. $k_\varepsilon^w(P)$) denote the minimum-size of an ε -kernel (resp. weak ε -kernel) of P . We present an $O(nk_\varepsilon(P) \log n)$ -time algorithm for computing an ε -kernel of P of size $k_\varepsilon(P)$, and an $O(n^2 \log n)$ -time algorithm for computing a weak ε -kernel of P of size $k_\varepsilon^w(P)$. We also present a fast algorithm for the Hausdorff variant of this problem.

In addition, we introduce the notion of ε -core, a convex polygon lying inside $\text{ch}(P)$, prove that it is a good approximation of the optimal ε -kernel, present an efficient algorithm for computing it, and use it to compute an ε -kernel of small size.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Coreset, approximation, kernel

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.4

Related Version *Full Version*: <https://arxiv.org/abs/2207.07211> [1]

Funding *Pankaj K. Agarwal*: Work on this paper was partially supported by NSF grants IIS-18-14493, CCF-20-07556, and CCF-2223870.

Sariel Har-Peled: Work on this paper was partially supported by a NSF AF award CCF-1907400.

1 Introduction

Coresets have been successfully used as geometric summaries to develop fast approximation algorithms for a wide range of geometric optimization problems. Agarwal et al. [2] introduced the notions of ε -kernels/coresets for approximating the convex hull of a point set P in \mathbb{R}^d : For an interval $J = [a, b]$, let $(1 - \varepsilon)J = [a + (\varepsilon/2)|J|, b - (\varepsilon/2)|J|]$ be its scaling down by a factor of $1 - \varepsilon$ around its center. For a direction $v \in \mathbb{S}$, let $I_v(P)$ denote the projection of $\text{ch}(P)$ in direction v , which is an interval. A subset $C \subseteq P$ is an ε -kernel if $I_v(C) \supseteq (1 - \varepsilon)I_v(P)$ for all directions $v \in \mathbb{S}$, see Definition 4. The *weak* ε -kernels impose a weaker requirement that $|I_v(C)| \geq (1 - \varepsilon)|I_v(P)|$ for all $v \in \mathbb{S}$, see Definition 6. See Figure 1.

It is known that there exists an ε -kernel (as well as a weak ε -kernel) of P of size $O(\varepsilon^{-(d-1)/2})$ and that it can be computed efficiently [2]. However there may exist an ε -kernel of P of much smaller size, as is often the case in practice, see, e.g. [23]. Let $k_\varepsilon(P)$ be the minimum size of an ε -kernel of P . An interesting question is whether an ε -kernel of P of size k_ε can be computed efficiently, i.e., computing an *instance-optimal* ε -kernel. A similar question can be asked for weak ε -kernels. These problems are known to be NP-Hard for $d \geq 3$. Although it is generally believed that an instance-optimal ε -kernel or weak ε -kernel in the plane can be computed in polynomial time using dynamic programming, we are unaware of any paper that presents such an algorithm. See below for related work on this problem. In this paper, we settle this question by presenting fast algorithms for computing instance-optimal ε -kernels and weak ε -kernels for $d = 2$.



© Pankaj K. Agarwal and Sariel Har-Peled;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

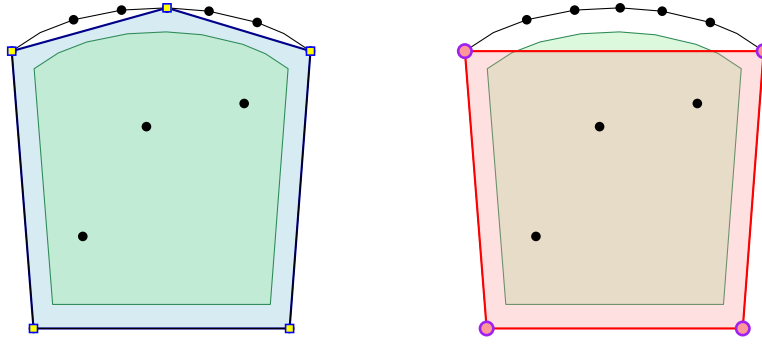
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 4; pp. 4:1–4:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Somewhat oversimplifying the difference, a regular kernel has to conceptually include a “shrunk” middle portion (left), while the weak kernel (right) only has to approximate the projections. Specifically, on the left, the projection interval of the approximation has to include the projection interval of the green region. On the right, the approximation projection interval needs to be sufficiently long but it does not have the inclusion constraint.

Related work. As mentioned above, Agarwal et al. [2] proved the existence of an ε -kernel of size $O(\varepsilon^{-(d-1)/2})$ for any set of points in \mathbb{R}^d and presented fast algorithms for computing such an ε -kernel. These algorithms were subsequently improved and generalized, see [9, 5, 3]. Yu et al. [23] studied practical algorithms for computing coresets/kernels, and suggested an incremental algorithm that seems to provide a good approximation to the optimal kernel.

The NP-Hardness of computing an instance-optimal kernel in \mathbb{R}^3 follows from that of polytope approximation [12], see also [4, 8]. Clarkson [11] studied the problem of polytope approximation as a hitting-set problem, providing a logarithmic approximation in the optimal size, that can be used for approximating the optimal kernel. For $d = 3$, the approximation factor can be improved to $O(1)$ [7]. Using a greedy approach, Blum et al. [6] studied the problem of approximating optimal kernels in high dimensions, and presented polynomial-time algorithms for computing an ε -kernel of size $O(dk_\varepsilon \log k_\varepsilon)$ or an $(\varepsilon + 8\varepsilon^{1/3})$ -kernel of size $O(k_\varepsilon \varepsilon^{-2/3})$.

More recently, there has been some work on computing variants of ε -kernels of minimum size, though *none* of them compute an instance-optimal ε -kernel. Wang et al. [22] use a different definition of kernel, so comparing the results of this paper to their work is somewhat confusing. Specifically, Wang et al. [22] presented a cubic-time algorithm that computes a minimum-size subset Q of P with the property that $\max_{p \in P} (1 - \varepsilon) \langle v, p \rangle \leq \max_{q \in Q} \langle v, q \rangle$, assuming that P is α -fat for some constant α ; they refer to such a subset as a ε -core-set of P . A shortcoming of this definition is that it is neither translation nor non-uniform-scaling invariant. However, it can be shown that their algorithm computes an ε -kernel of size at most $k_\varepsilon/3$ (observe that $k_\varepsilon/3$ can be much larger than k_ε). Klimenko and Raichel [16] provided an $O(n^{2.53})$ time algorithm for computing a minimum-size subset Q such that $H(\text{ch}(P), \text{ch}(Q))$, the Hausdorff distance between $\text{ch}(P)$ and $\text{ch}(Q)$, is at most ε .¹ They also tackle the case when P is convex, which they solve in $O(n \log^2 n)$ time. The standard approach for computing small kernels, is to apply an affine transformation to the point set to make it “fat”, then apply an algorithm for Hausdorff approximation, with parameter ε/c where c depends on the fatness of the mapped point set and its diameter. Using the algorithm

¹ Recall that for two sets $A, B \in \mathbb{R}^2$, $H(A, B) = \max\{h(A, B), h(B, A)\}$, where $h(X, Y) = \max_{x \in X} \min_{y \in Y} \|x - y\|$.

in [16], an ε -kernel of size at most $k_{\varepsilon/2}$ can be computed in $O(n^{2.53})$ time. We note that since ε is an absolute error, the size of Hausdorff-approximation can be $\Omega(n)$ in the worst case. If we set the error parameter to be $\varepsilon \cdot \text{diam}(P)$, then there exists an ε -Hausdorff approximation Q of size $O(\varepsilon^{-(d-1)/2})$ but Q may not be an ε -kernel since for a direction $v \in \mathbb{S}$, $|I_v(Q)|$ maybe as small as $|I_v(P)| - \varepsilon \text{diam}(P)$, while ε -kernel requires $I_v(Q) \supseteq (1 - \varepsilon)I_v(P)$. As such while the width or minimum-enclosing-box of an ε -kernel approximates that of P , a Hausdorff approximation does not offer such a guarantee and thus not always suitable for approximating extent measures of P .

There is also some connection between our problem and minimum-link distance and polygon approximation, see [13, 14, 18, 19, 21, 20] for some relevant results.

Our results. Let P be a set of n points in \mathbb{R}^2 , and let $\varepsilon > 0$ be a parameter. There are three main results in this paper:

- OPTIMAL KERNEL. We present (in Section 4) an $O(k_{\varepsilon} n \log n)$ -time algorithm for computing an ε -kernel of P of size $k_{\varepsilon} := k_{\varepsilon}(P)$; recall that $k_{\varepsilon} = O(\varepsilon^{-1/2})$.
- OPTIMAL WEAK KERNEL. We present (in Section 5) an $O(n^2 \log n)$ -time algorithm for computing a weak ε -kernel of P of size $k_{\varepsilon}^w(P)$, the minimum size of a weak ε -kernel of P .

Our algorithm for computing the optimal kernel can be adapted to computing an optimal Hausdorff approximation of $\text{ch}(P)$:

- OPTIMAL HAUSDORFF APPROXIMATION. We present (in [1]) an $O(k_{\varepsilon}^h n \log n)$ -time algorithm for computing a set $Q \subseteq P$ of size k_{ε}^h such that $H(\text{ch}(P), \text{ch}(Q)) \leq \varepsilon$, where k_{ε}^h is the size of the minimum such subset.

We obtain these results by reducing the computation of (weak) optimal kernel to the following two covering problems, which are of independent interest:

- OPTIMAL ARC COVER. Given a set Ξ of n arcs of the unit circle \mathbb{S} , compute its smallest subset that covers \mathbb{S} . Lee and Lee [17] had presented an $O(n \log n)$ -time algorithm for this problem, which is optimal in the worst case. Here we present a somewhat simpler algorithm with the same running time (see [1]), which is more intuitive and which we adapt to the computation of weak kernels.
- OPTIMAL STAR COVER. Given a polygon \mathcal{P} that is star shaped with respect to the origin \mathbf{o} and a set of lines L , compute a smallest subset of lines (i.e., cuts) in L that separate \mathbf{o} from $\partial\mathcal{P}$. Alternatively, this can be interpreted as covering $\partial\mathcal{P}$ by the (outer) halfplanes defined by the lines of L . We reduce this problem to the above arc-cover problem, but the number of candidate arcs can be quadratic. We use a greedy algorithm to prune the number of candidate arcs to $O(kn)$, in $O(kn \log n)$ time, where k is the size of the optimal solution, and then compute an arc cover in $O(kn \log n)$ time using the above algorithm. We reduce the computation of ε -kernel to this covering problem by using the polarity transform (see Section 4)

Finally, we introduce (in [1]) the concept of core of a point set, prove its properties, and describe an algorithm for computing it. A convex body C can be represented as the intersection of all the minimal slabs that contains it. The ε -core is the result of intersecting all these slabs after shrinking them by a factor of $1 - \varepsilon$. It induces an affine-invariant inner approximation of C . For a point set P , its ε -core is a convex polygon lying inside $\text{ch}(P)$. We describe an $O(n \log n)$ -time algorithm for computing the ε -core of P .

We show that the convex hull of any ε -kernel of P contains the ε -core of P , and that any subset $C \subseteq P$ whose convex hull contains the ε -core is a 4ε -kernel of P , see [1]. Thus the ε -core is an approximation to the optimal ε -kernel, which has the benefit of being well

defined for any bounded convex shape. We believe this notion of ε -core is new, and is of independent interest. We present an $O(n \log n)$ -time algorithm for computing the smallest subset of P such that its convex-hull contains the $\varepsilon/4$ -core of P , which yields an ε -kernel of P of size at most $k_{\varepsilon/4}$.

2 Preliminaries

Let P be a set of n points in \mathbb{R}^2 , and let $\varepsilon \in (0, 1)$ be a parameter. Without loss of generality assume that the origin \mathbf{o} lies in the interior of $\text{ch}(P)$, where $\text{ch}(P)$ denotes the convex-hull of P (if $\mathbf{o} \notin \text{ch}(P)$, one can choose three arbitrary points of P and translate P so that their centroid becomes \mathbf{o}).

Normal diagram. A direction in \mathbb{R}^2 can be represented as a unit vector in \mathbb{R}^2 . The set of unit vectors (directions) in \mathbb{R}^2 is denoted by $\mathbb{S} = \{p \in \mathbb{R}^2 \mid \|p\| = 1\}$.

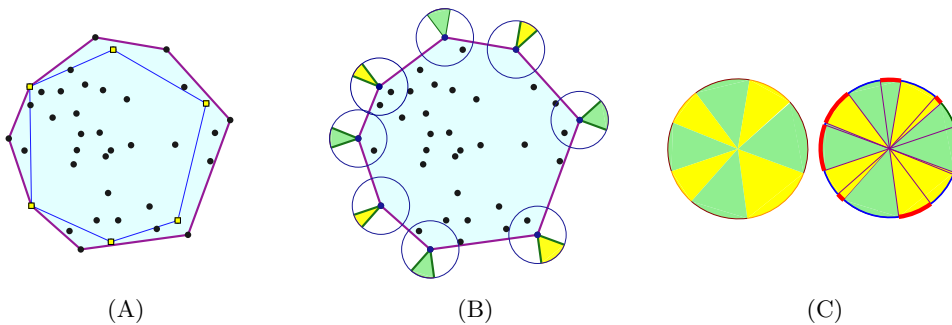
► **Definition 1.** For a line ℓ not passing through the origin, let $\mathfrak{h} = \mathfrak{h}(\ell)$ (resp. $\bar{\mathfrak{h}} = \bar{\mathfrak{h}}(\ell)$) be the (closed) halfplane bounded by ℓ and containing (resp. not containing) the origin.

For a direction $v \in \mathbb{S}$ and a point $q \in \mathbb{R}^2$, let $\mathfrak{h}_v(q)$ be the halfplane that is bounded by the line normal to direction v and passing through q , and that contains \mathbf{o} .

► **Definition 2 (Extremal point, supporting line).** For a direction $v \in \mathbb{S}$, let p_v be the **extremal point** of P in the direction v . That is $p_v = \arg \max_{p \in P} \langle v, p \rangle$. The point p_v is unique if v is not the outer normal of an edge of $\text{ch}(P)$. Similarly, let ℓ_v be the supporting line of $\text{ch}(P)$ normal to v and passing through p_v . Let $\mathfrak{h}_v = \mathfrak{h}(\ell_v)$ and $\bar{\mathfrak{h}}_v = \bar{\mathfrak{h}}(\ell_v)$. Observe that $\text{ch}(P) \subset \mathfrak{h}_v$.

For a real number ψ , let $\mathfrak{h}_v \ominus \psi$ and $\bar{\mathfrak{h}}_v \ominus \psi$ be the halfplanes formed by translating \mathfrak{h}_v and $\bar{\mathfrak{h}}_v$, respectively, towards the origin by distance ψ .

► **Definition 3.** The **normal diagram** of P is the partition of \mathbb{S} into maximal intervals so that the extremal point p_v remains the same for all directions within an interval. The endpoints of these intervals correspond to the outer normals of the edges of $\text{ch}(P)$. The normal diagram can be further refined so that for all directions v within each interval, both p_v and p_{-v} remain the same. Such a pair of points are **antipodal pairs**. Let $\mathcal{N} = \mathcal{N}(P)$ denote this **refinement** of the normal diagram, and observe that $|\mathcal{N}| \leq 2n$. See Figure 2.



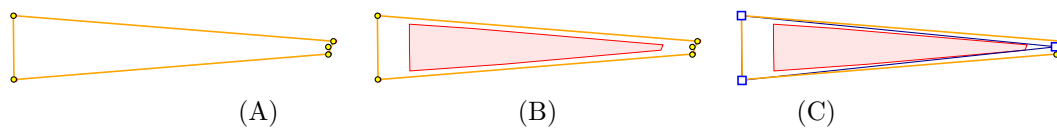
► **Figure 2** (A) Point set P , $\text{ch}(P)$, and ε -kernel of P (say, for $\varepsilon = 0.2$). (B) Directions in which a point is extremal. (C) Normal diagram of P and its refinement $\mathcal{N}(P)$, I_v .

Directional width and ε -kernel. For a direction $v \in \mathbb{S}$, let

$$I_v(P) = \left[\min_{p \in P} \langle v, p \rangle, \max_{p \in P} \langle v, p \rangle \right]$$

denote the **projection interval** of P in direction v . Its length $\bar{\omega}(v, P) = \|I_v\|$ is the **directional width** of P in the direction of v . Note that $I_v = -I_{-v}$ and $\bar{\omega}(v, P) = \bar{\omega}(-v, P)$. For an $\varepsilon \in (0, 1)$ and an interval $J = [x, y]$, let $(1 - \varepsilon)J$ be the shrinking of J by a factor of $(1 - \varepsilon)$, i.e., $(1 - \varepsilon)J = [x + (\varepsilon/2)|J|, y - (\varepsilon/2)|J|]$.

► **Definition 4.** A set $X \subseteq \text{ch}(P)$ is an **ε -approximation** of P if $I_u(X) \supseteq (1 - \varepsilon)I_u(P)$ for all directions $u \in \mathbb{S}$. A subset $C \subseteq P$ is a “strong” **ε -kernel** of P if it is an ε -approximation of P . Let $k_\varepsilon(P)$ denote the minimum size of an ε -kernel of P . See Figure 3 for an example.



■ **Figure 3** (A) A point set and its convex hull. (B) Its 0.2-core. (C) Its optimal 0.2-kernel – observe that it contains points that are not on the convex-hull.

We emphasize that the shrinking here is done for every direction individually around the center of the projection interval – in particular, there is no center point of the $\text{ch}(P)$ around which we do the scaling – to some extent this gives rise to most of the technical difficulties in constructing and approximating an optimal kernel. The following property of ε -approximation will be useful later on.

► **Lemma 5** ([2]). Let P be a point set in \mathbb{R}^d , $X \subseteq \text{ch}(P)$, and T an affine map in \mathbb{R}^d . X is an ε -approximation for $P \iff T(X)$ is an ε -approximation of $T(P)$.

A slightly weaker notion of ε -kernel was used by Agarwal et al. [2], that is potentially (significantly) smaller than their “strong” counterparts but somewhat harder to compute.

► **Definition 6.** A subset $C \subseteq P$ is a **weak ε -kernel** of P if $\bar{\omega}(u, C) \geq (1 - \varepsilon)\bar{\omega}(u, P)$ for all $u \in \mathbb{S}$.

This weaker definition was sufficient for the purposes of Agarwal et al.. However, it is less intuitive than the stronger variant, and it is harder to compute the optimal weak kernel.

Computing optimal circular arc cover. Let Ξ denote a set of n circular arcs on \mathbb{S} , each of length less than π , that cover \mathbb{S} .² As mentioned in the introduction, an $O(n \log n)$ -time algorithm for computing the smallest subset of Ξ that cover \mathbb{S} was proposed in [17]. In the full version [1] we present an alternative $O(n \log n)$ time algorithm for computing the smallest-size arc cover from Ξ , which we believe is simpler and more intuitive. The basic idea is to use the greedy algorithm. Picking a start arc, and then going counterclockwise as far one can adding arcs in a greedy fashion results in a cover of size $k + 1$, where k is the optimal size. After an $O(n \log n)$ preprocessing, the greedy algorithm can be executed in $O(k)$ time. To reduce the size of the solution to k , one has to guess a starting arc that belongs to the optimal solution. We show that the least covered point on the circle is covered

² By computing the union of arcs in Ξ , we can decide, in $O(n \log n)$ time, whether Ξ covers \mathbb{S} .

by $O(n/k)$ intervals. This implies that one has to try only $O(n/k)$ starting arcs and thus run the greedy algorithm $O(n/k)$ times. The overall running time is thus $O(n \log n)$. See [1] for full details. We will use this algorithm as a subroutine in Section 4 and [1] and a variant of it in Section 5. In particular, we get the following result:

► **Theorem 7.** *Let Ξ be a set of n circular arcs on \mathbb{S} . The optimal cover of \mathbb{S} by the arcs of Ξ , if there exists one, can be computed in $O(n \log n)$ time.*

3 Covering a Star Polygon by Halfplanes

The input is a set of L of n lines and a polygon \mathcal{Z} with $O(n)$ vertices that is star-shaped with respect to the origin \mathfrak{o} (i.e., for every point $p \in \mathcal{Z}$, $\mathfrak{o}p \subseteq \mathcal{Z}$). Formally, the task at hand is to compute a minimum set of lines $C \subseteq L$, such that for any point $p \in \partial\mathcal{Z}$, $\text{int}(\mathfrak{o}p)$ intersects a line of C . Geometrically, $\mathbb{F}_0(C) := \bigcap_{\ell \in C} \bar{h}(\ell)$, the intersection of inner halfplanes bounded by lines in C , is contained in \mathcal{Z} . An alternative interpretation of this problem is that $\partial\mathcal{Z} \subset \bigcup_{\ell \in C} \bar{h}(\ell)$.

3.1 Reduction to arc cover

$\partial\mathcal{Z}$ can be viewed as the image of a function $\mathcal{Z} : \mathbb{S} \rightarrow \mathbb{R}^2$. Specifically, for a direction $u \in \mathbb{S}$, $\mathcal{Z}(u)$ is the intersection point of $\partial\mathcal{Z}$ with the ray from the origin in direction u . A line ℓ **blocks** the direction u if ℓ intersects the segment $\mathfrak{o}\mathcal{Z}(u)$. A subset $G \subseteq L$ is a **blocking set** of \mathcal{Z} if each direction in \mathbb{S} is blocked by at least one line of G (i.e., $\mathbb{F}_0(G) \subset \mathcal{Z}$).

Fix a line $\ell \in L$. Let $\ell \cap \mathcal{Z}$ denote the set of connected components (i.e., segments) of $\ell \cap \mathcal{Z}$. For a segment $s \in \ell \cap \mathcal{Z}$, let $\mathfrak{x}s = \{\mathfrak{o}p / \|\mathfrak{o}p\| \in \mathbb{S} \mid p \in s\}$ be the circular arc induced by s . All directions in $\mathfrak{x}s$ are blocked by ℓ . Let $\mathfrak{x}\ell = \{\mathfrak{x}s \mid s \in \ell \cap \mathcal{Z}\}$ be the set of all circular arcs that are induced by blocking segments of ℓ . Let $\Xi = \bigcup_{\ell \in L} \mathfrak{x}\ell$ be the set of all circular arcs defined by the lines of L . For a subset $\Gamma \subseteq \Xi$, let $L(\Gamma) = \{\ell \in L \mid \gamma \in \mathfrak{x}\ell, \gamma \in \Gamma\}$ be the original subset of lines of L supporting the arcs of Γ .

► **Lemma 8.** (i) *If $\Gamma \subseteq \Xi$ is an arc cover, i.e., $\bigcup \Gamma = \mathbb{S}$, then $L(\Gamma)$ is a blocking set.*
(ii) *There is an arc cover $\Gamma \subseteq \Xi$ of size k if and only if there is a blocking set $G \subseteq L$ of size k .*

Proof. (i) If Γ is an arc cover, then for every direction $u \in \mathbb{S}$, there is an arc $\mathfrak{x}s \in \Gamma$ that blocks the direction u . If $\mathfrak{x}s \in \mathfrak{x}\ell$, for a line $\ell \in L(\Gamma)$, then the segment $\mathfrak{o}\mathcal{Z}(u)$ intersects ℓ . Since this condition holds for all directions in \mathbb{S} , it follows $L(\Gamma) \subseteq L$ is a blocking set.

(ii) If there is an arc cover $\Gamma \subseteq \Xi$ of size k , then by part (i), $L(\Gamma)$ is a blocking set of size at most k . Conversely, let G be a blocking set for \mathcal{Z} . Without loss of generality, we can assume that each line of G appears as an edge on the boundary of the face F of $\mathcal{A}(G)$ that contains the origin, because otherwise we can remove the line from G . For each line $\ell \in G$, let $s_\ell \in \ell \cap \mathcal{Z}$ be the segment that contains the edge of F lying on ℓ . Since $F \subseteq \mathcal{Z}$, the segment $\mathfrak{o}\mathcal{Z}(u)$ intersects an edge of F for every $u \in \mathbb{S}$. Hence, $\{\mathfrak{x}s_\ell \mid \ell \in G\}$ is an arc cover of size at most $|G|$. ◀

By Lemma 8, it suffices to compute smallest-size arc cover from Ξ . But $|\Xi| = \Theta(n^2)$ in the worst case. Therefore computing Ξ explicitly and then using Theorem 7 to compute an arc cover take $O(n^2 \log n)$ time. In the following, we show how to improve the running time to $O(nk \log n)$, where k is the optimal solution size.

3.2 Computing an almost-optimal blocking set

We extend the greedy algorithm used in the circular arc cover (see Section 2 and [1]) to compute an arc cover in Ξ without computing Ξ explicitly. For clarity, we describe the greedy algorithm in terms of computing a blocking set.

For a pair of directions $u, v \in \mathbb{S}$, let $\mathcal{Z}(u, v] \subseteq \mathcal{Z}$ be the semiopen subchain of \mathcal{Z} from $\mathcal{Z}(u)$ to $\mathcal{Z}(v)$ in the counterclockwise direction, which contains the endpoint $\mathcal{Z}(v)$ but not $\mathcal{Z}(u)$. As such, we have $\mathcal{Z}(u, u] = \mathcal{Z}$.

We define a (partial) function $\mathfrak{s} : \mathbb{S} \times L \rightarrow \mathbb{R}^4$, as follows. For a pair $u \in \mathbb{S}$ and a line $\ell \in L$, if ℓ does not intersect the segment $\circ\mathcal{Z}(u)$, then $\mathfrak{s}(u, \ell)$ is not defined. Otherwise, it is the segment of $\ell \cap \mathcal{Z}$ that intersects $\circ\mathcal{Z}(u)$. Similarly, we define a (partial) function $\mathfrak{f} : \mathbb{S} \times L \rightarrow \mathbb{S}$, that is the first point of $\mathfrak{s}(u, \ell)$ in the counter-clockwise direction after $\mathcal{Z}(u)$ (note, that ℓ might intersect the boundary \mathcal{Z} many times). Set $\lambda(u) = \arg \max_{\ell \in L} \mathfrak{f}(u, \ell)$, i.e., among the feasible segments that intersect $\circ\mathcal{Z}(u)$, $\lambda(u)$ is the last one to exit \mathcal{Z} in the counterclockwise direction.

The algorithm consists of the following steps: Set $v_0 := (1, 0)$, $\ell_0 := \lambda(v_0)$, $G := \{\ell_0\}$, and $i := 1$. In the i th iteration, the algorithm does the following: it sets $v_i = \mathfrak{f}(v_{i-1}, \ell_{i-1})$, $\ell_i = \lambda(v_i)$, and $G = G \cup \{\ell_i\}$. The algorithm then continues to the next iteration till $\mathbb{F}_0(G) \subseteq \text{int}(\mathcal{Z})$. Let v'_1 be the first intersection point of ℓ_0 with \mathcal{Z} in the clockwise direction from v_0 , i.e., the segment $\mathcal{Z}(v_1)\mathcal{Z}(v'_1)$ lies inside \mathcal{Z} . Then the terminating condition is the same as $\mathfrak{f}(v_i, \ell_i)$ lying after v'_1 (from v_i) in the counterclockwise direction. By construction, $\mathbb{F}_0(G) \subset \mathcal{Z}$. Since this is a greedy algorithm for computing an arc cover, $|G| \leq k + 1$. The polygon \mathcal{Z} can be preprocessed, in $O(n \log n)$ time, into a data structure of linear size so that for a pair $u \in \mathbb{S}$ and a line $\ell \in L$, $\mathfrak{f}(u, \ell)$ can be computed in $O(\log n)$ time [10, 15]. The algorithm performs $O(nk)$ such queries, so the total running time is $O(nk \log n)$.

► **Lemma 9.** *Let L be a set of n lines, \mathcal{Z} be a polygon with $O(n)$ vertices that is star shaped with respect to \circ and that contains $\mathbb{F}_0(L)$, and let k be the size of the smallest blocking set in L for \mathcal{Z} . A blocking set $G \subseteq L$ of size at most $k + 1$ can be computed in $O(kn \log n)$ time.*

3.3 Computing an optimal solution

Let G be the blocking set computed by the above greedy algorithm. For each line $\ell \in L$ we compute its intersection points with the lines of G . For each such intersection point ξ , if ξ lies inside \mathcal{Z} , let $s_\xi \in \mathcal{P} \cap \ell$ be the segment that contains ξ . Let S_1 be the set of resulting $O(nk)$ segments. Let

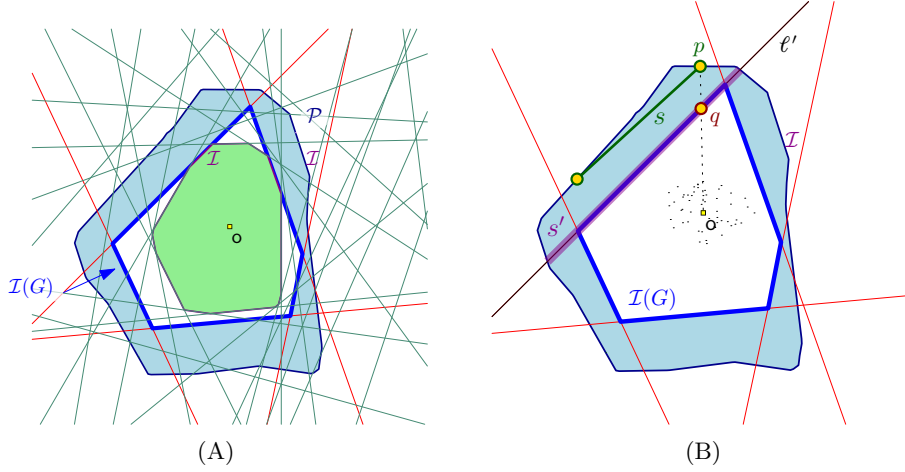
$$S_2 = \bigcup_{\ell \in G} \ell \cap \mathcal{P}$$

be the set of all segments induced by the lines of G . Set $S = S_1 \cup S_2$. The computes the set $\Gamma = \{\mathfrak{x}s \mid s \in S\}$, and then computes the minimal size arc cover C of \mathbb{S} by the arcs of Γ . The returned set is $K = \{\ell \in L \mid \mathfrak{x}s \in C, s \subset \ell\}$.

► **Lemma 10.** *The set Γ contains an arc cover of size k .*

Proof. Suppose for the sake of contradiction that Γ does not contain an arc cover of size k . Let $C \subset \Xi$ be an arc cover of size k . Then C contains an arc $\mathfrak{x}s$ such that s lies on a line of $L \setminus G$ and s does not intersect any line of G , i.e., it lies in the interior of a face of $\mathcal{A}(G)$, the arrangement of G .

If s lies in the face corresponding to $\mathbb{F}_0(G)$, then s must intersect $\partial\mathbb{F}_0(G)$, as the endpoints of s lies on $\partial\mathcal{Z}$ and $\mathcal{I}(G) \subseteq \mathcal{Z}$, contradicting the assumption that s does not intersect any line of G .



■ **Figure 4** (A) A hitting set G of size at most $k + 1$. (B) Illustration of the proof of Lemma 10.

Next, suppose s lies in some other face of $\mathcal{A}(G)$. Let p be an endpoint of s . The segment po must intersect a line $\ell' \in G$ at a point q . In particular, let $s' \in \ell' \cap \mathcal{Z}$ be the segment of ℓ' containing q . Clearly, s' is a blocker for all the points on s , so we can obtain another optimal solution by replacing $\mathcal{A}s$ with $\mathcal{A}s'$ (see Figure 4), and this solution has one more arc of Γ , a contradiction.

Hence, we can conclude that Γ contains an optimal arc cover. ◀

Computing the set G takes $O(nk \log n)$ time. Observe that $|S_1| = O(nk)$, as each line of L induces at most $k + 1$ segments in this set. Similarly, as $|G| = k + 1$, we have that $|S_2| = O(nk)$. It follows that computing S_1 and S_2 requires $O(nk)$ ray-shooting queries in \mathcal{Z} , and these queries overall take $O(nk \log n)$ time. Hence, we obtain the following:

► **Lemma 11.** *Let L be a set of n lines in the plane, and let \mathcal{Z} be a polygon with $O(n)$ vertices that is star shaped with respect to \circ and that contains $\mathbb{F}_\circ(L)$. Then a blocking set from L of \mathcal{Z} of size k can be computed, in $O(kn \log n)$ time, where k is the size of the optimal solution.*

4 Computing Optimal ε -Kernel

Let P be a set of n points in \mathbb{R}^2 and $\varepsilon \in (0, 1)$ a parameter. We describe an $O(nk_\varepsilon \log n)$ -time algorithm for computing an ε -kernel of size k_ε . We use polarity to construct a set L of n lines and a star polygon \mathcal{Z} that contains $\mathbb{F}_\circ(L) = \bigcap_{\ell \in C} \mathfrak{h}(\ell)$. An ε -kernel of P corresponds to a blocking set in L for \mathcal{Z} .

► **Definition 12** (ε -shifted supporting line). *For a direction $u \in \mathbb{S}$ and a parameter $\varepsilon > 0$, let $\ell_{u,\varepsilon}$ be the boundary line of $\mathfrak{h}_{u,\varepsilon} = \mathfrak{h}_u \ominus (\varepsilon/2)\overline{\omega}(u, P)$, see Definition 2. Let $\overline{\mathfrak{h}}_{u,\varepsilon}$ be the (closed) complement halfplane to $\mathfrak{h}_{u,\varepsilon}$.*

Set $\mathcal{H}_\varepsilon = \{\overline{\mathfrak{h}}_{u,\varepsilon} \mid u \in \mathbb{S}\}$. The following lemma is immediate from the definition of ε -kernel.

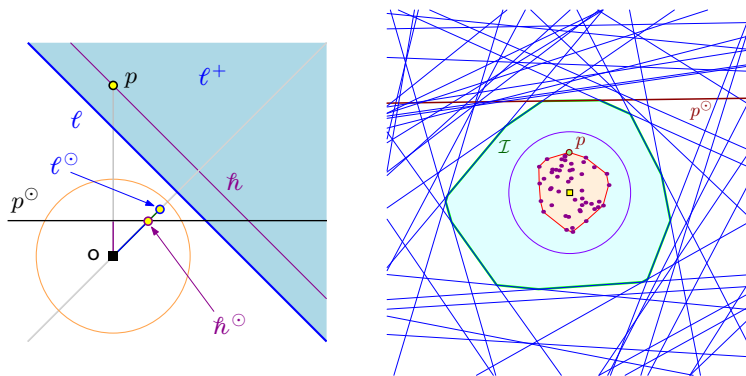
► **Lemma 13.** *Given a point set P in \mathbb{R}^2 and a parameter $\varepsilon \in (0, 1)$, a subset $C \subseteq P$ is an ε -kernel of P if and only if $\overline{\mathfrak{h}}_{v,\varepsilon} \cap C \neq \emptyset$ for all $u \in \mathbb{S}$, i.e., C is a hitting set of \mathcal{H}_ε .*

The problem of computing an ε -kernel thus reduces to computing a minimum-size hitting set of the infinite set \mathcal{H}_ε . It will be convenient to use the polarity transform and work in the mapped plane, so we first describe the polar of ε -kernel and then describe the algorithm.

Polarity. For a point $p \neq o$, its **inversion**, through the unit circle, is the point $p^{-1} = p / \|p\|^2$. Observe that p, p^{-1}, o are collinear, $\|p\| \|p^{-1}\| = 1$, and p and p^{-1} are on the same side of the origin on this line. We use the *polarity transform*, which maps a point $p = (a, b) \neq o$ to the line

$$p^\circ \equiv ax + by - 1 = 0 \equiv \langle p, (x, y) \rangle - 1 = 0 \equiv \left\langle p, (x, y) - \frac{p}{\|p\|^2} \right\rangle = 0.$$

Namely, the line p° is orthogonal to the vector op , and the closest point on p° to the origin is p^{-1} . Geometrically, a point p is being mapped to the line passing through the inverted point p^{-1} and orthogonal to the vector op^{-1} . Similarly, for a line ℓ , its *polar* point ℓ° is q^{-1} , where q is the closest point to the origin on ℓ . Observe that $(\ell^\circ)^\circ = \ell$ and $(p^\circ)^\circ = p$ for any line ℓ and any point p .



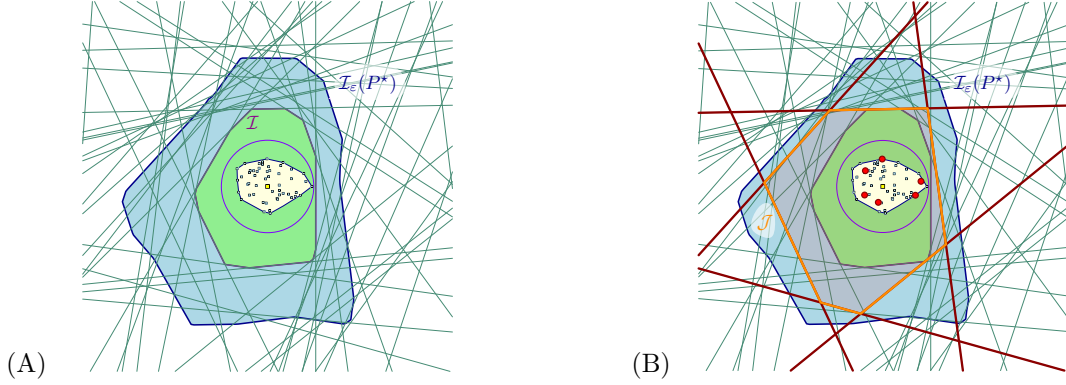
■ **Figure 5** Left: A point p lies in the halfplane $\bar{h}(\ell) \iff p^\circ$ intersects the segment $o\ell^\circ$. Right: A convex hull of a point set, and the corresponding “polar” polygon formed by the intersection of halfplanes.

If a point p lies on a line ℓ then $\ell^\circ \in p^\circ$. If p lies in the halfplane $\bar{h}(\ell)$ (by Definition 1, we have $o \notin \bar{h}(\ell)$) if and only if p° intersects the segment $o\ell^\circ$, see Figure 5 (left). Set $P^\circ = \{p^\circ \mid p \in P\}$ and $\bar{F}_0 := \bar{F}_0(P^\circ) = \bigcap_{p \in P} \bar{h}(p^\circ)$. Then the polygon \bar{F}_0 is the polar of $\text{ch}(P)$, namely:

- I. If $p \in P$ is a vertex of $\text{ch}(P)$ then p° contains an edge of \bar{F}_0 , see Figure 5 (right).
- II. The polar of line ℓ missing (resp. intersecting) $\text{ch}(P)$ is a point lying in (resp. out) \bar{F}_0 .
- III. For a point $p \in \text{ch}(P)$, $\bar{F}_0 \subset \bar{h}(p^\circ)$.

Consider any direction $u \in \mathbb{S}$. Let p_u be the extremal point of P in direction u , and let ℓ_u be the corresponding supporting line, see Definition 2. The point ℓ_u° lies on the edge of \bar{F}_0 supported by p_u° , and $\ell_u^\circ / \|\ell_u^\circ\| = u$. Similarly, the polar of the shifted supporting line $\ell_{u,\varepsilon}$ (see Definition 12), is the point $\ell_{u,\varepsilon}^\circ$ which lies outside \bar{F}_0 on the ray induced by u (starting at the origin).

Kernel and polarity. Returning to ε -kernels, let \mathcal{N} be the refinement of the normal diagram of $\text{ch}(P)$, see Definition 3. Recall that \mathcal{N} is centrally symmetric. The supporting lines ℓ_u and ℓ_{-u} support the same pair of vertices of $\text{ch}(P)$ for all directions u lying inside an interval of \mathcal{N} . For each interval $\gamma \in \mathcal{N}$, let $-\gamma$ denote its antipodal interval. For each interval $\gamma \in \mathcal{N}$, let p_γ be the supporting vertex of $\text{ch}(P)$ for all directions in γ .



■ **Figure 6** (A) $\text{ch}(P)$, $\mathcal{I}(P^\circ)$, $\mathcal{I}_\varepsilon(P^\circ)$. (B) ε -kernel C and its polar C° ; $\text{ch}(P) \subseteq \mathcal{I}P^\circ \subseteq \mathcal{I}_\varepsilon(C^\circ)$.

Let $\bar{p}_{\gamma,\varepsilon} = (1 - \varepsilon/2)p_\gamma + (\varepsilon/2)p_{-\gamma}$. It can be verified that the line $\ell_{v,\varepsilon}$ for $v \in \gamma$ passes through $\bar{p}_{\gamma,\varepsilon}$. Therefore the polar of the set of lines $\{\ell_{v,\varepsilon} \mid v \in \gamma\}$ is a segment e_γ that lies on the line $(\bar{p}_{\gamma,\varepsilon})^\circ$ and outside \mathbb{F}_\circ . The sequence $\langle e_\gamma \mid \gamma \in \Gamma \rangle$ forms the boundary of a polygon $\mathcal{I}_\varepsilon(P^\circ)$ that is star shaped with respect to \circ and that contains \mathbb{F}_\circ in its interior. See Figure 6. Putting everything together, we obtain the following lemma, which characterizes the ε -kernel after polarity.

► **Lemma 14.** *Let P be a set of n points in \mathbb{R}^2 and $\varepsilon \in (0, 1)$ a parameter. The star-shaped polygon $\mathcal{I}_\varepsilon(P^\circ)$ can be computed in $O(n \log n)$ time. Furthermore, a subset $C \subseteq P$ is an ε -kernel of P if and only if C° is a blocking set for $\mathcal{I}_\varepsilon(P^\circ)$ (see Figure 6).*

Computing the smallest set $C \subseteq P$ thus reduces to the star-polygon-cover problem. Using Lemma 11 and that there is an ε -kernel of size $O(\varepsilon^{-1/2})$ [2], we obtain the following:

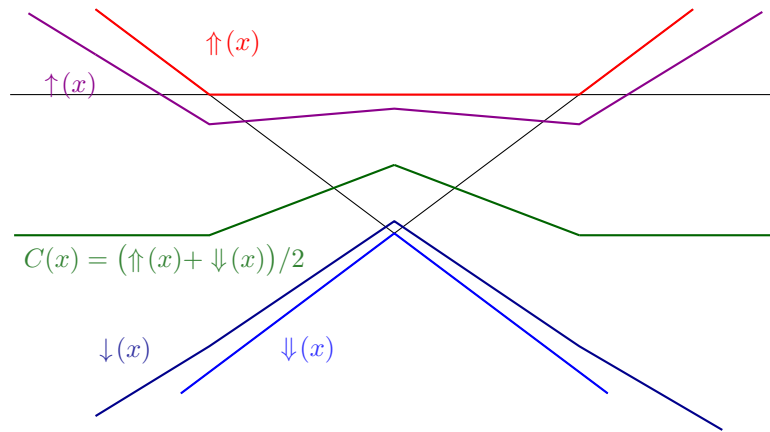
► **Theorem 15.** *Let P be a set of n points in \mathbb{R}^2 , and let $\varepsilon \in (0, 1)$ a parameter. An optimal ε -kernel of P of size k can be computed in $O(kn \log n)$ time. In the worst case, $k = O(\varepsilon^{-1/2})$, and the running time is $O(\varepsilon^{-1/2}n \log n)$.*

Below we show that there exists a set P of points such that there are quadratic number of intersections between P° and $\mathcal{I}_\varepsilon(P^\circ)$. This suggest that our somewhat more involved algorithm using greedy algorithm to prune the set of arcs used is necessary even in this case. It will be more convenient to use the duality transform instead of polarity for describing the lower-bound construction.

Duality and ε -kernel. The duality transform provides a similar mapping to polarity. The **dual point** to the line $\ell \equiv y = ax + b$ is the point $\ell^* = (a, -b)$. Similarly, for a point $p = (c, d)$ its **dual line** is $p^* \equiv y = cx - d$. Namely, for $p = (a, b)$, the dual line is $p^* \equiv y = ax - b$, and for a line $\ell \equiv y = c'x + d'$ the dual point is $\ell^* = (c', -d')$. The following interpretation of kernels in the dual is standard, and goes back to the original work of Agarwal et al. [2]. As such, we state the problem in these settings without proving the equivalence.

For a set of lines $L = P^* = \{p^* \mid p \in P\}$ in the plane (i.e., L is a set of affine functions from \mathbb{R} to \mathbb{R}), let

$$\uparrow_L(x) = \max_{f \in L} f(x) \quad \text{and} \quad \downarrow_L(x) = \min_{f \in L} f(x),$$



■ **Figure 7** Lower and upper envelopes, and their ε -approximations.

be the upper and lower envelopes of L , respectively. The function $\uparrow(x)$ is convex, while $\downarrow(x)$ is concave. The **extent** of L is

$$\Downarrow_L(x) = \uparrow_L(x) - \downarrow_L(x).$$

For a fixed $\varepsilon \in (0, 1)$, the ε -upper envelope and ε -lower envelope are

$$\uparrow_L(x) = \uparrow_L(x) - \frac{\varepsilon}{2} \Downarrow_L(x) = \left(1 - \frac{\varepsilon}{2}\right) \uparrow_L(x) + \frac{\varepsilon}{2} \downarrow_L(x)$$

$$\downarrow_L(x) = \downarrow_L(x) + \frac{\varepsilon}{2} \Downarrow_L(x) = \frac{\varepsilon}{2} \uparrow_L(x) + \left(1 - \frac{\varepsilon}{2}\right) \downarrow_L(x),$$

respectively. Unfortunately, these functions are not necessarily convex, as demonstrated in Figure 7.

Computing an optimal ε -kernel for P is equivalent to computing a set of lines $M \subseteq L$, such that $\uparrow_M(x)$ lies above $\uparrow_L(x)$ (and of course below $\uparrow_L(x)$), for all x . And similarly, $\downarrow_M(x)$ lies below $\downarrow_L(x)$, for all x .

Lower-bound construction. Here we show that in the worst case the set $\bigcup_{\ell \in L} (\ell \cap P)$ can have quadratic size. In particular, we construct a set of lines L , where the lines of L have quadratic number of intersections with $\uparrow(\cdot)$ and $\downarrow(\cdot)$.

Consider the parabolas $f(x) = \frac{2}{\varepsilon}(x^2 + 1)$ and $g(x) = -\frac{1}{1-\varepsilon/2}(x^2 + 1)$. Fix parameters n and ε . Let $p_i = (i/2n, f(i/2n))$ and $q_i = (i/2n, g(i/2n))$, for $i = 0, \dots, 2n$. For a pair of distinct points $p, q \in \mathbb{R}^2$, let $\ell(p, q)$ denote the line passing through p and q . Let

$$L_f = \{\ell(p_i, p_{i+2}) \mid i = 0, 2, 2n - 2\} \quad \text{and} \quad L_g = \{\ell(q_i, q_{i+2}) \mid i = 1, 3, 2n - 3\}.$$

The upper envelope of L_f in the range $[0, 1]$ is above $f(x)$, except for touching it at the points p_0, p_2, \dots, p_{2n} . Similarly, the lower envelope of L_g , in the range $I = [1/2n, 1 - 1/2n]$ lies below g , except for touching it at the points $q_1, q_3, \dots, q_{2n-1}$.

It is easy to verify that the lines of L_f and L_g do not intersect each other in the range $x \in [0, 1]$. As such, the upper envelope (resp. lower envelope) of $L = L_f \cup L_g$ in this range is realized by the upper envelope (resp. lower envelope) of L_f (resp. L_g).

Consider a value $x \in \{1/2n, 3/2n, \dots, (2n - 1)/2n\}$. We have that $\uparrow_L(x) > f(x)$ and $\downarrow_L(x) = g(x)$. As such, we have

$$\uparrow_L(x) = \frac{\varepsilon}{2} \uparrow_L(x) + \left(1 - \frac{\varepsilon}{2}\right) \downarrow_L(x) > \frac{\varepsilon}{2} f(x) + \left(1 - \frac{\varepsilon}{2}\right) g(x) = x^2 + 1 - (x^2 + 1) = 0.$$

4:12 Computing Instance-Optimal Kernels in Two Dimensions

Similarly, for $x \in \{2/2n, 4/2n, \dots, (2n-2)/2n\}$, we have $\uparrow_L(x) = f(x)$ and $\downarrow_L(x) < g(x)$. As such, we have

$$\uparrow_L(x) = \frac{\varepsilon}{2} \uparrow_L(x) + \left(1 - \frac{\varepsilon}{2}\right) \downarrow_L(x) < \frac{\varepsilon}{2} f(x) + \left(1 - \frac{\varepsilon}{2}\right) g(x) = 0.$$

We thus obtain the following.

► **Lemma 16.** *For any $\varepsilon > 0$ and for any $n \geq 1$, there exists a set of $2n$ lines in \mathbb{R}^2 whose ε -upper envelope crosses the x -axis at least $2n - 2$ times.*

Next, we replicate the x -axis by sufficiently close (almost parallel) n lines that lie between the lower and upper envelopes of L , and we add them to L . Then there are $\Omega(n^2)$ intersection points between \uparrow_L and the lines of L . We thus get the following result.

► **Lemma 17.** *There exists a set L of n lines in \mathbb{R}^2 such that the number of intersection points between $\mathcal{I}_\varepsilon(L)$ and L is $\Omega(n^2)$.*

5 Optimal Weak Kernel

The above results dealt with the stronger notion of a kernel, but the original work of Agarwal et al. [2] defined a weaker notion of a kernel, see Definition 6. In this section, we present an $O(n^2 \log n)$ -time algorithm for computing an optimal weak ε -kernel, by reducing it to computing a smallest arc cover, with some additional properties, in a set of $O(n^2)$ unit arcs (i.e., arcs on the unit circle).

Let P be a set of n points in \mathbb{R}^2 and $\varepsilon \in (0, 1)$ a parameter. We parametrize \mathbb{S} with the orientation in the range $[-\pi, \pi]$ (with the two endpoints of this interval being glued together), and let $u(\theta) = (\cos \theta, \sin \theta)$. Recall that a subset $C \subseteq P$ is an weak ε -kernel of P if

$$\bar{\omega}(u(\theta), C) \geq (1 - \varepsilon) \bar{\omega}(u(\theta), P) \quad (1)$$

for all $\theta \in [-\pi, \pi]$. Since $\bar{\omega}(u(\theta), P) = \bar{\omega}(u(-\theta), P)$, it suffices to satisfy Eq. (1) for the angular interval $[-\pi/2, \pi/2]$. However, it will be convenient to work with the entire \mathbb{S} , so let

$$\Downarrow_P(\theta) = \bar{\omega}(u(\theta/2), P) \quad \text{for } \theta \in [-\pi, \pi].$$

A subset $C \subseteq P$ is a **weak ε -kernel** if and only if

$$\Downarrow_C(\theta) \geq (1 - \varepsilon) \Downarrow_P(\theta) \quad \forall \theta \in [-\pi, \pi].$$

For a pair $1 \leq i < j \leq n$ and $\theta \in [-\pi, \pi]$, we define $\gamma_{ij} \in [-\pi, \pi] \rightarrow \mathbb{R}_{\geq 0}$ as

$$\gamma_{ij}(\theta) := |\langle u(\theta/2), p_i - p_j \rangle| = |(a_i - a_j) \cos(\theta/2) + (b_i - b_j) \sin(\theta/2)|,$$

where $p_i = (a_i, b_i)$. Set $\Gamma = \{\gamma_{ij} \mid 1 \leq i < j \leq n\}$. It is easily seen that $\uparrow_\Gamma(\theta) = \Downarrow_P(\theta)$. For a pair $1 \leq i < j \leq n$, we define $I_{ij} = \{\theta \in [-\pi, \pi] \mid \gamma_{ij}(\theta) \geq (1 - \varepsilon) \uparrow_\Gamma(\theta)\}$.

► **Lemma 18.** *The set I_{ij} is a single connected circular arc.*

Proof. It is convenient to reparameterize γ_{ij} . More precisely, we define the function $\xi_{ij} : \mathbb{R} \rightarrow \mathbb{R}_{\geq 0}$ as

$$\xi_{ij}(x) = |(a_i - a_j) + (b_i - b_j)x| \quad \text{for } x \in \mathbb{R}. \quad (2)$$

Set

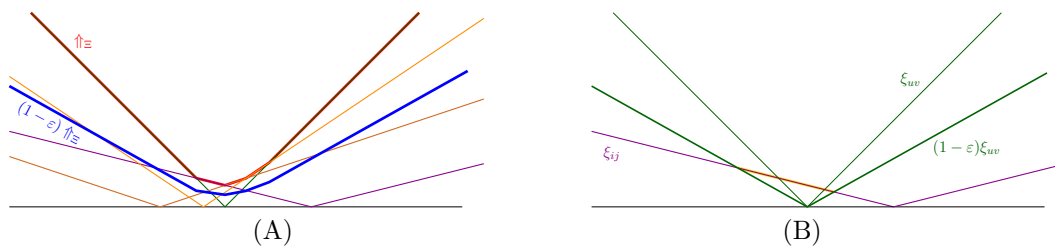
$$\Xi = \{\xi_{ij} \mid 1 \leq i < j \leq n\} \quad \text{and} \quad J_{ij} = \{x \in \mathbb{R} \mid \xi_{ij}(x) \geq (1 - \varepsilon) \uparrow_{\Xi}(x)\}.$$

Note that

$$\gamma_{ij}(\theta) = \frac{1}{\sqrt{1 + \tan^2(\theta/2)}} \xi_{ij}(\tan(\theta/2)),$$

therefore $\tan(\theta/2) \in J_{ij}$ if and only if $\theta \in I_{ij}$.

The graph of ξ_{ij} is a cone with axis of symmetry around the y -axis and apex on the x -axis – specifically, there are two numbers α_{ij}, β_{ij} such that $\xi_{ij}(x) = \alpha_{ij}|x - \beta_{ij}|$. The number α_{ij} is the **slope** of ξ_{ij} . The function $(1 - \varepsilon) \uparrow_{\Xi}$ is a convex chain, which is the upper envelope of the functions $(1 - \varepsilon)\xi_{ij}$, see Figure 8 (A).



■ **Figure 8** Illustration of the proof of Lemma 18. (A) Upper envelope \uparrow_{Ξ} , and lower-bound curve $(1 - \varepsilon) \uparrow_{\Xi}$. (B) A cone with higher slope “buries” at least one leg of the other cone.

Since the graph of ξ_{ij} is composed of two rays, J_{ij} is potentially the union of two intervals (potentially infinite rays). If J_{ij} does not contain any finite interval, i.e., consists of two rays, then I_{ij} is a single arc containing the orientation π . So assume that J_{ij} contains a finite interval, see Figure 8 (B). This implies that there are indices u, v , such that $(1 - \varepsilon)\xi_{uv}$ has higher slope than ξ_{ij} . But then $(1 - \varepsilon)\xi_{uv}$ is completely above one of the two rays forming the image of ξ_{ij} , implying that J_{ij} can only be a single interval in this case. This in turn implies that I_{ij} consists of a single arc. This completes the proof of the lemma. ◀

A 2-approximation algorithm. Let $\mathcal{I} = \{I_{ij} \mid 1 \leq i < j \leq n\}$. Using the algorithm of Theorem 7, we compute, in $O(n^2 \log n)$ time, a minimum arc cover $\mathcal{J} \subseteq \mathcal{I}$. Each interval $I_{ij} \in \mathcal{J}$ corresponds to two points p_i, p_j of P . Set $C := \{p_i, p_j \mid I_{ij} \in \mathcal{J}\}$.

▶ **Lemma 19.** C is an weak ε -kernel of size at most twice the optimal size.

Proof. Since \mathcal{J} is an arc cover, for any $\theta \in [-\pi, \pi]$, there is pair $p_i, p_j \in C$ such that $\gamma_{ij}(\theta) \geq (1 - \varepsilon) \uparrow_{\Gamma}(\theta)$. Therefore $\downarrow_C(\theta) \geq (1 - \varepsilon) \uparrow_{\Gamma}(\theta) = (1 - \varepsilon) \downarrow_P(\theta)$, implying that C is a weak ε -kernel.

Conversely, let C^* be an optimal weak ε -kernel. We construct an arc cover \mathcal{J}^* as follows. The points in C^* are in convex position. Consider $\mathcal{N} = \mathcal{N}(C^*)$ the refined normal diagram of C , which is a centrally symmetric partition of \mathbb{S} into $2|C^*|$ intervals such that each pair of antipodal intervals of is associated with an antipodal pair of points $p_i, p_j \in C^*$. For each such pair p_i, p_j , we add the interval I_{ij} to \mathcal{J}^* ; $|\mathcal{J}^*| = |C^*|$. For $\theta \in [-\pi, \pi]$, suppose p_i, p_j is the supporting pair in directions $u(\theta/2)$ and $-u(\theta/2)$, respectively. Then $\gamma_{ij}(\theta) = \downarrow_{C^*}(\theta)$. Since $\downarrow_{C^*}(\theta) \geq (1 - \varepsilon) \uparrow_{\Gamma}(\theta)$, $\theta \in I_{ij} \in \mathcal{J}^*$. Hence, \mathcal{J}^* is an arc cover.

We can thus conclude that $|C| \leq 2|C^*|$. ◀

An exact algorithm. The above algorithm is a 2-approximation because it uses two potentially new points for each interval. We can change the arc-cover problem to account for this. We label every arc in \mathcal{I} by two indices $i, j \in \llbracket n \rrbracket$ – indices of the pair of points in P that define it. An arc cover $\mathcal{J} \subset \mathcal{I}$ of \mathbb{S} is **admissible** if every pair of intersecting arcs in \mathcal{J} share exactly one label. For any admissible arc cover \mathcal{J} , the size of the set $\{p_i, p_j \mid I_{ij} \in \mathcal{J}\}$ is at most $|\mathcal{J}|$. Furthermore, the arc cover constructed from a weak kernel in the proof of Lemma 19 is admissible. Therefore it suffices to compute a minimum-size admissible arc cover in \mathcal{I} .

To compute the smallest admissible arc cover, we follow the ideas in the algorithm of for the arc-cover [1]. While $|\mathcal{I}| = O(n^2)$, there must be a direction $u \in \mathbb{S}$ that is covered by at most $O(n^2/k)$ intervals of \mathcal{I} , where k is the size of the optimal weak ε -kernel. Let $\mathcal{J} \subseteq \mathcal{I}$ be the set of intervals covering u (u and \mathcal{J} can be computed in $O(n^2 \log n)$ time). For each one of these intervals, we now perform the greedy algorithm, as in [1]. The only difference is that instead of having a global data structure for all intervals, we break them into n groups. Specifically, for $i = 1, \dots, n$, let $\mathcal{I}_i \subset \mathcal{I}$ be the set of all arcs I with i being one of the two indices in its label. Now, we build the necessary data-structure used in [1] for each such group. Now, if the current interval is I_{ij} , the algorithm uses the data-structures for \mathcal{I}_i and \mathcal{I}_j to generate two candidate intervals to be used by the greedy algorithm. The algorithm uses the one that extends further clockwise. The rest of the algorithm is the same as in [1]. This algorithm computes the smallest admissible circular arc cover \mathcal{J}^* . We return the set $\{p_i, p_j \mid I_{ij} \in \mathcal{J}^*\}$, which in view of the above discussion is an optimal weak ε -kernel. Putting everything together we obtain the following:

► **Theorem 20.** *Given a set P of n points in the plane and a parameter $\varepsilon \in (0, 1)$, an optimal weak ε -kernel of P can be computed in $O(n^2 \log n)$ time.*

6 Conclusions

In this paper, we studied the problem of computing optimal kernels in the plane, both in the strong and weak sense. Surprisingly, this very natural problem had not received much attention when kernels were developed around twenty years ago. The problem has surprisingly non-trivial structure, and getting near linear running time to compute them exactly required non-trivial ideas and care. A natural open question is whether an instance-optimal ε -kernel of n points in \mathbb{R}^2 can be computed in $O(n \log n)$ time.

References

- 1 Pankaj K. Agarwal and Sariel Har-Peled. Computing optimal kernels in two dimensions. *CoRR*, abs/2207.07211, 2022. doi:10.48550/arXiv.2207.07211.
- 2 Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *J. Assoc. Comput. Mach.*, 51(4):606–635, 2004. doi:10.1145/1008731.1008736.
- 3 Pankaj K. Agarwal, Sariel Har-Peled, and Hai Yu. Robust shape fitting via peeling and grating coresets. *Discret. Comput. Geom.*, 39(1-3):38–58, 2008. doi:10.1007/s00454-007-9013-2.
- 4 Pankaj K. Agarwal, Nirman Kumar, Stavros Sintos, and Subhash Suri. Efficient algorithms for k -regret minimizing sets. In Costas S. Iliopoulos, Solon P. Pissis, Simon J. Puglisi, and Rajeev Raman, editors, *16th Int. Symp. Exper. Alg.*, (SEA), pages 7:1–7:23, 2017. doi:10.4230/LIPIcs.SEA.2017.7.
- 5 Pankaj K. Agarwal and Hai Yu. A space-optimal data-stream algorithm for coresets in the plane. In Jeff Erickson, editor, *Proc. 23rd Annu. Sympos. Comput. Geom.* (SoCG), pages 1–10. ACM, 2007. doi:10.1145/1247069.1247071.

- 6 Avrim Blum, Sariel Har-Peled, and Benjamin Raichel. Sparse approximation via generating point sets. *ACM Trans. Algo.*, 15(3):32:1–32:16, 2019. doi:10.1145/3302249.
- 7 Hervé Brönnimann and Michael T. Goodrich. Almost optimal set covers in finite vc-dimension. *Discrete Comput. Geom.*, 14(4):463–479, 1995. doi:10.1007/BF02570718.
- 8 Wei Cao, Jian Li, Haitao Wang, Kangning Wang, Ruosong Wang, Raymond Chi-Wing Wong, and Wei Zhan. k-regret minimizing set: Efficient algorithms and hardness. In *20th Int. Conf. Data. Theory, (ICDT)*, pages 11:1–11:19, 2017. doi:10.4230/LIPIcs.ICDT.2017.11.
- 9 Timothy M. Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Comput. Geom. Theory Appl.*, 35(1-2):20–35, 2006. doi:10.1016/j.comgeo.2005.10.002.
- 10 Bernard Chazelle and Leonidas J. Guibas. Visibility and intersection problems in plane geometry. *Discret. Comput. Geom.*, 4:551–581, 1989. doi:10.1007/BF02187747.
- 11 Kenneth L. Clarkson. Algorithms for polytope covering and approximation. In Frank K. H. A. Dehne, Jörg-Rüdiger Sack, Nicola Santoro, and Sue Whitesides, editors, *Proc. 3th Workshop Algorithms Data Struct. (WADS)*, volume 709 of *Lect. Notes in Comp. Sci.*, pages 246–252. Springer, 1993. doi:10.1007/3-540-57155-8_252.
- 12 Gautam Das and Michael T. Goodrich. On the complexity of optimization problems for 3-dimensional convex polyhedra and decision trees. *Comput. Geom.*, 8:123–137, 1997. doi:10.1016/S0925-7721(97)00006-0.
- 13 Subir Kumar Ghosh and Anil Maheshwari. An optimal algorithm for computing a minimum nested nonconvex polygon. *Information Processing Letters*, 36(6):277–280, 1990. doi:10.1016/0020-0190(90)90038-Y.
- 14 Leonidas J. Guibas, John Hershberger, Joseph S. B. Mitchell, and Jack Snoeyink. Approximating polygons and subdivisions with minimum-link paths. *Int. J. Comput. Geom. Appl.*, 3(04):383–415, 1993. doi:10.1142/S0218195993000257.
- 15 John Hershberger and Subhash Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *J. Algorithms*, 18(3):403–431, 1995. doi:10.1006/jagm.1995.1017.
- 16 Georgiy Klimenko and Benjamin Raichel. Fast and exact convex hull simplification. In Mikolaj Bojanczyk and Chandra Chekuri, editors, *Proc. 41th Conf. Found. Soft. Tech. Theoret. Comput. Sci. (FSTTCS)*, volume 213 of *LIPIcs*, pages 26:1–26:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.26.
- 17 C. C. Lee and D. T. Lee. On a circle-cover minimization problem. *Inf. Process. Lett.*, 18(2):109–115, 1984. doi:10.1016/0020-0190(84)90033-4.
- 18 Joseph SB Mitchell and Valentin Polishchuk. Minimum-perimeter enclosures. *Information Processing Letters*, 107(3-4):120–124, 2008. doi:10.1016/j.ipl.2008.02.007.
- 19 Joseph SB Mitchell and Subhash Suri. Separation and approximation of polyhedral objects. *Comput. Geom. Theory Appl.*, 5(2):95–114, 1995. doi:10.1016/0925-7721(95)00006-U.
- 20 Cao An Wang. Finding minimal nested polygons. *BIT Numerical Mathematics*, 31(2):230–236, 1991. doi:10.1007/bf01931283.
- 21 Cao An Wang and Edward P. F. Chan. Finding the minimum visible vertex distance between two non-intersecting simple polygons. In Alok Aggarwal, editor, *Proc. 2nd Annu. Sympos. Comput. Geom. (SoCG)*, pages 34–42. ACM, 1986. doi:10.1145/10515.10519.
- 22 Yanhao Wang, Michael Mathioudakis, Yuchen Li, and Kian-Lee Tan. Minimum coresets for maxima representation of multidimensional data. In Leonid Libkin, Reinhard Pichler, and Paolo Guagliardo, editors, *Proc. 40th Symp. Principles Database Sys. PODS*, pages 138–152. ACM, 2021. doi:10.1145/3452021.3458322.
- 23 Hai Yu, Pankaj K. Agarwal, Raghunath Poreddy, and Kasturi R. Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets. *Algorithmica*, 52(3):378–402, 2008. doi:10.1007/s00453-007-9067-9.

Line Intersection Searching Amid Unit Balls in 3-Space

Pankaj K. Agarwal  

Department of Computer Science, Duke University, Durham, NC, USA

Esther Ezra  

School of Computer Science, Bar Ilan University, Ramat Gan, Israel

Abstract

Let \mathcal{B} be a set of n unit balls in \mathbb{R}^3 . We present a linear-size data structure for storing \mathcal{B} that can determine in $O^*(n^{1/2})$ time whether a query line intersects any ball of \mathcal{B} and report all k such balls in additional $O(k)$ time. The data structure can be constructed in $O(n \log n)$ time. (The $O^*(\cdot)$ notation hides subpolynomial factors, e.g., of the form $O(n^\epsilon)$, for arbitrarily small $\epsilon > 0$, and their coefficients which depend on ϵ .)

We also consider the dual problem: Let \mathcal{L} be a set of n lines in \mathbb{R}^3 . We preprocess \mathcal{L} , in $O^*(n^2)$ time, into a data structure of size $O^*(n^2)$ that can determine in $O^*(1)$ time whether a query unit ball intersects any line of \mathcal{L} , or report all k such lines in additional $O(k)$ time.

2012 ACM Subject Classification Theory of computation; Theory of computation \rightarrow Computational geometry

Keywords and phrases Intersection searching, cylindrical range searching, partition trees, union of cylinders

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.5

Funding Pankaj K. Agarwal: Work partially supported by NSF grants IIS-18-14493, CCF-20-07556, and CCF-22-23870.

Esther Ezra: Work partially supported by Grants 824/17 and 800/22 from the Israel Science Foundation.

1 Introduction

Let $\mathcal{B} := \{B_1, \dots, B_n\}$ be a set of n unit-radius balls in \mathbb{R}^3 . We wish to preprocess \mathcal{B} into a data structure that supports various *line-intersection* queries. That is, given a query line ℓ in \mathbb{R}^3 , determine whether ℓ intersects a ball in \mathcal{B} , report all balls of \mathcal{B} that ℓ intersects, count the number of such balls, or compute some aggregate function on the balls intersected by ℓ . Since all balls in \mathcal{B} have the same radius, this problem can be reformulated as the *unit-cylinder range-searching* problem: Consider the set P of the centers of balls in \mathcal{B} . Preprocess P into a data structure so that we can quickly answer range queries for a query unit-radius cylinder C , such as determine whether $C \cap P = \emptyset$ (referred to as *emptiness query*), report $C \cap P$ (*reporting query*), or compute $|C \cap P|$ (*counting query*). We also consider the dual problem where the input is a set \mathcal{L} of n lines in \mathbb{R}^3 , and we wish to answer *unit-ball-intersection* queries, i.e., does a query unit ball intersect any line of \mathcal{L} .

Related work The intersection-searching problem asks to preprocess a set \mathcal{O} of geometric objects in \mathbb{R}^d into a data structure, so that one can quickly report or count all objects of \mathcal{O} intersected by a query object γ , or just test whether γ intersects any object of \mathcal{O} at all. Intersection queries are generalization of range queries (in which the input objects are points) and point-enclosure queries (in which the query objects are points).



© Pankaj K. Agarwal and Esther Ezra;

licensed under Creative Commons License CC-BY 4.0

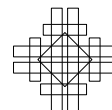
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 5; pp. 5:1–5:14

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Intersection-searching problems in 2D have been studied since the early 1990s, see, e.g., [7] and surveys [1, 3], but these problems mostly reduce to 2D or 3D range searching. In general, intersection-searching queries can be formulated as semi-algebraic range queries or point-enclosure queries in an appropriate parametric space, but the storage and query time are large because the parametric space tends to be much higher dimensional than the ambient space [2]. For example, using semi-algebraic range searching data structures and multi-level partition trees based on geometric cuttings (see e.g. [13]), a line-intersection query, and its generalizations such as segment-intersection and ray-shooting queries, amid n triangles or balls in \mathbb{R}^3 can be answered in $O^*(n^{3/4})$ time using $O^*(n)$ storage, in $O(\log n)$ time using $O^*(n^4)$ storage, or in $O^*(n/s^{1/4})$ time using $O^*(s)$ storage, for any $n \leq s \leq n^4$, by combining the first two solutions [13, 14].¹ Recently, Ezra and Sharir [8] proposed a new approach for answering ray-shooting queries amid triangles in \mathbb{R}^3 , using the polynomial-partitioning scheme by Guth [9]. This approach was extended to 3D intersection-searching in a fairly general setting by Agarwal *et al.* [2].

Analogous to halfspace-emptiness and halfspace-reporting queries, intersection-detection and intersection-reporting queries in some cases can be answered more quickly than intersection-counting queries using the concept of shallow cutting [11]. For example, a line/segment intersection-detection query amid n balls in \mathbb{R}^3 can be answered in $O^*(n/s^{1/3})$ time using $O^*(s)$ storage, for $n \leq s \leq n^3$ [12, 13, 14, 16], while the best known data structure for answering intersection-counting queries takes $O^*(n/s^{1/4})$ time, as mentioned above.

Our results. In this paper we make progress toward intersection queries between lines and unit balls in \mathbb{R}^3 . Our first main result (Sections 2–4) is a linear-size data structure for answering line-intersection queries amid unit balls in \mathbb{R}^3 :

► **Theorem 1.** *Let \mathcal{B} be a set of n unit balls in \mathbb{R}^3 . \mathcal{B} can be preprocessed, in $O(n \log n)$ time, into a linear-size data structure so that for a query line ℓ in \mathbb{R}^3 , a line-intersection-detection query can be answered in $O^*(n^{1/2})$ time, and a line-intersection-reporting query can be answered in additional $O(k)$ time, where k is the output size.*

We preprocess the centers of \mathcal{B} into a data structure for answering unit-cylinder range emptiness/reporting queries (Section 2). Our main observation is that if the centers lie in a narrow slab, the region bounded by two parallel planes within distance 2 from each other, then a query unit cylinder C can be replaced by $O(1)$ *cylindrical prisms*, each of which is of the form $\tau \oplus r_u$, where τ is a carefully chosen portion of ∂C , u is one of $O(1)$ canonically chosen directions in \mathbb{R}^3 , and r_u is the ray emanating from the origin in the direction u (Section 3). An advantage of working with such cylindrical prisms is that we can combine the theory of lower envelopes of bivariate functions [15] with Matoušek’s [11] shallow-cutting technique to construct a linear-size data structure with $O^*(\sqrt{n})$ query time. One stumbling block in applying his technique to our setting is the construction of the so-called *test set*. Roughly speaking, a test set is a small representative set of all query cylindrical prisms, in the sense that if the data structure has small query time for the test set, it also has a similar query time for any cylindrical prism. The construction of a test set Q in [11] for half-space range searching heavily relies on the linearity of hyperplanes. Agarwal and Matoušek had proposed an approach for constructing a test set for semi-algebraic ranges [4], but unfortunately it gives a weaker bound on the query time. Sharir and Shaul [16] were able to overcome this challenge by proposing a different approach for constructing a test set, which is fairly general. We adapt their approach to our setting for constructing a desired test set (Section 4).

¹ As in the abstract, the $O^*(\cdot)$ notation hides subpolynomial factors, e.g., of the form $O(n^\varepsilon)$, for arbitrarily small $\varepsilon > 0$, and their coefficients which depend on ε .

Our second main result (Section 5) is an $O^*(n^2)$ -size data structure for answering fixed-radius neighbor queries amid a set of lines in \mathbb{R}^3 :

► **Theorem 2.** *Let \mathcal{L} be a set of n lines in \mathbb{R}^3 . \mathcal{L} can be preprocessed, in time $O^*(n^2)$, into a data structure of size $O^*(n^2)$ that can answer in $O(\log n)$ time whether a query point in \mathbb{R}^3 lies within unit distance from any of the lines of \mathcal{L} . Reporting the subset of these lines costs additional $O(k)$ time, where k is the output size.*

This problem is equivalent to answering point-enclosure queries amid a set of unit cylinders in \mathbb{R}^3 . Using a two-dimensional geometric cutting, we reduce the problem to the case when the query point lies inside a narrow slab. We then replace each input cylinder with $O(1)$ cylindrical prisms and check whether the query point lies in any of the cylindrical prisms.

2 Unit-Cylinder Range Searching

Let $P \subset \mathbb{R}^3$ be a set of n points in \mathbb{R}^3 . We wish to preprocess P into a linear-size data structure so that a range-emptiness or a range-reporting query for a unit cylinder C can be answered quickly. For simplicity, we assume that the axis of C is not parallel to the yz -plane. A similar but simpler data structure can answer queries for unit cylinders whose axes are parallel to the yz -plane; we omit the details from this version. Let \mathbb{C} be the family of unit cylinders whose axes are not parallel to the yz -plane. A cylinder $C_p \in \mathbb{C}$ can be represented by a point $p = (p_1, p_2, p_3, p_4) \in \mathbb{R}^4$ where (p_1, p_2) and (p_3, p_4) are intersection points of the axis of C_p with the planes $x = 0$ and $x = 1$, respectively. We thus identify \mathbb{C} with \mathbb{R}^4 .

We construct a two-level partition tree Ψ on P , as follows. For a point $p \in P$, let p^* be its xy -projection, and let $P^* = \{p^* \mid p \in P\}$. Without loss of generality, we assume that no two points in P project to the same point. Our top-level tree is a two-dimensional partition tree, based on simplicial partition, and some of its nodes store a second-level partition tree.

Let S be a set of n points in \mathbb{R}^2 , and let $r > 0$ be a parameter. A *simplicial $(1/r)$ -partition* for P with respect to the parameter r is a collection $\Pi = \{(S_1, \Delta_1), \dots, (S_m, \Delta_m)\}$, where $m \leq r$ is an integer, such that (i) $\{S_1, \dots, S_m\}$ is a partition of S (into pairwise-disjoint subsets) satisfying $n/r \leq |S_i| \leq 2n/r$, for each i , and (ii) each Δ_i is a (possibly degenerate) triangle, referred to as a *cell*, that contains S_i . In general, the cells Δ_i need not be disjoint. The *crossing number* of Π for a line ℓ in \mathbb{R}^2 is the number of its cells that are crossed by ℓ . The crossing number of Π is defined as the maximum crossing numbers over all lines ℓ . Matoušek [11] described an algorithm for constructing a simplicial partition whose crossing number is $O(\sqrt{r})$. If $r = O(1)$, the running time of his algorithm is $O(n)$; see also [5].

We choose r to be a sufficiently large constant. By constructing simplicial partitions recursively and stopping the recursion as soon as the number of points becomes smaller than some sufficiently large constant n_0 , we construct a two-dimensional partition tree on P^* , which is the primary tree of Ψ . See [1, 5, 11] for details. Each node $v \in \Psi$ is associated with a cell Δ_v and a subset $P_v^* \subseteq P^* \cap \Delta_v$. If v is the root then $\Delta_v = \mathbb{R}^2$ and $P_v^* = P^*$. Let $P_v = \{p \in P \mid p^* \in P_v^*\}$ be the subset of P corresponding to P_v^* . Set $n_v = |P_v|$. Let $\Delta_v^\uparrow = \Delta_v \times \mathbb{R}$ be the vertical prism erected over the cell Δ_v . Then $P_v \subset \Delta_v^\uparrow$.

The *width* of a planar point set X is the minimum distance between two parallel supporting lines of X . We call a cell of Ψ *narrow* if its width is at most 2 and *wide* otherwise. For a node $v \in \Psi$, if Δ_v is narrow, we build a second-level partition tree Σ_v on P_v for answering range queries with a unit cylinder, using the algorithm described in Section 3. By Theorem 12, P_v can be preprocessed, in $O(n \log n)$ time, into a linear-size data structure so that an emptiness query for a unit cylinder can be answered in $O^*(n^{1/2})$ time. This data structure can also report all k points of P_v lying in a query cylinder in an additional $O(k)$ time. This completes the description of the data structure.

Query procedure. Let C be a query unit cylinder whose axis is not parallel to the z -axis, and let C^* be its xy -projection, which is a strip of width 2 bounded by two parallel lines. For simplicity, we describe the procedure answering the emptiness query with C . We visit Ψ recursively in a top-down manner, starting from its root. Suppose we are at a node $v \in \Psi$. If $C^* \cap \Delta_v = \emptyset$, then we simply return. If v is a leaf, then we check all points of P_v and return yes if any of them lies in C and no otherwise. So assume that v is an internal node and $\Delta_v \cap C^* \neq \emptyset$. If Δ_v is narrow then we use the secondary data structure Σ_v stored at v to test whether $P_v \cap C \neq \emptyset$; see Section 3. On the other hand, if Δ_v is wide (i.e., its width is more than 2), then we recursively visit the children of v . Note that if Δ_v is wide then it intersects at least one of the two boundary lines of C^* . For the emptiness query, the query procedure can terminate as soon as a point of P inside C is found. But for the reporting query, we continue with recursive calls until we have reported all the points.

Analysis. The height of Ψ is $O(\log n)$, and some of its nodes store a linear-size secondary structure and others use $O(1)$ space, so the overall size of Ψ is $O(n \log n)$. A similar argument shows that the preprocessing time is $O(n \log^2 n)$.

Concerning the query time, we present the analysis for emptiness queries. Reporting queries can be analyzed in a similar manner, where we gain an additional factor of $O(k)$ in the query time. Denote by $Q(n)$ the maximum emptiness query time for the two-level data structure on a set of n points. For $n \leq n_0$, $Q(n) = O(n)$. If Δ_v is a narrow cell then we use the secondary data structure stored at v and answer an emptiness query in $O^*(\sqrt{n_v})$ time. On the other hand, if Δ_v is wide then as mentioned above, one of the boundary lines of C^* intersects Δ_v . Since the crossing number of a simplicial partition is $O(\sqrt{r})$, the query is answered recursively at $O(\sqrt{r})$ children. The query procedure spends $O^*(\sqrt{n_v})$ time for each of the remaining children of v . We therefore, obtain the following recurrence for $n > n_0$.

$$Q(n) = O(\sqrt{r})Q(2n/r) + O^*(rn^{1/2}),$$

Since r is a sufficiently large constant, the solution is $Q(n) = O^*(n^{1/2})$.

The $\log n$ factor can be removed from the space and preprocessing time using a standard technique of storing the second-level structure at every $\varepsilon \cdot \log n$ level of the primary structure, for a sufficiently small constant $\varepsilon > 0$. This adds a factor of $O(n^\varepsilon)$ to the query time, which is subsumed by our $O^*(\cdot)$ notation. This completes the proof of Theorem 1.

3 Range Queries for Narrow Cells

Let $P \subset \mathbb{R}^3$ be a set of n points lying in a vertical slab σ of width at most 2, and let \mathbb{C} be the set of all unit cylinders whose axes are not parallel to the yz -plane. In this section, we describe a data structure for answering range emptiness and reporting queries on P with a unit cylinder in \mathbb{C} . Let H^-, H^+ be the two parallel boundary planes of σ with $x(H^-) < x(H^+)$. Without loss of generality, we assume that H^-, H^+ are normal to the x -axis. We also assume that the width of σ is at most $w_0 = \sin^2(1/16)$ because otherwise we partition σ into $O(1)$ slabs, each of width at most w_0 , and build a separate data structure for each of them.

We first show that a range query on such a set P with a unit cylinder in \mathbb{C} can be reduced to answering range queries with $O(1)$ ‘‘cylindrical prisms,’’ each of which is erected in one of the $O(1)$ carefully chosen canonical directions; see Section 3.1 for a precise problem formulation and the reduction. We then apply the machinery developed in [11, 16] to build the desired data structure. As mentioned in Section 1, a critical ingredient of this machinery

is the construction of a test set, which, roughly speaking, is a small-size representative set of query cylindrical prisms. As in [4, 16], each range in the test set is not a cylindrical prism but a *generalized cylindrical prism*, the union of an infinite family of cylindrical prisms. We describe the notion of test set in Section 3.2 but postpone its construction to Section 4. We finally adapt the machinery of [11] for answering range queries with cylindrical prisms.

3.1 Reduction to cylindrical-prism queries

Let \mathbb{S}^2 be the unit sphere of directions in \mathbb{R}^3 . For a direction $\mathbf{u} \in \mathbb{S}^2$, let $r_{\mathbf{u}}$ be the ray emanating from the origin in direction \mathbf{u} . Set $\bar{\mathbf{u}} = -\mathbf{u}$. For a point $p \in \mathbb{R}^3$, $p + r_{\mathbf{u}}$ is the ray in direction \mathbf{u} emanating from p , and $p - r_{\mathbf{u}} = p + r_{\bar{\mathbf{u}}}$ is the ray emanating from p in direction $\bar{\mathbf{u}}$. Let $\kappa > 16\pi$ be a sufficiently large constant. By choosing two orthogonal families of $O(\kappa)$ great circles, we partition \mathbb{S}^2 into “spherical grid” cells so that the (spherical) distance between any two points within a grid cell is at most $\frac{\pi}{\kappa}$. Let \mathcal{G} be the set of these $O(\kappa)$ great circles, and let $\mathcal{A}(\mathcal{G})$ be the grid formed by the arrangement of \mathcal{G} .

Cylindrical patches and prisms. Let $C \in \mathbb{C}$ be a unit cylinder with axis ℓ . Let the unit circle C^* be the orthogonal projection ∂C on a two-dimensional plane orthogonal to ℓ , i.e., $C = C^* \times \ell$. For a point $p^* \in C^*$, the line $\{p^*\} \times \ell \subset \partial C$ is called a *generator line* of C . A *cylindrical patch* $\tau \subset \partial C$ is a portion of ∂C bounded by two of its generator lines, i.e., $\tau = \delta \times \ell$, where $\delta \subset C^*$ is a unit arc spanning less than a semi-circle. We partition ∂C into a family $\mathcal{P}(C)$ of $O(1)$ *canonical patches* using the grid $\mathcal{A}(\mathcal{G})$, as follows. Each patch in $\mathcal{P}(C)$ is the maximal portion of ∂C whose (inner) normals lie within the same grid cell of $\mathcal{A}(\mathcal{G})$. The normals of C form a great circle C^\perp orthogonal to ℓ . The generator lines on ∂C at which normals of C are the intersection points of C^\perp with the great circles of \mathcal{G} form the boundary lines of the canonical patches. The portion of ∂C between two consecutive boundary lines forms a canonical patch. By construction, the normals within a canonical patch vary by at most π/κ . Although $\mathcal{A}(\mathcal{G})$ has $O(\kappa^2)$ cells, a cylinder in \mathbb{C} has only $O(\kappa)$ canonical patches.

Good directions. A direction $\mathbf{u} \in \mathbb{S}^2$ is called *good* for a canonical patch $\tau \in \mathcal{P}(C)$ if the following two conditions hold:

- (G1) The angle between \mathbf{u} and the (inner) normal of either of the planes H^-, H^+ does not lie in the range $[\frac{\pi}{2} - \frac{\pi}{\kappa}, \frac{\pi}{2} + \frac{\pi}{\kappa}]$, i.e., if $\mathbf{u} = (u_x, u_y, u_z)$ then $|u_x| \geq \sin \frac{\pi}{\kappa}$. Recall that the inner normals of H^-, H^+ are $(\pm 1, 0, 0)$. This condition says that \mathbf{u} is not “nearly parallel” to the plane H^- (or H^+).
- (G2) The angle between \mathbf{u} and the *inner* normal \mathbf{n}_p for any $p \in \tau$ is at most $\frac{\pi}{2} - \frac{\pi}{\kappa}$, i.e., $\langle \mathbf{n}_p, \mathbf{u} \rangle \geq \sin \frac{\pi}{\kappa}$. This condition says that for any point $p \in \tau$, the ray $p + r_{\mathbf{u}}$ enters C and it is not “nearly parallel” to the tangent plane of C at p .

(G1)–(G2) imply that for a point p on a canonical patch τ , $p + r_{\mathbf{u}}$ enters C and exits slab σ before it exits C . We make this notion more precise in Lemma 5 below.

► **Lemma 3.** *There exists a constant $\delta := \delta(\kappa) \geq \frac{\pi}{\kappa}$ such that the set of good directions for any canonical patch $\tau(C)$ of a cylinder $C \in \mathbb{C}$ contains a spherical cap of radius δ .*

Proof. We show that there is a direction \mathbf{u} that is at least δ far away from all bad directions for τ , which would imply the lemma.

Let $B^\oplus \subset \mathbb{S}^2$ be the set of all directions that are within (spherical) distance δ from a bad direction for τ . B^\oplus is the union of two sets B_0^\oplus, B_1^\oplus , formed by the complement of the two aforementioned conditions (G1) and (G2) of good directions. More precisely,

- (I) Let B_0 be the set of all points on \mathbb{S}^2 that lie within (spherical) distance $\frac{\pi}{\kappa}$ from the great circle normal to the vector $(1, 0, 0)$, i.e., parallel to the planes H^-, H^+ . B_0^\oplus is the set of points that lie within distance δ from B_0 . The area of B_0^\oplus is at most $4\pi \sin(\frac{\pi}{\kappa} + \delta)$.
- (II) Let B_1 be the set of directions that make an angle of more than $\frac{\pi}{2} - \frac{\pi}{\kappa}$ from some point of τ . Since the normals within τ vary by at most π/κ , B_1 is a spherical cap of angular opening at most $\frac{\pi}{2} + \frac{2\pi}{\kappa}$. B_1^\oplus is the set of points on \mathbb{S}^2 within distance δ from B_1 and thus spherical cap of radius $\frac{\pi}{2} + \frac{2\pi}{\kappa} + \delta$. Hence, the area of B_1^\oplus is at most $2\pi + 2\pi \sin(\frac{2\pi}{\kappa} + \delta)$.

Summing these areas, we obtain

$$\text{Area}(B^\oplus) \leq 2\pi \left(1 + 2 \sin\left(\frac{\pi}{\kappa} + \delta\right) + \sin\left(\frac{2\pi}{\kappa} + \delta\right) \right) < 4\pi,$$

provided we choose $\delta = \frac{\pi}{\kappa}$ and $\kappa \geq 16\pi$. As such $\mathbb{S}^2 \setminus B^\oplus \neq \emptyset$, and there exists a direction \mathbf{u} such that all directions in the spherical cap of radius δ centered at \mathbf{u} are good for τ . ◀

In the following, we set $\delta = \frac{\pi}{\kappa}$. Let $B_0 \subset \mathbb{S}^2$ be, as above, the set of directions that violate the condition (G1). We choose $\mathcal{Z} \subset \mathbb{S}^2 \setminus B_0$ to be a set of $O(1/\delta^2)$ points that is a δ -net for $\mathbb{S}^2 \setminus B_0$, i.e., for any point on $\mathbb{S}^2 \setminus B_0$, there is a point in \mathcal{Z} within distance δ . For simplicity, we assume that \mathcal{Z} is centrally symmetric. Lemma 3, the definition of B_0 and the choice of δ immediately imply the following:

► **Corollary 4.** *For any point $\mathbf{v} \in \mathbb{S}^2$, there is a point $\mathbf{u} \in \mathcal{Z}$ within distance $\frac{2\pi}{\kappa}$ from \mathbf{v} .*

Reduction to cylindrical prisms. We first prove a key property of good directions:

► **Lemma 5.** *Let $C \in \mathbb{C}$ be a unit cylinder, let $\tau \in \mathcal{P}(C)$ be a canonical patch, and let \mathbf{u} be a good direction for τ . Then any line parallel to \mathbf{u} intersects τ in at most one point. Moreover, for any point $p \in \tau \cap \sigma$, the ray $p + r_{\mathbf{u}}$ exits σ before exiting C .*

Proof. If a line ℓ parallel to \mathbf{u} intersects τ twice, then by the Intermediate Value Theorem, τ must have a tangent in the \mathbf{u} -direction, which contradicts property (G2) of \mathbf{u} being a good direction. Hence, ℓ intersects τ at most once.

We next prove the second assertion of the lemma. Let p' be the other intersection point of ∂C and the ray $p + r_{\mathbf{u}}$. We observe that $|pp'|$ is minimized when pp' is orthogonal to the axis of C and forms an angle of $\frac{\pi}{\kappa}$ with the tangent plane to C at p . It thus follows that $|pp'| \geq 2 \sin(\frac{\pi}{\kappa})$. On the other hand, by property (G1) of good direction, $r_{\mathbf{u}}$ forms an angle of at most $\frac{\pi}{2} - \frac{\pi}{\kappa}$ with the outer normal \mathbf{n} to the plane H^- (or H^+). Consequently, the length of the projection of pp' on \mathbf{n} is at least

$$|pp'| \cos\left(\frac{\pi}{2} - \frac{\pi}{\kappa}\right) = |pp'| \sin\left(\frac{\pi}{\kappa}\right) \geq 2 \sin^2\left(\frac{\pi}{\kappa}\right) \geq 2 \sin^2\left(\frac{1}{16}\right) > w_0.$$

Thus p' lies outside σ , which completes the proof of the lemma. ◀

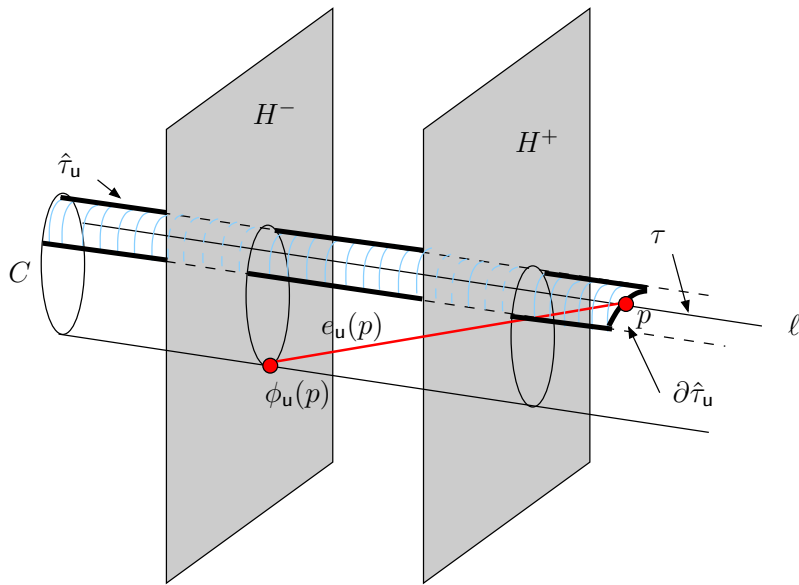
For a direction $\mathbf{u} \in \mathcal{Z}$, let $\mathcal{P}_{\mathbf{u}}(C) \subset \mathcal{P}(C)$ denote the subset of canonical patches of C for which \mathbf{u} is a good direction. We construct a *canonical prism* $\tau_{\mathbf{u}}^\uparrow$ on every patch $\tau \in \mathcal{P}_{\mathbf{u}}(C)$. We construct $O(1)$ canonical prisms for every patch $\tau \in \mathcal{P}(C)$, one for every direction z that is good for τ . The construction of the prism is somewhat delicate because we wish to meet two conflicting constraints: (i) we want to ensure that $\tau_{\mathbf{u}}^\uparrow \cap \sigma$ lies inside $C \cap \sigma$, and (ii) the union of the canonical prisms over all canonical patches and over all their good directions in

\mathcal{Z} covers $C \cap \sigma$. At a high level, we carefully clip τ by a constant-complexity semi-algebraic curve lying on τ so that any generator line of τ intersects the curve exactly once. Let $\hat{\tau}_u \subset \tau$ be the *clipped patch* of τ with respect to direction u . We set $\tau_u^\uparrow := \hat{\tau}_u \oplus r_u$. We now describe the construction of $\hat{\tau}_u$.

We fix a direction $u \in \mathcal{Z}$. For a point $p \in \tau$, let $e_u(p) = (p + r_u) \cap \sigma$ be the segment of the ray $p + r_u$ that lies inside σ ; $e_u(p)$ may be empty. Let $\phi_u(p)$ be the other endpoint of $e_u(p)$ if $e_u(p) \neq \emptyset$. We set

$$\hat{\tau}_u := \{p \in \tau \mid e_u(p) \subset C\}, \quad \tau_u^\uparrow = \hat{\tau}_u \oplus r_u, \quad \text{and} \quad \partial\hat{\tau}_u = \{p \in \tau \mid \phi(p) \in \partial C\}. \tag{1}$$

(Here we use the convention that if $e_u(p) = \emptyset$ then it lies inside C .) The patch τ is clipped along the arc $\partial\tau_u$. We note that the points $\phi_u(p)$ for $p \in \partial\tau_u$ lie on $\partial C \cap H^-$ (resp. $\partial C \cap H^+$) if the x -component of u is positive (resp. negative); see Figure 1.



■ **Figure 1** A clipped canonical patch $\hat{\tau}_u$ and its boundary arc $\partial\tau_u$; segment $e_u(p)$ and its endpoint $\phi_u(p)$ for a point $p \in \tau$.

Fix one of the generator lines ℓ' of τ . For any point $p \in \ell'$, the intersection point $\bar{p} = p + r_u$ with ∂C lies on another generator line of C , say, ℓ'' . As we translate p along ℓ' , \bar{p} also translates along ℓ'' (with the segment $p\bar{p}$ being in direction u), and thus there is a unique point $p_{\ell'} \in \ell'$ for which $\bar{p}_{\ell'} = \phi_u(p_{\ell'})$, i.e., $p_{\ell'} \in \partial\hat{\tau}_u$. The following lemma easily follows from the convexity of C :

► **Lemma 6.**

- (i) Each generator line ℓ' of τ contains exactly one point $p_{\ell'}$ of $\partial\hat{\tau}_u$. If the x -component of u is positive (resp. negative), then $\{p \in \ell \mid x(p) \geq x(p_{\ell'})\}$ (resp. $\{p \in \ell \mid x(p) \leq x(p_{\ell'})\}$) is the portion of ℓ' that lies in $\hat{\tau}_u$.
- (ii) $\tau_u^\uparrow \cap \sigma \subset C \cap \sigma$.

We construct the canonical prisms for every patch in $\mathcal{P}_u(C)$, and we repeat this step for all directions in \mathcal{Z} . Finally, we set

$$\Lambda(C) := \bigcup_{u \in \mathcal{Z}} \{\tau_u^\uparrow \mid \tau \in \mathcal{P}_u(C)\}$$

to be the set of all canonical prisms erected over the patches of $\mathcal{P}_u(C)$, and set

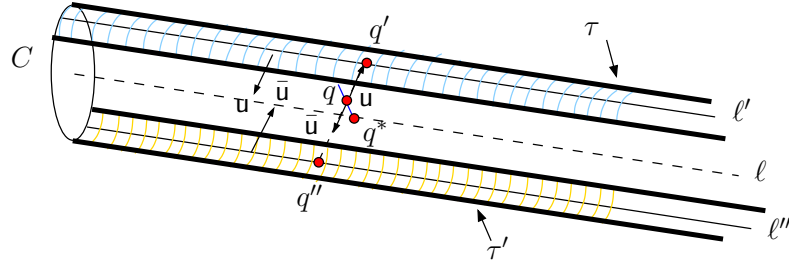
$$\mathcal{W}(C) := \bigcup_{\tau \in \Lambda(C)} \tau^\uparrow, \quad (2)$$

to be the union of these canonical prisms.

► **Lemma 7.** *For any cylinder $C \in \mathbb{C}$, $\mathcal{W}(C) \cap \sigma = C \cap \sigma$.*

Proof. It follows immediately from Lemma 6 that $\mathcal{W}(C) \cap \sigma \subseteq C \cap \sigma$. Therefore it remains to prove that $\mathcal{W}(C) \cap \sigma \supseteq C \cap \sigma$. Let q be a point in $C \cap \sigma$. We show that there exists a direction $u \in \mathcal{Z}$ and a patch $\tau \in \mathcal{P}_u(C)$ such that $q \in \tau_u^\uparrow$.

Let q^* be the projection of q on the axis of C . Let $u \in \mathcal{Z}$ be the direction closest to q^*q , and let $\bar{u} = -u$ which is also in \mathcal{Z} . By the construction of \mathcal{Z} , u, \bar{u} satisfy (G1). Let q' (resp. q'') be the intersection point of the ray $q + r_u$ (resp. $q + r_{\bar{u}} = q - r_u$) with ∂C , and let τ' (resp. τ'') be the canonical (unclipped) patch of C containing q' (resp. q''). Let $\ell' \subset \tau'$ (resp. $\ell'' \subset \tau''$) be the generator line of C containing q' (resp. q''). See Figure 2.



■ **Figure 2** Points q, q^*, q' , and q'' ; generator lines and patches containing q' and q'' .

We first claim that \bar{u} is a good direction for τ' . By Corollary 4, $\angle(qq^*, \bar{u}) \leq \frac{2\pi}{\kappa}$. Furthermore, $\angle(qq^*, \mathbf{n}_{q'}) \leq \angle(qq^*, \bar{u}) \leq \frac{2\pi}{\kappa}$. Therefore

$$\angle(\mathbf{n}_{q'}, \bar{u}) \leq \angle(qq^*, \bar{u}) + \angle(qq^*, \mathbf{n}_{q'}) \leq \frac{4\pi}{\kappa}.$$

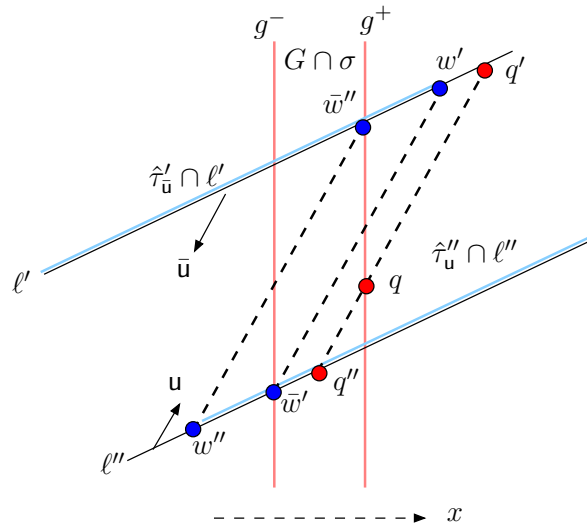
However, a direction that violates (G2) for patch τ (i.e., it is in B_1) makes an angle of at least $\frac{\pi}{2} - \frac{2\pi}{\kappa}$ with $\mathbf{n}_{q'}$, which is more than $\frac{4\pi}{\kappa}$ by our choice of κ . Hence, \bar{u} is a good direction for τ' . A similar argument shows that u is a good direction for τ'' . If $q' \in \hat{\tau}'$ then q lies in the canonical prism τ_u^\uparrow , and similarly if $q'' \in \hat{\tau}''$ then q lies in the prism τ_u^\uparrow . We therefore argue that at least one of these conditions holds.

We claim that if q' does not lie in the clipped patch $\hat{\tau}'$, then q'' lies in the clipped patch $\hat{\tau}''$. Without loss of generality, assume that the x -component of u is positive. Let G be the plane spanned by ℓ' and ℓ'' ; the segment $q'q''$ lies in G . Let $g^- = G \cap H^-$ and $g^+ = G \cap H^+$ be the intersection lines of G with the boundary planes of σ ; $x(g^+) > x(g^-)$. Then $G \cap \sigma$ is the strip lying between the parallel lines g^-, g^+ . By definition, $q \in G \cap \sigma$. Let $w' = \ell' \cap \partial\hat{\tau}'_{\bar{u}}$ (resp. $w'' = \ell'' \cap \partial\hat{\tau}''_{\bar{u}}$) be the point on ℓ' (resp. ℓ'') that lies on the boundary arc $\partial\hat{\tau}'_{\bar{u}}$ (resp. $\partial\hat{\tau}''_{\bar{u}}$). See Figure 3.

Since the x -component of u is assumed to be positive (and thus the x -component of \bar{u} is negative), by Lemma 6,

$$\hat{\tau}'_{\bar{u}} \cap \ell' = \{p \in \ell' \mid x(p) \leq x(w')\} \quad \text{and} \quad \hat{\tau}''_{\bar{u}} \cap \ell'' = \{p \in \ell'' \mid x(p) \geq x(w'')\}. \quad (3)$$

Let $\bar{w}' = \phi_{\bar{u}}(w')$ and $\bar{w}'' = \phi_u(w'')$ be the other endpoints of the segments $e_{\bar{u}}(w')$ and $e_u(w'')$, respectively. By definition, $\bar{w}' = \ell'' \cap g^-$ and $\bar{w}'' = \ell' \cap g^+$. Furthermore, the segments



■ **Figure 3** Illustration of at least one of q' and q'' lying on the clipped patch.

$q'q''$, $w'\bar{w}'$, and $w''\bar{w}''$ are parallel to each other, with their endpoints lying on l' and l'' . By Lemma 5, $x(w') > x(g^+) = x(\bar{w}'')$ because the ray $\bar{w}' + r_u \subset G$ exits σ before exiting C , and similarly $x(w'') < x(g^-) = x(\bar{w}')$. See Figure 3. If $q' \notin \hat{\tau}'_u$ then by (3), $x(q') > x(w')$ and thus $x(q') > x(\bar{w}'')$. Since $q'q''$ and $w''\bar{w}''$ are parallel segments, we conclude that $x(q'') > x(w'')$ and therefore by (3), $q'' \in l'' \cap \hat{\tau}''_u$. Hence, if $q' \notin \hat{\tau}'_u$ then $q'' \in \hat{\tau}''_u$. This completes the proof of the lemma. ◀

Recall that $P \subset \sigma$, therefore by Lemma 7, we can answer an emptiness (or reporting) query with a unit cylinder $C \in \mathbb{C}$ by answering emptiness (or reporting) queries with all cylindrical prisms in $\Lambda(C)$ (see (2)). Fix a grid cell $\varphi \in \mathcal{A}(\mathcal{G})$ and a direction $u \in \mathcal{Z}$ that is good for patches corresponding to φ . Let $\mathbb{C}_{u,\varphi}$ be the set of all cylindrical prisms τ_u^\uparrow erected in direction u over the canonical patches τ corresponding to the grid cell ϕ of unit cylinders in \mathbb{C} . We build a separate partition tree $\mathcal{T}_{u,\varphi}$ for answering range queries with prisms in $\mathbb{C}_{u,\varphi}$. In the rest of the section, we describe how we build $\mathcal{T}_{u,\varphi}$ by adapting the approach in [16].

3.2 Test set for cylindrical prisms

Throughout this section, let $r > 1$ be a fixed parameter, which we will choose to be a sufficiently large constant. We call a semi-algebraic set Δ $(1/r)$ -shallow (or simply *shallow* if the value of r is clear from the context) with respect to P if $|P \cap \Delta| \leq n/r$.

Following the terminology in [16], we call a family \mathcal{Q} of constant-complexity semi-algebraic sets, which we will refer to as *generalized prisms*, a test set for $\mathbb{C}_{u,\phi}$ with respect to P and r if the following properties hold:

- (C1) *Compactness*: $|\mathcal{Q}| = r^{O(1)}$.
- (C2) *Shalowness*: Each generalized prism $\pi^\uparrow \in \mathcal{Q}$ is $(1/r)$ -shallow with respect to P .
- (C3) *Containment*: Each $(1/r)$ -shallow cylindrical prism $\tau_u^\uparrow \in \mathbb{C}_{u,\phi}$ is contained in a single generalized cylindrical prism π^\uparrow of \mathcal{Q} , i.e., $\tau_u^\uparrow \subseteq \pi^\uparrow$.
- (C4) *Efficiency*: There exists a small bound on the associated function $\zeta(m)$, bounding the size of a partition of the *free space*, the complement of the union, of any subset of m generalized cylindrical prisms of \mathcal{Q} into elementary cells.

Each set in \mathcal{Q} will be the union of cylindrical prisms erected in direction \mathbf{u} over an infinite family of clipped canonical patches – see Section 4 for details. Informally, properties (C1)-(C3) imply that instead of considering the whole family of cylindrical prisms in $\mathbb{C}_{\mathbf{u},\phi}$, we can consider a small finite set \mathcal{Q} of “representative queries” from a more general set, each of which is shallow with respect to P , such that if the partition tree we build has a small query time for a range in \mathcal{Q} then it also has roughly the same query time for any cylindrical prism. Property (C4) bounds the query time for a range in the test set. We describe, in Section 4, the construction of a test set \mathcal{Q} of size $O(r^4)$ with $\zeta(m) = O^*(m^2)$ (cf. Lemma 13), and \mathcal{Q} can be constructed in $O(n)$ time if r is a constant.

3.3 Data structure

With a small-size test set at hand, we are now ready to describe the algorithm for constructing an elementary-cell partition of P and the partition tree by closely following the mechanism in [11, 16]. Let P and r be the same as above.

Geometric cuttings. Given a family Γ of n constant-complexity semi-algebraic sets in \mathbb{R}^d , a weight function $\omega : \Gamma \rightarrow \mathbb{R}^+$, and a parameter $r > 1$, a $(1/r)$ -cutting for Γ is a partition of space (or a portion thereof) into elementary cells, such that total weight of sets crossed by each cell is at most $\omega(\Gamma)/r$. The following lemma is taken from [16].

► **Lemma 8.** *Let Γ be a collection of n semi-algebraic sets of constant complexity in \mathbb{R}^d , let $\omega : \Gamma \rightarrow \mathbb{R}^+$ a weight function, and $r > 1$ a parameter. Assume that the free space of any subset of m sets in Γ can be partitioned into at most $\zeta(m)$ elementary cells, where $\zeta(\cdot)$ is a super-linear function. Then there exists a $(1/r)$ -cutting Ξ of Γ of size $O(\zeta(r))$ that covers the free space of Γ . Furthermore, the free space of Ξ is covered by the union of $O(r)$ sets of Γ . Ξ can be constructed in $O(n)$ time if r is a constant.*

By combining Lemmas 8 and 13, we obtain the following:

► **Corollary 9.** *Let \mathcal{Q} be a collection of n generalized prisms in \mathbb{R}^3 satisfying (C1)-(C4). Let $\omega : \mathcal{Q} \rightarrow \mathbb{R}^+$ be a weight function, and let $r \in [1, n]$ be a parameter. There exists a $(1/r)$ -cutting Ξ of \mathcal{Q} of size $O^*(r^2)$ that covers the free space of \mathcal{Q} . Furthermore, the free space of Ξ can be covered by $O(r)$ generalized prisms in \mathcal{Q} . Ξ can be constructed in $O(n)$ time if r is a constant.*

Elementary-cell partition and partition tree. Let P be a set of n points in \mathbb{R}^3 , and let $r > 1$ be a parameter. We extend the notion of simplicial partition reviewed in Section 2 to answering queries with cylindrical prisms in $\mathbb{C}_{\mathbf{u},\phi}$, as follows.

An *elementary-cell $(1/r)$ -partition* of P is a collection $\Phi = \{(P_1, \Delta_1), \dots, (P_m, \Delta_m)\}$, for some integer $m = O(r)$, such that (i) each Δ_i is an *elementary cell*, (ii) $\{P_1, \dots, P_m\}$ is a partition of P , s.t. $P_i \subset \Delta_i$, and $n/r \leq |P_i| \leq 2n/r$. The cells Δ_i may overlap. The *crossing number* of Φ for a range R is the number of elementary cells of Φ crossed by R , i.e., the number of elementary cells that intersect ∂R . The following lemma is a slight adaptation of the argument in [16] and its proof exploits Corollary 9:

► **Lemma 10.** *Let P be a set of n points in \mathbb{R}^3 lying in the slab σ of width at most $\sin^2(1/16)$, let $r > 1$ be a fixed parameter, and let \mathcal{Q} be a family of generalized prisms satisfying (C1)-(C4). Then there exists an elementary-cell $(1/r)$ -partition Φ of P such that the crossing number of Φ for any range in \mathcal{Q} is $O(r/\zeta^{-1}(r) + \log r \log |\mathcal{Q}|)$. Φ can be computed in $O(n)$ time if r is a constant.*

Plugging Lemma 13 and Corollary 9 in Lemma 10, we obtain the following corollary:

► **Corollary 11.** *let P be a set of n points lying in the slab σ of width at most $\sin^2(1/16)$, and let $r \geq 1$ be a fixed parameter. Then there exists an elementary cell $(1/r)$ -partition Φ of P such that the crossing number of Φ for any $(1/r)$ -shallow cylindrical prism in $\mathbb{C}_{u,\varphi}$ is $O^*(r^{1/2})$. Φ can be constructed in $O(n)$ time if r is a constant.*

By applying Corollary 11 recursively in a standard manner—see [11, 16]—we can build the partition tree $\mathcal{T}_{u,\varphi}$ of size $O(n)$ in $O(n \log n)$ time for answering emptiness or reporting queries with the cylindrical prisms in $\mathbb{C}_{u,\varphi}$. Since the crossing number of the elementary-cell partition is $O^*(r^{1/2})$, the query time for an emptiness query is $O^*(n^{1/2})$, and all k points lying in a query range can be reported in an additional $O(k)$ time. Omitting all the details, which can be found in [11, 16], we obtain the following result:

► **Theorem 12.** *Let P be a set of n points in \mathbb{R}^3 lying inside a vertical slab of width at most 2. P can be preprocessed, in $O(n \log n)$ time, into a data structure of size $O(n)$, so that for a (unit) cylinder $C \in \mathbb{C}$, an emptiness query can be answered in $O^*(n^{1/2})$ time, and all k points of $C \cap P$ can be reported in additional $O(k)$ time.*

4 Test-Set Construction

We now describe the construction of a test set for cylindrical prisms in $\mathbb{C}_{u,\varphi}$, for a fixed grid cell $\varphi \in \mathcal{A}(\mathcal{G})$ and $u \in \mathcal{L}$, that satisfies (C1)–(C4). Recall that the space of cylinders in \mathbb{C} is identified with \mathbb{R}^4 . For a fixed φ and u , a cylindrical prism π^\uparrow is uniquely defined by the cylinder $C \in \mathbb{C}$ whose boundary contains π^\uparrow , so the space of cylindrical prisms in $\mathbb{C}_{u,\varphi}$ can also be identified with \mathbb{R}^4 . For a prism $p \in \mathbb{R}^4$, let π_p^\uparrow be the cylindrical prism defined by p , i.e., the prism erected in direction u over the canonical patch of the unit cylinder $C_p \in \mathbb{C}$ corresponding to the grid cell φ . If C_p does not contain any canonical patch corresponding to φ , then we regard π_p^\uparrow as an empty set.

For a point $a \in \mathbb{R}^3$, we define the region $R_a := \{p \in \mathbb{R}^4 \mid a \in \pi_p^\uparrow\}$ to be the locus of all points in \mathbb{R}^4 representing cylindrical prisms that contain a . R_a is a semi-algebraic set of constant complexity. We choose a random subset $N \subseteq P$ of $O(r \log r)$ points, with an appropriate constant of proportionality. We then form the set of regions $\mathcal{R} := \{R_p \mid p \in N\}$ and construct their arrangement $\mathcal{A}(\mathcal{R})$. Set $k = c \ln r$, where $c > 0$ is an appropriate constant of proportionality. Let $\mathcal{A}_{\leq k}(\mathcal{R})$ be the set of all points of $\mathcal{A}(\mathcal{R})$ at level at most k , that is, these points represent all cylindrical prisms of $\mathbb{C}_{u,\varphi}$ that contain at most k points of P . We compute the vertical decomposition of the cells of $\mathcal{A}_{\leq k}(\mathcal{R})$ [15], which decomposes each cell of $\mathcal{A}_{\leq k}(\mathcal{R})$ into elementary cells (each of which, in fact, is a pseudo-prism). Let $\mathcal{A}_{\leq k}^\nabla(\mathcal{R})$ be set of resulting elementary cells; $|\mathcal{A}_{\leq k}^\nabla(\mathcal{R})| = O^*(r^4)$ [10].

For an elementary cell $\Delta \in \mathcal{A}_{\leq k}^\nabla(\mathcal{R})$, let $\pi_\Delta^\uparrow = \bigcup_{p \in \Delta} \pi_p^\uparrow$ be the *generalized (cylindrical) prism*, which is the union of an infinite family of cylindrical prisms defined by the points in Δ . The generalized prism π^\uparrow is a constant-complexity semi-algebraic set that is unbounded in direction u and has the property that for any $q \in \pi^\uparrow$, $q + r_u \subseteq \pi^\uparrow$. We set $\Pi^\uparrow := \Pi^\uparrow(\mathcal{R}) = \{\pi_\Delta^\uparrow \mid \Delta \in \mathcal{A}_{\leq k}^\nabla(\mathcal{R})\}$ to be the family of $O^*(r^4)$ generalized prisms corresponding to the cells in $\mathcal{A}_{\leq k}^\nabla(\mathcal{R})$. Following a straightforward argument, as in [16], it can be shown that Π^\uparrow satisfies (C1)–(C3). It thus suffices to prove (C4), namely, that the free space $\mathcal{H}(\mathcal{P}^\uparrow)$ of any subset $\mathcal{P}^\uparrow \subseteq \Pi^\uparrow$ of m generalized prisms can be partitioned into $O^*(m^2)$ elementary cells.

In the following, without loss of generality, we assume that $u = (0, 0, 1)$. Let $\mathcal{P}^\uparrow \subseteq \Pi$ be a subset of m generalized prisms. Let π denote the lower boundary of a $\pi^\uparrow \in \mathcal{P}^\uparrow$, i.e., the set of points $p \in \pi^\uparrow$ for which the open ray $p - r_u$ emanating from p in the $(-u)$ -direction is

disjoint from π^\uparrow ; $\pi^\uparrow = \pi \oplus r_u$. We refer to π as a *generalized (cylindrical) patch*, which is a constant-complexity two-dimensional xy -monotone semi-algebraic set. The patch π can be viewed as the graph of a partially defined bivariate function, also denote by π . (The value of the function is set to $+\infty$ for every point $(x, y) \in \mathbb{R}^2$ at which π is not defined.) Set $\mathcal{P} = \{\pi \mid \pi^\uparrow \in \mathcal{P}^\uparrow\}$. The *lower envelope* of \mathcal{P} is defined as the graph of the function

$$\mathcal{E}(x, y) = \min_{\pi \in \mathcal{P}} \pi(x, y),$$

which, with a slight abuse of notation, is also denoted by \mathcal{E} . It induces a partition of \mathbb{R}^2 into maximal connected regions such that \mathcal{E} is attained by a single generalized patch of \mathcal{P} (or by none of them) over the interior of each such region. The boundary of such a region consists of points at which \mathcal{E} is attained by at least two of the generalized patches in \mathcal{P} , or by the boundary of one of them. Let \mathcal{M} denote this planar subdivision, called the *minimization diagram* of \mathcal{P} . The combinatorial complexity of \mathcal{E} and \mathcal{M} is the number of faces of all dimensions in \mathcal{M} , and it is bounded by $O^*(m^2)$ [15]. The free space $\mathcal{H}(\mathcal{P}^\uparrow)$ is the set of points in \mathbb{R}^3 lying below the lower envelope \mathcal{E} .

We partition $\mathcal{H}(\mathcal{P}^\uparrow)$ into elementary cells, as follows. We first compute the two-dimensional vertical decomposition of every face f of \mathcal{M} , which partitions f into pseudo-trapezoids. Let \mathcal{M}^∇ denote the resulting refinement of \mathcal{M} . By construction, the same function of \mathcal{P} appears on \mathcal{E} for all points in a trapezoid of \mathcal{M}^∇ . For each trapezoid $\psi \in \mathcal{M}^\nabla$, we construct the prism $\psi^\downarrow := \{(x, y, z) \in \mathbb{R}^3 \mid (x, y) \in \psi \text{ and } z \in (-\infty, \mathcal{E}(x, y))\}$; ψ^\downarrow is unbounded in the $(-z)$ -direction and bounded by the graph of \mathcal{E} from above. It is easily seen that $\{\psi^\downarrow \mid \psi \in \mathcal{M}^\nabla\}$ is a partition of $\mathcal{H}(\mathcal{P}^\uparrow)$ into elementary cells. Furthermore, since $|\mathcal{M}^\nabla| = O^*(m^2)$, the number of elementary cells in the partition is $O^*(m^2)$. Hence, we obtain the following:

► **Lemma 13.** *Let $P \subset \mathbb{R}^3$ be a set of n points in \mathbb{R}^3 , and let $r \geq 1$ be a parameter. A test set of size $O^*(r^4)$ for $\mathbb{C}_{u,\varphi}$ with respect to P and r that satisfies (C1)–(C4) with $\zeta(m) = O^*(m^2)$ can be computed in $O^*(r^4)$ time.*

5 The Dual Problem

In this section, we consider the *dual problem* mentioned in Section 1: Given a set \mathcal{L} of n lines in \mathbb{R}^3 , preprocess \mathcal{L} into a data structure that supports efficient unit-ball intersection detection (as well as reporting) queries. This problem can be formulated as a *point-enclosure* problem among a set of unit cylinders: Let \mathcal{C} be the set of unit cylinders whose axes are the lines of \mathcal{L} . Preprocess \mathcal{C} into a data structure that can quickly determine whether a query point $q \in \mathbb{R}^3$ lies in the union of the cylinders in \mathcal{C} , or report all such cylinders.

Data structure. We project the cylinders in \mathcal{C} onto the xy -plane. (As in Section 2, we assume that none of the axes of the cylinders in \mathcal{C} are parallel to the yz -plane.) Let \mathcal{B} denote the set of boundary (silhouette) lines in \mathbb{R}^2 of the strips corresponding to the xy -projections of the cylinders in \mathcal{C} . Let $r > 1$ be a sufficiently large constant. We construct in $O(n^2)$ time a *hierarchical $(1/r)$ -cutting* of \mathcal{B} using the algorithm by Chazelle [6]. That is, we construct $s = O(\log n)$ cuttings Ξ_1, \dots, Ξ_s so that Ξ_i is a $(1/r^i)$ -cutting of \mathcal{B} of size $O(r^{2i})$, each triangle of Ξ_i is contained in a triangle of Ξ_{i-1} , and each triangle $\Delta \in \Xi_{i-1}$ contains a constant number of triangles of Ξ_i , which we refer to as *children* cells of Δ . Each cell of Ξ_s is crossed by $O(1)$ lines of \mathcal{B} . The algorithm also constructs the subset of lines of \mathcal{B} crossing every cell of Ξ_i for all $i \leq s$.

Fix a cell Δ of Ξ_i for some $i \leq s$. Let $\Delta^\uparrow := \Delta \times \mathbb{R}$ be the vertical slab erected over Δ . Let $\mathcal{C}_\Delta \subseteq \mathcal{C}$ be the set of cylinders that intersect the slab Δ^\uparrow . Following our definitions in Section 2, we call Δ (and Δ^\uparrow) *narrow* if its width is at most 2 and *wide* otherwise. If a unit cylinder intersects a wide slab Δ^\uparrow , then at least one of its two *silhouette* lines crosses Δ . Hence, by the cutting property, for a wide cell Δ of Ξ_i , $|\mathcal{C}_\Delta| \leq n/r^i$. If a cell $\Delta \in \Xi_i$ is narrow, then we construct a secondary data structure Ψ_Δ for answering point-enclosure queries on \mathcal{C}_Δ , as described below. Furthermore, we remove all cells of Ξ_j , for $j > i$, that are contained in Δ , for they will never be visited by the query procedure.

We now describe the secondary data structure constructed on a narrow cell Δ . We assume that the width of Δ is at most $\sin^2(1/16)$, otherwise we split Δ into $O(1)$ subcells, each of width at most $\sin^2(1/16)$ and construct a separate secondary data structure for each of them. Let \mathcal{L} and $\Lambda(C)$, for a unit cylinder C , be the same as defined in Section 3. For each cylinder $C \in \mathcal{C}_\Delta$, we construct the collection $\Lambda(C)$ of canonical cylindrical prisms, as described in Section 3.1. Recall that each prism in $\Lambda(C)$ is erected in one of directions in \mathcal{L} , i.e., it has the following form $\tau_u^\uparrow = \hat{\tau} \times r_u$ where $\hat{\tau}$ is a clipped canonical patch of C . By Lemma 7, for a point $q \in \Delta^\uparrow$, $q \in C$ if and only if $q \in \mathcal{U}(\Lambda(C))$. We thus build a data structure for answering point-enclosure queries in the set $\bigcup_{C \in \mathcal{C}_\Delta} \Lambda(C)$.

We fix a direction $u \in \mathcal{L}$ and let $\mathcal{P}_u^\uparrow \subseteq \bigcup_{C \in \mathcal{C}_\Delta} \Lambda(C)$ be the subset of canonical prisms of cylinders in \mathcal{C}_Δ erected in direction u . We build a separate data structure $\Psi_{\Delta,u}$ for answering point-enclosure queries in $\mathcal{P}_{\Delta,u}^\uparrow$, for every $u \in \mathcal{L}$, as follows. Without loss of generality, assume that u is the $(+z)$ -direction. Let $\mathcal{P}_{\Delta,u} = \{\hat{\tau}_u \mid \tau_u^\uparrow \in \mathcal{P}_{\Delta,u}^\uparrow\}$ be the set of clipped canonical patches corresponding to the prisms in $\mathcal{P}_{\Delta,u}^\uparrow$, which, as in Section 4, we regard as a set of partially-defined bivariate functions. Let $\mathcal{E}_{\Delta,u}$ be the lower envelope of $\mathcal{P}_{\Delta,u}$, and $\mathcal{M}_{\Delta,u}$ its minimization diagram. Their complexity is $O(|\mathcal{P}_{\Delta,u}|^2)$. A point $q = (q_x, q_y, q_z) \in \mathcal{U}(\mathcal{P}_{\Delta,u}^\uparrow)$ if and only if $q_z \geq \mathcal{E}_{\Delta,u}(q_x, q_y)$. We construct $\mathcal{M}_{\Delta,u}$ and preprocess it for answering planar point-location queries. Summing over all directions in \mathcal{L} , the total size of the data structure Ψ_Δ is $O^*(|\mathcal{C}_\Delta|^2)$ and it can be constructed in time $O^*(|\mathcal{C}_\Delta|^2)$. Summing these bounds over all narrow cells of the hierarchical cuttings, the total size and the preprocessing time of the overall data structure are $O^*(n^2)$.

Query procedure. Let $q = (q_x, q_y, z_z)$ be a query point. We visit the cuttings Ξ_1, Ξ_2, \dots in order. Suppose we are visiting Ξ_i , and let $\Delta \in \Xi_i$ be the cell containing $q^* = (q_x, q_y)$. If $i = s$, we answer the query in $O(1)$ time by testing q with all cylinders of \mathcal{C}_Δ . If Δ is narrow, we query the secondary data structure Ψ_Δ , as follows. For each direction $u \in \mathcal{L}$, we check whether $q \in \mathcal{U}(\mathcal{P}_u^\uparrow)$ by locating (q_x, q_y) in \mathcal{M}_u and testing whether $q_z \geq \mathcal{E}_u(q_x, q_y)$. If the answer is yes for one such u , we conclude that $q \in \mathcal{U}(\mathcal{C}_\Delta)$ and return yes. Otherwise, we return no. Finally, if Δ is wide, we recursively visit the child cell of Ξ_{i+1} that contains q . The overall query time is $O(\log n)$. This completes the proof of Theorem 2.

References

- 1 Pankaj K. Agarwal. Range searching. In J. E. Goodman, J. O'Rourke, and C. D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, chapter 40, pages 1057–1092. Chapman and Hall/CRC, third edition, 2017. URL: <https://www.taylorfrancis.com/chapters/edit/10.1201/9781315119601-40/range-searching-pankaj-agarwal>.
- 2 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir. Intersection queries for flat semi-algebraic objects in three dimensions and related problems. In *Proc. 38th Int. Sympos. Comput. Geom.*, volume 224 of *LIPICs*, pages 4:1–4:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SocG.2022.4.

- 3 Pankaj K. Agarwal and Je Erickson. Geometric range searching and its relatives. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, pages 1–56. Amer. Math. Soc., 2007.
- 4 Pankaj K. Agarwal and Jiri Matousek. On range searching with semialgebraic sets. *Discret. Comput. Geom.*, 11:393–418, 1994. doi:10.1007/BF02574015.
- 5 Timothy M. Chan. Optimal partition trees. *Discret. Comput. Geom.*, 47(4):661–690, 2012. doi:10.1007/s00454-012-9410-z.
- 6 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discret. Comput. Geom.*, 9:145–158, 1993. doi:10.1007/BF02189314.
- 7 David P. Dobkin and Herbert Edelsbrunner. Space searching for intersecting objects. *J. Algorithms*, 8(3):348–361, 1987. doi:10.1016/0196-6774(87)90015-0.
- 8 Esther Ezra and Micha Sharir. On ray shooting for triangles in 3-space and related problems. *SIAM J. Comput.*, 51(4):1065–1095, 2022. doi:10.1137/21m1408245.
- 9 Larry Guth. Polynomial partitioning for a set of varieties. *Math. Proc. Camb. Phil. Soc.*, 159:459–469, 2015. URL: <http://hdl.handle.net/1721.1/110285>.
- 10 Vladlen Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. *J. ACM*, 51(5):699–730, 2004. doi:10.1145/1017460.1017461.
- 11 Jiri Matousek. Reporting points in halfspaces. *Comput. Geom.*, 2:169–186, 1992. doi:10.1016/0925-7721(92)90006-E.
- 12 Shai Mohaban and Micha Sharir. Ray shooting amidst spheres in three dimensions and related problems. *SIAM J. Comput.*, 26(3):654–674, 1997. doi:10.1137/S0097539793252080.
- 13 Marco Pellegrini. Ray shooting on triangles in 3-space. *Algorithmica*, 9(5):471–494, 1993. doi:10.1007/BF01187036.
- 14 Marco Pellegrini. Ray shooting and lines in space. In Jacob E. Goodman, Joseph O’Rourke, and Csaba D. Tóth, editors, *Handbook of Discrete and Computational Geometry*, pages 1093–1112. CRC Press, third edition edition, 2017.
- 15 Micha Sharir and Pankaj K. Agarwal. *Davenport-Schinzel sequences and their geometric applications*. Cambridge University Press, 1995.
- 16 Micha Sharir and Hayim Shaul. Semialgebraic range reporting and emptiness searching with applications. *SIAM J. Comput.*, 40(4):1045–1074, 2011. doi:10.1137/090765092.

Drawings of Complete Multipartite Graphs up to Triangle Flips

Oswin Aichholzer ✉ 
Technische Universität Graz, Austria

Hung P. Hoang ✉ 
Department of Computer Science, ETH Zürich,
Switzerland

Jan Kynčl ✉ 
Charles University, Prague, Czech Republic

Birgit Vogtenhuber ✉ 
Technische Universität Graz, Austria

Man-Kwun Chiu ✉ 
Wenzhou-Kean University, Wenzhou, China

Michael Hoffmann ✉ 
Department of Computer Science, ETH Zürich,
Switzerland

Yannic Maus ✉ 
Technische Universität Graz, Austria

Alexandra Weinberger ✉ 
Technische Universität Graz, Austria

Abstract

For a drawing of a labeled graph, the rotation of a vertex or crossing is the cyclic order of its incident edges, represented by the labels of their other endpoints. The extended rotation system (ERS) of the drawing is the collection of the rotations of all vertices and crossings. A drawing is simple if each pair of edges has at most one common point. Gioan's Theorem states that for any two simple drawings of the complete graph K_n with the same crossing edge pairs, one drawing can be transformed into the other by a sequence of triangle flips (a.k.a. Reidemeister moves of Type 3). This operation refers to the act of moving one edge of a triangular cell formed by three pairwise crossing edges over the opposite crossing of the cell, via a local transformation.

We investigate to what extent Gioan-type theorems can be obtained for wider classes of graphs. A necessary (but in general not sufficient) condition for two drawings of a graph to be transformable into each other by a sequence of triangle flips is that they have the same ERS. As our main result, we show that for the large class of complete multipartite graphs, this necessary condition is in fact also sufficient. We present two different proofs of this result, one of which is shorter, while the other one yields a polynomial time algorithm for which the number of needed triangle flips for graphs on n vertices is bounded by $O(n^{16})$. The latter proof uses a Carathéodory-type theorem for simple drawings of complete multipartite graphs, which we believe to be of independent interest.

Moreover, we show that our Gioan-type theorem for complete multipartite graphs is essentially tight in the following sense: For the complete bipartite graph $K_{m,n}$ minus two edges and $K_{m,n}$ plus one edge for any $m, n \geq 4$, as well as K_n minus a 4-cycle for any $n \geq 5$, there exist two simple drawings with the same ERS that cannot be transformed into each other using triangle flips. So having the same ERS does not remain sufficient when removing or adding very few edges.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory; Human-centered computing → Graph drawings

Keywords and phrases Simple drawings, simple topological graphs, complete graphs, multipartite graphs, k -partite graphs, bipartite graphs, Gioan's Theorem, triangle flips, Reidemeister moves

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.6

Related Version *Full Version*: <https://arxiv.org/abs/2303.07401v1>

Funding *Oswin Aichholzer*: partially supported by the Austrian Science Fund (FWF): W1230.
Man-Kwun Chiu: partially supported by ERC StG 757609. Part of the work was done while Chiu was at FU Berlin.

Michael Hoffmann: supported by the Swiss National Science Foundation within the collaborative D-A-CH project *Arrangements and Drawings* as SNSF project 200021E-171681.

Jan Kynčl: supported by the grant no. 22-19073S of the Czech Science Foundation (GAČR).



© Oswin Aichholzer, Man-Kwun Chiu, Hung P. Hoang, Michael Hoffmann, Jan Kynčl, Yannic Maus, Birgit Vogtenhuber, and Alexandra Weinberger;
licensed under Creative Commons License CC-BY 4.0

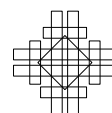
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 6; pp. 6:1–6:16



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Birgit Vogtenhuber: partially supported by the Austrian Science Fund within the collaborative D-A-CH project *Arrangements and Drawings* as FWF project I 3340-N35.

Alexandra Weinberger: supported by the Austrian Science Fund (FWF): W1230.

1 Introduction

Gioan’s Theorem states that any two simple drawings of the complete graph K_n in which the same pairs of edges cross can be transformed into each other (up to strong isomorphism) via a sequence of triangle flips. Informally, a triangle flip is the act of moving one edge of a triangular cell formed by three pairwise crossing edges over the opposite crossing of the cell; see Figure 1 for an illustration of this operation and Section 2 for the formal definition.



■ **Figure 1** A sketch of a triangle flip.

Gioan’s Theorem can be seen as a generalization of results on pseudolines by Ringel [29] from 1955 and Roudneff [30] from 1988 to simple drawings of K_n . Gioan’s conference paper [15] from 2005 contained a proof sketch only. A full proof was first published in 2017 by Arroyo, McQuillan, Richter, and Salazar [4], who also coined the name “Gioan’s Theorem”. In 2021, Schaefer [31] generalized Gioan’s Theorem to slightly sparser graphs, namely, simple drawings of K_n minus any non-perfect matching. A full version of Gioan’s proof [16] finally appeared in 2022.

A priori it is not clear how to generalize Gioan’s Theorem beyond Schaefer’s result. For transforming drawings of general graphs via triangle flips, it is not sufficient to only have the same crossing edge pairs. We should also consider the rotation of a vertex or edge crossing, which is defined as the cyclic order of emanating edges. For example, Figure 2 shows two simple drawings of the complete bipartite graph $K_{3,3}$ with the same crossing edge pairs and the same rotations of vertices, but different rotations of the crossings involving b_1r_3 . Observe that triangle flips do not change the rotations of crossings or vertices. A take-away from this observation is that for a Gioan-type theorem to hold, the rotations of all crossings and vertices must be the same in both drawings. A concept capturing exactly this necessity is the extended rotation system. The extended rotation system (ERS) of a drawing of a graph is the collection of the rotations of all vertices and crossings. In this light, one of the contributions of Gioan’s Theorem is that for drawings of the complete graph, having the same crossing edge pairs is equivalent to having the same ERS (up to global inversion) [15, 16]. This fact has been first stated by Gioan [15]; the first published proofs are by Kynčl [22, 23]. An analogous statement for K_n minus any non-perfect matching has been shown by Schaefer [31]. For complete multipartite graphs, this equivalence does not hold; see again Figure 2.

As our main result, we show that having the same ERS is sufficient to transform simple drawings of complete multipartite graphs into each other via triangle flips. We thus obtain a Gioan-type theorem for a large class of graphs that includes the before studied graphs, namely complete graphs [4, 15, 16, 31] and complete graphs minus a non-perfect matching [31].

► **Theorem 1.** *Let D_1 and D_2 be two simple drawings of a complete multipartite graph on the sphere S^2 with the same ERS. Then there is a sequence of triangle flips that transforms D_1 into D_2 .*



■ **Figure 2** Two simple drawings of $K_{3,3}$ with the same crossing edge pairs and same rotations at all vertices but different rotations at all crossings involving the edge $b_1 r_3$ and hence different ERSs.

We also show that Theorem 1 is essentially tight in the sense that having the same ERS does not remain sufficient when removing or adding very few edges.

► **Theorem 2.** *For any $m, n \geq 3$ and $K_{m,n}$ minus any two edges, there exist two simple drawings with the same ERS that cannot be transformed into each other using triangle flips. The same holds for any $n \geq 5$ and K_n minus any four-cycle C_4 , as well as for any $m \geq 4, n \geq 1$ and $K_{m,n}$ plus one edge between vertices in the bipartition class of size m .*

The first part of Theorem 2 implies that an analogue to Schaefer’s generalization of Gioan’s Theorem for K_n minus a non-perfect matching cannot be achieved for complete bipartite graphs, not even for $K_{m,n}$ minus a matching of size two. Note that $K_{m,n}$ with $m \geq 4$ and $n \geq 1$ is a subgraph of K_{n+m} minus a 4-cycle. Hence, the second part of Theorem 2 implies that – perhaps counterintuitively – the set of graphs for which a Gioan-type theorem holds is not closed under adding edges. From the proof of Theorem 2 it follows that Theorem 1 cannot be extended to any graph that contains a K_5 minus a four-cycle C_4 or a $K_{3,2}$ minus two edges incident to the same vertex of the smaller partition class, as an induced subgraph.

We present two different proofs of Theorem 1. Our first proof uses a similar approach as the proof of Gioan’s Theorem by Schaefer [31]. His proof heavily relies on a (plane) spanning star as a basis for transforming one drawing into the other. While plane spanning stars exist in any simple drawing of K_n , also minus a non-perfect matching, this is in general not the case for complete multipartite graphs. However, any simple drawing of a complete multipartite graph G contains a plane spanning tree [2]. We show that for drawings of G with the same ERS, such a plane spanning tree can be used for transforming one drawing into the other. The resulting proof is shorter and probably more elegant than the second proof. But it does not directly yield a polynomial time transformation algorithm, as it is still an open question [2] whether a plane spanning tree can be found in polynomial time.

Our second proof yields a polynomial time algorithm for the transformation. It uses a similar approach as the proof of Gioan’s Theorem by Arroyo, McQuillan, Richter, and Salazar [4]. Several ingredients of their proof are known properties of drawings of complete graphs or follow directly from such properties, while it was unknown whether analogous statements hold for drawings of other graphs. Hence, for our proof we discover a number of useful, fundamental properties of simple drawings of complete multipartite graphs. For example, we establish a Carathéodory-type theorem for them.

The classic Carathéodory Theorem states that if a point $p \in \mathbb{R}^2$ lies in the convex hull of a set $A \subset \mathbb{R}^2$ of $n \geq 3$ points, then there exists a triangle spanned by points of A that contains p . In the terminology of drawings, if a point p lies in a bounded cell of a straight-line drawing D of K_n in \mathbb{R}^2 , then there exists a 3-cycle C in D so that p lies in the bounded cell of C . This statement has been generalized to simple (not necessarily straight-line) drawings of K_n [6, 7]. However, it clearly does not generalize to arbitrary (non-complete) graphs; consider for example a simple drawing of a path with self-intersections that forms a bounded

cell. A natural question is, for which classes of graphs this statement, or a variation of it, holds. We show that it holds for complete multipartite graphs if in addition to 3-cycles – which might not exist in those graphs – we also allow 4-cycles to contain p .

► **Theorem 3** (Carathéodory-type theorem for simple drawings of complete multipartite graphs). *Let D be a simple drawing of a complete multipartite graph G in the plane. For every point p in a bounded cell of D , there exists a cycle C of length three or four in D such that p is contained in a bounded cell of C . This statement is tight in the sense that it may not hold for G minus one edge.*

Number of triangle flips. Schaefer [31, Remark 3.3] showed that for K_n , polynomially many triangle flips are sufficient and gave an upper bound of $O(n^{20})$ for the number of required flips. Using a different approach in our second proof of Theorem 1, we show an upper bound of $O(n^{16})$ triangle flips for complete multipartite graphs on n vertices. We further present drawings which, regardless of the approach, require at least $\Omega(n^6)$ triangle flips.

Motivation and related work. Originally, rotation systems were invented to investigate embeddings of graphs on higher-genus surfaces [17]. Nowadays they are widely used to represent drawings of graphs in the plane and to derive their structural properties. Gioan’s Theorem implies that for simple drawings of complete graphs, the set of crossing pairs of edges determines the drawing’s ERS. Conversely, for drawings of complete graphs, the rotation system determines which pairs of edges cross [22, 27]. These relations are crucial in the study of simple drawings of complete graphs, their generation and enumeration [1, 22, 24].

For non-complete graphs, the literature on rotation systems for simple drawings is rather sparse. Besides the recent work of Schaefer [31], we are only aware of work by Cardinal and Felsner [8], who investigate the realization of complete bipartite graphs as outer drawings. The main reason why there are no further results on rotation systems beyond drawings of complete graphs is the lack of known properties in these cases. Our work contributes towards the generalization of rotation systems to drawings of wider graph classes, not only by the main statement but also due to the structural results obtained along the way.

We note that rotation systems of drawings also play a role in a wider context. For example, they are crucial in a recent breakthrough result devising an algorithm for the subpolynomial approximation of the crossing number for non-simple drawings of general graphs [10].

The study of triangle flips has a long history in several different contexts. In addition to the mentioned work on Gioan’s Theorem [4, 15, 16, 31], this in particular includes work on arrangements of pseudolines [14, 29, 30, 32], knot theory [3, 20, 21, 25, 28, 35, 36], as well as on transforming curves on compact oriented surfaces [9].

Outline. In Section 2, we mainly state definitions, introduce notation, and give a characterization of complete multipartite graphs. In Sections 3 and 4 we sketch the proofs of the Carathéodory-type Theorem 3 and Theorem 2, respectively. Section 5 is devoted to proving Theorem 1, where the first proof is given nearly fully, and the second one is shortly sketched to explain the algorithm. In Section 6 we present bounds on the required number of triangle flips derived from the second proof. We conclude the paper with open questions in Section 7.

2 Definitions and preliminaries

A graph $G = (V, E)$ is *multipartite* if its vertex set V can be partitioned into k nonempty subsets V_1, \dots, V_k , for some $k \in \mathbb{N}$, such that each V_i , for $i \in \{1, \dots, k\}$, induces an independent set in G , that is, no two vertices in V_i are adjacent. A *complete multipartite*

graph $G = (V, E)$ contains *all* edges outside of the independent sets, that is, we have $E = \{v_i v_j : v_i \in V_i \wedge v_j \in V_j \wedge 1 \leq i < j \leq k\}$. For a multiset $\{n_1, \dots, n_k\}$ of natural numbers, there is a unique (up to isomorphism) complete multipartite graph K_{n_1, \dots, n_k} with $|V_j| = n_j$, for all $j \in \{1, \dots, k\}$. Note that both the empty graph on n vertices (with $k = 1$ and $n_1 = n$) and the complete graph K_n (with $k = n$ and $n_1 = \dots = n_k = 1$) are complete multipartite graphs. We also have the following useful characterization, whose proof is an easy graph-theoretic exercise.

► **Lemma 4.** *A graph $G = (V, E)$ is complete multipartite if and only if for every edge $uv \in E$ and every vertex $w \in V \setminus \{u, v\}$ we have $uw \in E$ or $vw \in E$ (or both).*

Drawings. A *drawing* γ of a graph $G = (V, E)$ is a geometric representation of G by points and curves on an oriented surface \mathcal{S} . More precisely, every vertex v of G is mapped to a point γ_v on \mathcal{S} and every edge uv of G is mapped to a simple (that is, continuous and not self-intersecting) curve γ_{uv} on \mathcal{S} with endpoints γ_u and γ_v , such that: (1) any two vertices are mapped to distinct points ($\gamma_u = \gamma_v \implies u = v$, for all $u, v \in V$), (2) no vertex is mapped to the relative interior of an edge ($\gamma_{uv} \cap \gamma_w = \emptyset$, for all $uv \in E$ and $w \in V \setminus \{u, v\}$), and (3) every pair of curves γ_e, γ_f , for $e \neq f$, intersects in at most finitely many points, each of which is either a common endpoint or a proper, transversal crossing.

In this paper, we consider drawings on the sphere \mathcal{S}^2 , except for a few places – specified explicitly – where we consider drawings in the plane \mathbb{R}^2 . All our graphs and drawings are labeled. Hence, we often identify vertices and edges with their geometric representation in a drawing. Any subgraph H of G induces a *subdrawing* $\gamma[H]$ that is obtained by restricting γ to the vertices and edges of H . For a graph F , an *F-subdrawing* of γ is a subdrawing $\gamma[H]$ that is induced by some subgraph H of G that is isomorphic to F . A drawing partitions \mathcal{S} into vertices (*endpoints*) and *crossings* of the curves $\{\gamma_e : e \in E\}$, *edge fragments* (the connected components of the curves $\{\gamma_e : e \in E\}$ after removing all vertices and crossings), and *cells* (the connected components of \mathcal{S} after removing all vertices, crossings, and edge fragments). For a cell C we denote by ∂C the *boundary* of C . A cell that is bounded by exactly three edge fragments is called a *tricell*.

The class of drawings of a graph is vast and for many purposes too rich to be directly useful. To begin with, it is not clear in general how to represent a drawing using a finite amount of space. Two natural approaches to address this concern are to (1) further restrict the class of drawings or (2) study drawings on a much coarser level, up to some notion of isomorphism. In this work, we use a combination of both of these approaches.

Simple drawings. An example for the first approach are *straight-line drawings* in the Euclidean plane (also known as *geometric graphs*), where the geometry of an edge is uniquely determined by the location of its endpoints; see the *Handbook of Discrete and Computational Geometry* [34, Chapter 10] and references therein. In this work, we consider a more general class of drawings, which appear in the literature as *simple drawings* [11], *good drawings* [5, 12], *topological graphs* [26], *simple topological graphs* [22], and even just as *drawings* [18]. In a simple drawing, every pair of edges has at most one point in common, either a common endpoint or a proper crossing. Additionally, we may assume that no three edges meet at a common point. Simple drawings are a combinatorial/topological generalization of straight-line drawings. If the graph G has n vertices, then every simple drawing of G has $O(n^4)$ crossings, edge fragments, and cells. Simple drawings are also important for crossing minimization because all crossing-minimal drawings are simple [33].



■ **Figure 3** Two drawings of $K_{3,3}$ that have same ERS but are not strongly isomorphic (because ux crosses vy and wz in different order). The shaded tricell is an invertible triangle.

Strong isomorphism. An example for the second approach is the notion of strong isomorphism for drawings, defined as follows. Two drawings γ and η of a graph $G = (V, E)$ are *strongly isomorphic*, denoted by $\gamma \cong \eta$, if there exists an orientation-preserving homeomorphism¹ of \mathcal{S} that maps γ to η , that is, $\gamma_v \mapsto \eta_v$, for all $v \in V$, and $\gamma_e \mapsto \eta_e$, for all $e \in E$. A combinatorial formulation, which is equivalent for connected drawings, can be obtained as follows [22]: (1) the same pairs of edges cross (this is called *weak isomorphism*); (2) the order of crossings along each edge is the same; and (3) at each vertex and crossing the *rotation*, that is, the clockwise circular order of incident edges, is the same (see next paragraph for more details). The notion of strong isomorphism encapsulates basically everything that can be said about a drawing from a topological or combinatorial point of view: the order of edges around vertices and cells, which pairs of edges cross, and in which order the crossings appear along an edge. For our purposes, we consider strongly isomorphic drawings to be equivalent.

Extended rotation systems. A coarser notion of equivalence can be obtained by requiring two drawings to have the same *rotation system*, which is the collection of the rotations of all vertices. Property (3) in the above-mentioned combinatorial description uses a slightly stronger notion of equivalence, where also the rotations at crossings are the same in both drawings. More formally, the *rotation of a crossing* χ is the clockwise cyclic order of the four vertices of the crossing edge pair which is induced by the cyclic order of edge fragments around χ . (In other words, the rotation of a crossing χ is the rotation of an additional degree-4 vertex v_χ obtained by splitting the crossing edge pair at χ and replacing χ by v_χ .) The *extended rotation system* (ERS) of a drawing is the collection of rotations of all vertices and crossings. Any two strongly isomorphic drawings have the same ERS [22]. But the converse is not true in general, as the example in Figure 3 demonstrates.

Crossing triangles. In fact, the only difference between the two drawings in Figure 3 with respect to strong isomorphism stems from the tricell formed by the triple ux, vy, wz of pairwise crossing edges, which is shaded gray in the figure: In the left drawing, this cell lies to the right of the oriented edge ux , whereas in the right drawing, it lies to the left of ux . Given a simple drawing, a tricell Δ in the subdrawing of three pairwise crossing edges e_1, e_2, e_3 is called a *crossing triangle*; the three edges e_1, e_2, e_3 are said to *span* Δ . Note that every edge triple in a simple drawing spans at most one crossing triangle. The following lemma shows that the crossing triangles are well-defined for complete multipartite graphs. It follows from the proof of Theorem 1, but can also be shown directly (and with a much shorter proof).

¹ Strong isomorphism can also be defined for unlabeled drawings; then a mapping for the vertex sets is needed. The homeomorphism is sometimes not required to be orientation-preserving; then, e.g., mirror-images of drawings are also considered to be strongly isomorphic.

► **Lemma 5.** *In every simple drawing of a complete multipartite graph, the set of edge triples that span crossing triangles is uniquely determined by the ERS.*

Invertible triangles and triangle flips. To formally define the triangle flip operation, globally fix an orientation π of the edges of the abstract graph G . This orientation can be arbitrary, but once we fix the graph, we also fix its orientation. With this orientation π , we can assign every crossing triangle a parity as follows. The *parity* of a crossing triangle Δ in a drawing is the parity (odd or even) of the number of bounding edges of Δ such that Δ lies to the left of the edge (when going along the edge according to its orientation). See Figure 3 for two drawings with even (left) and odd (right) parity of the crossing triangle. A crossing triangle Δ in a drawing γ is *invertible* if there exists another simple drawing $\gamma' \neq \gamma$ of the same graph G with the same edge orientation π and with the same ERS in which Δ appears with the opposite parity. We will show that any invertible triangle in a drawing of a complete multipartite graph is empty of vertices.

Locally redrawing the edges of an empty crossing triangle and thereby changing its parity is an elementary operation to transform a given drawing, say, the one in Figure 3 (left), into a new drawing, such as the one in Figure 3 (right). Up to strong isomorphism, there is a unique way for the redrawing. This operation is referred to as *triangle flip* [4], *triangle mutation* [15], *slide move* [31], *homotopy move* [9, 20], or *Reidemeister move of Type 3*, where the latter name has been extensively used² in knot theory [3, 21, 25, 28, 35, 36].

Triangle flip graphs. Based on the triangle flip as an elementary operation, we can define a meta graph whose vertices are drawings and whose edges correspond to triangle flips. We fix a graph G and consider all simple drawings of G on \mathcal{S} up to strong isomorphism; these are the vertices of the triangle flip graph $\mathcal{T}(G)$. Any two such drawings γ, η are connected by an edge in $\mathcal{T}(G)$ if η can be obtained from γ by a single triangle flip. As triangle flips are reversible, edges are symmetric. So we consider $\mathcal{T}(G)$ as an undirected graph.

Observe that a triangle flip does not change the rotation of any vertex or crossing, only the order of crossings along the edges changes. Therefore only drawings that have the same ERS can be in the same component of $\mathcal{T}(G)$. In general, the flip graph $\mathcal{T}(G)$ may be disconnected. Consider, for instance, the two drawings of a path depicted in in Figure 4. As neither drawing contains any crossing triangle, both are isolated vertices in $\mathcal{T}(G)$.



■ **Figure 4** Two drawings of a path with the same ERS, but the order of crossings along the edge cd differs, thus, the drawings are not strongly isomorphic. Neither drawing contains any tricell to flip.

3 A Carathéodory-type theorem for complete multipartite graphs

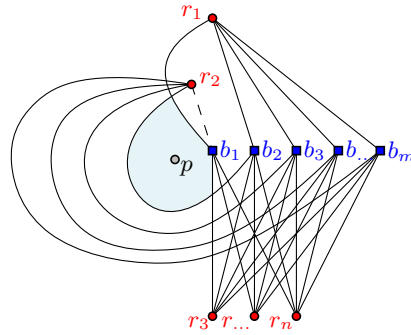
This section is devoted to a proof outline of the Carathéodory-type Theorem 3. The corresponding statement for simple drawings of K_n , which is a direct generalization of the

² albeit in the context of knots also an above/below relationship among the curves is relevant

classic theorem for convex sets in \mathbb{R}^2 , was shown by Balko, Fulek, and Kynčl [6]. A simpler proof was given later by Bergold, Felsner, Scheucher, Schröder, and Steiner [7], whose proof idea we follow.

Sketch of Proof. If G is empty or a star $K_{1,n}$, then the statement is vacuously true. So we assume that G is neither, and thus every pair of distinct vertices $u, v \in V$ with $uv \notin E$ has at least two distinct common neighbors. By studying a minimal counter-example we prove Theorem 3 by contradiction. To that aim, we consider a simple drawing D of G and a point p , such that the following holds: (1) p is in a bounded cell of D , (2) p is not contained in a bounded cell of any induced C_i -subdrawing of D , for $i \in \{3, 4\}$, and (3) when removing any vertex from D , the point p lies in the unbounded cell.

Let a be a vertex of G , and let O be the smallest set of edges incident to a such that removal of all edges of O from D puts p into the unbounded cell of the resulting drawing D^- . Then in D^- one can draw a simple curve P from p to the interior of the unbounded cell of D so that P does not intersect any vertex or edge of D^- . Subject to this constraint, we select P to minimize the number of crossings with edges of D . We show that we can assume every edge in O crosses P exactly once. Finally we consider an edge $ab \in O$, which crosses P in a point p_{ab} , and analyze two cases depending on whether ab crosses another edge between a and p_{ab} or not. We show that in both cases, p is contained in a bounded cell of an induced C_i -subdrawing of D , for $i \in \{3, 4\}$.



■ **Figure 5** Drawing of $K_{m,n}$ minus one edge ($r_2 b_1$, drawn dashed), based on Figure 6. The point p lies in a bounded cell, but in no C_i , for $i \in \{3, 4, 5\}$.

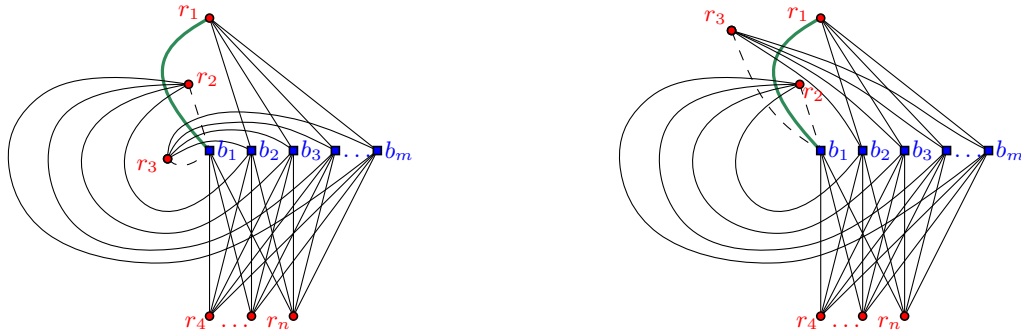
To see that the theorem may not hold if we remove one edge from G , consider the simple drawing of $K_{m,n}$, $m, n \geq 2$, depicted in Figure 5. When removing the edge $b_1 r_2$, the point p still lies in a bounded cell, but any cycle that encloses p has at least six vertices. ◀

4 Theorem 1 is essentially tight

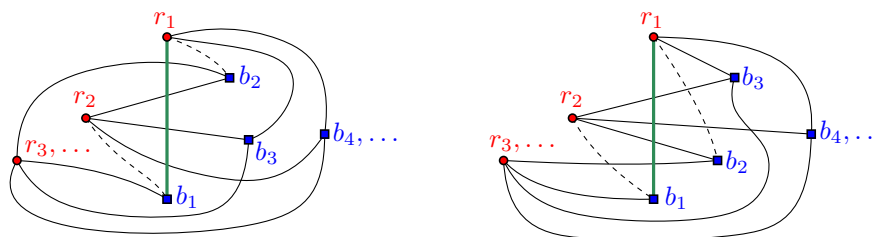
Theorem 2 implies that Theorem 1 is essentially tight: The removal or addition of very few edges may yield a graph for which the theorem does not hold. This implies that the class of graphs for which this Gioan-type theorem holds is not closed under the operation of taking (non-induced) subgraphs or supergraphs. We sketch the proof of Theorem 2 by depicting the drawings we use to show tightness.

Each of Figures 6–9 contains two simple drawings of a graph with the same ERS. In all of them, the crossing order along $b_1 r_1$ differs between the two drawings. This order cannot be changed via triangle flips because the edges crossing $b_1 r_1$ in different orders are pairwise

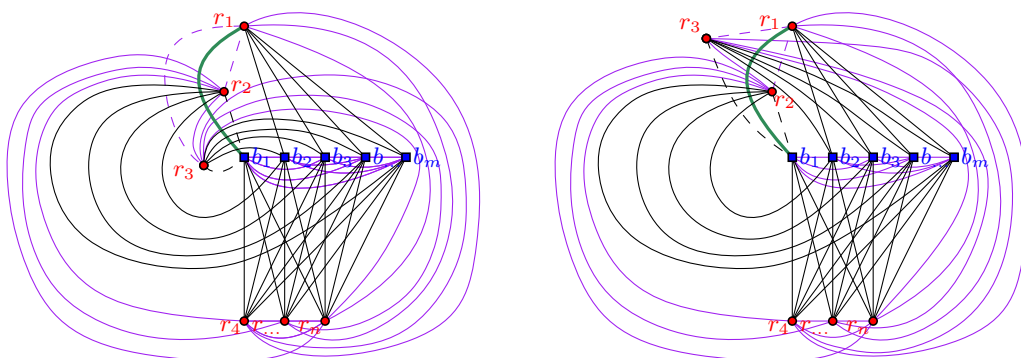
non-crossing. Figures 6 and 7 cover the case of $K_{m,n}$ minus two adjacent or disjoint edges, Figure 8 is an extension of Figure 6 to K_m minus a 4-cycle, and Figure 9 shows subdrawings of Figure 8 that form a $K_{m-1,n+1}$ plus one edge.



■ **Figure 6** Two drawings of $K_{m,n}$ minus two adjacent edges b_1r_2 and b_1r_3 (drawn as dashed lines) that have the same ERS but cannot be transformed into each other via triangle flips.

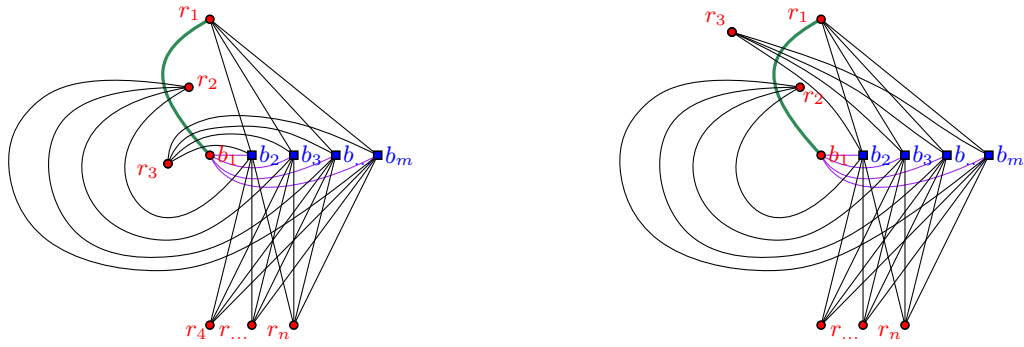


■ **Figure 7** Two drawings of $K_{m,n}$ minus two independent edges b_2r_1 and b_1r_2 (drawn dashed) that have the same ERS but cannot be transformed into each other via triangle flips.



■ **Figure 8** Two drawings of K_m minus a 4-cycle (drawn dashed) that have the same ERS, but cannot be transformed into each other via triangle flips.

We remark that also two simple drawings with the same ERS that cannot be transformed into each other via triangle flips exist for any graph that contains (1) a K_5 minus a 4-cycle, or (2) a $K_{2,3}$ minus two edges sharing a vertex in the bipartition class of cardinality two (where the list of induced subgraphs is not exhaustive). This can be shown by choosing appropriate subdrawings in the construction from Figure 8.



■ **Figure 9** Two drawings of $K_{m-1, n+1}$ plus one edge $(b_1 r_1)$ that cannot be transformed into each other via triangle flips.

5 A Gioan-type theorem for complete multipartite graphs

In this section, we present our two proofs of Theorem 1 and include a short algorithmic discussion of the second one.

5.1 First proof of Theorem 1

For our first proof of Theorem 1, we use the same general approach as Schaefer [31]. To closely follow the lines of Schaefer, we also use homeomorphisms in this proof.

Proof. Let G be a complete multipartite graph, and let D_1 and D_2 be two simple drawings of G on \mathcal{S}^2 with the same ERS. Let $R = \{r_1, r_2, \dots, r_n\}$ be a maximal independent set in G and let $B = \{b_1, b_2, \dots, b_m\}$ denote the set of the remaining vertices. Note that the graph on the vertex set $R \cup B$ together with all edges with an endpoint in R and one in B forms a complete bipartite graph $K_{n, m}$, and the set R is an independent set in G while B might not necessarily be an independent set.

By [2], the subdrawing of D_1 spanned by this $K_{n, m}$ contains a spanning tree T which is drawn crossing-free in this subdrawing and hence also in D_1 . As D_1 and D_2 have the same crossing edge pairs, T is drawn crossing-free in D_2 as well. Since the rotation systems of D_1 and D_2 are the same by assumption, the drawings of T in D_1 and D_2 are homeomorphic. Thus there exists a drawing $D \cong D_1$ with the following properties.

1. The drawing of T is the same for both drawings D and D_2 , implying that also the vertex locations are the same in both drawings.
2. Considering the set of the vertices and edges of D and D_2 together as the *combined drawing* of D and D_2 , we denote the cyclical order of edges in D and D_2 emanating from a vertex as *combined rotation* at that vertex. For each edge e of G (not in T) and each vertex v of e , the two drawings of e are consecutive in the combined rotation at v .
3. For each edge e of G , the two drawings of e are either identical or have only finitely many points in common (two are its endpoints and the others are proper crossings).

Our goal is to change D via triangle flips (and orientation-preserving homeomorphisms) until we obtain $D = D_2$. Since the vertex locations in both drawings are the same, we can speak about two drawings of an edge, one in D , and one in D_2 , being the same or not. As in Schaefer's proof, we iteratively reduce the number of edges that are drawn differently in D and D_2 . Let $E_=_$ be the set of edges whose drawings in D and D_2 are the same. Initially, $E_=_$ contains at least all edges of T . If $E_=_$ contains all edges of G then we are done.

So suppose that this is not the case and consider an edge e that is drawn differently in D and D_2 . Let e_1 and e_2 denote the curves representing e in D and D_2 , respectively. Since D and D_2 have the same ERS, e_1 and e_2 cross the same edges of T and they do so with the same crossing rotations. Moreover, the following lemma implies that they also cross those edges in the same order. The lemma can be proven relying on Lemma 4 and using a case distinction for drawings with six vertices.

► **Lemma 6.** *Let D be a simple drawing of a complete multipartite graph G on S^2 and let vw be an edge of G . Then for any pair of adjacent or disjoint edges crossed by vw , the ERS of D determines the order in which vw crosses them.*

Hence e_1 and e_2 are equivalent with respect to the drawing of T (which is the same in D and D_2), that is, e_1 has the same sequence of directed crossings with T as e_2 . Let $\Gamma = e_1 \cup e_2$ be the (not necessarily simple) closed curve formed by e_1 and e_2 . A *lens* in Γ is a cell of Γ whose boundary is formed by exactly two edge fragments of Γ , where one is from e_1 and one is from e_2 . Next, consider the drawing D_T of T plus the drawings e_1 and e_2 of e . A lens of Γ is called *empty* if it contains no vertices of T (and hence also no vertices of G) in its interior. With the next lemma, we show that Γ forms an *empty lens*. This lemma is a special case of a result of Hass and Scott on intersecting curves on surfaces [19, Lemma 3.1], which is also known as the *bigon criterion* [13, Section 1.2.4]. Schaefer [31, Lemma 3.2] gives an elementary proof in the planar (or spherical) case when the plane spanning tree T is a star. However, he only uses that the star is a spanning subdrawing that is crossing-free and that e_1 and e_2 are equivalent with respect to the star. Thus, we can follow the proof line by line to obtain the result for any plane spanning tree T .

► **Lemma 7** ([13, 19, 31]). *Let D_1 and D_2 be two simple drawings of a graph on S^2 that contain the same crossing-free drawing D_T of a spanning tree T as a subdrawing. Let e be an edge for which the drawings e_1 and e_2 differ, but are equivalent with respect to D_T . Then $\Gamma = e_1 \cup e_2$ forms an empty lens.*

Let L be an empty lens of Γ , which is formed by the edge fragments γ_1 of e_1 and γ_2 of e_2 , respectively. Each of the two points of $\gamma_1 \cap \gamma_2$ is either an endpoint or a crossing between e_1 and e_2 . Recall that, in the combined drawing of D and D_2 , e_1 and e_2 are consecutive in the combined rotation at each of their endpoints. Hence, independent of whether the points of $\gamma_1 \cap \gamma_2$ are crossings or endpoints, γ_2 is what Schaefer calls a “*homotopic detour* of γ_1 on e_1 ”. We next need his detour lemma, which we restate here using slightly different terminology (and for drawings on the sphere instead of in the plane).

► **Lemma 8** (detour lemma [31, Lemma 2.1]). *Let γ_2 be a homotopic detour of the arc γ_1 on the edge e_1 in a simple drawing of a graph. Let F be the set of edges which cross γ_2 at least twice. Then we can apply a sequence of triangle flips and homeomorphisms of the sphere S^2 so that in the resulting drawing, γ_1 is routed arbitrarily close to γ_2 , without intersecting it. The triangle flips and homeomorphisms only affect a small open neighborhood of the region bounded by $\gamma_1 \cup \gamma_2$, and only edges in F and the γ_1 part of e_1 are redrawn.*

Note that the set F of edges that are affected by the transformation is disjoint from E_- , because any edge of E_- is identical in D and D_2 and hence intersects γ_2 at most once.

If at least one of the points of $\gamma_1 \cap \gamma_2$ is a crossing, then after applying the detour lemma, we can redraw e_1 (via a homeomorphism) to have at least one fewer crossing with e_2 and repeat the process of applying Lemmas 7 and 8 with the redrawn edge.

If none of the points of $\gamma_1 \cap \gamma_2$ is a crossing, then $e_1 \cup e_2$ is a simple closed curve and $\gamma_1 = e_2$ is a homotopic detour of $\gamma_2 = e_1$. Hence, after one final application of Lemma 8, we can redraw e_1 to be identical to e_2 . With this step, e_2 is added to E_- and we have reduced the number of edges differing between D and D_2 by one.

Repeating this process for the remaining differing edges we obtain two identical drawings. Omitting the homeomorphisms, the process yields a sequence of triangle flips for transforming D_1 into D_2 (up to strong isomorphism), which completes the proof of the theorem. ◀

5.2 Second Proof of Theorem 1

Our second proof of Theorem 1, which we briefly outline here, uses the same general framework as the proof of Gioan’s Theorem by Arroyo, McQuillan, Richter, and Salazar [4].

Sketch of Proof. We consider two simple drawings D_1 and D_2 of a complete (multipartite) graph $G = (V, E)$ with the same ERS, and one of them, say $D := D_1$, is iteratively transformed to become “more similar” to the other. Similarity is measured using a subgraph X of G for which we demand as an invariant that the induced subdrawings $D[X]$ and $D_2[X]$ are strongly isomorphic. In each iteration, we will add one edge to X and then perform a sequence of triangle flips in D so as to reestablish the invariant.

Initially, we establish the invariant in the following way. As in the first proof, we consider an independent set $R \subseteq V$ of vertices such that G contains a complete bipartite subgraph between R and $B := V \setminus R$. If G is complete, then R contains a single vertex only; in general, it may contain several vertices. We then pick one vertex $r_0 \in R$ and start by taking X to be the maximal induced substar of G centered at r_0 (which includes all vertices of B). Then the invariant holds because both drawings have the same rotation system by assumption.

We then consider the (possibly) remaining vertices of R in an arbitrary order. Let $r \in R$ be the next vertex to be considered. First, we show that the position of r in the induced – strongly isomorphic, by the invariant – subdrawings $D[X]$ and $D_2[X]$ is consistent, that is, the vertex r lies in the same (according to isomorphism) face of these drawings. (The proof of this statement uses the Carathéodory-type Theorem 3.)

We add the edges incident to r one by one to X . When adding an edge rb to X to obtain $X' = X \cup \{rb\}$, the drawings $D[X']$ and $D_2[X']$ may not be strongly isomorphic because the edge rb may cross other edges in a different order in both drawings. We consider a sort of overlay O of both drawings $D[X']$ and $D_2[X']$, in which the two versions of rb together form a closed curve Γ with $O(|V(X')|^4)$ self-crossings, where $|V(X')|$ is the number of vertices of X' . In Γ , we can identify a nice substructure, which we refer to as a *free lens*, and show that it always exists. A lens in Γ is *free* if it does not contain any vertex of O ; it may contain edge crossings, though. Each such edge crossing corresponds to an invertible triangle in D . Invertible triangles are empty of vertices not only of the vertices in X but also of the (possibly) not yet considered vertices of R . Hence, the edges of D that cross an invertible triangle Δ behave similarly to a collection of pseudolines inside Δ , except that not all pairs need to cross. Let m be the number of edges that cross Δ . Using a classic sweeping algorithm by Hershberger and Snoeyink [32, Lemma 3.1], all m edges can be “swept” out of Δ via triangle flips in D , where the total number of flips is bounded by $O(m^3)$. After these flips, Δ has become a crossing triangle and can be flipped in D . Processing all invertible triangles inside a selected free lens in this fashion effectively destroys this lens. And after iteratively destroying all free lenses, the resulting drawing $D[X']$ is strongly isomorphic to $D_2[X']$.

After all vertices in R and the complete bipartite subgraph of G between R and B have been added to X , we add the remaining edges (the ones with both endpoints in B) in exactly the same fashion as described above. ◀

While the outline of the above proof mostly follows the one for K_n [4], its core challenges lie in the proofs of several statements, whose analogues are known for K_n but not for complete multipartite graphs. Among others, these include the arguments about the existence of a free lens and that invertible triangles are empty.

Algorithmic complexity. The above proof yields an algorithm that can be implemented using standard computational geometry data structures. Its runtime is polynomial in the size of the input and the number of performed triangle flips.

6 On the number of triangle flips

The *flip distance* between two different drawings of a complete multipartite graph with the same ERS is the minimum number of triangle flips that are required to transform one drawing into the other. This section is devoted to obtain bounds on the flip distance.

For an upper bound, Schaefer [31, Remark 3.3] showed that any two simple drawings of K_n with the same rotation system can be transformed into each other with at most $O(n^{20})$ triangle flips. Using our second proof of Theorem 1, we can obtain an upper bound of $O(n^{16})$ on the flip distance between two simple drawings of any complete multipartite graph with n vertices and the same ERS (and thus also for such drawings of K_n).

► **Theorem 9.** *Let D_1 and D_2 be two simple drawings of a complete multipartite graph G on S^2 with n vertices and with the same ERS. Then D_1 can be transformed into D_2 via a sequence of $O(n^{16})$ triangle flips, obtained via the algorithm in the second proof of Theorem 1.*

Proof. We analyze the number of flips performed through the second proof of Theorem 1. Recall that in this proof, we iteratively consider the edges of G . We perform flips in a drawing D (initially set to D_1) so that the subdrawings of D and D_2 induced by the already considered edges become (strongly) isomorphic.

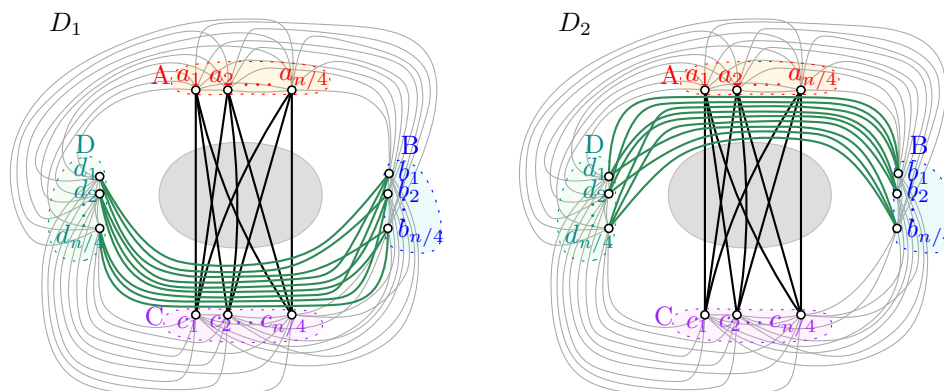
When considering a new edge e , we imagine to add both versions of it (the one from D and the one from D_2) to the already isomorphic subdrawing X of D and D_2 . In the full version, we show that this can be done such that in the combined drawing, the two copies of e have $O(|V(X)|^4) = O(n^4)$ crossings, where $|V(X)|$ is the number of vertices of X .

Let C be the closed curve formed by the two copies of e . In order to transform D to make the drawing of e in D isomorphic to the one in D_2 , we iteratively resolve a free lens of C . At every iteration, we reduce the number of crossings of C , except for the very last iteration (i.e., for the very last lens). Hence, the number of lenses we need to resolve when processing e is bounded by $O(n^4)$ as well. To resolve a free lens, we need to flip all inverted triangles in this lens that have e as an edge, of which there are at most $O(n^4)$ many. For one inverted triangle Δ intersected by $m = O(n^2)$ edges, this can be done with $O(m^3) = O(n^6)$ flips. Hence resolving one free lens can be achieved with $O(n^4) \cdot O(n^6) = O(n^{10})$ flips.

Repeating this for all lenses of C and for each of the $O(n^2)$ edges of G , we obtain an upper bound of $O(n^2) \cdot O(n^4) \cdot O(n^{10}) = O(n^{16})$ for the total number of triangle flips. ◀

► **Theorem 10.** *Let G be a multipartite graph G with n vertices that contains two vertex-disjoint subgraphs each forming a $K_{m,m}$ for some $m = \Theta(n)$. Then G admits two drawings D_1 and D_2 with the same ERS that have flip distance $\Omega(n^6)$.*

Proof idea. To transform the two drawings of K_n in Figure 10 into each other, each of the $\Theta(n^2)$ edges $b_i d_j$ needs to be moved over the $\Theta(n^4)$ crossings formed by edges $a_k c_\ell$, yielding the $\Omega(n^6)$ lower bound. An according example of two drawings of a $K_{m,m}$ can be obtained by disregarding all edges $a_i b_j$ and $c_i d_j$. ◀



■ **Figure 10** Two simple drawings of K_n with the same ERS whose flip distance is $\Omega(n^6)$.

7 Conclusion & open questions

We have shown that Gioan's Theorem holds for complete multipartite graphs (Theorem 1), extending previous results [4, 15, 16, 31]. Further, we have shown that the class of graphs for which an analogue statement holds is not closed under addition or removal of edges (Theorem 2). We also provide several obstructions such that Gioan's Theorem does not hold for any graph that contains any of these obstructions as a substructure. However, the list of obstructions is probably incomplete. A full characterization of graphs for which a Gioan-type statement for drawings with the same ERS holds remains open.

► **Question 1.** *Can we completely characterize all graphs for which a Gioan-type theorem holds for drawings with the same ERS?*

Further, having the same ERS is not the only necessary condition for a Gioan-type statement to hold. Another example of such a condition is that incident or disjoint edges must have the same crossing orders over all drawings. The constructions in the proof of Theorem 2 rely on violating this condition.

► **Question 2.** *Can we characterize all graphs for which a Gioan-type theorem holds for classes of drawings which fulfill (subsets of) obviously necessary conditions?*

In Section 3, we have proven a Carathéodory-type theorem for simple drawings of complete multipartite graphs with the same ERS (Theorem 3). It would be interesting to know for which further classes of graphs a similar statement is true.

Naturally, we would also like to narrow or even close the gap between the lower bound of $\Omega(n^6)$ and the upper bound of $O(n^{16})$ for the flip distance, obtained in Section 6.

► **Question 3.** *What is the worst case flip distance between two simple drawings of a complete multipartite graph on n vertices with a given ERS?*

References

- 1 Bernado M. Ábrego, Oswin Aichholzer, Silvia Fernández-Merchant, Thomas Hackl, Jürgen Pammer, Alexander Pilz, Pedro Ramos, Gelasio Salazar, and Birgit Vogtenhuber. All Good Drawings of Small Complete Graphs. In *Proc. 31st European Workshop on Computational Geometry EuroCG '15*, pages 57–60, Ljubljana, Slovenia, 2015. URL: http://www.csun.edu/~sf70713/publications/all_good_drawings_2015.pdf.

- 2 Oswin Aichholzer, Alfredo García, Irene Parada, Birgit Vogtenhuber, and Alexandra Weinberger. Shooting stars in simple drawings of $K_{m,n}$. In Patrizio Angelini and Reinhard von Hanxleden, editors, *Graph Drawing and Network Visualization*, pages 49–57, Cham, 2023. Springer International Publishing. doi:10.1007/978-3-031-22203-0_5.
- 3 James W. Alexander and Garland B. Briggs. On types of knotted curves. *Ann. Math.*, 28(1/4):562–586, 1927. doi:10.2307/1968399.
- 4 Alan Arroyo, Dan McQuillan, R. Bruce Richter, and Gelasio Salazar. Drawings of K_n with the same rotation scheme are the same up to triangle-flips (Gioan’s theorem). *Australas. J. Comb.*, 67(2):131–144, 2017. URL: https://ajc.maths.uq.edu.au/pdf/67/ajc_v67_p131.pdf.
- 5 Alan Arroyo, Dan McQuillan, R. Bruce Richter, and Gelasio Salazar. Levi’s lemma, pseudolinear drawings of K_n , and empty triangles. *J. Graph Theory*, 87(4):443–459, 2018. doi:10.1002/jgt.22167.
- 6 Martin Balko, Radoslav Fulek, and Jan Kynčl. Crossing numbers and combinatorial characterization of monotone drawings of K_n . *Discrete Comput. Geom.*, 53:107–143, 2015. doi:10.1007/s00454-014-9644-z.
- 7 Helena Bergold, Stefan Felsner, Manfred Scheucher, Felix Schröder, and Raphael Steiner. Topological drawings meet classical theorems from convex geometry. In *Proc. 28th Internat. Sympos. Graph Drawing*, volume 12590 of *Lecture Notes Comput. Sci.*, pages 281–294. Springer-Verlag, 2020. doi:10.1007/978-3-030-68766-3_22.
- 8 Jean Cardinal and Stefan Felsner. Topological drawings of complete bipartite graphs. *J. Comput. Geom*, 9(1):213–246, 2018. doi:10.20382/jocg.v9i1a7.
- 9 Hsien-Chih Chang, Jeff Erickson, David Letscher, Arnaud de Mesmay, Saul Schleimer, Eric Sedgwick, Dylan Thurston, and Stephan Tillmann. Tightening curves on surfaces via local moves. In *Proc. 29th ACM-SIAM Sympos. Discrete Algorithms*, pages 121–135, 2018. doi:10.1137/1.9781611975031.8.
- 10 Julia Chuzhoy and Zihan Tan. A subpolynomial approximation algorithm for graph crossing number in low-degree graphs. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 303–316, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3519984.
- 11 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.
- 12 Paul Erdős and Richard K. Guy. Crossing number problems. *Am. Math. Mon.*, 88:52–58, 1973. doi:10.2307/2319261.
- 13 Benson Farb and Dan Margalit. *A primer on mapping class groups*, volume 49 of *Princeton Mathematical Series*. Princeton University Press, Princeton, NJ, 2012.
- 14 Stefan Felsner, Alexander Pilz, and Patrick Schnider. Arrangements of approaching pseudolines. *Discrete Comput. Geom.*, 67:380–402, March 2022. doi:10.1007/s00454-021-00361-w.
- 15 Emeric Gioan. Complete graph drawings up to triangle mutations. In *Proc. 31st Internat. Workshop Graph-Theoret. Concepts Comput. Sci.*, volume 3787 of *Lecture Notes Comput. Sci.*, pages 139–150. Springer, 2005. doi:10.1007/11604686_13.
- 16 Emeric Gioan. Complete graph drawings up to triangle mutations. *Discrete Comput. Geom.*, 67:985–1022, 2022. doi:10.1007/s00454-021-00339-8.
- 17 Jonathan L. Gross and Thomas W. Tucker. *Topological graph theory*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons Inc., New York, 1987. A Wiley-Interscience Publication.
- 18 Heiko Harborth. Empty triangles in drawings of the complete graph. *Discrete Math.*, 191:109–111, 1998. doi:10.1016/S0012-365X(98)00098-3.
- 19 Joel Hass and Peter Scott. Intersections of curves on surfaces. *Isr. J. Math.*, 51(1-2):90–120, 1985.
- 20 Noboru Ito and Yusuke Takimura. (1,2) and weak (1,3) homotopies on knot projections. *J. Knot Theory Ramif.*, 22(14), 2013. doi:10.1142/S0218216513500855.

- 21 Louis H. Kauffman. Invariants of graphs in three-space. *Trans. Am. Math. Soc.*, 311(2):697–710, 1989. doi:10.1090/S0002-9947-1989-0946218-0.
- 22 Jan Kynčl. Enumeration of simple complete topological graphs. *Eur. J. Comb.*, 30:1676–1685, 2009. doi:10.1016/j.ejc.2009.03.005.
- 23 Jan Kynčl. Simple realizability of complete abstract topological graphs in P. *Discrete Comput. Geom.*, 45:383–399, 2011. doi:10.1007/s00454-010-9320-x.
- 24 Jan Kynčl. Improved enumeration of simple topological graphs. *Discrete Comput. Geom.*, 50:727–770, 2013. doi:10.1007/s00454-013-9535-8.
- 25 Marc Lackenby. A polynomial upper bound on Reidemeister moves. *Ann. Math.*, 82(2):491–564, 2015. doi:10.4007/annals.2015.182.2.3.
- 26 János Pach, József Solymosi, and Géza Tóth. Unavoidable configurations in complete topological graphs. *Discrete Comput. Geom.*, 30(2):311–320, 2003. doi:10.1007/s00454-003-0012-9.
- 27 János Pach and Géza Tóth. How many ways can one draw a graph? *Combinatorica*, 26(5):559–576, 2006. doi:10.1007/s00493-006-0032-z.
- 28 Kurt Reidemeister. Elementare Begründung der Knotentheorie. *Abh. Math. Sem. Univ. Hamburg*, 5:24–32, 1927. doi:10.1007/BF02952507.
- 29 Gerhard Ringel. Teilungen der Ebene durch Geraden oder topologische Geraden. *Math. Z.*, 64:79–102, 1956. doi:10.1007/BF01166556.
- 30 Jean-Pierre Roudneff. Tverberg-type theorems for pseudoconfigurations of points in the plane. *Eur. J. Comb.*, 9(2):189–198, 1988. doi:10.1016/S0195-6698(88)80046-5.
- 31 Marcus Schaefer. Taking a detour; or, Gioan’s theorem, and pseudolinear drawings of complete graphs. *Discrete Comput. Geom.*, 66:12–31, 2021. doi:10.1007/s00454-021-00296-2.
- 32 Jack Snoeyink and John Hershberger. Sweeping arrangements of curves. In *Discrete and Computational Geometry: Papers from the DIMACS Special Year*, volume 6 of *DIMACS*, pages 309–349. AMS, 1991. doi:10.1090/dimacs/006/21.
- 33 László A. Székely. A successful concept for measuring non-planarity of graphs: the crossing number. *Discr. Math.*, 276(1):331–352, 2004. 6th International Conference on Graph Theory. doi:10.1016/S0012-365X(03)00317-0.
- 34 Csaba D. Tóth, Joseph O’Rourke, and Jacob E. Goodman, editors. *Handbook of Discrete and Computational Geometry, Third Edition*. Chapman and Hall/CRC, 2017. URL: <https://www.routledge.com/9781498711395>.
- 35 Bruce Trace. On the Reidemeister moves of a classical knot. *Proc. Amer. Math. Soc.*, 89(4):722–724, 1983. doi:10.1090/S0002-9939-1983-0719004-4.
- 36 Shûji Yamada. An invariant of spatial graphs. *J. Graph Theory*, 13(5):537–551, 1989. doi:10.1002/jgt.3190130503.

Decomposition of Zero-Dimensional Persistence Modules via Rooted Subsets

Ángel Javier Alonso  

Technische Universität Graz, Austria

Michael Kerber  

Technische Universität Graz, Austria

Abstract

We study the decomposition of zero-dimensional persistence modules, viewed as functors valued in the category of vector spaces factorizing through sets. Instead of working directly at the level of vector spaces, we take a step back and first study the decomposition problem at the level of sets.

This approach allows us to define the combinatorial notion of *rooted subsets*. In the case of a filtered metric space M , rooted subsets relate the clustering behavior of the points of M with the decomposition of the associated persistence module. In particular, we can identify intervals in such a decomposition quickly. In addition, rooted subsets can be understood as a generalization of the elder rule, and are also related to the notion of constant conqueror of Cai, Kim, Mémoli and Wang. As an application, we give a lower bound on the number of intervals that we can expect in the decomposition of zero-dimensional persistence modules of a density-Rips filtration in Euclidean space: in the limit, and under very general circumstances, we can expect that at least 25% of the indecomposable summands are interval modules.

2012 ACM Subject Classification Mathematics of computing → Topology; Theory of computation → Computational geometry

Keywords and phrases Multiparameter persistent homology, Clustering, Decomposition of persistence modules, Elder Rule

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.7

Related Version *Full Version*: <https://arxiv.org/abs/2303.06118>

Funding This research has been supported by the Austrian Science Fund (FWF) grant P 33765-N.

Acknowledgements The authors thank Jan Jendrysiak for helpful discussions. We are also grateful to the anonymous reviewers for their careful reading of our manuscript and their detailed comments and suggestions.

1 Introduction

Multiparameter persistent homology is an active research area in topological data analysis. The motivation is that in many datasets there are multiple parameters that deserve attention in a multiscale analysis [11, 18, 31]. Concretely, when analyzing point clouds, we want to consider the distances between points, but also potentially remove points of low density.

A central object of persistent homology is the *persistence module*, which tracks algebraically how the topological features of the data change as we move through the parameter space. In the single-parameter case, every persistence module decomposes into a collection of intervals, called the *persistence barcode* [20], where each interval represents the lifetime of a topological feature in the data. In the multiparameter setting, there is a generalized notion of interval, which again represents the lifetime of a topological feature, but decomposing a multiparameter persistence module into intervals is not always possible, and one might be left with non-interval indecomposable persistence modules that lead to complications, both theoretically [12, 13, 18, 32] and computationally [1, 4, 23].



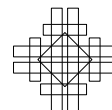
© Ángel Javier Alonso and Michael Kerber;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 7; pp. 7:1–7:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Table 1** Number of intervals in the decomposition of zero-dimensional persistence modules for density-Rips filtrations. We tried both *clustered* samples where the points were sampled by a multivariate Gaussian distribution around 5 peaks, and *uniform* samples in the unit square. The density parameter was computed via a Gaussian kernel density estimate (*kde*) or a *random* density was assigned. The table shows the number of intervals for 5 independent test runs; for n points, the module is interval-decomposable if the number of intervals is n . This only happens for one run.

Sample	Densities	100 points					500 points				
		Run 1	2	3	4	5	Run 1	2	3	4	5
clustered	kde	100	98	95	98	98	474	487	478	479	479
uniform	kde	88	88	86	88	86	444	447	433	453	457
clustered	random	77	86	87	88	76	397	381	390	380	386
uniform	random	76	79	75	75	70	376	361	366	355	377

In fact, the classification of such indecomposable persistence modules is thought to be out of reach: certain involved posets are of *wild representation type*, even when accounting for certain simplifications [2]. Moreover, infinite families of complicated indecomposable persistence modules can be realized by simple geometric constructions [13], and, most recently, it has been shown in [3] that multiparameter persistence modules are, generically, close to being indecomposable, under the interleaving metric (we refer to [3] for a precise statement).

Still, the mentioned complications do not imply that the persistence modules that come up in practice are close to indecomposable, or that they are not decomposable into intervals. Indeed, is the decomposition of multiparameter persistence modules as badly behaved in practice as we can expect in theory? The authors of [2] and those of [3] state similar questions.

As an initial test, we computed the decomposition of persistence modules for a standard zero-dimensional construction, using a prototypical implementation of the algorithm by Dey and Xin [23] (this implementation will be discussed in another paper). As we see in Table 1, the assumption that persistence modules can be decomposed completely into intervals seems to be false most of the time, at least in this setting. However, Table 1 also shows that in all tested instances, *most* indecomposable summands are indeed intervals.

This begs the question whether we can provably expect many intervals in general. In addition, knowledge of the intervals can greatly simplify and speed up computational tasks for persistence modules: for instance, a popular way to analyze 2-parameter persistence modules is by considering 1-dimensional restrictions, so-called slices, resulting in a parameterized family of persistent barcodes [29, 31, 33, 34]. Every interval of the 2-dimensional persistence module gives one bar in the barcode of the slice, by intersecting the slice with the interval. Thus, by knowing the intervals, existing algorithms can focus on the non-interval “core” of the problem, which is typically of much smaller size.

The practical problem of the described approach is that decomposing a multiparameter persistence module is costly, despite ongoing efforts [23]. However, to leverage the knowledge of intervals there is no need to compute a total decomposition, or to even identify all intervals. It suffices to have a method to “peel off” intervals from a persistence module quickly. Thus, we pose the question whether there exist methods that work very fast in practice and still are capable of detecting many intervals.

Contributions. We focus on the case of zero-dimensional persistence modules. Already this case is of practical interest because of its connection to hierarchical clustering methods (see the Related work section below), and has received attention recently [2, 10, 14, 34]. In this context, we give some answers to the questions stated above:

For a point cloud M , a *nearest neighbor pair* is a pair $(x, y) \in M \times M$ such that y is the nearest neighbor of x and x is the nearest neighbor of y (breaking ties with a fixed total order). The theory we develop says that for a zero-dimensional persistence module of the density-Rips bifiltration (for any density estimation function), there are at least as many intervals as there are nearest neighbor pairs in M . These intervals are easily determined by the nearest neighbor pairs, and we refer to them as *NN-intervals*. Since all nearest neighbor pairs can be computed in $O(n \log n)$ time [19, 37], this yields a fast method to compute all NN-intervals of the decomposition. Moreover, we can expect many NN-intervals: using previous results on nearest neighbor graphs, we show that if M is sampled independently from an arbitrary, almost continuous density function, at least a quarter of the summands in the decomposition are intervals as $n \rightarrow \infty$. To our knowledge, this is the first result proving a non-constant lower bound on the number of intervals in a decomposition.

To arrive at this result, we use the following main idea: Instead of studying the decomposition of the persistence module directly in the category of (graded) vector spaces, we work in the category of *persistent sets*, whose objects can be interpreted as a two-parameter hierarchical clustering. The decomposition of a persistence module is governed by its idempotent endomorphisms, so we look for idempotent endomorphisms not of persistence modules, but of persistent sets, which are simpler. We show that such idempotent endomorphisms can be translated into *rooted subsets*, which are subsets of points that get consistently merged with a fixed point in the hierarchical clustering. Moreover, rooted subsets with a single element correspond to intervals in the associated persistence module.

Instead of peeling off intervals from the persistence module, we peel off rooted subsets from the persistent set. The advantage is that the remaining structure is still a hierarchical clustering, and the process can be iterated.

Related work. Multiparameter persistent sets and zero-dimensional persistence modules, as we will study them here, are related to a multiparametric approach to the clustering problem first considered by Carlsson and Mémoli [16]. The need for multiple parameters, density and scale, is justified by an axiomatic approach to clustering [15, 17, 28]. The application of techniques from multiparameter persistence homology, like persistence modules and interleavings, to this setting has attracted attention recently [2, 14, 31, 33, 34].

Cai, Kim, Mémoli, and Wang [14] define a useful summary for zero-dimensional persistence modules coming from density-Rips, called the *elder-rule-staircode*, inspired by the elder rule [24]. They also introduce the related concept of constant conqueror, and they ask whether a constant conqueror induces an interval in the decomposition of the associated persistence module. We answer this question in the negative with Example 22, and, in contrast, we show that a *rooted generator*, as introduced here, does induce an interval in the decomposition (Corollary 13).

Brodzki, Burfitt, and Pirashvili [10] also study the decomposition of zero-dimensional persistence modules. They identify a class of persistence modules, called *semi-component modules*, that may appear as summands in the decomposition of zero-dimensional modules, but that are still hard to classify. Their methods have been of great inspiration, and in Theorem 24 we give another proof, within the theory we develop, of a theorem of theirs.

2 Preliminaries

Persistent sets and persistence modules. In what follows, we let P be a finite poset, which we will view as a category. A **persistence module** (over P) is a functor from P to the category Vec of finite dimensional vector spaces, over a fixed field K . Such a functor

$F: P \rightarrow \mathbf{Vec}$ associates to each **grade** $p \in P$ a finite dimensional vector space F_p and to each morphism $p \leq q$ in P a linear map $F_{p \rightarrow q}: F_p \rightarrow F_q$, in such a way that $F_{p \rightarrow p} = \text{id}$ and composition is preserved. We see persistence modules as the objects of the functor category \mathbf{Vec}^P , where natural transformations are the morphisms. In this sense, a morphism $f: F \rightarrow G$ of persistence modules is a family of maps $\{f_p: F_p \rightarrow G_p\}_{p \in P}$ such that for every two $p \leq q$ the following diagram commutes

$$\begin{array}{ccc} F_p & \xrightarrow{F_{p \rightarrow q}} & F_q \\ \downarrow f_p & & \downarrow f_q \\ G_p & \xrightarrow{G_{p \rightarrow q}} & G_q. \end{array}$$

Similarly, a **persistent set** (over P) is a functor from P to \mathbf{Set} , the category \mathbf{Set} of finite sets, and morphisms of persistent sets are natural transformations as above.

We can obtain a persistence module from a persistent set by the application of the **linearization functor** $\mathbf{Set} \rightarrow \mathbf{Vec}$ that takes each set to the free vector space generated by it. This linearization functor induces a functor $\mathcal{L}: \mathbf{Set}^P \rightarrow \mathbf{Vec}^P$ by postcomposition.

From geometry to persistent sets. Let (M, d) be a finite metric space, and consider a function $f: M \rightarrow \mathbb{R}$. We can understand f as an assignment of a *density* to each of the points of M ; that is, a density estimation function [36]. We assume that f assigns lower values to points of *higher* density. Following [14], we call the triple (M, d, f) an **augmented metric space**. We construct a persistent set, the **density-Rips persistent set** of (M, d, f) , that tracks how the clustering of points of M changes as we change the density and scale parameters, in a sense that we make precise shortly.

First, for a fixed scale parameter $\varepsilon \geq 0$, we define the **geometric graph of M at ε** , denoted by $\mathcal{G}_\varepsilon(M)$, as the undirected graph on the vertex set M and edges (x, y) where $d(x, y) \leq \varepsilon$. The connected components of $\mathcal{G}_\varepsilon(M)$, as ε goes from 0 to ∞ , form the clusters of the dendrogram obtained via the single-linkage clustering method.

To introduce the density, for each $\sigma \in \mathbb{R}$ we let $M_\sigma := \{x \in M \mid f(x) \leq \sigma\} \subseteq M$ be the metric subspace of points with (co)density below σ . For any two $\sigma \leq \sigma'$, $M_\sigma \subseteq M_{\sigma'}$ and by taking each (ε, σ) to the graph $\mathcal{G}_\varepsilon(M_\sigma)$, we obtain a functor $\mathcal{G}(M, f): \mathbb{R}_{\geq 0} \times \mathbb{R} \rightarrow \mathbf{Graph}$, where the order in $\mathbb{R}_{\geq 0} \times \mathbb{R}$ is given by $(\varepsilon, \sigma) \leq (\varepsilon', \sigma')$ if and only if $\varepsilon \leq \varepsilon'$ and $\sigma \leq \sigma'$. We then consider the **connected components functor** $\pi_0: \mathbf{Graph} \rightarrow \mathbf{Set}$, that takes each graph to its set of connected components. In this way, we obtain a functor $\pi_0 \circ \mathcal{G}(M, f): \mathbb{R}_{\geq 0} \times \mathbb{R} \rightarrow \mathbf{Set}$.

► **Remark 1.** The linearized persistence module $\mathcal{L}(\pi_0 \circ \mathcal{G}(M, f)): \mathbb{R}_{\geq 0} \times \mathbb{R} \rightarrow \mathbf{Vec}$ is isomorphic to the persistence module obtained by applying zero-dimensional homology at graph level, $H_0 \circ \mathcal{G}(M, f): \mathbb{R}_{\geq 0} \times \mathbb{R} \rightarrow \mathbf{Vec}$. In this sense, the construction we have described is the zero-dimensional level of the density-Rips filtration, which is standard in multiparameter persistent homology (see [5, 18] and also [14]).

We can understand the functor $\pi_0 \circ \mathcal{G}(M, f): \mathbb{R}_{\geq 0} \times \mathbb{R} \rightarrow \mathbf{Set}$ as a persistent set $S: P \rightarrow \mathbf{Set}$ indexed by a finite grid $P \subseteq \mathbb{R}_{\geq 0} \times \mathbb{R}$ in the following way. We consider the set of distances $D := \{d(x, y) \mid x, y \in M\}$ and densities $T := \{f(x) \mid x \in M\}$, and define a finite grid $P := D \times T \subset \mathbb{R}_{\geq 0} \times \mathbb{R}$. Finally, we define the persistent set $S: P \rightarrow \mathbf{Set}$ by taking each $(\varepsilon, \sigma) \in P$ to $(\pi_0 \circ \mathcal{G}(M, f))_{(\varepsilon, \sigma)}$, and similarly for the morphisms.

► **Definition 2.** Let (M, d, f) be an augmented metric space. We define its **density-Rips persistent set** as the functor $S: P \rightarrow \mathbf{Set}$, constructed as above.

Decomposition of persistence modules. We can study persistence modules via their decomposition. For two persistence modules F and G their direct sum $F \oplus G$ is the persistence module given by taking direct sums pointwise, $(F \oplus G)_p = F_p \oplus G_p$. A persistence module is **indecomposable** if $F \cong F_1 \oplus F_2$ implies that either $F_1 = 0$ or $F_2 = 0$. Since persistence modules are actual modules (see, for instance, [8, Lemma 2.1]), by the Krull-Schmidt theorem, a decomposition of a persistence module $F = F_1 \oplus F_2 \oplus \dots \oplus F_n$ into indecomposable summands is unique up to permutation and isomorphism of the summands.

Let I be a non-empty connected subposet of a poset P such that for any two $i, j \in I$ and any $l \in P$, if $i \leq l \leq j$ then $l \in I$. The **interval module supported on I** , $\mathcal{I}(I): P \rightarrow \text{Vec}$, is the indecomposable (by, e.g. [7, Proposition 2.2]) persistence module given by

$$\mathcal{I}(I)_p = \begin{cases} K, & \text{if } p \in I, \\ 0, & \text{otherwise,} \end{cases} \quad \text{with internal maps } \mathcal{I}(I)_{p \rightarrow q} = \begin{cases} \text{id}, & \text{if } p, q \in I, \\ 0, & \text{otherwise.} \end{cases}$$

If P is a totally ordered set, every persistence module over P decomposes as a direct sum of interval modules [6], but such a nice decomposition does not exist in general for other posets.

Decomposition and endomorphisms. A direct sum $X = X_1 \oplus X_2$ of persistence modules is characterized up to isomorphism by morphisms $\iota_i: X_i \rightarrow X$ and $\pi_i: X \rightarrow X_i$ for $i = 1, 2$ such that $\pi_i \circ \iota_i = \text{id}_{X_i}$ and $\iota_1 \circ \pi_1 + \iota_2 \circ \pi_2 = \text{id}_X$ (see, for instance, [30]). In this case, for each $i = 1, 2$, the maps ι_i and π_i induce an endomorphism $\varphi_i := \iota_i \circ \pi_i$ of X . Such an endomorphism $\varphi_i: X \xrightarrow{\pi_i} X_i \xrightarrow{\iota_i} X$ is also split:

► **Definition 3.** In any category, we say that an endomorphism $\varphi: X \rightarrow X$ is **split** if there exists an object Y and a factorization $\varphi: X \xrightarrow{\pi} Y \xrightarrow{\iota} X$ such that $\pi \circ \iota = \text{id}_Y$.

We will use the following standard fact about split endomorphisms (proof in the full version):

► **Lemma 4.** Let $\varphi: X \rightarrow X$ be a split endomorphism that has two factorizations $X \xrightarrow{\pi} Y \xrightarrow{\iota} X$ and $X \xrightarrow{\pi'} Y' \xrightarrow{\iota'} X$ with $\pi \circ \iota = \text{id}_Y$ and $\pi' \circ \iota' = \text{id}_{Y'}$. Then Y and Y' are isomorphic.

Every split endomorphism $\varphi: X \xrightarrow{\pi} Y \xrightarrow{\iota} X$ is also **idempotent**, meaning that $\varphi \circ \varphi = \varphi$. Moreover, in our categories of interest, namely persistent sets Set^P and persistence modules Vec^P , every idempotent endomorphism splits through its **image**, see below. In these two categories, we define the image of a morphism f , $\text{img } f$, by taking the image pointwise, that is, $(\text{img } f)_p = f_p(S_p)$. The following two lemmas are standard (proof in the full version).

► **Lemma 5.** Let $\varphi: X \rightarrow X$ be an idempotent endomorphism in Vec^P or Set^P . Then f splits through its image: there exists a factorization $f: X \xrightarrow{\pi} \text{img } \varphi \xrightarrow{\iota} X$ with $\pi \circ \iota = \text{id}_{\text{img } \varphi}$.

► **Lemma 6.** Let $F: P \rightarrow \text{Vec}$ be a persistence module, and let $\varphi: F \rightarrow F$ be an idempotent endomorphism. Then F decomposes as $\text{img}(\text{id}_F - \varphi) \oplus \text{img } \varphi$.

3 Endomorphisms of persistent sets and rooted subsets

As seen above, the decomposition of a persistence module is intimately related to its idempotent endomorphisms. Our main idea is that, when studying the decomposition of persistence modules of the form $\mathcal{L}S$, for a persistent set $S: P \rightarrow \text{Set}$, we look for idempotent endomorphisms of S and study their image under the linearization functor \mathcal{L} .

► **Definition 7.** Given a persistent set S , a **generator** is a pair (p_x, x) with $x \in S_{p_x}$ such that x is not in the image of any morphism $S_{q \rightarrow p_x}$ for any $q < p_x$. When it is clear, we will often suppress the grade p_x from the notation, and directly write that $x \in S_{p_x}$ is a generator.

There is an induced preorder on the generators of S : for two generators $x \in S_{p_x}$ and $y \in S_{p_y}$ we say that $(p_x, x) \leq (p_y, y)$ if and only if $p_x \leq p_y$. This relation might not be antisymmetric, and so in general the preordered set of generators is not a poset.

Generators are useful because an endomorphism φ of a persistent set $S: P \rightarrow \text{Set}$ is uniquely determined by the image of its generators: for each $z \in S_q$ we have $\varphi_q(z) = S_{p_x \rightarrow q} \circ \varphi_{p_x}(x)$ for some generator $x \in S_{p_x}$, by the commutativity property.

In linear algebra, an idempotent endomorphism can be thought as a projection onto its image, that is, onto its fixed points. This point of view and the concept of generators above motivates the following definition, which plays a fundamental role in our work.

► **Definition 8.** A **rooted subset** A is a non-empty subset of the generators of S such that there exists an idempotent endomorphism φ of S whose set of generators that are not fixed is precisely A . If a rooted subset is a singleton, $A = \{x\}$, we say that x is a **rooted generator**.

► **Remark 9.** In the case of an augmented metric space (M, d, f) and its density-Rips persistent set S of Definition 2 there exists a bijection between the points of M and the generators of S . A point $x \in M$ first appears in the graph $\mathcal{G}_0(M_{f(x)})$, where x is always its own connected component. In what follows, we will often identify a point $x \in M$ with its generator $x \in S_{p_x}$. In this sense, we can understand an endomorphism of S as an endomorphism of the set of points that is compatible with the connected components of all graphs $\mathcal{G}_\varepsilon(M_\sigma)$.

We are especially interested in persistent sets obtained from (augmented) metric spaces, and our objective is to relate rooted generators to the geometry of these objects. Considering an augmented metric space (M, d, f) and its density-Rips persistent set, Proposition 10 below characterizes rooted generators by the clustering behavior of the points of M .

► **Proposition 10.** Let (M, d, f) be an augmented metric space and consider a point $x \in M$. If there exist some other point $y \in M$ such that

1. $f(y) \leq f(x)$ (i.e. y is “denser” than x), and
2. whenever x is in a cluster of more than one point, $y \in M$ is in the same cluster: for every $\mathcal{G}_\varepsilon(M_\sigma)$, if x is path-connected to some other point then x is path-connected to y , then the generator (p_x, x) of the density-Rips persistent set S of M is a rooted generator.

Conversely, if x is a rooted generator of S , then there exists a point $y \in M$ that satisfies conditions 1 and 2 above.

Proof. Before going into the proof, recall that, by the way we construct S and the inclusion $P \hookrightarrow \mathbb{R}_{\geq 0} \times \mathbb{R}$, for each $q \in P$ there is an associated graph $\mathcal{G}_\varepsilon(M_\sigma)$, for some $(\varepsilon, \sigma) \in \mathbb{R}_{\geq 0} \times \mathbb{R}$. Each element $z \in S_q$ is a connected component of this graph $\mathcal{G}_\varepsilon(M_\sigma)$, and the generators $x \in S_{p_x}$ such that $S_{p_x \rightarrow q}(x) = z$ are precisely the points in that connected component.

The first part follows from Proposition 11 below, which proves it in more generality.

For the converse, let φ be an idempotent of S whose only generator that is not fixed is $x \in S_{p_x}$. This means that there exists a generator $y \in S_{p_y}$, different from x , such that $\varphi_{p_x}(x) = S_{p_y \rightarrow p_x}(y)$. And clearly $\varphi_{p_z}(z) = z$ for any other generator $z \in S_{p_z}$. From the fact that $\varphi_{p_x}(x) = S_{p_y \rightarrow p_x}(y)$ we deduce that $f(y) \leq f(x)$, since $p_y \leq p_x$ in P . To see that the second condition holds, pick a $q \geq p_x$ and suppose that there exists a generator $w \in S_{p_w}$ such that $S_{p_x \rightarrow q}(x) = S_{p_w \rightarrow q}(w)$. This means that in the graph $\mathcal{G}_\varepsilon(M_\sigma)$ associated to q both x and w are in the same connected component, and we claim that y is also in this component. Indeed, by the definition of φ we have $S_{p_x \rightarrow q} \circ \varphi_{p_x}(x) = S_{p_y \rightarrow q}(y) = S_{p_w \rightarrow q}(w)$. ◀

► **Proposition 11.** *Let (M, d, f) be an augmented metric space. If for a set of points $A \subset M$ there exists a point $y \notin A$ such that for every $x \in A$, $f(y) \leq f(x)$ and for each $\mathcal{G}_\varepsilon(M_\sigma)$ either:*

- *x is path-connected to y , or*
- *the set of points that are path-connected to x is contained in A ,*

then the set of generators $\{(p_x, x) \mid x \in A\}$ is a rooted subset in the density-Rips persistent set S of M .

Proof. To show that A is a rooted subset, we need to define an appropriate idempotent φ of S . Recalling that an endomorphism is uniquely determined by the image of its generators, we define φ by setting

$$\varphi_{p_x}(x) = \begin{cases} S_{p_y \rightarrow p_x}(y), & \text{if } x \in A, \\ x, & \text{otherwise,} \end{cases} \tag{1}$$

for every generator $x \in S_{p_x}$. We need to show that φ is indeed well-defined, which means that the image of $z \in S_q$, $\varphi_q(z) = S_{p_x \rightarrow q} \circ \varphi_{p_x}(x)$, is the same no matter the generator $x \in S_{p_x}$ we choose. Fix a $q \in P$ and a $z \in S_q$, and consider the set G of generators whose image in S_q is z , $G := \{(p_x, x) \mid p_x \leq q, S_{p_x \rightarrow q}(x) = z\}$.

Then, to check that φ is well-defined, for every two $(p_x, x), (p_w, w) \in G$ it must hold that

$$S_{p_x \rightarrow q} \circ \varphi_{p_x}(x) = S_{p_w \rightarrow q} \circ \varphi_{p_w}(w). \tag{2}$$

If both (p_x, x) and (p_w, w) are not in A , or if both (p_x, x) and (p_w, w) are in A , then Equation (2) above trivially holds, by the way we have defined φ in Equation (1).

Thus, the only interesting case is that only one of (p_x, x) or (p_w, w) is in A . Say that $(p_x, x) \in A$ and $(p_w, w) \notin A$. Then, by assumption both x and w need to be path-connected to y at the graph $\mathcal{G}_\varepsilon(M_\sigma)$ associated to q , which means that, as desired,

$$S_{p_x \rightarrow q} \circ \varphi_{p_x}(x) = S_{p_y \rightarrow q}(y) = S_{p_w \rightarrow q}(w) = S_{p_w \rightarrow q} \circ \varphi_{p_w}(w).$$

Now, φ is idempotent, because for every $x \in A$ we have $\varphi_{p_x}^2(x) = \varphi_{p_x}(S_{p_y \rightarrow p_x}(y)) = S_{p_y \rightarrow p_x}(\varphi_{p_y}(y)) = S_{p_y \rightarrow p_x}(y)$. And it is clear that the only generators that are not fixed by φ are those in A . We conclude that, effectively, A is a rooted subset. ◀

Decomposition induced by rooted subsets. As we have seen, rooted subsets are related to the clustering behavior of the points. They are also related to the decomposition of the linearized persistence module: they induce summands.

► **Theorem 12.** *Let φ be an idempotent endomorphism of a persistent set S . Then the persistence module $\mathcal{L}S$ decomposes into*

$$\text{img}(\text{id}_{\mathcal{L}S} - \mathcal{L}\varphi) \oplus \mathcal{L}(\text{img } \varphi).$$

Proof. Since φ is idempotent, $\mathcal{L}\varphi$ is idempotent. By Lemma 6, this induces a decomposition

$$\mathcal{L}S \cong \text{img}(\text{id} - \mathcal{L}\varphi) \oplus \text{img } \mathcal{L}\varphi.$$

It is left to show that $\text{img } \mathcal{L}\varphi \cong \mathcal{L}(\text{img } \varphi)$. Applying Lemma 5 to $\mathcal{L}\varphi$, we have a factorization $\mathcal{L}\varphi: \mathcal{L}S \xrightarrow{\pi} \text{img } \mathcal{L}\varphi \xrightarrow{\iota} \mathcal{L}S$ with $\pi \circ \iota = \text{id}$. Applying Lemma 5 again, this time to φ , we have a factorization $\varphi: S \xrightarrow{\pi'} \text{img } \varphi \xrightarrow{\iota'} S$ with $\pi' \circ \iota' = \text{id}$. Now, split endomorphisms are

preserved by every functor: in the diagram $\mathcal{L}S \xrightarrow{\mathcal{L}\pi'} \mathcal{L}(\text{img } \varphi) \xrightarrow{\mathcal{L}l'} \mathcal{L}S$ it holds $\mathcal{L}l' \circ \mathcal{L}\pi' = \mathcal{L}\varphi$ and $\mathcal{L}\pi' \circ \mathcal{L}l' = \text{id}$. Thus, the endomorphism $\mathcal{L}\varphi$ splits in two ways:

$$\begin{array}{ccc}
 & \text{img } \mathcal{L}\varphi & \\
 \nearrow \pi & \downarrow \cong & \searrow l \\
 \mathcal{L}S & & \mathcal{L}S \\
 \searrow \mathcal{L}\pi' & \downarrow & \nearrow \mathcal{L}l' \\
 & \mathcal{L}(\text{img } \varphi) &
 \end{array}$$

where the middle arrow exists and is an isomorphism by Lemma 4, finishing the proof. ◀

Combining the above theorem with the Krull-Schmidt theorem, we obtain the following:

► **Corollary 13.** *A rooted subset of a persistent set S induces a summand in the decomposition of $\mathcal{L}S$. A rooted generator $x \in S_{p_x}$ induces an interval summand, and all other summands can be obtained by decomposing $\mathcal{L}(\text{img } \varphi)$, where φ is the endomorphism associated to x .*

This allows to iteratively *peel off* intervals of a persistence module of the form $\mathcal{L}S$: find a rooted generator of S , with associated idempotent φ , and continue considering $\text{img } \varphi$ instead of S . In the setting of an augmented metric space (M, d, f) and its density-Rips persistent set, the intervals that are peeled off are easily interpretable through the clustering behavior of the points M , by Proposition 10. Moreover, the conditions we describe actually happen in practice, as we see in Section 5.

Neighborly rooted points. In fact, certain points of an augmented metric space (M, d, f) can be seen to be rooted by looking at the nearest neighbors, which will be useful in Section 5. In what follows we fix a total order on M compatible with the order induced by f . Recall that the **nearest neighbor** of x is the element $x' \neq x$ of minimum distance to x , where ties have been broken by the fixed total order on M .

► **Definition 14.** *Let (M, d, f) be an augmented metric space. An element x is **neighborly rooted** if its nearest neighbor $y \in M$ satisfies $f(y) \leq f(x)$.*

► **Lemma 15.** *With the notation as above, if a point $x \in M$ is neighborly rooted then x is a rooted generator in the density-Rips persistent set of (M, d, f) .*

Proof. It is clear that the nearest neighbor of x satisfies the conditions of Proposition 10. ◀

► **Remark 16.** We can identify all neighborly rooted points in the time it takes to solve the all-nearest-neighbor problem. Naturally, the all-nearest-neighbor problem can be solved in $O(n^2)$, where n is the number of points, by checking all possible pairs. When the points are in Euclidean space, the running time can be improved to $O(n \log n)$ time [19, 37].

Two notable intervals in the decomposition. The concept of rooted generators allows us to prove that, in certain cases, we can find at least two intervals in the decomposition of $\mathcal{L}S$, as in Theorem 18 below. We first prove Theorem 17, which has already appeared in [10, Theorem 5.3], where the proof method is to directly construct an endomorphism of the persistence module, as we also do after composing with the linearization functor.

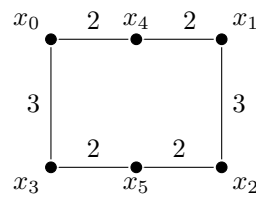
► **Theorem 17.** *Let S be a persistent set. Suppose that the preordered set of generators of S has a bottom \perp (that is, one has $\perp \leq x$ for any other generator x). Then the decomposition of $\mathcal{L}S$ consists of at least one interval, induced by \perp .*

Proof. Let $\perp \in S_{p_\perp}$ be a bottom and let $x \in S_{p_x}$ be a generator of S . Since $\perp \in S_{p_\perp}$ is a bottom, we have $p_\perp \leq p_x$. We can define an idempotent $\varphi: S \rightarrow S$ by $\varphi_{p_x}(x) = S_{p_\perp \rightarrow p_x}(\perp)$ for every generator $x \in S_{p_x}$ of S . This endomorphism is well-defined and its image has only one generator, namely \perp , and thus $\mathcal{L}(\text{img } \varphi)$ is isomorphic to an interval module. ◀

► **Theorem 18.** *Let (M, d, f) be an augmented metric space, and let $S: P \rightarrow \text{Set}$ be its density-Rips persistent set, as in Definition 2. If $|M| \geq 2$ then the decomposition of $\mathcal{L}S$ into indecomposable summands consists of at least two intervals.*

Proof. Consider a point $\top \in M$ of maximal function value, that is, $f(\top) \geq f(x)$ for any other $x \in M$. Let y be the nearest neighbor of \top . Since $M_{f(\top)} = M_\sigma$ for any $\sigma \geq f(\top)$, it is clear that \top and its nearest neighbor y satisfy the conditions of Proposition 10, and thus \top is a rooted generator, yielding the first interval. For the second interval, we note that there is at least one point $\perp \in M$ of minimal density value and apply Theorem 17. ◀

► **Example 19.** Not every summand of an indecomposable decomposition can be obtained by taking rooted subsets and applying Corollary 13. As an example, consider the augmented metric space given by six points $\{x_0, \dots, x_5\}$ in the plane as in Figure 1. Note that x_4 and x_5 are rooted in the associated density-Rips persistent set, and that they can be peeled off. After peeling, we obtain a persistent set $S: P \rightarrow \text{Set}$ with $P := \{0, 2, 3, 4\} \times \{0, 1, 2, 3, 4, 5\} \subset \mathbb{R}^2$, which we describe in Figure 2. This example is an adaptation of [14, Example 4.12], which is introduced in the context of *conquerors* that we discuss in Section 4.



■ **Figure 1** The augmented metric space (M, d, f) of Example 19, with $f(x_i) = i$. These are six points $\{x_0, \dots, x_5\}$ in the plane, where the distances are given by the numbers next to each line.

We claim that the persistence module $\mathcal{L}S: P \rightarrow \text{Vec}$ decomposes into four summands, all of them interval modules. We denote these summands by I_0, I_1, I_2 and I_3 , where each I_i is associated to the generator (p_i, x_i) of S , where $p_i = (0, i) \in P$. For each $i = 0, \dots, 3$, we set $(I_i)_p = 0$ for any $p < p_i$ and $(I_i)_{p_i} = K$, and we define $\iota_i: I_i \rightarrow \mathcal{L}S$ by

$$\begin{aligned} (\iota_0)_{p_0}(1) &= [x_0], & (\iota_1)_{p_1}(1) &= [x_1] - [x_0], \\ (\iota_2)_{p_2}(1) &= [x_2] - [x_1], & (\iota_3)_{p_3}(1) &= [x_3] - [x_0] + [x_1] - [x_2]. \end{aligned}$$

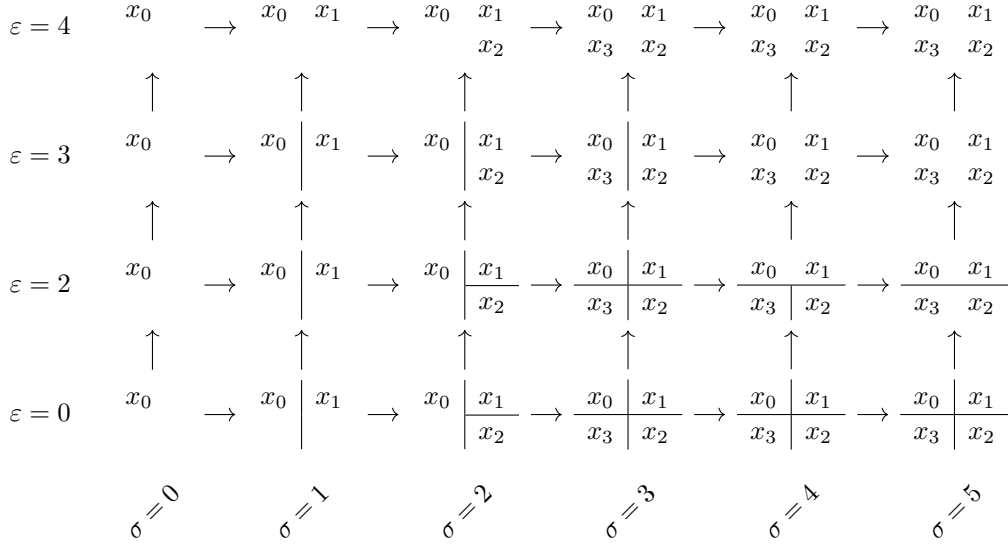
The support of each I_i are the grades $p \geq p_i$ such that $((\mathcal{L}S)_{p_i \rightarrow p} \circ (\iota_i)_{p_i})(1)$ is not zero. It can be seen that these maps induce a decomposition $\mathcal{L}S \cong I_0 \oplus I_1 \oplus I_2 \oplus I_3$.

However, no subset of the generators other than $\{x_1, x_2, x_3\}$ is rooted because each of the connected components given by $\{x_1, x_0\}$, $\{x_1, x_2\}$, $\{x_2, x_3\}$, and $\{x_0, x_3\}$ appear in S .

4 Rooted generators as a generalization of the elder rule

Single-parameter case. We now suppose that the poset P is a finite totally ordered poset. In this setting, the theory of rooted generators allows us to recover the *elder rule* [24] (see also [22] and [14]).

7:10 Decomposition of Zero-Dimensional Persistence Modules via Rooted Subsets



■ **Figure 2** The persistent set $S: P \rightarrow \text{Set}$ of Example 19 obtained by taking the density-Rips persistence set of Figure 1 and removing x_4 and x_5 . Each node in the grid represents a partition of the x_i , where x_i and x_j are in the same partition if they are not separated by a line. The arrows are the functions that send the partition of x_i in one node to the partition of x_i in the other.

► **Proposition 20.** *Let P be a finite totally ordered poset and let $S: P \rightarrow \text{Set}$ be a persistent set. Suppose that S has at least two generators and that S_\top is a singleton, where \top is the maximum element of P . Then every maximal generator (in the preorder of Definition 7) is rooted.*

Proof. Let $x \in S_{p_x}$ be a maximal generator, and define

$$I_x := \{q \in P \mid q \geq p_x \text{ and, for any other generator } w \in S_{p_w}, S_{p_w \rightarrow q}(w) \neq S_{p_x \rightarrow q}(x)\}.$$

Since $p_x \in I_x$, I_x is not empty, and we can consider the set $U \subset P$ of upper bounds of I_x . Moreover, since $S_\top = \{*\}$ and there are at least two generators by assumption, the set $U \setminus I_x$ is not empty. Let α be the least element in $U \setminus I_x$. By construction of I_x and $U \setminus I_x$, there is a generator $y \in S_{p_y}$ such that $S_{p_y \rightarrow \alpha}(y) = S_{p_x \rightarrow \alpha}(x)$. Now, since x is maximal, it holds that $p_y \leq p_x$, and we can define an idempotent $\varphi: S \rightarrow S$ by $\varphi_{p_x}(x) = S_{p_y \rightarrow p_x}(y)$, and $\varphi_{p_z}(z) = z$ for any other generator $z \in S_{p_z}$. Such an idempotent is well-defined by the way we have defined α : if there is any other generator $w \in S_{p_w}$ such that $S_{p_w \rightarrow q}(w) = S_{p_x \rightarrow q}(x)$ then $\alpha \leq q$ and also $S_{p_y \rightarrow q}(y) = S_{p_x \rightarrow q}(x)$. We conclude that x is rooted, as desired. ◀

Thus, when P is a total order, we can decompose any persistence module $\mathcal{L}S$ by peeling off rooted generators, following Theorem 12 and by iteratively considering maximal generators.

Relation to constant conquerors. Let (M, d, f) be an augmented metric space. Cai, Kim, Mémoli and Wang [14] define the concept of a constant conqueror as follows. First, define an ultrametric on M : $u(x, x') := \min\{\varepsilon \in [0, \infty) \mid x \text{ and } x' \text{ are path-connected in } \mathcal{G}_\varepsilon(M)\}$.

Now fix a total order \prec on M and let $x \in M$ be a non-minimal element with respect to this order. A **conqueror** of x in M is another point $x' \in M$ such that (1) $x' \prec x$, and (2) for

any x'' with $x'' \prec x$ one has $u(x, x') \leq u(x, x'')$. Given a function $f: M \rightarrow \mathbb{R}$, a **conqueror function** of a non-minimal $x \in M$, with respect to \prec , is a function $c_x: [f(x), \infty) \rightarrow M$ that sends each σ to a conqueror of x in M_σ . For the minimal element \perp of M we define $c_\perp: [f(\perp), \infty) \rightarrow M$ to be the constant function at \perp .

Also, in the same paper [14], given a point $x \in M$, and assuming that $f: M \rightarrow \mathbb{R}$ is injective, the authors define the **staircode** of x as the set given by

$$I_x := \{(\varepsilon, \sigma) \in \mathbb{R}_{\geq 0} \times \mathbb{R} \mid x \in M_\sigma \text{ and } x \text{ is the oldest in } [x]_{(\varepsilon, \sigma)}\},$$

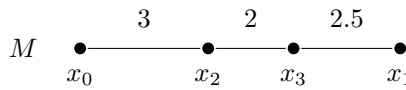
where $[x]_{(\varepsilon, \sigma)}$ is the set of points that are path-connected to x in $\mathcal{G}_\varepsilon(M_\sigma)$ and being the “oldest” means $f(x) < f(x')$ for any other $x' \in [x]_{(\varepsilon, \sigma)}$. The authors also define an analogous notion when f is not injective, which we do not reproduce here.

Finally, the authors ask the following question:

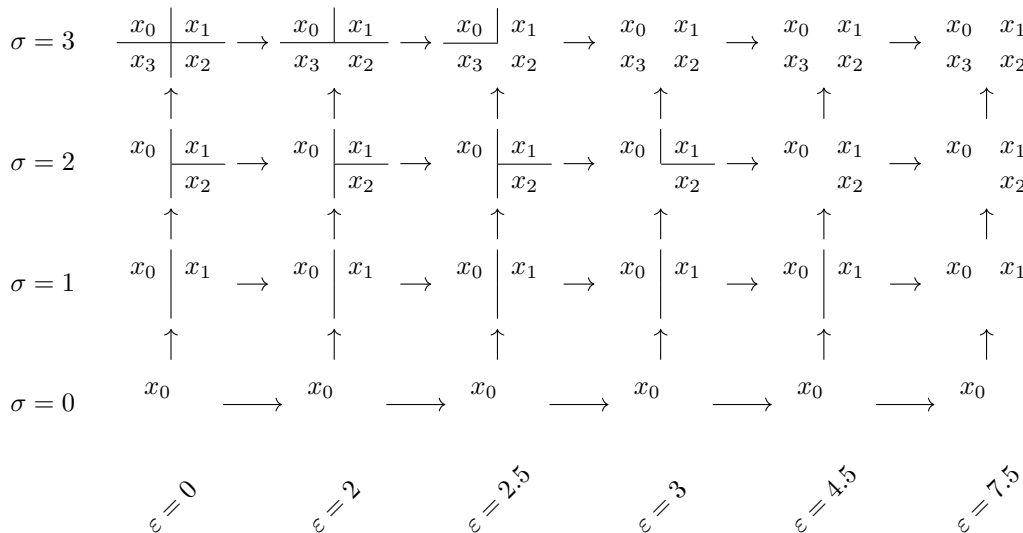
► **Question 21.** *Let (M, d, f) be an augmented metric space. If $x \in M$ has a constant conqueror function, is the interval module supported by I_x a summand of its density-Rips persistence module?*

If we replace constant conqueror by rooted generator then the answer is yes, by Corollary 13. The next example shows that the same cannot hold as originally stated in the question above.

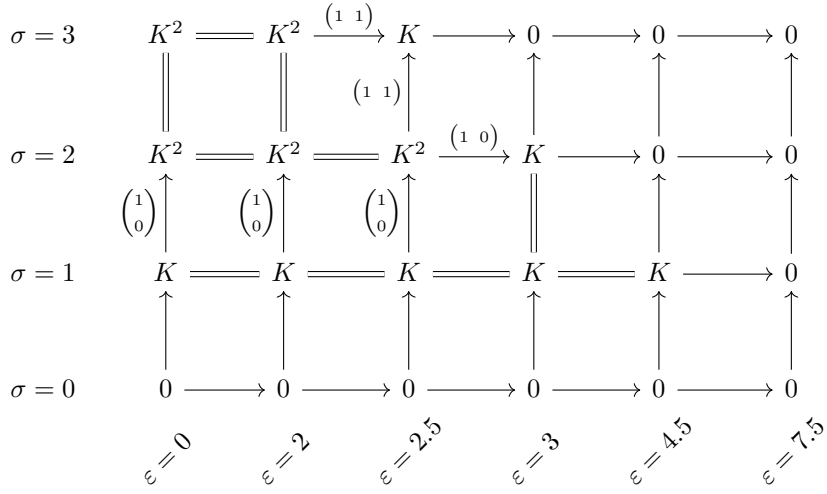
► **Example 22.** Consider the subset M of \mathbb{R} given by the points $x_0 = 0, x_1 = 7.5, x_2 = 3$ and $x_3 = 5$. Under the metric induced by the Euclidean distance on \mathbb{R} , M is a metric space, and can be made into an augmented metric space by defining $f(x_i) = i$, see Figure 3.



■ **Figure 3** The augmented metric space (M, d, f) of Example 22, with $M \subset \mathbb{R}$ and $f(x_i) = i$.



■ **Figure 4** We picture the density-Rips persistent set of Figure 3.



■ **Figure 5** An indecomposable persistence module $F: P \rightarrow \text{Vec}$, as referenced in Example 22.

Consider the only total order \prec on M compatible with f , $x_0 \prec x_1 \prec x_2 \prec x_3$. The point x_1 has a constant conqueror: x_0 is the only candidate, and it is clear that, for every $i = 1, \dots, 3$ and $x' \prec x_1$, $u_i(x_1, x_0) \leq u_i(x_1, x')$, where u_i is the ultrametric of M_i , precisely because x_0 is the only point that satisfies $x' \prec x_1$.

Let $S: P \rightarrow \text{Set}$ be the density-Rips persistent set constructed from the augmented metric space (M, d, f) . Here, P is the subsubset of \mathbb{R}^2 given by $\{0, 2, 2.5, 3, 4.5, 7.5\} \times \{0, 1, 2, 3\}$, where the first coordinate represents the distances and the second coordinate the densities. We picture S in Figure 4. Now we proceed to decompose $\mathcal{L}S$. First, note that x_3 is a rooted generator, and consider an associated idempotent $\varphi: S \rightarrow S$. By Theorem 12, there is an interval $I := \text{img}(\text{id}_{\mathcal{L}S} - \mathcal{L}\varphi)$ in the decomposition, and we can continue considering the persistent set $\text{img} \varphi$. In $\text{img} \varphi$, x_0 is a minimal generator. By Theorem 17 (and its proof) there is an idempotent $\psi: \text{img} \varphi \rightarrow \text{img} \varphi$ such that $I' := \text{img} \mathcal{L}\psi$ is an interval. Applying Theorem 12 again, we obtain a decomposition of $\mathcal{L}S$ of the form

$$I \oplus I' \oplus \text{img}(\text{id}_{\mathcal{L} \text{img} \varphi} - \mathcal{L}\psi).$$

By direct computation, it can be seen that $\text{img}(\text{id}_{\mathcal{L} \text{img} \varphi} - \mathcal{L}\psi)$ is isomorphic to the persistence module described in Figure 5. Moreover, this persistence module is indecomposable, which can be checked by looking at its endomorphisms: a persistence module F is indecomposable if and only if every endomorphism of F is either nilpotent or an isomorphism (see [9], and [10]).

Note that x_1 is not a rooted generator in $\mathcal{L}S$. In M_1 , x_1 is its own connected component during $\varepsilon \in [0, 7.5)$, until x_0 joins the connected component. And in M_3 it is by itself during $\varepsilon \in [0, 2.5)$ and then joins the connected component of x_3 , which is not connected to x_0 at that point. Similarly, x_2 is not rooted.

► **Remark 23.** Note that in Condition (2) of the definition of conqueror, we require that $x'' \prec x$. This requirement measures part of the difference between constant conqueror function and rooted generator for augmented metric spaces. If we drop this requirement, denoting the resulting concept by **conqueror***, we suppose that f is injective, and that \prec is compatible with the order induced by f , then a non-minimal, with respect to \prec , point $x \in M$ has a constant **conqueror*** function if and only if x is a rooted generator, as in Proposition 10.

5 A lower bound on the number of expected intervals

We apply the theory we have developed to the study of how a typical decomposition of a persistence module coming from density-Rips might look like. In particular, suppose we sample independently n points from a common density function $f(x)$ in \mathbb{R}^d , obtaining a finite metric space $M \subset \mathbb{R}^d$. We can then consider the augmented metric space (M, d_M, f) , where f , rather than being an estimated density, is the true underlying density function. This setting resembles actual practice, but is more suitable to theoretical study. Let S be the density-Rips persistent set of M . Then, how many intervals can we expect in the decomposition of $\mathcal{L}S$? The following theorem says that, under very general conditions on f , regardless of d , and as n goes to infinity, we can at least expect 25% of the summands to be intervals.

► **Theorem 24.** *Let X_1, \dots, X_n be i.i.d. points taking values in \mathbb{R}^d , sampled from a common density function $f(x)$ that is continuous almost everywhere with respect to the Lebesgue measure.*

Consider the finite augmented metric space $(M = \{X_1, \dots, X_n\}, d_M, f)$, where d_M is induced by the Euclidean metric in \mathbb{R}^d , and let S be its density-Rips persistent set.

Let \mathfrak{I}_n be the random variable that counts the number of intervals in the indecomposable decomposition of $\mathcal{L}S$, and let \mathfrak{S}_n be the random variable that counts the total number of summands in the same decomposition. We have

$$\liminf_{n \rightarrow \infty} \mathbb{E} \left[\frac{\mathfrak{I}_n}{\mathfrak{S}_n} \right] \geq c(d), \tag{3}$$

where $c(d)$ is a constant that depends on d , and $c(1) = \frac{1}{3}$, $c(2) \approx 0.31$ and $c(d) \downarrow \frac{1}{4}$ as $d \rightarrow \infty$.

The rest of the section is dedicated to proving this theorem. The nearest neighbor graph of a metric space plays a fundamental role.

► **Definition 25.** *The **nearest neighbor graph** of M is the directed graph on M given by the directed edges of the form (x, x') , where x' is the nearest neighbor of x .*

Now, we are interested in estimating the number of neighborly rooted elements, as in Definition 14, as they induce an interval in the decomposition of $\mathcal{L}S$. However, in general being neighborly rooted depends on f . To do without the condition on f we have:

► **Lemma 26.** *Let (M, d_M, f) be an augmented metric space and let S be its density-Rips persistent set. There are at least as many intervals in the indecomposable decomposition of $\mathcal{L}S$ as 2-cycles in the nearest neighbor graph of M .*

Proof. We can assume without loss of generality that $|M| \geq 2$. Let G be the nearest neighbor graph of M . The only cycles in this graph are precisely the 2-cycles, and each weakly connected component of G contains exactly one 2-cycle (see [25]).

Let C_1, \dots, C_k be the weakly connected components of G . Fix $i \in \{1, \dots, k\}$, and let $x, y \in M$ be such that (x, y) and (y, x) is the 2-cycle in C_i . Either $f(y) \leq f(x)$ or $f(x) \leq f(y)$, and either x is neighborly rooted, y is neighborly rooted, or both are neighborly rooted. Say x is neighborly rooted, and define an endomorphism $\varphi_i: S \rightarrow S$ by setting

$$(\varphi_i)_{p_z}(z) = \begin{cases} S_{p_y \rightarrow p_x}(y), & \text{if } x = z, \\ z, & \text{otherwise,} \end{cases}$$

for every generator $z \in S_{p_z}$. Such an endomorphism is well-defined as shown in Proposition 10.

Constructing, for each i , an idempotent φ_i as above, it is clear that we can iteratively peel off the associated intervals, yielding the desired conclusion. ◀

Naturally, the number of 2-cycles is half the number of points that are the nearest neighbor of its nearest neighbor. The problem of estimating the probability for a point to be the nearest neighbor of its nearest neighbor, assuming a random point process, has been studied by multiple authors (see [21, 25, 26, 27, 35]).

In our case, when we have X_1, \dots, X_n i.i.d. points in \mathbb{R}^d sampled from a common density function f under the conditions of Theorem 24, by [27, Theorem 1.1], and letting $N_{i,n}$ denote the probability event that X_i is the nearest neighbor of its nearest neighbor, we have

$$\lim_{n \rightarrow \infty} \mathbb{P}(N_{i,n}) = b(d), \quad (4)$$

where $b(d)$ is the volume of a unit d -sphere divided by the volume of the union of two unit spheres with centers at distance 1. In fact, $b(1) = \frac{2}{3}$, $b(2) \approx 0.621$, and $b(d) \downarrow \frac{1}{2}$ as $d \rightarrow \infty$ (see [35, Table 2]), and we define $c(d) := \frac{b(d)}{2}$.

We are now ready to finish the proof of Theorem 24 at the start of the section. Applying Lemma 26 and the linearity of expectation, it holds

$$\mathbb{E}[\mathcal{J}_n] \geq \mathbb{E} \left[\sum_{i=1}^n \frac{I(N_{i,n})}{2} \right] = \sum_{i=1}^n \frac{\mathbb{E}[I(N_{i,n})]}{2} = \sum_{i=1}^n \frac{\mathbb{P}(N_{i,n})}{2},$$

where $I(N_i)$ is the indicator random variable of $N_{i,n}$. By Equation (4) we have

$$\liminf_{n \rightarrow \infty} \mathbb{E} \left[\frac{\mathcal{J}_n}{n} \right] \geq \frac{b(d)}{2} = c(d).$$

Finally, noting that the number of summands in the decomposition is bounded by the number of points, $\mathfrak{S}_n \leq n$ (see full version), Equation (3) of Theorem 24 follows, finishing the proof.

6 Discussion

Although we have focused our attention to augmented metric spaces and density-Rips, rooted subsets can be applied to other persistent sets. Of special interest for us is the degree-Rips filtration [5] of a metric space, where we filter by the degree of the vertices in the underlying geometric graphs. To accommodate this situation, one could modify condition 1 of Proposition 10 to take into account the evolution of the degrees, rather than the density. We leave an in-depth treatment of this case for future work.

We have seen, both in our lower bound of Section 5 and in preliminary experimental evaluation, that we can expect to find many intervals in the decomposition of those persistence modules coming from geometry, at least in the cases considered here. This is in contrast to the purely algebraic setting, where, in light of recent developments [2, 3], looking for a decomposition might fall short.

References

- 1 Hideto Asashiba, Mickaël Buchet, Emerson G. Escolar, Ken Nakashima, and Michio Yoshikawa. On interval decomposability of 2D persistence modules. *Computational Geometry*, 105/106:Paper No. 101879, 33, 2022. doi:10.1016/j.comgeo.2022.101879.
- 2 Ulrich Bauer, Magnus B. Botnan, Steffen Oppermann, and Johan Steen. Cotorsion torsion triples and the representation theory of filtered hierarchical clustering. *Advances in Mathematics*, 369:107171, 51, 2020. doi:10.1016/j.aim.2020.107171.
- 3 Ulrich Bauer and Luis Scoccola. Generic two-parameter persistence modules are nearly indecomposable, November 2022. arXiv:2211.15306.

- 4 Håvard Bakke Bjerkevik, Magnus Bakke Botnan, and Michael Kerber. Computing the interleaving distance is NP-hard. *Foundations of Computational Mathematics*, 20(5):1237–1271, 2020. doi:10.1007/s10208-019-09442-y.
- 5 Andrew J. Blumberg and Michael Lesnick. Stability of 2-parameter persistent homology. *Foundations of Computational Mathematics*, 2022. doi:10.1007/s10208-022-09576-6.
- 6 Magnus Bakke Botnan and William Crawley-Boevey. Decomposition of persistence modules. *Proceedings of the American Mathematical Society*, 148(11):4581–4596, 2020. doi:10.1090/proc/14790.
- 7 Magnus Bakke Botnan and Michael Lesnick. Algebraic stability of zigzag persistence modules. *Algebraic & Geometric Topology*, 18(6):3133–3204, 2018. doi:10.2140/agt.2018.18.3133.
- 8 Magnus Bakke Botnan, Steffen Oppermann, Steve Oudot, and Luis Scoccola. On the bottleneck stability of rank decompositions of multi-parameter persistence modules, July 2022. arXiv:2208.00300.
- 9 Michel Brion. Representations of quivers. In *Geometric methods in representation theory. I*, volume 24 of *Sémin. Congr.*, pages 103–144. Soc. Math. France, Paris, 2012.
- 10 Jacek Brodzki, Matthew Burfitt, and Mariam Pirashvili. On the complexity of zero-dimensional multiparameter persistence, August 2020. arXiv:2008.11532.
- 11 Mickaël Buchet, Frédéric Chazal, Tamal K. Dey, Fengtao Fan, Steve Y. Oudot, and Yusu Wang. Topological analysis of scalar fields with outliers. In *31st International Symposium on Computational Geometry*, volume 34 of *LIPICs. Leibniz Int. Proc. Inform.*, pages 827–841, 2015. doi:10.4230/LIPICs.S0CG.2015.827.
- 12 Mickaël Buchet and Emerson G. Escolar. Every 1D persistence module is a restriction of some indecomposable 2D persistence module. *Journal of Applied and Computational Topology*, 4(3):387–424, 2020. doi:10.1007/s41468-020-00053-z.
- 13 Mickaël Buchet and Emerson G. Escolar. Realizations of indecomposable persistence modules of arbitrarily large dimension. *Journal of Computational Geometry*, 13(1):298–326, 2022. doi:10.20382/jocg.v13i1a12.
- 14 Chen Cai, Woojin Kim, Facundo Mémoli, and Yusu Wang. Elder-rule-staircodes for augmented metric spaces. *SIAM Journal on Applied Algebra and Geometry*, 5(3):417–454, 2021. doi:10.1137/20M1353605.
- 15 Gunnar Carlsson and Facundo Mémoli. Characterization, stability and convergence of hierarchical clustering methods. *Journal of Machine Learning Research*, 11:1425–1470, 2010.
- 16 Gunnar Carlsson and Facundo Mémoli. Multiparameter hierarchical clustering methods. In *Classification as a tool for research*, Stud. Classification Data Anal. Knowledge Organ., pages 63–70. Springer, Berlin, 2010. doi:10.1007/978-3-642-10745-0_6.
- 17 Gunnar Carlsson and Facundo Mémoli. Classifying clustering schemes. *Foundations of Computational Mathematics*, 13(2):221–252, 2013. doi:10.1007/s10208-012-9141-9.
- 18 Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009. doi:10.1007/s00454-009-9176-0.
- 19 Kenneth L. Clarkson. Fast algorithms for the all nearest neighbors problem. In *24th Annual Symposium on Foundations of Computer Science*, pages 226–232. IEEE, 1983. doi:10.1109/SFCS.1983.16.
- 20 Anne Collins, Afra Zomorodian, Gunnar Carlsson, and Leonidas J. Guibas. A barcode shape descriptor for curve point cloud data. *Computers & Graphics*, 28(6):881–894, 2004. doi:10.1016/j.cag.2004.08.015.
- 21 Trevor F. Cox. Reflexive nearest neighbours. *Biometrics*, 37(2):367, 1981. doi:10.2307/2530424.
- 22 Justin Curry. The fiber of the persistence map for functions on the interval. *Journal of Applied and Computational Topology*, 2(3-4):301–321, 2018. doi:10.1007/s41468-019-00024-z.
- 23 Tamal K. Dey and Cheng Xin. Generalized persistence algorithm for decomposing multiparameter persistence modules. *Journal of Applied and Computational Topology*, 6(3):271–322, 2022. doi:10.1007/s41468-022-00087-5.

- 24 Herbert Edelsbrunner and John L. Harer. *Computational topology: an introduction*. American Mathematical Society, Providence, RI, 2010. doi:10.1090/mbk/069.
- 25 David Eppstein, Michael S. Paterson, and Frances F. Yao. On nearest-neighbor graphs. *Discrete & Computational Geometry*, 17(3):263–282, 1997. doi:10.1007/PL00009293.
- 26 Norbert Henze. On the probability that a random point is the j th nearest neighbour to its own k th nearest neighbour. *Journal of Applied Probability*, 23(1):221–226, 1986. doi:10.2307/3214132.
- 27 Norbert Henze. On the fraction of random points by specified nearest-neighbour interrelations and degree of attraction. *Advances in Applied Probability*, 19(4):873–895, 1987. doi:10.2307/1427106.
- 28 Jon Kleinberg. An impossibility theorem for clustering. In *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002.
- 29 Michael Lesnick and Matthew Wright. Interactive visualization of 2-d persistence modules, December 2015. arXiv:1512.00180.
- 30 Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer New York, 1978. doi:10.1007/978-1-4757-4721-8.
- 31 Leland McInnes and John Healy. Accelerated hierarchical density clustering. In *2017 IEEE International Conference on Data Mining Workshops*, pages 33–42, 2017. doi:10.1109/ICDMW.2017.12.
- 32 Samantha Moore. Hyperplane restrictions of indecomposable n -dimensional persistence modules. *Homology, Homotopy and Applications*, 24(2):291–305, 2022. doi:10.4310/HHA.2022.v24.n2.a14.
- 33 Alexander Rolle and Luis Scoccola. Persistable: Persistent and stable clustering. URL: <https://github.com/LuisScoccola/persistable>.
- 34 Alexander Rolle and Luis Scoccola. Stable and consistent density-based clustering, July 2021. arXiv:2005.09048.
- 35 Mark F. Schilling. Mutual and shared neighbor probabilities: finite- and infinite-dimensional results. *Advances in Applied Probability*, 18(2):388–405, 1986. doi:10.2307/1427305.
- 36 Bernard W. Silverman. *Density estimation for statistics and data analysis*. Monographs on Statistics and Applied Probability. Chapman & Hall, London, 1986.
- 37 Pravin M. Vaidya. An $O(n \log n)$ algorithm for the all-nearest-neighbors problem. *Discrete & Computational Geometry*, 4(2):101–115, 1989. doi:10.1007/BF02187718.

On Helly Numbers of Exponential Lattices

Gergely Ambrus ✉

Department of Geometry, Bolyai Institute, University of Szeged, Hungary
Alfréd Rényi Institute of Mathematics, Budapest, Hungary

Martin Balko ✉

Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic

Nóra Frankl ✉

School of Mathematics and Statistics, The Open University, Milton Keynes, UK
Alfréd Rényi Institute of Mathematics, Budapest, Hungary

Attila Jung ✉

Institute of Mathematics, ELTE Eötvös Loránd University, Budapest, Hungary

Márton Naszódi ✉

Department of Geometry, ELTE Eötvös Loránd University, Budapest, Hungary
MTA-ELTE Lendület Combinatorial Geometry Research Group, Budapest, Hungary

Abstract

Given a set $S \subseteq \mathbb{R}^2$, define the *Helly number* of S , denoted by $H(S)$, as the smallest positive integer N , if it exists, for which the following statement is true: for any finite family \mathcal{F} of convex sets in \mathbb{R}^2 such that the intersection of any N or fewer members of \mathcal{F} contains at least one point of S , there is a point of S common to all members of \mathcal{F} .

We prove that the Helly numbers of *exponential lattices* $\{\alpha^n : n \in \mathbb{N}_0\}^2$ are finite for every $\alpha > 1$ and we determine their exact values in some instances. In particular, we obtain $H(\{2^n : n \in \mathbb{N}_0\}^2) = 5$, solving a problem posed by Dillon (2021).

For real numbers $\alpha, \beta > 1$, we also fully characterize exponential lattices $L(\alpha, \beta) = \{\alpha^n : n \in \mathbb{N}_0\} \times \{\beta^n : n \in \mathbb{N}_0\}$ with finite Helly numbers by showing that $H(L(\alpha, \beta))$ is finite if and only if $\log_\alpha(\beta)$ is rational.

2012 ACM Subject Classification Mathematics of computing → Combinatoric problems

Keywords and phrases Helly numbers, exponential lattices, Diophantine approximation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.8

Related Version *Full Version*: <https://arxiv.org/abs/2301.04683>

Funding *Gergely Ambrus*: Partially supported by ERC Advanced Grant “GeoScape”, by the Hungarian National Research grant no. NKFIH KKP-133819, and by project no. TKP2021-NVA-09, which has been implemented with the support provided by the Ministry of Innovation and Technology of Hungary from the National Research, Development and Innovation Fund, financed under the TKP2021-NVA funding scheme.

Martin Balko: Supported by the grant no. 21/32817S of the Czech Science Foundation (GAČR) and by the Center for Foundations of Modern Computer Science (Charles University project UNCE/S-CI/004). This article is part of a project that has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 810115).

Nóra Frankl: Partially supported by ERC Advanced Grant “GeoScape”.

Attila Jung: Supported by the Rényi Doctoral Fellowship of the Rényi Institute.

Márton Naszódi: Supported by the János Bolyai Scholarship of the Hungarian Academy of Sciences.

Acknowledgements This research was initiated at the 11th Emléktábla workshop on combinatorics and geometry. We would like to thank Géza Tóth for interesting discussions about the problem during the early stages of the research



© Gergely Ambrus, Martin Balko, Nóra Frankl, Attila Jung, and Márton Naszódi; licensed under Creative Commons License CC-BY 4.0

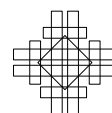
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 8; pp. 8:1–8:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Helly's theorem [11] is one of the most classical results in combinatorial geometry. It states that, for each $d \in \mathbb{N}$, if the intersection of any $d + 1$ or fewer members of a finite family \mathcal{F} of convex sets in \mathbb{R}^d is nonempty, then the entire family \mathcal{F} has nonempty intersection. There have been numerous variants and generalizations of this famous result; see [1, 13] for example. One active direction of this research with rich connections to the theory of optimization, in particular to integer programming and LP-type problems [1, 4], is the study of variants of Helly's theorem with coordinate restrictions, which is captured by the following definition.

Let d be a positive integer. The *Helly number* of a set $S \subseteq \mathbb{R}^d$, denoted by $H(S)$, is the smallest positive integer N , if it exists, such that the following statement is true for every finite family \mathcal{F} of convex sets in \mathbb{R}^d : if the intersection of any N or fewer members of \mathcal{F} contains at least one point of S , then $\bigcap \mathcal{F}$ contains at least one point of S . If no such number N exists, then we write $H(S) = \infty$. Helly's theorem in this language can be restated as $H(\mathbb{R}^d) = d + 1$.

A classical result of this sort is *Doignon's theorem* [8] where the set S is the integer lattice \mathbb{Z}^d . This result, which was also independently discovered by Bell [3] and by Scarf [15], states that $H(\mathbb{Z}^d) \leq 2^d$. This is tight as for $Q = \{0, 1\}^d$ the intersection of any $2^d - 1$ sets in the family $\{\text{conv}(Q \setminus \{x\}) : x \in Q\}$ contains a lattice point, but the intersection of all 2^d sets does not.

The theory of Helly numbers of general sets is developing quickly and there are many results of this kind [1, 13]. For example, De Loera, La Haye, Oliveros, and Roldán-Pensado [5] and De Loera, La Haye, Rolnick, and Soberón [6] studied the Helly numbers of differences of lattices and Garber [9] considered Helly numbers of crystals or cut-and-project sets.

The Helly number of a set S is closely related to the maximum size of a set that is empty in S . A subset $X \subseteq S$ is *intersect-empty* if $(\bigcap_{x \in X} \text{conv}(X \setminus \{x\})) \cap S = \emptyset$. A convex polytope P with vertices in S is *empty in S* if P does not contain any points of S other than its vertices. In particular, an empty polytope does not contain points of S in the interior of its edges. For a discrete set S , we use $h(S)$ to denote the maximum number of vertices of an empty polytope in S . If there are empty polytopes in S with arbitrarily large number of vertices, then we write $h(S) = \infty$.

The following result by Hoffman [12] (which was essentially already proved by Doignon [8]) shows the close connection between intersect-empty sets and empty polytopes in S and the S -Helly numbers; see also [2].

► **Proposition 1** ([12]). *If $S \subseteq \mathbb{R}^d$, then $H(S)$ is equal to the maximum cardinality of an intersect-empty set in S . If S is discrete, then $H(S) = h(S)$.*

Since all the sets S studied in this paper are discrete, we state all of our results using $h(\alpha)$ but, due to Proposition 1, our results apply to $H(\alpha)$ as well.

Very recently, Dillon [7] proved that the Helly number of a set S is infinite if S belongs to a certain collection of *product sets*, which are sets of the form $S = A^d$ with a certain kind of discrete set $A \subseteq \mathbb{R}$. His result shows, for example, that whenever p is a polynomial of degree at least 2 and $d \geq 2$, then $h(\{p(n) : n \in \mathbb{N}_0\}^d) = \infty$. However, there are sets for which Dillon's method gives no information, for example $\{2^n : n \in \mathbb{N}_0\}^2$. Thus, Dillon [7] posed the following question, which motivated our research.

► **Problem 1** (Dillon, [7]). *What is $h(\{2^n : n \in \mathbb{N}_0\}^2)$?*

In this paper, we study the Helly numbers of *exponential lattices* $L(\alpha)$ and $L(\alpha, \beta)$ in the plane where $L(\alpha) = \{\alpha^n : n \in \mathbb{N}_0\}^2$ and $L(\alpha, \beta) = \{\alpha^n : n \in \mathbb{N}_0\} \times \{\beta^n : n \in \mathbb{N}_0\}$ for real numbers $\alpha, \beta > 1$. In particular, we prove that Helly numbers of exponential lattices $L(\alpha)$

are finite and we provide several estimates that give exact values for α sufficiently large, solving Problem 1. We also show that Helly numbers of exponential lattices $L(\alpha, \beta)$ are finite if and only if $\log_\alpha(\beta)$ is rational.

2 Our results

For a real number $\alpha > 1$ and the exponential lattice $L(\alpha) = \{\alpha^n : n \in \mathbb{N}_0\}^2$, we abbreviate $h(L(\alpha))$ by $h(\alpha)$.

As our first result, we provide finite bounds on the numbers $h(\alpha)$ for any $\alpha > 1$. The upper bounds are getting smaller as α increases and reach their minimum at $\alpha = 2$.

► **Theorem 2.** *For every real $\alpha > 1$, the maximum number of vertices of an empty polygon in $L(\alpha)$ is finite. More precisely, we have $h(\alpha) \leq 5$ for every $\alpha \geq 2$, $h(\alpha) \leq 7$ for every $\alpha \in [\frac{1+\sqrt{5}}{2}, 2)$, and*

$$h(\alpha) \leq 3 \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha - 1} \right) \right\rceil + 3$$

for every $\alpha \in (1, \frac{1+\sqrt{5}}{2})$.

We note that if $\alpha = 1 + \frac{1}{x}$ for $x \in (0, \infty)$, then the bound from Theorem 2 becomes $h(1 + \frac{1}{x}) \leq O(x \log_2(x))$. Moreover, we show that the breaking points of α for our upper bounds are determined by certain polynomial equations; see Section 3.

We also consider the lower bounds on $h(\alpha)$ and provide the following estimate.

► **Theorem 3.** *We have $h(\alpha) \geq 5$ for every $\alpha \geq 2$ and $h(\alpha) \geq 7$ for every $\alpha \in [\frac{1+\sqrt{5}}{2}, 2)$. For every $\alpha \in (1, \frac{1+\sqrt{5}}{2})$, we have*

$$h(\alpha) \geq \left\lfloor \sqrt{\frac{1}{\alpha - 1}} \right\rfloor.$$

If $\alpha = 1 + \frac{1}{x}$ where $x \in (0, \infty)$, then the lower bound from Theorem 3 becomes $h(1 + \frac{1}{x}) \geq \lfloor \sqrt{x} \rfloor$. So with decreasing α , the parameter $h(\alpha)$ indeed grows to infinity.

By combining Theorems 2 and 3, we get the precise value of the Helly numbers of $L(\alpha)$ with $\alpha \geq (1 + \sqrt{5})/2$. In particular, for $\alpha = 2$, we obtain a solution to Problem 1.

► **Corollary 4.** *We have $h(\alpha) = 5$ for every $\alpha \geq 2$ and $h(\alpha) = 7$ for every $\alpha \in [\frac{1+\sqrt{5}}{2}, 2)$.*

We prove the following result which shows that even a slight perturbation of S can affect the value $h(S)$ drastically (note that this also follows by adding large empty polygons to S without changing its asymptotic density). The proof is omitted here. We use the *Fibonacci numbers* $(F_n)_{n \in \mathbb{N}_0}$, which are defined as $F_0 = 1, F_1 = 1$ and $F_n = F_{n-1} + F_{n-2}$ for every integer $n \geq 2$.

► **Proposition 5.** *We have $h(\{F_n : n \in \mathbb{N}_0\}^2) = \infty$.*

We recall that $F_n = \frac{\varphi^{n+1} - \psi^{n+1}}{\sqrt{5}}$ for every $n \in \mathbb{N}_0$, where $\varphi = \frac{1+\sqrt{5}}{2}$ is the *golden ratio* and $\psi = \frac{1-\sqrt{5}}{2} = 1 - \varphi$ is its conjugate. Since $\psi < 1$, this formula shows that the points of $\{F_n : n \in \mathbb{N}_0\}^2$ are approaching the points of the scaled exponential lattice $\frac{\varphi}{\sqrt{5}} \cdot L(\varphi) = \{\frac{\varphi}{\sqrt{5}} \cdot \varphi^n : n \in \mathbb{N}_0\}^2$. Thus, Proposition 5 is in sharp contrast with the fact

8:4 On Helly Numbers of Exponential Lattices

that $h(\frac{\varphi}{\sqrt{5}} \cdot L(\varphi)) = h(\varphi) \leq 7$, which follows from Theorem 2 and from the fact that affine transformations of any set $S \subseteq \mathbb{R}^d$ do not change $h(S)$. We also note Dillon's method [7] does not imply $h(\{F_n : n \in \mathbb{N}_0\}^2) = \infty$.

We also consider the more general case of exponential lattices where the rows and the columns might use different bases. For real numbers $\alpha > 1$ and $\beta > 1$, let $L(\alpha, \beta)$ be the set $\{\alpha^n : n \in \mathbb{N}_0\} \times \{\beta^n : n \in \mathbb{N}_0\}$. Note that $L(\alpha) = L(\alpha, \alpha)$ for every $\alpha > 1$.

As our last main result, we fully characterize exponential lattices $L(\alpha, \beta)$ with finite Helly numbers $h(L(\alpha, \beta))$, settling the question of finiteness of Helly numbers of planar exponential lattices completely.

► **Theorem 6.** *Let $\alpha > 1$ and $\beta > 1$ be real numbers. Then $h(L(\alpha, \beta))$ is finite if and only if $\log_\alpha(\beta)$ is a rational number.*

Moreover, if $\log_\alpha(\beta) \in \mathbb{Q}$, that is, $\beta = \alpha^{p/q}$ for some $p, q \in \mathbb{N}$, then

$$\left\lfloor \frac{1}{pq} \left\lfloor \sqrt{\frac{1}{\alpha^{1/q} - 1}} \right\rfloor \right\rfloor \leq h(L(\alpha, \beta)) \leq pq \cdot h(\alpha^p).$$

The proof of the 'only if' part of Theorem 6 is based on the theory of continued fractions and Diophantine approximation. The details are discussed in Section 5. The proof of the 'if' part of Theorem 6 is based on Theorem 2 and is omitted here.

Open problems

First, it is natural to try to close the gap between the upper bound from Theorem 2 and the lower bound from Theorem 3 and potentially obtain new precise values of $h(\alpha)$.

Second, we considered only the exponential lattice in the plane, but it would be interesting to obtain some estimates on the Helly numbers of exponential lattices $\{\alpha^n : n \in \mathbb{N}_0\}^d$ in dimension $d > 2$.

We also mention the following conjecture of De Loera, La Haye, Oliveros, and Roldán-Pensado [5], which inspired the research of Dillon [7].

► **Conjecture 7** ([5]). *If \mathcal{P} is the set of prime numbers, then $h(\mathcal{P}^2) = \infty$.*

Using computer search, Summers [16] showed that $h(\mathcal{P}^2) \geq 14$.

3 Proof of Theorem 2

Here, we prove Theorem 2 by showing that the number $h(\alpha)$ is finite for every $\alpha > 1$. This follows from the upper bounds $h(\alpha) \leq 5$ for $\alpha \geq 2$, $h(\alpha) \leq 7$ for every $\alpha \geq [\frac{1+\sqrt{5}}{2}, 2)$, and

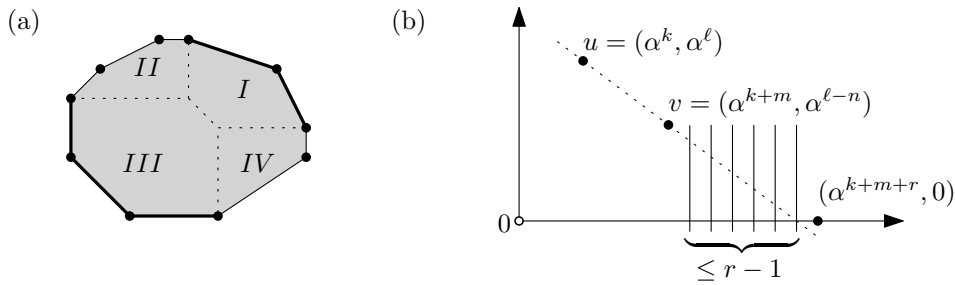
$$h(\alpha) \leq 3 \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha - 1} \right) \right\rceil + 3$$

for any $\alpha \in (1, \frac{1+\sqrt{5}}{2})$.

We start by introducing some auxiliary definitions and notation. Let $\alpha > 1$ be a real number and consider the exponential lattice $L(\alpha)$. For $i \in \mathbb{N}_0$, the i th column of $L(\alpha)$ is the set $\{\alpha^i, \alpha^n : n \in \mathbb{N}_0\}$. Analogously, the i th row of $L(\alpha)$ is the set $\{\alpha^n, \alpha^i : n \in \mathbb{N}_0\}$.

For a point p in the plane, we write $x(p)$ and $y(p)$ for the x - and y -coordinates of p , respectively. Let P be an empty convex polygon in $L(\alpha)$. Let e be an edge of P connecting vertices u and v where $x(u) < x(v)$ or $y(u) < y(v)$ if $x(u) = x(v)$. We use \bar{e} to denote the line determined by e and oriented from u to v . The slope of e is the slope of \bar{e} , that is, $\frac{y(v)-y(u)}{x(v)-x(u)}$.

We distinguish four types of edges of P ; see part (a) of Figure 1. First, assume $x(u) \neq x(v)$ and $y(u) \neq y(v)$. We say that e is of *type I* if the slope of e is negative and P lies to the right of \bar{e} . Similarly, e is of *type II* if the slope of e is positive and P lies to the right of \bar{e} . An edge e has *type III* if the slope of e is negative and P lies to the left of \bar{e} . Finally, *type IV* is for e with positive slope and with P lying to the left of \bar{e} . It remains to deal with horizontal and vertical edges of P . A horizontal edge e is of type II if P lies below \bar{e} and is of type III otherwise. Similarly, a vertical edge e is of type IV if P lies to the left of \bar{e} and is of type III otherwise.



■ **Figure 1** (a) The four types of edges of a convex polygon. (b) An illustration of the proof of Lemma 8.

Note that each edge of P has exactly one type and that the types partition the edges of P into four convex chains. We first provide an upper bound on the number of edges of those chains of P and then derive the bound on the total number of edges of P by summing the four bounds. We start by estimating the number of edges of P of type I.

► **Lemma 8.** *The polygon P has at most $\lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \rceil$ edges of type I.*

Proof. First, let $r = \lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \rceil$ and note that $r \geq 1$ as $\alpha > 1$. Let e be the left-most edge of P of type I and let u and v be vertices of e . Since e is of type I, we have $u = (\alpha^k, \alpha^\ell)$ and $v = (\alpha^{k+m}, \alpha^{\ell-n})$ for some positive integers k, ℓ, m , and n .

We will show that the point $(\alpha^{k+m+r}, 0)$ lies above the line \bar{e} . Since there are at most $r - 1$ columns of $L(\alpha)$ between the vertical line containing v and the vertical line containing $(\alpha^{k+m+r}, 0)$ and the point $(\alpha^{k+m+r}, 0)$ is below the lowest row of $L(\alpha)$, it then follows that there are at most r edges of P of type I; see part (b) of Figure 1.

Since the line \bar{e} contains u and v , we see that

$$\bar{e} = \{(x, y) \in \mathbb{R}^2 : (\alpha^\ell - \alpha^{\ell-n})x + (\alpha^{k+m} - \alpha^k)y = \alpha^{k+\ell+m} - \alpha^{k+\ell-n}\}.$$

It suffices to check that by substituting the coordinates of the point $(\alpha^{k+m+r}, 0)$ into the equation of the line \bar{e} results in a left side that is at least $\alpha^{k+\ell+m} - \alpha^{k+\ell-n}$. The left side equals $\alpha^{k+\ell+m+r} - \alpha^{k+\ell+m-n+r}$ and thus we want

$$\alpha^{k+\ell+m+r} - \alpha^{k+\ell+m-n+r} \geq \alpha^{k+\ell+m} - \alpha^{k+\ell-n}.$$

By dividing both sides by $\alpha^{k+\ell}$ and by rearranging the terms, we can rewrite this expression as

$$\alpha^{-n}(1 - \alpha^{m+r}) \geq \alpha^m - \alpha^{m+r}.$$

Since $m, r > 0$ and $\alpha > 1$, we get $(1 - \alpha^{m+r}) < 0$ and thus the left side is increasing as n increases, so we can assume $n = 1$, leading to

$$\alpha^{-1} - \alpha^{m+r-1} \geq \alpha^m - \alpha^{m+r}.$$

8:6 On Helly Numbers of Exponential Lattices

We can again rearrange the inequality as

$$\alpha^r - \alpha^{r-1} - 1 \geq -\alpha^{-1-m},$$

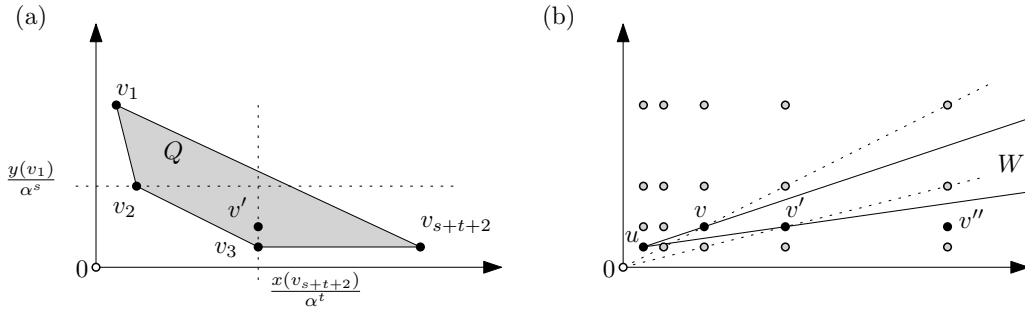
where the right side is negative and approaches 0 as m tends to infinity, so we can replace it by 0, obtaining

$$\alpha^r - \alpha^{r-1} \geq 1.$$

This inequality is satisfied by our choice of r . ◀

We now estimate the number of edges of P that are of type III.

► **Lemma 9.** *The polygon P has at most $2\lceil \log_\alpha \left(\frac{\alpha+1}{\alpha}\right) \rceil + 1$ edges of type III for $1 < \alpha < 2$ and at most 2 such edges for $\alpha \geq 2$.*



■ **Figure 2** (a) An illustration of the proof of Lemma 9 for $s = 1 = t$. (b) An illustration of Lemma 10.

Proof. Let $t = \lceil \log_\alpha \left(\frac{\alpha+1}{\alpha}\right) \rceil$ and $s = t + 1$ for $\alpha \in (1, 2)$ and $t = 1 = s$ for $\alpha \geq 2$. Suppose for contradiction that there are $s + t + 1$ edges of P of type III. Let v_1, \dots, v_{s+t+2} be the vertices of the convex chain that is formed by edges of P of type III. We use Q to denote the convex polygon with vertices v_1, \dots, v_{s+t+2} . Note that Q is empty in $L(\alpha)$ as P is empty and $Q \subseteq P$.

Let v' be the point $(x(v_{s+2}), \alpha \cdot y(v_{s+2}))$, that is, v' is the point of $L(\alpha)$ that lies just above v_{s+2} ; see part (a) of Figure 2. We will show that the point v' lies below the line $\overline{v_1 v_{s+t+2}}$. Since v' lies in the same column of $L(\alpha)$ as v_{s+2} , this then implies that v' lies in the interior of Q , contradicting the fact that Q is empty in $L(\alpha)$.

Note that $x(v') \leq \frac{x(v_{s+t+2})}{\alpha^t}$ and $y(v') \leq \frac{y(v_1)}{\alpha^s}$ as all edges $v_i v_{i+1}$ are of type III and thus the x - and y -coordinates decrease by a multiplicative factor at least α for each such edge. Since the only vertical edge might be $v_1 v_2$ and the only horizontal edge might be $v_{s+t+1} v_{s+t+2}$, the x - or y -coordinates indeed decrease by the factor α at each step.

Let $v_1 = (\alpha^k, \alpha^\ell)$ and $v_{s+t+2} = (\alpha^{k+m}, \alpha^{\ell-n})$ for some positive integers k, ℓ, m, n . Note that $m, n \geq s + t$. The line determined by v_1 and v_{s+t+2} is then

$$\{(x, y) \in \mathbb{R}^2 : (\alpha^\ell - \alpha^{\ell-n})x + (\alpha^{k+m} - \alpha^k)y = \alpha^{k+\ell+m} - \alpha^{k+\ell-n}\}.$$

Since $x(v') \leq \frac{x(v_{s+t+2})}{\alpha^t}$ and $y(v') \leq \frac{y(v_1)}{\alpha^s}$, it suffices to check

$$(\alpha^\ell - \alpha^{\ell-n})\frac{\alpha^{k+m}}{\alpha^t} + (\alpha^{k+m} - \alpha^k)\frac{\alpha^\ell}{\alpha^s} < \alpha^{k+\ell+m} - \alpha^{k+\ell-n}.$$

After dividing by $\alpha^{k+\ell+m}$, this can be rewritten as

$$\alpha^{-t} + \alpha^{-s} < 1 - \alpha^{-m-n} + \alpha^{-t-n} + \alpha^{-s-m}.$$

Since $m, n \geq s + t$, the right hand side is decreasing with increasing m and n and thus we only need to prove

$$\alpha^{-s} + \alpha^{-t} \leq 1.$$

If $\alpha \geq 2$, then $s = 1 = t$ and this inequality becomes $2/\alpha \leq 1$, which is true. If $\alpha \in (1, 2)$, then $s = t + 1$ and the inequality becomes $1 + 1/\alpha \leq \alpha^t$ which holds by our choice of t . ◀

It remains to bound the number of edges of P that are of types II and IV. Observe that if we switch the x - and y - coordinates of P , then edges of type II become edges of type IV and vice versa. Since the exponential lattice $L(\alpha)$ is symmetric with respect to the line $x = y$, we see that it suffices to estimate the number of edges of type II. To do so, we use the following auxiliary result, the proof of which is omitted here.

► **Lemma 10.** *Let u be a point of $L(\alpha)$ and let v and v' be two points of $L(\alpha)$ that are consecutive in a row R of $L(\alpha)$ that lies above the row containing u ; see part (b) of Figure 2.*

Then, all points of $L(\alpha)$ that lie above R in the interior of the wedge W spanned by the lines \overline{uv} and $\overline{uv'}$ lie on at most $\left\lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \right\rceil$ lines containing the origin.

Now, we can apply Lemma 10 to obtain an upper bound on the number of edges of P of type II.

► **Lemma 11.** *The polygon P has at most $\left\lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \right\rceil + 1$ edges of type II.*

Proof. Again, let $r = \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \right\rceil$. Let u be the leftmost vertex of the convex chain C determined by the edges of P of type II. Similarly, let v be the second leftmost vertex of C . Note that since the edge uv is of type II, the vertex v lies in a row R of $L(\alpha)$ above the row containing u . Let v' be the point $(\alpha \cdot x(v), y(v))$, that is, point of $L(\alpha)$ that is to the right of v on R .

Then, by Lemma 10, all points of $L(\alpha)$ that lie above R and in the interior of the wedge W spanned by the lines \overline{uv} and $\overline{uv'}$ lie on at most r lines containing the origin.

Since P is empty in $L(\alpha)$, all vertices of C besides u and v and possibly v' lie in W above R . Since all edges of C are of type II, every line determined by the origin and by a point of $L(\alpha)$ from the interior of W contains at most one vertex of C .

Note that if v' is a vertex of C , then the only vertices of C are u, v, v' . Thus, in total C has at most $r + 2$ vertices and therefore at most $r + 1$ edges. ◀

We recall that, by symmetry, the same bound applies for edges of type IV and thus we get the following result.

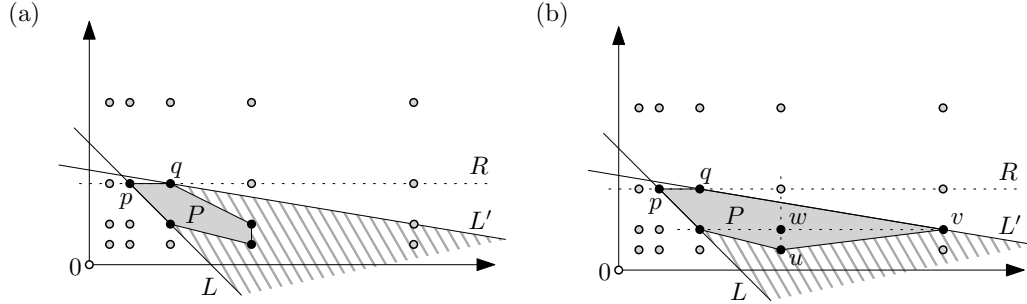
► **Corollary 12.** *The polygon P has at most $\left\lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \right\rceil + 1$ edges of type IV.* ◀

Since each edge of P is of one of the types I-IV, it immediately follows from Lemmas 8, 9, 11, and from Corollary 12 that the number of edges of P is at most

$$3 \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \right\rceil + 2 + 2 \left\lceil \log_\alpha \left(\frac{\alpha+1}{\alpha} \right) \right\rceil + 1 \leq 5 \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha-1} \right) \right\rceil + 3,$$

as $\log_x \left(\frac{x}{x-1} \right) \geq \log_x \left(\frac{x+1}{x} \right)$ for every $x > 1$. In particular, this gives $h(2) \leq 8$ and $h \left(\frac{1+\sqrt{5}}{2} \right) \leq 13$. To obtain better bounds that are tight for $\alpha \geq \frac{1+\sqrt{5}}{2}$, we observe that not all types can appear simultaneously. To show this, we will use one last auxiliary result.

Let p and q be (not necessarily different) points lying on the same row R of $L(\alpha)$, each contained in an edge of P . Let L and L' be two lines containing p and q , respectively. If the slopes of L and L' are negative, then we call the part of the plane between L and L' below R a *slice of negative slope*; see part (a) of Figure 3. Analogously, a *slice of positive slope* is the part of the plane between L and L' above R if L and L' have positive slope.



■ **Figure 3** (a) An example of a slice of negative slope. The slice is denoted by dark gray stripes. (b) An illustration of the proof of Lemma 13 for negative slopes.

► **Lemma 13.** *If the empty polygon P is contained in a slice of negative slope, then there is no non-vertical edge of P of type IV. Similarly, if P is contained in a slice of positive slope, then there is no edge of type I.*

Proof. By symmetry, it suffices to prove the statement for slices of negative slope. Suppose for contradiction that there is a non-vertical edge uv of type IV in a slice of negative slope determined by lines L and L' and points p and q as in the definition of a slice. Without loss of generality, we assume $x(u) < x(v)$.

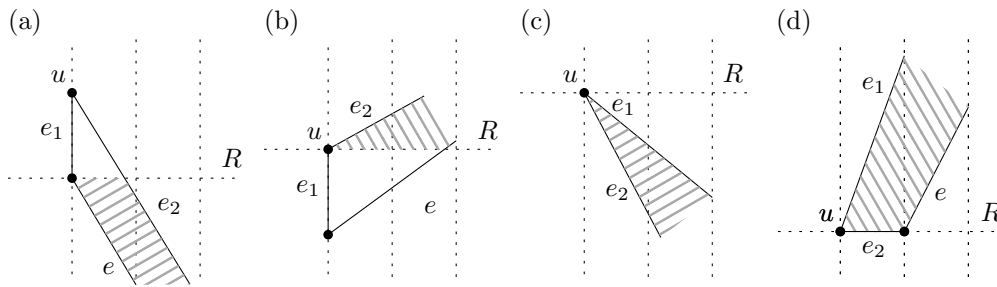
Consider the point $w = (x(u), y(v))$ of $L(\alpha)$. Since uv is non-vertical, we have $w \notin \{u, v\}$. We claim that w is in the interior of P , contradicting the assumption that P is empty in $L(\alpha)$. Since uv is of type IV, the point u lies below the row containing w . However, since p is contained in an edge of P and P is in the slice, the boundary of P intersects this row to the left of w . Analogously, v is to the right of the column containing w and thus the boundary of P intersects this column above w . Then, however, w lies in the interior of P . ◀

Finally, we can now finish the proof of Theorem 2.

Proof of Theorem 2. First, we observe that if all vertices of P lie on two columns of $L(\alpha)$, then P can have at most four vertices. So we assume that this is not the case. Let u be the leftmost vertex of P with the highest y -coordinate among all leftmost vertices of P . Let e_1 and e_2 be the edges of P incident to u . We denote the other edge of P incident to e_1 as e . We also use t_I, t_{II}, t_{III} , and t_{IV} to denote the number of edges of P of type I, II, III, and IV, respectively.

First, assume that e_1 is vertical. If e_2 is horizontal, then, since u is the top vertex of e_1 and P is not contained in two columns of $L(\alpha)$, the point $(\alpha \cdot x(u), y(u)/\alpha)$ of $L(\alpha)$ lies in the interior of P , which is impossible as P is empty in $L(\alpha)$.

If e_1 is vertical and the slope of e_2 is negative, then there is no edge of type II. Thus, the edge e intersects the row R of $L(\alpha)$ containing the other vertex of e_1 and \bar{e} has negative



■ **Figure 4** An illustration of the proof of Theorem 2.

slope. Then, the part of P below R is contained in the slice of negative slope determined by \bar{e}_2 and \bar{e} ; see part (a) of Figure 4. By Lemma 13, there is no non-vertical edge of type IV in P . By Lemmas 8 and 9, the total number of edges of P is thus at most

$$t_I + t_{III} + 1 \leq \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha - 1} \right) \right\rceil + 2 \left\lceil \log_\alpha \left(\frac{\alpha + 1}{\alpha} \right) \right\rceil + 2$$

for $\alpha \in (1, 2)$ and is by one smaller for $\alpha \geq 2$.

If e_1 is vertical and the slope of e_2 is positive, then, since P is empty, there is no edge of type III besides e_1 as otherwise the point $(\alpha \cdot x(u), y(u))$ of $L(\alpha)$ is in the interior of P . The edge e intersects the row R of $L(\alpha)$ containing u and \bar{e} has positive slope. Thus, the part of P above R is contained in the slice of positive slope determined by \bar{e}_2 and \bar{e} ; see part (b) of Figure 4. By Lemma 13, there is no edge of type I in P . By Lemma 11 and Corollary 12, the total number of edges of P is then at most

$$t_{II} + 1 + t_{IV} \leq 2 \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha - 1} \right) \right\rceil + 3.$$

In the rest of the proof, we can now assume that none of the edges e_1 and e_2 is vertical. We can label them so that the slope of e_1 is larger than the slope of e_2 .

First, assume that the slope of e_1 is positive and the slope of e_2 is negative. Then, since the vertices of P do not lie on two columns of $L(\alpha)$, the point $(\alpha \cdot x(u), y(u))$ is contained in the interior of P , which is impossible as P is empty in $L(\alpha)$.

If the slopes of e_1 and e_2 are both non-positive, then there is no edge of type II besides the possibly horizontal edge e_1 as u is the leftmost vertex of P . By Lemma 13, there is also no non-vertical edge of type IV as P is contained in the slice of negative slopes determined by \bar{e}_1 and \bar{e}_2 or by \bar{e} and \bar{e}_2 if e_1 is horizontal; see part (c) of Figure 4. Thus, by Lemmas 8 and 9, the number of edges of P is at most

$$t_I + 1 + t_{III} + 1 \leq \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha - 1} \right) \right\rceil + 2 \left\lceil \log_\alpha \left(\frac{\alpha + 1}{\alpha} \right) \right\rceil + 3$$

for $\alpha \in (1, 2)$ and is by one smaller for $\alpha \geq 2$.

If the slopes of e_1 and e_2 are both non-negative, then there is no edge of type III besides the possibly horizontal edge e_2 (note that a vertical edge of type III would have u as its bottom vertex, which is impossible by the choice of u). Then, P is contained in the slice of positive slope determined by \bar{e}_1 and \bar{e}_2 or, if e_2 is horizontal, by \bar{e}_1 and \bar{e} ; see part (d) of Figure 4. Lemma 13 then implies that there is also no edge of type I. We thus have at most

$$t_{II} + 1 + t_{IV} \leq 2 \left\lceil \log_\alpha \left(\frac{\alpha}{\alpha - 1} \right) \right\rceil + 3$$

8:10 On Helly Numbers of Exponential Lattices

edges of P by Lemma 11 and Corollary 12.

Altogether, the upper bound on the number of edges of P is

$$\max \left\{ \left\lceil \log_{\alpha} \left(\frac{\alpha}{\alpha-1} \right) \right\rceil + 2 \left\lceil \log_{\alpha} \left(\frac{\alpha+1}{\alpha} \right) \right\rceil + 3, 2 \left\lceil \log_{\alpha} \left(\frac{\alpha}{\alpha-1} \right) \right\rceil + 3 \right\}$$

for $\alpha \in (1, 2)$ and the first term is smaller by 1 for $\alpha \geq 2$. This becomes 5 for $\alpha \geq 2$, $h(\alpha) \leq 7$ for $\alpha \geq [\frac{1+\sqrt{5}}{2}, 2)$, and at most $3 \left\lceil \log_{\alpha} \left(\frac{\alpha+1}{\alpha-1} \right) \right\rceil + 3$ otherwise, since $\left\lceil \log_{\alpha} \left(\frac{\alpha+1}{\alpha} \right) \right\rceil \leq \left\lceil \log_{\alpha} \left(\frac{\alpha}{\alpha-1} \right) \right\rceil$ for every $\alpha \in (1, \frac{1+\sqrt{5}}{2})$. ◀

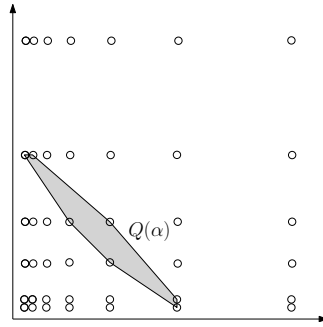
4 Proof of Theorem 3

We prove the lower bounds on $h(\alpha)$ through the following three propositions.

► **Proposition 14.** *For every $\alpha \geq 2$, we have $h(\alpha) \geq 5$.*

Proof. It is easy to check that $\text{conv}\{(1, \alpha^2), (\alpha, \alpha), (\alpha^2, 1), (\alpha^2, \alpha), (\alpha, \alpha^2)\}$ is an empty polygon in $L(\alpha)$ with 5 vertices for any α . ◀

► **Proposition 15.** *For every $\alpha \in [\frac{1+\sqrt{5}}{2}, 2)$, we have $h(\alpha) \geq 7$.*



■ **Figure 5** An illustration of the proof of Proposition 15.

Proof. Let $k = k(\alpha)$ be a sufficiently large integer, and let

$$Q(\alpha) = \{(1, \alpha^k), (\alpha^{k-2}, \alpha^{k-1}), (\alpha^{k-1}, \alpha^{k-2}), (\alpha^k, 1), (\alpha^k, \alpha), (\alpha^{k-1}, \alpha^{k-1}), (\alpha, \alpha^k)\};$$

see Figure 5. We will show that $\text{conv}(Q(\alpha))$ is an empty polygon in $L(\alpha)$ with 7 vertices.

First, we show that $Q(\alpha) \setminus \{(\alpha^{k-1}, \alpha^{k-1})\}$ is in convex position. For this, by symmetry, it is enough to check that the vector $(\alpha^{k-1}, \alpha^{k-2}) - (\alpha^k, 1)$ is to the left of $(1, \alpha^k) - (\alpha^k, 1)$. This is the case exactly if $\alpha^{k-1} - \alpha^k + \alpha^{k-2} - 1 < 0$. By rearranging we get $\alpha^{k-2}(\alpha + 1 - \alpha^2) < 1$, which holds for any k , since $\alpha + 1 - \alpha^2 \leq 0$ as $\alpha \geq (1 + \sqrt{5})/2$.

Now, to show that the set $Q(\alpha)$ is in convex position, it is sufficient to check that $(\alpha^{k-1}, \alpha^{k-1}) - (\alpha^k, \alpha)$ is to the left of $(1, \alpha^k) - (\alpha^k, \alpha)$. This holds exactly if $\alpha^{k-1} - \alpha^k + \alpha^{k-1} - \alpha \geq 0$. By rearranging we get $2\alpha^{k-2}(2 - \alpha) \geq 1$. Since $1 < \alpha < 2$, this holds if k is sufficiently large.

Thus, $\text{conv}(Q(\alpha))$ has 7 vertices. To show that $\text{conv}(Q(\alpha))$ is empty in $L(\alpha)$, we remark that points of the exponential lattice $L(\alpha)$ with at least one coordinate smaller than α^{k-1} are below the line through $(\alpha^{k-1}, \alpha^{k-2})$ and $(\alpha^{k-2}, \alpha^{k-1})$. Further, points with at least one coordinate larger than α^{k-1} are either above the line through $(1, \alpha^k)$ and (α, α^k) or to the right of the line through $(\alpha^k, 1)$ and (α^k, α) . ◀

► **Proposition 16.** For every $\alpha > 1$, we have $h(\alpha) \geq \lfloor \sqrt{\frac{1}{\alpha-1}} \rfloor$.

Proof. For a positive integer k , let $P(k) = \{(\alpha^i, \alpha^{k-i}) : 1 \leq i \leq k\}$. Since $P(k)$ is contained in the hyperbola $h = \{(x, y) \in \mathbb{R}^2 : x, y > 0, xy = \alpha^k\}$, the points of $P(k)$ are in convex position, and $\text{conv}(P(k))$ has k vertices. We will show that if $k \leq \sqrt{\frac{1}{\alpha-1}}$, then $\text{conv}(P(k))$ is empty.

For points (x, y) of $L(\alpha)$ above h , we have $xy \geq \alpha^{k+1}$. Further, points (x, y) of $L(\alpha)$ with $xy \geq \alpha^{k+2}$ are separated from h by the hyperbola $h' = \{(x, y) \in \mathbb{R}^2 : x, y > 0, xy = \alpha^{k+1}\}$. Thus, it is sufficient to check that h' is above the line ℓ connecting $(1, \alpha^k)$ with $(\alpha^k, 1)$. The closest point of h' to ℓ is $(\alpha^{(k+1)/2}, \alpha^{(k+1)/2})$, thus it is sufficient to check that this point is above ℓ . This holds if $2\alpha^{(k+1)/2} - \alpha^k - 1 \geq 0$ and we show that this inequality is satisfied for $k \leq \sqrt{\frac{1}{\alpha-1}}$.

Let $\alpha = 1 + s^2$ with some $s \in (0, 1)$. In this notation, $k \leq 1/s$ and we need to prove that $2(1 + s^2)^{(k+1)/2} \geq (1 + s^2)^k + 1$. Since $(1 + s^2)^{(k+1)/2} \geq 1 + s^2 \frac{k+1}{2}$ by the Bernoulli inequality, and $(1 + s^2)^k \leq e^{s^2 k}$, it is sufficient to prove the stronger inequality $2(1 + s^2 \frac{k+1}{2}) \geq e^{s^2 k} + 1$. The worst case, when $k = 1/s$, is equivalent to $1 + s + s^2 \geq e^s$, which holds for $s \in (0, 1)$ as can be seen by the Taylor expansion of e^s . ◀

5 Proof of 'only if part' of Theorem 6

Let $\alpha, \beta > 1$ be two real numbers. We prove that if $\log_\alpha(\beta)$ is irrational, then $h(L(\alpha, \beta))$ is not finite.

To do so, we will find a subset of $L(\alpha, \beta)$ forming empty convex polygon in $L(\alpha, \beta)$ with arbitrarily many vertices. To do so, we use a theory of continued fractions, so we first introduce some definitions and notation.

5.1 Continued fractions

Here, we recall mostly basic facts about so-called continued fractions, which we use in the proof. Most of the results that we state can be found, for example, in the book by Khinchin [14].

For a positive real number r , the (simple) continued fraction of r is an expression of the form

$$r = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \dots}}}$$

where $a_0 \in \mathbb{N}_0$ and a_1, a_2, \dots are positive integers. The simple continued fraction of r can be written in a compact notation as

$$[a_0; a_1, a_2, a_3, \dots].$$

For every $n \in \mathbb{N}_0$, if we denote $\frac{p_n}{q_n} = [a_0; a_1, a_2, \dots, a_n]$ and set $p_{-1} = 1, p_0 = a_0, q_{-1} = 0, q_0 = 1$, then the numbers p_n and q_n satisfy the recurrence

$$p_n = a_n p_{n-1} + p_{n-2} \quad \text{and} \quad q_n = a_n q_{n-1} + q_{n-2} \tag{1}$$

for each $n \in \mathbb{N}$. Observe that if r is irrational, then its continued fraction has infinitely many coefficients. Also, it follows from (1) that $\frac{p_n}{q_n} < r$ for n even and $\frac{p_n}{q_n} > r$ for n odd.

For example, if $r = \log_2(3)$, we get the continued fraction $[1; 1, 1, 2, 2, 3, 1, 5, 2, 23, \dots]$ and the sequence $\left(\frac{p_n}{q_n}\right)_{n \in \mathbb{N}_0} = \left(\frac{1}{1}, \frac{2}{1}, \frac{3}{2}, \frac{8}{5}, \frac{19}{12}, \frac{65}{41}, \frac{84}{53}, \frac{485}{306}, \dots\right)$. For $r = \frac{1+\sqrt{5}}{2}$, we have $[1; 1, 1, 1, \dots]$ and $\left(\frac{p_n}{q_n}\right)_{n \in \mathbb{N}_0} = \left(\frac{1}{1}, \frac{2}{1}, \frac{3}{2}, \frac{5}{3}, \frac{8}{5}, \frac{13}{8}, \frac{21}{13}, \frac{34}{21}, \dots\right)$.

We will call the fractions $\frac{p_n}{q_n}$ the *convergents* of r . A *semi-convergent* of r is a number $\frac{p_{n-1}+ip_n}{q_{n-1}+iq_n}$ where $i \in \{0, 1, \dots, a_{n+1}\}$. Note that each convergent of r is also a semi-convergent of r . The names are motivated by the use of convergents and semi-convergents as rational approximations of an irrational number r .

A rational number $\frac{p}{q}$ is a *best approximation* of an irrational number r , if any fraction $\frac{p'}{q'} \neq \frac{p}{q}$ with $q' < q$ satisfies

$$\left|q' \left(r - \frac{p'}{q'}\right)\right| > \left|q \left(r - \frac{p}{q}\right)\right|.$$

A rational number $\frac{p}{q}$ is a *best lower approximation* of r if

$$q' \left(r - \frac{p'}{q'}\right) > q \left(r - \frac{p}{q}\right) \geq 0$$

for all rational numbers $\frac{p'}{q'}$ with $\frac{p'}{q'} \leq r$, $\frac{p}{q} \neq \frac{p'}{q'}$, and $0 < q' \leq q$. Similarly, $\frac{p}{q}$ is a *best upper approximation* of r if

$$q' \left(r - \frac{p'}{q'}\right) < q \left(r - \frac{p}{q}\right) \leq 0$$

for all rational numbers $\frac{p'}{q'}$ with $\frac{p'}{q'} \geq r$, $\frac{p}{q} \neq \frac{p'}{q'}$, and $0 < q' \leq q$.

It is a well known fact that convergents are best approximations of r [14]. The following lemma about best lower and upper best approximations is a recent result of Hančl and Turek [10].

► **Lemma 17** ([10]). *Let r be a real number with $r = [a_0; a_1, a_2, \dots]$ and let $\frac{p_n}{q_n}$ be the n th convergent of r for each $n \in \mathbb{N}_0$. Then, the following three statements hold.*

1. *The set of best lower approximations of r consists of semi-convergents $\frac{p_{n-1}+ip_n}{q_{n-1}+iq_n}$ of r with n odd and $0 \leq i < a_{n+1}$.*
2. *The set of best upper approximations of r consists of semi-convergents $\frac{p_{n-1}+ip_n}{q_{n-1}+iq_n}$ of r with n even and $0 \leq i < a_{n+1}$, except for the pair $(n, i) = (0, 0)$.*

Finally, a real number r is *restricted* if there is a positive integer M such that all the partial denominators a_i from the continued fraction of r are at most M . The restricted numbers are exactly those numbers r that are badly approximable by rationals [14], that is, there is a constant $c > 0$ such that for every $\frac{p}{q} \in \mathbb{Q}$ we have $\left|r - \frac{p}{q}\right| > \frac{c}{q^2}$.

We divide the rest of the proof of Theorem 6 into two cases, depending on whether $\log_\alpha(\beta)$ is restricted or not.

5.2 Unrestricted case

First, we assume that $\log_\alpha(\beta)$ is not restricted. Let $[a_0; a_1, a_2, a_3, \dots]$ be the continued fraction of $\log_\alpha(\beta)$ with $\frac{p_n}{q_n} = [a_0; a_1, \dots, a_n]$ for every $n \in \mathbb{N}_0$. Then, for every positive integer m , there is a positive integer $n(m)$ such that $a_{n(m)+1} \geq m$. We use this assumption to construct, for every positive integer m , a convex polygon with at least m vertices from $L(\alpha, \beta)$ that is empty in $L(\alpha, \beta)$.

For a given m , consider the integer $n(m)$ and let W be the set of points

$$w_i = (\alpha^{p_{n(m)-1+i}p_{n(m)}}, \beta^{q_{n(m)-1+i}q_{n(m)}})$$

where $i \in \{0, 1, \dots, a_{n(m)+1}\}$. That is, we consider points where the exponents form semi-convergents $\frac{p_{n(m)-1+i}p_{n(m)}}{q_{n(m)-1+i}q_{n(m)}}$ to $\log_\alpha(\beta)$. We abbreviate $p_{n,i} = p_{n(m)-1+i}p_{n(m)}$ and $q_{n,i} = q_{n(m)-1+i}q_{n(m)}$. Observe that $|W| \geq m$. We will show that W is the vertex set of an empty convex polygon in $L(\alpha, \beta)$. To do so, we assume without loss of generality that $n(m)$ is even so that $\frac{\beta^{q_{n(m)}}}{\alpha^{p_{n(m)}}} > 1$. The other case when $n(m)$ is odd is analogous.

First, we show that W is in convex position. In fact, we prove that all triples $(w_{i_1}, w_{i_2}, w_{i_3})$ with $i_1 < i_2 < i_3$ are oriented counterclockwise. It suffices to show this for every triple (w_i, w_{i+1}, w_{i+2}) . To do so, we need to prove the inequality

$$\frac{y(w_{i+2}) - y(w_{i+1})}{x(w_{i+2}) - x(w_{i+1})} = \frac{\beta^{q_{n,i+2}} - \beta^{q_{n,i+1}}}{\alpha^{p_{n,i+2}} - \alpha^{p_{n,i+1}}} > \frac{\beta^{q_{n,i+1}} - \beta^{q_{n,i}}}{\alpha^{p_{n,i+1}} - \alpha^{p_{n,i}}} = \frac{y(w_{i+1}) - y(w_i)}{x(w_{i+1}) - x(w_i)}.$$

After dividing by $\frac{\beta^{q_{n(m)-1}}}{\alpha^{p_{n(m)-1}}}$, this can be written as

$$\frac{\beta^{(i+2)q_{n(m)}} - \beta^{(i+1)q_{n(m)}}}{\alpha^{(i+2)p_{n(m)}} - \alpha^{(i+1)p_{n(m)}}} > \frac{\beta^{(i+1)q_{n(m)}} - \beta^{iq_{n(m)}}}{\alpha^{(i+1)p_{n(m)}} - \alpha^{ip_{n(m)}}}.$$

If divide both sides by $\frac{\beta^{(i+1)q_{n(m)}} - \beta^{iq_{n(m)}}}{\alpha^{(i+1)p_{n(m)}} - \alpha^{ip_{n(m)}}}$, then the above inequality becomes

$$\frac{\beta^{q_{n(m)}}}{\alpha^{p_{n(m)}}} > 1.$$

This is true as $n(m)$ is even.

It remains to prove that the polygon Q with the vertex set W is empty in $L(\alpha, \beta)$. Suppose for contradiction that there is a point (α^p, β^q) of $L(\alpha, \beta)$ lying in the interior of Q . Let i be the minimum positive integer from $\{1, \dots, a_{n(m)+1}\}$ such that $q < q_{n,i}$. Such an i exists as (α^p, β^q) is in the interior of Q . We then have $q_{n,i-1} < q < q_{n,i}$. Since (α^p, β^q) is in the interior of Q and W lies below the line $x = y$, we have $\frac{p}{q} > \log_\alpha(\beta)$. So it is enough to prove that (α^p, β^q) does not lie above the line $\overline{w_{i-1}w_i}$.

We have $p_{n,i} - \log_\alpha(\beta)q_{n,i} < p_{n,i-1} - \log_\alpha(\beta)q_{n,i-1}$ as $\frac{p_{n,i}}{q_{n,i}}$ is a best upper approximation of $\log_\alpha(\beta)$ and $q_{n,i-1} < q_{n,i}$. This implies $\frac{\beta^{q_{n,i-1}}}{\alpha^{p_{n,i-1}}} < \frac{\beta^{q_{n,i}}}{\alpha^{p_{n,i}}}$, or equivalently that w_i lies above the line determined by w_{i-1} and the origin.

Now if (α^p, β^q) lies above the line $\overline{w_{i-1}w_i}$, then it also lies above the line determined by w_{i-1} and the origin. Thus, $\frac{\beta^{q_{n,i-1}}}{\alpha^{p_{n,i-1}}} < \frac{\beta^q}{\alpha^p}$, implying

$$p - \log_\alpha(\beta)q < p_{n,i-1} - \log_\alpha(\beta)q_{n,i-1},$$

which means that $\frac{p}{q}$ is a better upper approximation of $\log_\alpha(\beta)$ than $\frac{p_{n,i-1}}{q_{n,i-1}}$. Thus, there exists a best upper approximation $\frac{p^*}{q^*}$ of $\log_\alpha(\beta)$ with $q_{n,i-1} < q^* < q_{n,i}$. This contradicts part (c) of Lemma 17 as $\frac{p^*}{q^*}$ is not a semi-convergent of $\log_\alpha(\beta)$.

5.3 Restricted case

Now, assume that the number $\log_\alpha(\beta)$ is restricted. Let $[a_0; a_1, a_2, a_3, \dots]$ be the continued fraction of $\log_\alpha(\beta)$ with $\frac{p_n}{q_n} = [a_0; a_1, \dots, a_n]$ for every $n \in \mathbb{N}_0$. Let $M = M(\alpha, \beta)$ be a number satisfying

$$a_n \leq M \tag{2}$$

8:14 On Helly Numbers of Exponential Lattices

for every $n \in \mathbb{N}_0$ and let $c = c(\alpha, \beta) > 0$ be a constant such that

$$\left| \log_\alpha(\beta) - \frac{p}{q} \right| > \frac{c}{q^2} \quad (3)$$

holds for every $\frac{p}{q} \in \mathbb{Q}$. Recall that $\frac{\alpha^{p_n}}{\beta^{q_n}} < 1$ for even n and $\frac{\alpha^{p_n}}{\beta^{q_n}} > 1$ for odd n . Note also that the sequence $\left(\frac{\alpha^{p_n}}{\beta^{q_n}}\right)_{n \in \mathbb{N}_0}$ converges to 1 as $\left(\frac{p_n}{q_n}\right)_{n \in \mathbb{N}_0}$ converges to $\log_\alpha(\beta)$. Moreover, the terms of $\left(\frac{p_n}{q_n}\right)_{n \in \mathbb{N}_0}$ with odd indices form a decreasing subsequence and the terms with even indices determine an increasing subsequence.

Let $n_0 = n_0(\alpha, \beta)$ be a sufficiently large positive integer and let V be the set of points $v_n = (\alpha^{p_n}, \beta^{q_n})$ for every odd $n \geq n_0$. Note that V is a subset of $L(\alpha, \beta)$.

We first show that V is in convex position. In fact, we prove a stronger claim by showing that the orientation of every triple $(v_{n_1}, v_{n_2}, v_{n_3})$ with $n_1 < n_2 < n_3$ is counterclockwise. It suffices to show this for every triple (v_{n-4}, v_{n-2}, v_n) . To do so, we prove that the slopes of the lines determined by consecutive points of V are increasing, that is,

$$\frac{y(v_n) - y(v_{n-2})}{x(v_n) - x(v_{n-2})} = \frac{\beta^{q_n} - \beta^{q_{n-2}}}{\alpha^{p_n} - \alpha^{p_{n-2}}} > \frac{\beta^{q_{n-2}} - \beta^{q_{n-4}}}{\alpha^{p_{n-2}} - \alpha^{p_{n-4}}} = \frac{y(v_{n-2}) - y(v_{n-4})}{x(v_{n-2}) - x(v_{n-4})}$$

for every even $n \geq n_0$. By dividing both sides of the inequality with $\frac{\beta^{q_{n-2}}}{\alpha^{p_{n-2}}}$, we rewrite this expression as

$$\frac{\beta^{q_n - q_{n-2}} - 1}{\alpha^{p_n - p_{n-2}} - 1} > \frac{1 - \beta^{q_{n-4} - q_{n-2}}}{1 - \alpha^{p_{n-4} - p_{n-2}}}.$$

Using (1), this is the same as

$$\frac{\beta^{a_n q_{n-1}} - 1}{\alpha^{a_n p_{n-1}} - 1} > \frac{1 - \beta^{-a_{n-2} q_{n-3}}}{1 - \alpha^{-a_{n-2} p_{n-3}}}.$$

The above inequality can be rewritten as

$$(\beta^{a_n q_{n-1}} - 1)(1 - \alpha^{-a_{n-2} p_{n-3}}) > (\alpha^{a_n p_{n-1}} - 1)(1 - \beta^{-a_{n-2} q_{n-3}}),$$

where $\beta^{q_{n-1}} > \alpha^{p_{n-1}} > 1$ and $1 > \alpha^{-p_{n-3}} > \beta^{-q_{n-3}} > 0$ as $n-1$ and $n-3$ are even. Therefore, if the above inequality holds for $a_n = 1 = a_{n-2}$, then it holds for any a_n and a_{n-1} as both numbers are always at least 1. Thus, it suffices to show

$$(\beta^{q_{n-1}} - 1)(1 - \alpha^{-p_{n-3}}) > (\alpha^{p_{n-1}} - 1)(1 - \beta^{-q_{n-3}}). \quad (4)$$

We prove this using the following simple auxiliary lemma.

► **Lemma 18.** Consider the function $f: \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}$ given by $f(x, y) = (x-1)(1-1/y)$. Let $x, y, x', y' > 1$ be real numbers such that $1 - \frac{1}{y} - \frac{x}{x'} > 0$. Then, $f(x', y) > f(x, y')$.

Proof. We have

$$\begin{aligned} f(x', y) - f(x, y') &= (x' - 1) \left(1 - \frac{1}{y}\right) - (x - 1) \left(1 - \frac{1}{y'}\right) \\ &= x' - \frac{x' - 1}{y} - x + \frac{x - 1}{y'} > x' - \frac{x'}{y} - x = x' \left(1 - \frac{1}{y} - \frac{x}{x'}\right) > 0, \end{aligned}$$

where the last inequality follows from $1 - \frac{1}{y} - \frac{x}{x'} > 0$. ◀

Now, by choosing $x = \alpha^{p_{n-1}}$, $x' = \beta^{q_{n-1}}$, $y = \alpha^{p_{n-3}}$, and $y' = \beta^{q_{n-3}}$, the inequality (4) becomes $f(x', y) > f(x, y')$. In order to prove it, we just need to verify the assumptions of Lemma 18. We clearly have $x, x', y, y' > 1$. It now suffices to show $1 - \frac{1}{y} - \frac{x}{x'} > 0$. By (3), we obtain that $q_{n-1} \log_\alpha(\beta) - p_{n-1} \geq c/q_{n-1}$, thus

$$\frac{x}{x'} = \frac{\alpha^{p_{n-1}}}{\beta^{q_{n-1}}} \leq \alpha^{-c/q_{n-1}}.$$

Now, to bound q_{n-1} in terms of p_{n-3} , equation (1) gives

$$\begin{aligned} q_{n-1} &= a_{n-1}q_{n-2} + q_{n-3} \leq (M + 1)q_{n-2} = (M + 1)(a_{n-2}q_{n-3} + q_{n-4}) \\ &\leq (M + 1)^2q_{n-3} \leq 2 \log_\beta(\alpha)(M + 1)^2p_{n-3}, \end{aligned}$$

where we used (2) and $q_{n-4} \leq q_{n-3} \leq q_{n-2}$, $q_{n-3} \leq 2 \log_\beta(\alpha)p_{n-3}$ for n large enough. It follows that $q_{n-1} \leq M'p_{n-3}$ for a suitable constant $M' = M'(\alpha, \beta) > 0$. Thus,

$$1 - \frac{1}{y} - \frac{x}{x'} \geq 1 - \alpha^{-p_{n-3}} - \alpha^{-c/q_{n-1}} \geq 1 - \alpha^{-p_{n-3}} - \alpha^{-c/(M'p_{n-3})},$$

which is at least

$$\frac{c \ln \alpha}{2M'p_{n-3}} - \frac{1}{\alpha^{p_{n-3}}}$$

as $1 - c \ln \alpha / (2M'p_{n-3}) \geq e^{-2c \ln \alpha / (2M'p_{n-3})} = \alpha^{-c/(M'p_{n-3})}$ if $0 < c \ln \alpha / (2M'p_{n-3}) < 1/2$. The last expression is positive if $n \geq n_0$ and n_0 is sufficiently so that p_{n-3} is large enough.

It remains to show that the convex polygon P with the vertex set V is empty in $L(\alpha, \beta)$. We proceed analogously as in the unrestricted case. Suppose for contradiction that there is a point (α^p, β^q) of $L(\alpha, \beta)$ lying in the interior of P . Then, let $v_n = (\alpha^{p_n}, \beta^{q_n})$ be the lowest vertex of P that has (α^p, β^q) below. Such a vertex v_n exists, as V contains points with arbitrarily large y -coordinate. By the choice of v_n , we obtain $q_{n-2} < q < q_n$. Since (α^p, β^q) is in the interior of P and V lies below the line $x = y$, we have $\frac{p}{q} > \log_\alpha(\beta) > \frac{p_{n-1}}{q_{n-1}}$. Moreover, since all triples from V are oriented counterclockwise, the point (α^p, β^q) lies above the line $\overline{v_{n-2}v_n}$.

Let

$$w_i = (\alpha^{p_{n-2} + ip_{n-1}}, \beta^{q_{n-2} + iq_{n-1}})$$

where $i \in \{0, 1, \dots, a_n\}$ similarly as in the proof of the unrestricted case. There, it was shown that all the triples w_{i-1}, w_i, w_{i+1} are oriented counterclockwise, thus all the points w_i with $i \in \{1, \dots, a_n - 1\}$ lie below the line $\overline{v_{n-2}v_n}$. Thus, if (α^p, β^q) lies above the segment connecting v_{n-2} and v_n , then there is an i such that (α^p, β^q) lies above the segment connecting w_{i-1} and w_i . As in the last two paragraphs of the proof of the unrestricted case, the position of (α^p, β^q) implies the inequality $p - \log_\alpha(\beta)q < p_{n,i-1} - \log_\alpha(\beta)q_{n,i-1}$, and the contradiction follows from part (c) of Lemma 17, as there can be no best upper approximation of $\log_\alpha(\beta)$ which is not a semi-convergent of $\log_\alpha(\beta)$.

References

- 1 Nina Amenta, Jesús A. De Loera, and Pablo Soberón. Helly's theorem: new variations and applications. In *Algebraic and geometric methods in discrete mathematics*, volume 685 of *Contemp. Math.*, pages 55–95. Amer. Math. Soc., Providence, RI, 2017. doi:10.1090/conm/685.

- 2 Gennadiy Averkov, Bernardo González Merino, Ingo Paschke, Matthias Schymura, and Stefan Weltge. Tight bounds on discrete quantitative Helly numbers. *Adv. in Appl. Math.*, 89:76–101, 2017. doi:10.1016/j.aam.2017.04.003.
- 3 David E. Bell. A theorem concerning the integer lattice. *Studies in Appl. Math.*, 56(2):187–188, 1976/77. doi:10.1002/sapm1977562187.
- 4 Michele Conforti and Marco Di Summa. Maximal S -free convex sets and the Helly number. *SIAM J. Discrete Math.*, 30(4):2206–2216, 2016. doi:10.1137/16M1063484.
- 5 Jesús A. De Loera, Reuben N. La Haye, Déborah Oliveros, and Edgardo Roldán-Pensado. Helly numbers of algebraic subsets of \mathbb{R}^d and an extension of Doignon’s theorem. *Adv. Geom.*, 17(4):473–482, 2017. doi:10.1515/advgeom-2017-0028.
- 6 Jesús A. De Loera, Reuben N. La Haye, David Rolnick, and Pablo Soberón. Quantitative Tverberg theorems over lattices and other discrete sets. *Discrete Comput. Geom.*, 58(2):435–448, 2017. doi:10.1007/s00454-016-9858-3.
- 7 Travis Dillon. Discrete quantitative Helly-type theorems with boxes. *Adv. in Appl. Math.*, 129:Paper No. 102217, 17, 2021. doi:10.1016/j.aam.2021.102217.
- 8 Jean-Paul Doignon. Convexity in cristallographical lattices. *J. Geom.*, 3:71–85, 1973. doi:10.1007/BF01949705.
- 9 Alexey Garber. On Helly number for crystals and cut-and-project sets. Arxiv preprint arxiv.org/abs/1605.07881, 2017.
- 10 Jaroslav Hančl and Ondřej Turek. One-sided Diophantine approximations. *Journal of Physics A: Mathematical and Theoretical*, 52(4):045205, January 2019. doi:10.1088/1751-8121/aaf5d3.
- 11 Eduard Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresber. Deutsch. Math.-Verein.*, 32:175–176, 1923.
- 12 Alan J. Hoffman. Binding constraints and Helly numbers. In *Second International Conference on Combinatorial Mathematics (New York, 1978)*, volume 319 of *Ann. New York Acad. Sci.*, pages 284–288. New York Acad. Sci., New York, 1979.
- 13 Andreas Holmsen and Rephael Wenger. Helly-type theorems and geometric transversals. In *Handbook of Discrete and Computational Geometry (3rd ed.)*. CRC Press, 2017.
- 14 Aleksandr Ya. Khinchin. *Continued fractions*. Dover Publications, Inc., Mineola, NY, Russian edition, 1997. With a preface by B. V. Gnedenko, reprint of the 1964 translation.
- 15 Herbert E. Scarf. An observation on the structure of production sets with indivisibilities. *Proc. Nat. Acad. Sci. U.S.A.*, 74(9):3637–3641, 1977. doi:10.1073/pnas.74.9.3637.
- 16 Kevin Barrett Summers. The Helly Number of the Prime-coordinate Point Set. Bachelor’s thesis, University of California, 2015.

Optimal Volume-Sensitive Bounds for Polytope Approximation

Sunil Arya  

Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Hong Kong, China

David M. Mount  

Department of Computer Science and Institute for Advanced Computer Studies, University of Maryland, College Park, MD, USA

Abstract

Approximating convex bodies is a fundamental question in geometry and has a wide variety of applications. Consider a convex body K of diameter Δ in \mathbb{R}^d for fixed d . The objective is to minimize the number of vertices (alternatively, the number of facets) of an approximating polytope for a given Hausdorff error ε . It is known from classical results of Dudley (1974) and Bronshteyn and Ivanov (1976) that $\Theta((\Delta/\varepsilon)^{(d-1)/2})$ vertices (alternatively, facets) are both necessary and sufficient. While this bound is tight in the worst case, that of Euclidean balls, it is far from optimal for skinny convex bodies.

A natural way to characterize a convex object's skinniness is in terms of its relationship to the Euclidean ball. Given a convex body K , define its *volume diameter* Δ_d to be the diameter of a Euclidean ball of the same volume as K , and define its *surface diameter* Δ_{d-1} analogously for surface area. It follows from generalizations of the isoperimetric inequality that $\Delta \geq \Delta_{d-1} \geq \Delta_d$.

Arya, da Fonseca, and Mount (SoCG 2012) demonstrated that the diameter-based bound could be made surface-area sensitive, improving the above bound to $O((\Delta_{d-1}/\varepsilon)^{(d-1)/2})$. In this paper, we strengthen this by proving the existence of an approximation with $O((\Delta_d/\varepsilon)^{(d-1)/2})$ facets.

This improvement is a result of the combination of a number of new ideas. As in prior work, we exploit properties of the original body and its polar dual. In order to obtain a volume-sensitive bound, we explore the following more general problem. Given two convex bodies, one nested within the other, find a low-complexity convex polytope that is sandwiched between them. We show that this problem can be reduced to a covering problem involving a natural intermediate body based on the harmonic mean. Our proof relies on a geometric analysis of a relative notion of fatness involving these bodies.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Approximation algorithms, convexity, Macbeath regions

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.9

Related Version *Full Version*: <https://arxiv.org/abs/2303.09586>

Funding *Sunil Arya*: Research supported by the Research Grants Council of Hong Kong, China under project number 16213219 and 16214721.

Acknowledgements The authors would like to acknowledge the insights and feedback from Rahul Arya and Guilherme da Fonseca.

1 Introduction

Approximating convex bodies by polytopes is a fundamental problem, which has been extensively studied in the literature (see, e.g., Bronstein [11]). We are given a convex body K in Euclidean d -dimensional space and an error parameter $\varepsilon > 0$. The problem is to determine the minimum combinatorial complexity of a polytope that is ε -close to K according to



© Sunil Arya and David M. Mount;

licensed under Creative Commons License CC-BY 4.0

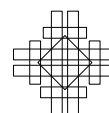
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 9; pp. 9:1–9:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



some measure of similarity. In this paper, we define similarity in terms of the Hausdorff distance [11], and we define combinatorial complexity in terms of the number of facets. Throughout, we assume that the dimension d is a constant.

Approximation bounds presented in the literature are of two common types. In both cases, it is shown that there exists $\varepsilon_0 > 0$ such that the bounds hold for all $\varepsilon \leq \varepsilon_0$. The first of these are *nonuniform bounds*, where the value of ε_0 may depend on properties of K , for example, bounds on its maximum curvature [8, 12, 16, 19, 29, 32]. This is in contrast to *uniform bounds*, where the value of ε_0 is independent of K (but may depend on d).

Examples of uniform bounds include the classical work of Dudley [13] and Bronshteyn and Ivanov [10]. Dudley showed that, for $\varepsilon \leq 1$, any convex body K can be ε -approximated by a polytope P with $O((\Delta/\varepsilon)^{(d-1)/2})$ facets, where Δ is K 's diameter. Bronshteyn and Ivanov showed the same bound holds for the number of vertices. Constants hidden in the O -notation depend only on d . These results have numerous applications in computational geometry, for example the construction of coresets [1, 3, 5].

The approximation bounds of both Dudley and Bronshteyn-Ivanov are tight in the worst case up to constant factors (specifically when K is a Euclidean ball) [11]. However, these bounds may be significantly suboptimal if K is “skinny”. A natural way to characterize a convex object’s skinniness is in terms of its relationship to the Euclidean ball. Given a convex body K , define its *volume diameter* Δ_d to be the diameter of a Euclidean ball of the same volume as K , and define its *surface diameter* Δ_{d-1} analogously for surface area. These quantities are closely related (up to constant factors) to the classical concepts of *quermassintegrals* and of *intrinsic volumes* of the convex body [20, 21]. It follows from generalizations of the isoperimetric inequality that $\Delta \geq \Delta_{d-1} \geq \Delta_d$ [21].

Arya, da Fonseca, and Mount [4] proved that the diameter-based bound could be made surface-area sensitive, improving the above bound to $O((\Delta_{d-1}/\varepsilon)^{(d-1)/2})$. In this paper, we strengthen this to the following volume-sensitive bound.

► **Theorem 1.1.** *Consider real d -space, \mathbb{R}^d . There exists a constant c_d (depending on d) such that for any convex body $K \subseteq \mathbb{R}^d$ and any $\varepsilon > 0$, if the width of K in any direction is at least ε , then there exists an ε -approximating polytope P whose number of facets is at most*

$$\left(\frac{c_d \Delta_d}{\varepsilon} \right)^{\frac{d-1}{2}}.$$

This bound is the strongest to date. For example, in \mathbb{R}^3 , the area-sensitive bound yields better bounds for pencil-like objects that are thin along two dimensions, while the volume-sensitive bound yields better bounds for pancake-like objects as well, which are thin in just one dimension.

The minimum-width assumption seems to be a technical necessity, since it is not difficult to construct counterexamples where this condition does not hold. But this is not a fundamental impediment. If the body’s width is less than ε in some direction, then by projecting the body onto a hyperplane orthogonal to this direction, it is possible to reduce the problem to a convex approximation problem in one lower dimension. This can be repeated until the body’s width is sufficiently large in all remaining dimensions, and the stated bound can be applied in this lower dimensional subspace, albeit with volume measured appropriate to this dimension.

While our uniform bound trivially holds in the nonuniform setting, we present a separate (and much shorter) proof that the same bounds hold in the nonuniform setting, assuming that K 's boundary is C^2 continuous. This is presented in the full version.

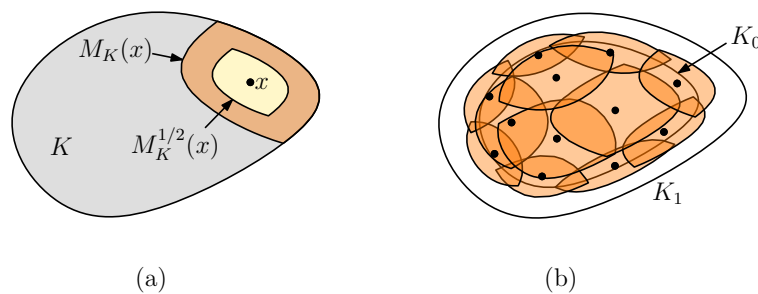
► **Theorem 1.2.** Consider real d -space, \mathbb{R}^d . There exists a constant c_d (depending on d) such that for any convex body $K \subseteq \mathbb{R}^d$ of C^2 boundary, as ε approaches zero, there exists an ε -approximating polytope P whose number of facets is at most

$$\left(\frac{c_d \Delta_d}{\varepsilon} \right)^{\frac{d-1}{2}}.$$

2 Overview of Techniques

Broadly speaking, the problem of approximating a convex body by a polytope involves “sandwiching” a polytope between two nested convex bodies, call them K_0 and K_1 . For example, K_0 may be the original body to be approximated and K_1 is an expansion based on the allowed error bound. Most of the prior work in this area has focused on the specific manner in which K_1 is defined relative to K_0 , which is typically confined to Euclidean space (for Hausdorff distance) or affine space (for the Banach-Mazur distance).

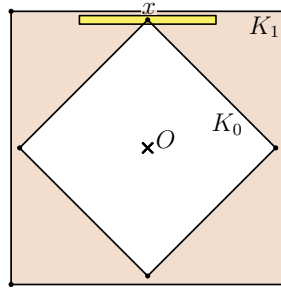
Recent approaches to convex approximation have been based on covering the body to be approximated with convex objects that respect the local shape of the body being approximated [2, 6]. Macbeath regions have been a key tool in this regard. Given a convex body K and a point x in K 's interior, the Macbeath region at x , $M_K(x)$, is the largest centrally symmetric body nested within K and centered at x (see Figure 1(a)). A Macbeath region that has been shrunk by some constant factor λ is denoted by $M_K^\lambda(x)$. Shrunk Macbeath regions have nice packing and covering properties, and they behave much like metric balls.



■ **Figure 1** (a) Macbeath regions and (b) covering K_0 by Macbeath regions.

A natural way to construct a sandwiching polytope between two nested bodies K_0 and K_1 is to construct a collection of shrunk Macbeath regions that cover K_0 but lie entirely within K_1 (see Figure 1(b)). If done properly, a sandwiching polytope can be constructed by sampling a constant number of points from each of these Macbeath regions, and taking the convex hull of their union. Thus, the number of Macbeath regions provides an upper bound on the number of vertices in the sandwiched polytope.

The “sandwiching” perspective described above yields additional new challenges. Consider the two bodies K_0 and K_1 shown in Figure 2, where K_0 is a diamond shape nested within the square K_1 . Consider 1/2-scaled Macbeath region centered at a point x that lies at the top vertex of K_0 . Observe that almost all of its volume lies outside of K_0 . This is problematic because our analysis is based on the number of Macbeath regions needed to cover the boundary of a body, in this case ∂K_0 . We want a significant amount of the volume of each Macbeath region to lie within K_0 . In cases like that shown in Figure 2, only a tiny fraction of the volume can be charged in this manner against K_0 .



■ **Figure 2** Relative fatness.

Intuitively, while the body K_0 is “fat” in a standard sense¹, it is not fat “relative” to the enclosing body K_1 . To deal with this inconvenience, we will replace K_1 with an intermediate body between K_0 and K_1 that satisfies this property. In Section 3.5 we formally define this notion of relative fatness, and we present an intermediate body, called the *harmonic-mean body*, that satisfies this notion of fatness. We will see that this body can be used as a proxy for the sake of approximation.

3 Preliminaries

In this section, we introduce terminology and notation, which will be used throughout the paper. This section can be skipped on first reading (moving directly to Section 4).

Let us first recall some standard notation. Given vectors $u, v \in \mathbb{R}^d$, let $\langle u, v \rangle$ denote their dot product, and let $\|v\| = \sqrt{\langle v, v \rangle}$ denote v ’s Euclidean length. Throughout, we will use the terms *point* and *vector* interchangeably. Given points $p, q \in \mathbb{R}^d$, let $\|pq\| = \|p - q\|$ denote the Euclidean distance between them. Let $\text{vol}(\cdot)$ and $\text{area}(\cdot)$ denote the d -dimensional and $(d - 1)$ -dimensional Lebesgue measures, respectively.

3.1 Polarity and Centrality Properties

Given a bounded convex body $K \subseteq \mathbb{R}^d$ that contains the origin O in its interior, define its *polar*, denoted K^* , to be the convex set

$$K^* = \{u : \langle u, v \rangle \leq 1, \text{ for all } v \in K\}.$$

The polar enjoys many useful properties (see, e.g., Eggleston [14]). For example, it is well known that K^* is bounded and $(K^*)^* = K$. Further, if K_1 and K_2 are two convex bodies both containing the origin such that $K_1 \subseteq K_2$, then $K_2^* \subseteq K_1^*$.

Given a nonzero vector $v \in \mathbb{R}^d$, we define its “polar” v^* to be the hyperplane that is orthogonal to v and at distance $1/\|v\|$ from the origin, on the same side of the origin as v . The polar of a hyperplane is defined as the inverse of this mapping. We may equivalently define K^* as the intersection of the closed halfspaces that contain the origin, bounded by the hyperplanes v^* , for all $v \in K$.

Given a convex body $K \subseteq \mathbb{R}^d$ and $x \in \text{int}(K)$, there are many ways to characterize the property that x is “central” within K [17, 31]. For our purposes, we will make it precise using the concept of Mahler volume. Define K ’s *Mahler volume*, denoted $\mu(K)$, to be the

¹ That is, the largest ball enclosed in K_0 and the smallest ball containing K_0 differ in size by a constant.

product $\text{vol}(K) \cdot \text{vol}(K^*)$. The Mahler volume is well studied (see, e.g. [28, 22, 30]). It is invariant under linear transformations, and it depends on the location of the origin within K . We say that K is *well-centered* with respect to a point $x \in \text{int}(K)$ if the Mahler volume $\mu(K - x)$ is at most $O(1)$. When x is not specified, it is understood to be the origin. We have the following lemma [6, 23].

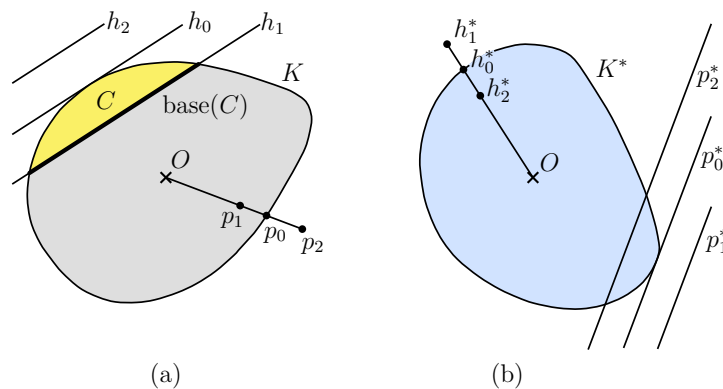
► **Lemma 3.1.** *Any convex body K is well-centered with respect to its centroid.*

Lower bounds on the Mahler volume have also been extensively studied and it is known that the following bound holds irrespective of the location of the origin [9, 18, 25].

► **Lemma 3.2.** *Given a convex body $K \subseteq \mathbb{R}^d$ whose interior contains the origin, $\mu(K) = \Omega(1)$.*

3.2 Caps, Rays, and Relative Measures

Consider a compact convex body K in d -dimensional space \mathbb{R}^d with the origin O in its interior. A *cap* C of K is defined to be the nonempty intersection of K with a halfspace. Letting h_1 denote a hyperplane that does not pass through the origin, let $\text{cap}_K(h_1)$ denote the cap resulting by intersecting K with the halfspace bounded by h_1 that does not contain the origin (see Figure 3(a)). Define the *base* of C , denoted $\text{base}(C)$, to be $h_1 \cap K$. Letting h_0 denote a supporting hyperplane for K and C parallel to h_1 , define an *apex* of C to be any point of $h_0 \cap K$.



■ **Figure 3** Convex body K and polar K^* with definitions used for width and ray.

We define the *absolute width* of cap C to be $\text{dist}(h_1, h_0)$. When a cap does not contain the origin, it will be convenient to define distances in relative terms. Define the *relative width* of such a cap C , denoted $\text{wid}_K(C)$, to be the ratio $\text{dist}(h_1, h_0) / \text{dist}(O, h_0)$ and, to simplify notation, define $\text{wid}_K(h_1) = \text{wid}_K(\text{cap}_K(h_1))$. Observe that as a hyperplane is translated from a supporting hyperplane to the origin, the relative width of its cap ranges from 0 to a limiting value of 1.

We also characterize the closeness of a point to the boundary in both absolute and relative terms. Given a point $p_1 \in K$, let p_0 denote the point of intersection of the ray Op_1 with the boundary of K . Define the *absolute ray distance* of p_1 to be $\|p_1 p_0\|$, and define the *relative ray distance* of p_1 , denoted $\text{ray}_K(p_1)$, to be the ratio $\|p_1 p_0\| / \|Op_0\|$. Relative widths and relative ray distances are both affine invariants, and unless otherwise specified, references to widths and ray distances will be understood to be in the relative sense.

We can also define volumes in a manner that is affine invariant. Recall that $\text{vol}(\cdot)$ denotes the standard Lebesgue volume measure. For any region $\Lambda \subseteq K$, define the *relative volume* of Λ with respect to K , denoted $\text{vol}_K(\Lambda)$, to be $\text{vol}(\Lambda) / \text{vol}(K)$.

With the aid of the polar transformation we can extend the concepts of width and ray distance to objects lying outside of K . Consider a hyperplane h_2 parallel to h_1 that lies beyond the supporting hyperplane h_0 (see Figure 3(a)). It follows that $h_2^* \in K^*$, and we define $\text{wid}_K(h_2) = \text{ray}_{K^*}(h_2^*)$ (see Figure 3(b)). Similarly, for a point $p_2 \notin K$ that lies along the ray Op_1 , it follows that the hyperplane p_2^* intersects K^* , and we define $\text{ray}_K(p_2) = \text{wid}_{K^*}(p_2^*)$. By properties of the polar transformation, it is easy to see that $\text{wid}_K(h_2) = \text{dist}(h_0, h_2) / \text{dist}(O, h_2)$. Similarly, $\text{ray}_K(p_2) = \|p_0 p_2\| / \|Op_2\|$. Henceforth, we will omit references to K when it is clear from context.

Some of our results apply only when we are sufficiently close to the boundary of K . Given $\alpha \leq \frac{1}{2}$, we say that a cap C is α -shallow if $\text{wid}(C) \leq \alpha$, and we say that a point p is α -shallow if $\text{ray}(p) \leq \alpha$. We will simply say *shallow* to mean α -shallow, where $\alpha \leq \frac{1}{2}$ is a sufficiently small constant.

3.3 Macbeath Regions and MNetS

Given a convex body K and a point $x \in K$, and a scaling factor $\lambda > 0$, the *Macbeath region* $M_K^\lambda(x)$ is defined as

$$M_K^\lambda(x) = x + \lambda((K - x) \cap (x - K)).$$

It is easy to see that $M_K^1(x)$ is the intersection of K with the reflection of K around x , and so $M_K^1(x)$ is centrally symmetric about x . Indeed, it is the largest centrally symmetric body centered at x and contained in K . Furthermore, $M_K^\lambda(x)$ is a copy of $M_K^1(x)$ scaled by the factor λ about the center x (see Figure 1(a)). We will omit the subscript K when the convex body is clear from the context. As a convenience, we define $M(x) = M^1(x)$.

The following lemma states that points in a shrunken Macbeath region all have similar ray distances. The proof appears in [7, Section 2.5].

► **Lemma 3.3.** *Let K be a convex body. If x is a $\frac{1}{2}$ -shallow point in K and $y \in M^{1/5}(x)$, then $\text{ray}(x)/2 \leq \text{ray}(y) \leq 2\text{ray}(x)$.*

The next lemma shows that translated copies of a Macbeath region act as proxies for Macbeath regions in the vicinity. The proof appears in Section 3.3 of the full version.

► **Lemma 3.4.** *Let $\lambda \leq 1/2$ and $\gamma \leq 1/10$. Let x be a point in a convex body K . Let $R = M(x) - x$. Let y be a point in $x + \lambda R$. Then $y + \gamma R \subseteq M^{2^\gamma}(y)$.*

We employ Macbeath region-based coverings in our polytope approximation scheme. In particular, we employ the concept of MNetS, as defined in [6]. Let $K \subseteq \mathbb{R}^d$ be a convex body, let Λ be an arbitrary subset of $\text{int}(K)$, and let $c \geq 5$ be any constant. Given $X \subseteq K$, define $\mathcal{M}_K^\lambda(X) = \{M_K^\lambda(x) : x \in X\}$. Define a (K, Λ, c) -MNet to be any maximal set of points $X \subseteq \Lambda$ such that the shrunken Macbeath regions $\mathcal{M}_K^{1/4c}(X)$ are pairwise disjoint. We refer to c as the expansion factor of the MNet. The following lemma, proved in [6], summarizes the key properties of MNetS.

► **Lemma 3.5 ([6]).** *Given a convex body $K \subseteq \mathbb{R}^d$, $\Lambda \subset \text{int}(K)$, and $c \geq 5$, a (K, Λ, c) -MNet X satisfies the following properties:*

- (Packing) *The elements of $\mathcal{M}_K^{1/4c}(X)$ are pairwise disjoint.*
- (Covering) *The union of $\mathcal{M}_K^{1/c}(X)$ covers Λ .*
- (Buffering) *The union of $\mathcal{M}_K(X)$ is contained within K .*

For the purposes of this paper, c will be any sufficiently large constant, specifically $c \geq 5$. To simplify notation, we use (K, Λ) -MNet to refer to such an MNet.

As mentioned before, we reduce our polytope approximation problem to that of finding a polytope which is sandwiched between two convex bodies. In turn we tackle this problem using M Nets as indicated in the next lemma. The proof appears in Section 3.3 of the full version.

► **Lemma 3.6.** *Let $K_0 \subset K_1$ be two convex bodies. Let X be a $(K_1, \partial K_0)$ -MNet. Then there exists a polytope P with $O(|X|)$ vertices such that $K_0 \subseteq P \subseteq K_1$.*

The following lemma bounds the sizes of M Nets in important special cases involving points at roughly the same ray distance. These bounds will be useful in obtaining our volume-sensitive bounds. The proof appears in Section 4 of the full version.

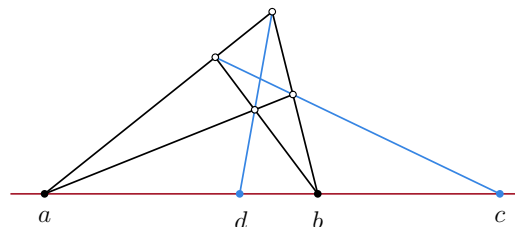
► **Lemma 3.7.** *Let $0 < \varepsilon \leq 1/2$ be sufficiently small and let $K \subseteq \mathbb{R}^d$ be a well-centered convex body. Let Λ be the points of K at ray distances between ε and 2ε , and let X be a (K, Λ) -MNet. Then:*

- (i) $|X| = O(1/\varepsilon^{(d-1)/2})$.
- (ii) *For any positive real $f \leq 1$, let $X_f \subseteq K$ be such that the total relative volume of the Macbeath regions of $\mathcal{M}^{1/4c}(X_f)$ is $O(f\varepsilon)$. Then $|X_f| = O(\sqrt{f}/\varepsilon^{(d-1)/2})$.*

3.4 Concepts from Projective Geometry

In this section we present some relevant standard concepts from projective geometry. For further details see any standard reference (e.g., [27]). Given four collinear points, a, b, c, d (not necessarily in this order), the *cross ratio* $(a, b; c, d)$ is defined to be $(\|ac\|/\|ad\|)/(\|bc\|/\|bd\|)$, where these are understood to be signed distances determined by the orientations of the segments along the line. We follow the convention of using symbols a, b, c, d, \dots for points, and the distinction from other uses (such as d for the dimension) should be clear from the context.

It is well known that cross ratios are preserved under projective transformations. If the cross ratio $(a, b; c, d)$ is -1 , we say that this quadruple of points forms a *harmonic bundle* (see Figure 4). This is an important special case which occurs frequently in constructions. In this case, the points lie on the line in the order of a, d, b, c and the ratio in which a divides c and d externally (i.e., $\|ac\|/\|ad\|$) is the same as the ratio in which b divides c and d internally (i.e., $\|bc\|/\|bd\|$). The sign is negative since bc and bd have opposite directions. If the point a is at infinity, the cross ratio degenerates to $\|bd\|/\|bc\|$, implying that b is midway between c and d .

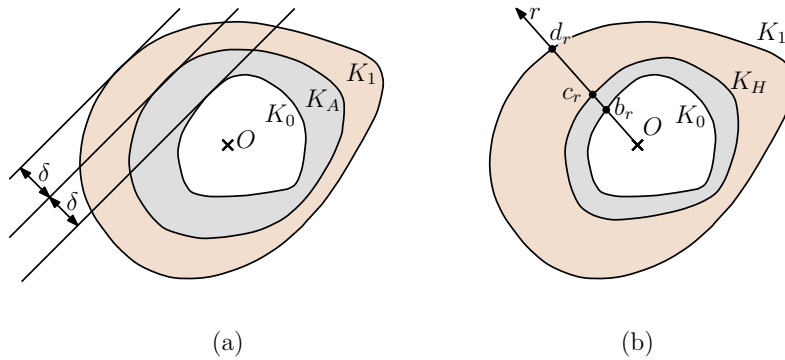


■ **Figure 4** Harmonic bundle (from the quadrilateral construction [27]).

3.5 Intermediate Bodies

In this section we explore the concept of relative fatness, which was introduced in Section 2. Given two convex bodies K_0 and K_1 such that $K_0 \subset K_1$ and $0 < \gamma < 1$, we say that K_0 is *relatively γ -fat* with respect to K_1 if, for any point $p \in \partial K_0$, and any scaling factor $0 < \lambda \leq 1$, at least a constant fraction γ of the volume of the Macbeath region $M = M_{K_1}^\lambda(p)$ lies within K_0 , that is, $\text{vol}(M \cap K_0) / \text{vol}(M) \geq \gamma$. We say that K_0 is *relatively fat* with respect to K_1 if it is relatively γ -fat for some constant γ . Relative fatness will play an important role in our analyses. Since an arbitrary nested pair $K_0 \subset K_1$ may not necessarily satisfy this property, it will be useful to define an intermediate body sandwiched between K_0 and K_1 that does.

There are a few natural ways to define such an intermediate body. Given two convex bodies K_0 and K_1 , where $K_0 \subseteq K_1$, the *arithmetic-mean body*, $K_A(K_0, K_1)$, is defined to be the convex body $\frac{1}{2}(K_0 \oplus K_1)$, where “ \oplus ” denotes Minkowski sum. Equivalently, for any unit vector u consider the two supporting halfspaces of K_0 and K_1 orthogonal to u , and take the halfspace that is midway between the two. The arithmetic-mean body is obtained by intersecting such halfspaces for all unit vectors u .



■ **Figure 5** (a) Arithmetic and (b) harmonic-mean bodies.

Another natural choice arises from a polar viewpoint. Assume that $K_0 \subset K_1$ and the origin $O \in \text{int}(K_0)$. The *harmonic-mean body*, $K_H(K_0, K_1)$, was introduced by Firey [15] and is defined as follows. For any ray r from the origin O , let b_r and d_r denote the points of intersection of r with ∂K_0 and ∂K_1 , respectively (see Figure 5(b)). Let c_r be the point on the ray such that $1/\|Oc_r\| = (1/\|Ob_r\| + 1/\|Od_r\|)/2$. Equivalently, the cross ratio $(O, c_r; d_r, b_r)$ equals -1 , that is, this quadruple forms a harmonic bundle. Clearly, c_r lies between b_r and d_r , and hence the union of these points over all rays r defines the boundary of a body that is sandwiched between K_0 and K_1 . This body is the harmonic-mean body. By considering the supporting hyperplanes orthogonal to the ray r , it is easy to see that the arithmetic-mean body of K_0 and K_1 is mapped to the harmonic-mean body of K_0^* and K_1^* under polarity, that is, $(K_A(K_0, K_1))^* = K_H(K_0^*, K_1^*)$. Therefore, $K_H(K_0, K_1)$ is convex. When K_0 and K_1 are clear from context, we will just write K_A and K_H , omitting references to their arguments.

In order to understand why these intermediate bodies are useful to us, recall the diamond and square bodies K_0 and K_1 from Figure 2 (see Figure 6(a)). Recall the issue that a large fraction of the volume of the Macbeath region $M_{K_1}^{1/2}(x)$ lies outside of K_0 . If we replace K_1 with $K_H = K_H(K_0, K_1)$ and compute the Macbeath region with respect to K_H instead (see Figure 6(b) and (c)), we see that a constant fraction of the volume of the Macbeath region lies within K_0 and so relative fatness is satisfied.

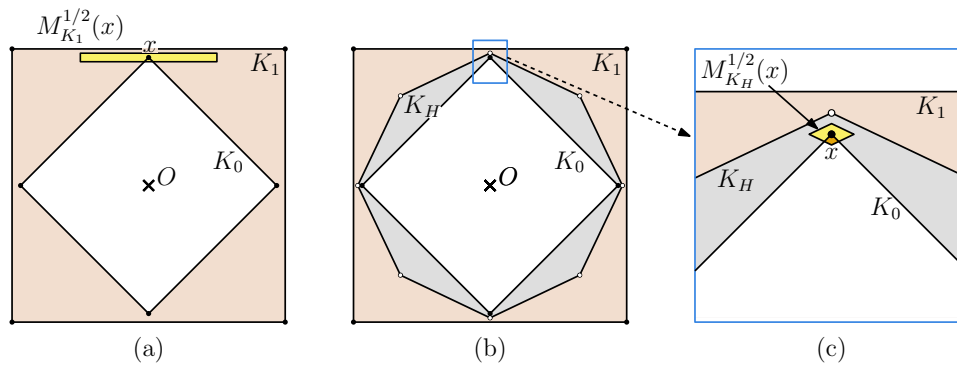


Figure 6 Relative fatness of K_H .

In Section 4, we will present an important result by showing that the inner body K_0 is relatively fat with respect to the harmonic-mean body $K_H(K_0, K_1)$. The proof makes heavy use of concepts from projective geometry, such as the harmonic bundle. This fact will be critical to establishing the volume-sensitive bounds given in this paper.

4 Relative Fatness and the Harmonic-Mean Body

In this section, we establish properties of the harmonic-mean body that are critical to the main results of this paper. In particular, given two bodies $K_0 \subset K_1$, we show that K_0 is relatively fat with respect to K_H . In fact, we present a stronger result in Lemma 4.4, which implies relative fatness as an immediate consequence. We will employ this stronger result in Section 5 to obtain our volume-sensitive bounds for polytope approximation.

The proof of Lemma 4.4 is based on the following technical lemma. For constant λ , it implies that for any point $b \in K_0$ that is not too close to the boundary of K_0 , the Macbeath regions centered at b with respect to K_0 and K_H , respectively, are roughly similar up to a constant scaling factor. This is formally stated in the corollary following the lemma.

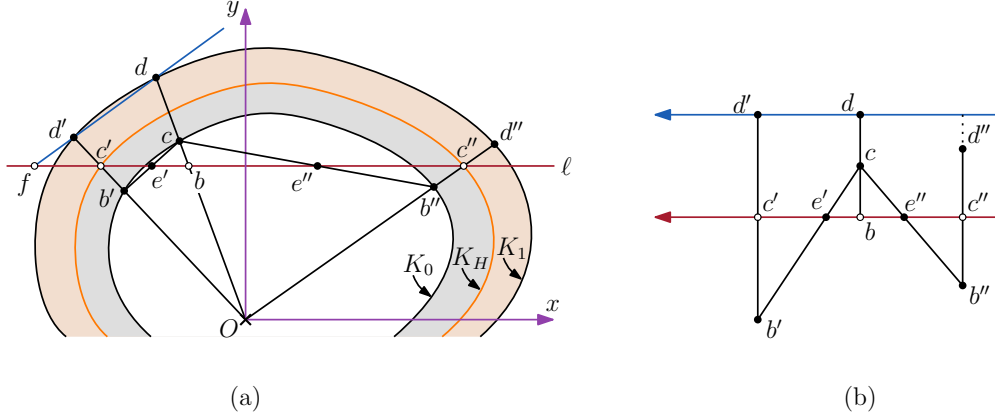
► **Lemma 4.1.** *Let $0 < \lambda < 1$ be a parameter. Let $K_0 \subset K_1$ be two convex bodies, where the origin O lies in the interior of K_0 . Let K_H denote the harmonic-mean body of K_0 and K_1 . Consider any ray emanating from the origin O . Let c and d denote the points of intersection of this ray with ∂K_0 and ∂K_1 , respectively (see figure). Let $b \in K_0$ be a point on this ray such that the cross ratio $(O, c; d, b) \leq -\lambda$. Consider any line passing through b . Let c' and c'' denote the points of intersection of this line with ∂K_H . Then*

$$\min(\|bc' \cap K_0\|, \|bc'' \cap K_0\|) \geq s(\lambda) \cdot \min(\|bc'\|, \|bc''\|), \quad \text{where } s(\lambda) = \lambda/6.$$

Proof. We sketch the key ideas and present a complete proof in the full version. Consider the two dimensional flat that contains the origin and the line ℓ that passes through the points c' , b , and c'' . Henceforth, let K_0, K_1, K_H refer to the two dimensional convex bodies obtained by intersecting the respective bodies with this flat. Let b' and d' denote the points of intersection of the ray Oc' with ∂K_0 and ∂K_1 , respectively, and define b'' and d'' analogously for Oc'' . All these points lie on the flat, and it follows from the definition of the harmonic-mean body that $(O, c'; d', b') = (O, c''; d'', b'') = -1$ (see Figure 7(a)).

By rotating space, we may assume that ℓ is horizontal and above the origin. Through an infinitesimal perturbation, we may assume that there is a supporting line for K_1 at d that is not parallel to ℓ . Without loss of generality, we may assume that it intersects ℓ to the left of

9:10 Optimal Volume-Sensitive Bounds for Polytope Approximation



■ **Figure 7** Lemma 4.1 and its proof.

b. Since c' and c'' are symmetrical in the statement of the lemma, we may assume that c' lies to left of b and c'' lies to its right. Let f denote the intersection point of the line dd' with ℓ (see Figure 7(a)). Clearly, the left-to-right order of points along ℓ is $\langle f, c', b, c'' \rangle$. Observe that the points $c, d, d',$ and d'' all lie strictly above ℓ , and the points b' and b'' lie strictly below.

Let e' denote the point of intersection of the segment cb' with segment bc' , and define e'' analogously for segment cb'' . Since c, b' and b'' all lie on ∂K_0 , by convexity, e' and e'' are contained in K_0 . Thus, to prove the lemma, it suffices to show that

$$\min(\|be'\|, \|be''\|) \geq s(\lambda) \cdot \min(\|bc'\|, \|bc''\|). \tag{1}$$

We begin by proving bounds on two cross ratios:

- (i) $-(f, e'; c', b) \geq \lambda/2$, and
- (ii) $-(f, e''; c'', b) \geq \lambda/2$.

Because projective transformations preserve cross ratios, it will be convenient to prove these bounds after first applying a projective transformation. In particular, this transformation maps O and f to infinity so that lines through O map to vertical lines and lines through f map to horizontal lines (see Figure 7(b)). After this transformation, $Oc', Oc,$ and Oc'' are vertical and directed upwards and $d'd$ and $c'b$ are horizontal and directed to the right. Clearly, $\|c'd'\| = \|bd\|$. Since d'' lies above ℓ and below the line $d'd$ we have $\|c''d''\| \leq \|bd\|$. By definition of b , we have $(O, c; d, b) = -1/(\|cd\|/\|cb\|) \leq -\lambda$. Since $\|cb\| + \|cd\| = \|bd\|$, we have $\|cb\| \geq \|bd\|\lambda/(1 + \lambda)$.

Given that f is at infinity, the above cross ratios reduce to simple ratios. Thus, it suffices to show:

- (i) $\|e'b\|/\|e'c'\| \geq \lambda/2$, and
- (ii) $\|e''b\|/\|e''c''\| \geq \lambda/2$.

To show (i), observe that since $(O, c'; d', b) = -1$ and since O is at infinity and c' lies between b' and d' , this is equivalent to $1/(\|c'd'\|/\|c'b'\|) = 1$, that is, $\|c'b'\| = \|c'd'\|$. By similar triangles $\triangle e'bc$ and $\triangle e'c'b'$, the fact that $\|c'b'\| = \|c'd'\| = \|bd\|$, and our bounds on λ , we have

$$\frac{\|e'c'\|}{\|e'b\|} = \frac{\|c'b'\|}{\|cb\|} \leq \frac{\|bd\|}{\|bd\|\lambda/(1 + \lambda)} = \frac{1 + \lambda}{\lambda} \leq \frac{2}{\lambda}, \tag{2}$$

which implies (i).

The analysis for (ii) is essentially the same as above. Since $(O, c'', d'', b'') = -1$ we have $\|c''b''\| = \|c''d''\|$. By similar triangles $\triangle e''bc$ and $\triangle e''c''b''$ and the fact that $\|c''b''\| = \|c''d''\| \leq \|bd\|$, the inequalities of Eq. (2) (with double primes for single primes) show that

$$\frac{\|e''c''\|}{\|e''b\|} \leq \frac{2}{\lambda},$$

which implies (ii).

These inequalities hold only in transformed configuration, but the cross ratios of (i) and (ii) hold unconditionally. Returning to the original configuration and using (i), we can show that $\|be'\|/\|bc'\| \geq \lambda/3$ and from (ii), we can show that either $\|be''\|/\|bf\| \geq \lambda/6$ or $\|be''\|/\|e''c''\| \geq \lambda/5$. We omit the details of this calculation, which can be found in the full version. In both cases, we are able to establish Eq. (1), as desired. ◀

The following corollary is immediate from the definition of Macbeath regions.

▶ **Corollary 4.2.** *Assume all entities to be as defined in the statement of Lemma 4.1. Then $M_{K_H}^{s(\lambda)}(b) \subseteq M_{K_0}(b)$, where $s(\lambda) = \lambda/6$.*

We have the following lemma which in conjunction with Corollary 4.2 will be useful in proving Lemma 4.4. The proof is presented in the full version.

▶ **Lemma 4.3.** *Let λ, K_0, K_1, K_H , the origin O , and points c and d be as in Lemma 4.1. Let h denote the point of intersection of the ray Oc with the boundary of K_H . Then:*

- (i) $\|Oc\| \geq \|hc\|$.
- (ii) *Let b be a point on segment Oc , which is not contained in the interior of $M_{K_H}^\lambda(c)$. Then $(O, c; d, b) \leq -\lambda/2$.*

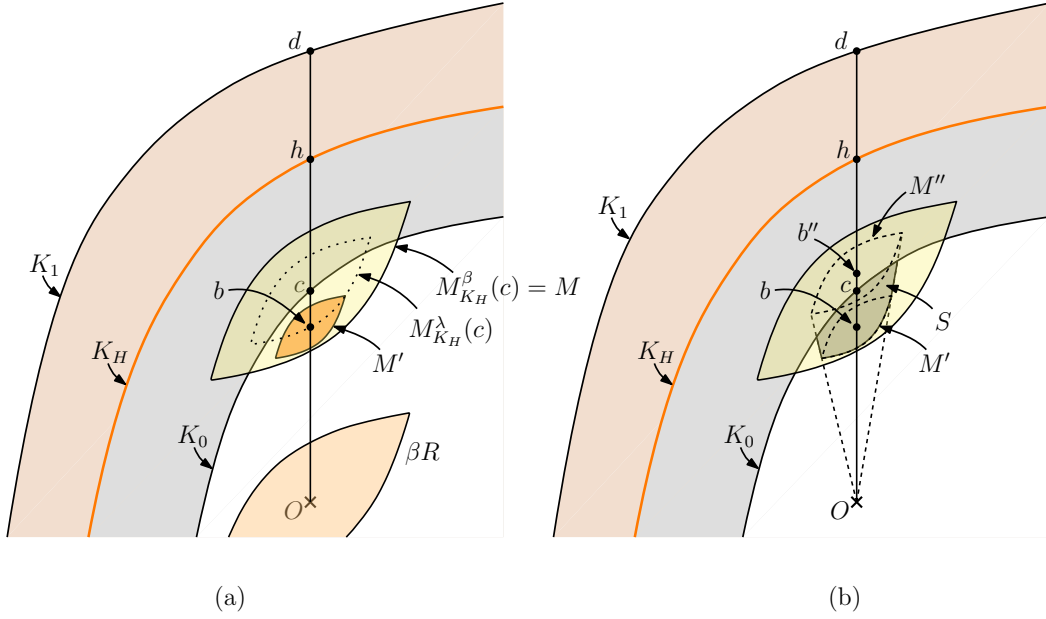
We now have all the key ingredients to present the main result of this section. The relative fatness of K_0 with respect to K_H is an immediate consequence of parts (i) and (ii) of this lemma. In order to state part (iii), we need a definition. Given a convex body K with the origin O in its interior and a region $R \subseteq K$, define the *shadow* of R with respect to K , denoted $\text{shadow}_K(R)$, to be the set of points $x \in K$ such that the segment Ox intersects R .

▶ **Lemma 4.4.** *Let $0 < \beta \leq 1$ be a real parameter. Let $K_0 \subset K_1$ be two convex bodies, let the origin O lie in the interior of K_0 , and let K_H denote the harmonic-mean body of K_0 and K_1 . Let c be any point on the boundary of K_0 and let $M = M_{K_H}^\beta(c)$. Then there exists a convex body M' such that*

- (i) $\text{vol}(M') = \Omega(\text{vol}(M))$,
- (ii) $M' \subseteq M \cap K_0$, and
- (iii) $\text{shadow}_{K_0}(M') \subseteq M$.

Proof. We sketch the proof of (i) and (ii) here. A complete proof appears in the full version. For the sake of convenience, assume that the ray Oc is directed vertically upwards. Let h be the point of intersection of the ray Oc with ∂K_H . Let $R = M_{K_H}(c) - c$ be the recentering of $M_{K_H}(c)$ about the origin. By definition, $M = M_{K_H}^\beta(c) = c + \beta R$. Let b be the point of intersection of the segment Oc with the boundary of $M_{K_H}^\lambda(c) = c + \lambda R$, where $\lambda = \beta/\kappa$ for a suitable large constant $\kappa \geq 2$ (independent of dimension). Recalling from Lemma 4.3(a) that $\|ch\| \leq \|Oc\|$, it follows that b is vertically below c at a distance of $\lambda\|ch\|$. Recalling $s(\lambda)$ from Corollary 4.2, let $M' = b + \gamma R$ for

$$\gamma = \frac{s(\lambda/2)}{10} = \frac{s(\beta/2\kappa)}{10} = \frac{\beta}{120\kappa}$$



■ **Figure 8** Proof of Lemma 4.4. (Objects are not drawn to scale.)

(see Figure 8(a)). Since M' and M are translated copies of R scaled by a factor of γ and β , respectively, we have $\text{vol}(M') = (\gamma/\beta)^d \text{vol}(M) = (1/120\kappa)^d \text{vol}(M)$. This proves (i).

To prove (ii), we will show that $M' \subseteq M$ and $M' \subseteq K_0$. Since $b \in c + \lambda R$ and $M' = b + \gamma R$, it follows that $M' \subseteq c + (\lambda + \gamma)R$. For large κ , we have $\lambda + \gamma \leq \beta$, and thus $M' \subseteq c + \beta R = M$.

Next we show that $M' \subseteq K_0$. Let d denote the point of intersection of the ray Oc with ∂K_1 . Applying Lemma 4.3(b), it follows that the cross ratio $(O, c; d, b) \leq -\lambda/2$. Applying Corollary 4.2 with $\lambda/2$ in place of λ and recalling that $s(\lambda/2) = 10\gamma$, we have $M_{KH}^{10\gamma}(b) \subseteq M_{K_0}(b)$. Also, by Lemma 3.4, we have $M' = b + \gamma R \subseteq M_{KH}^{2\gamma}(b)$. Thus $M' \subseteq M_{K_0}^{1/5}(b)$. By definition of Macbeath regions, $M_{K_0}(b) \subseteq K_0$, and so $M' \subseteq K_0$, as desired. ◀

The following corollary is immediate from parts (i) and (ii) of the above lemma.

► **Corollary 4.5.** *Let $K_0 \subset K_1$ be two convex bodies, let the origin O lie in the interior of K_0 , and let K_H denote the harmonic-mean body of K_0 and K_1 . Then K_0 is relatively fat with respect to K_H .*

5 Uniform Volume-Sensitive Bounds

In this section, we present the proof of Theorem 1.1. Let $\varepsilon > 0$ and let K_0 denote the convex body K described in this theorem. Let $K_1 = K_0 \oplus \varepsilon$ denote the Minkowski sum of K_0 with a ball of radius ε . Also recall that $\Delta_d(K_0)$ denotes the *volume diameter* of K_0 . Let $C(K_0, \varepsilon)$ be a shorthand for $(\Delta_d(K_0)/\varepsilon)^{(d-1)/2}$, the desired number of facets.

We will show that there exists a polytope with $O(C(K_0, \varepsilon))$ facets sandwiched between K_0 and K_1 . As mentioned above, we will transform the problem by mapping to the polar. Through an appropriate translation, we may assume that the origin O coincides with the centroid of K_0 . Note that the arithmetic-mean body K_A of K_0 and K_1 is given by $K_0 \oplus \frac{\varepsilon}{2}$, and recall from Section 3.5 that $K_H = K_A^*$ is the harmonic-mean body of K_1^* and K_0^* .

Our construction is based on Lemma 5.1, which shows that there is a $(K_H, \partial K_1^*)$ -MNet X of size $O(C(K_0, \varepsilon))$. Applying Lemma 3.6, it follows that there exists a polytope P sandwiched between K_1^* and K_H with $O(|X|)$ vertices. By polarity, this implies that P^* is a polytope sandwiched between K_A and K_1 having $O(|X|)$ facets. Since $K_0 \subseteq K_A$, this polytope is also sandwiched between K_0 and K_1 , which proves Theorem 1.1.

All that remains is showing that $|X| = O(C(K_0, \varepsilon))$. For this purpose, we will utilize the tools for bounding the sizes of MNet in conjunction with the relative fatness of the harmonic-mean body (established in Section 4).

► **Lemma 5.1.** *Let $\varepsilon > 0$ and let K_0, K_1, K_A, K_H be convex bodies as defined above. Let X be a $(K_H, \partial K_1^*)$ -MNet. Then $|X| = O(C(K_0, \varepsilon))$.*

Proof. We begin by showing that $\text{vol}(K_H) = \Omega(1/\text{vol}(K_0))$, and its Mahler volume $\mu(K_H)$ is at most $O(1)$ (implying that K_H is well-centered). To see this, recall that the width of K_0 in any direction is at least ε and $K_A = K_0 \oplus \frac{\varepsilon}{2}$. It is well-known that the ratio of the distances of the centroid from any pair of supporting hyperplanes is at most d [17, 24, 26]. It follows that a ball of radius $\varepsilon/(d+1)$ centered at the origin lies within K_0 . Thus, a constant-factor expansion of K_0 contains K_A , implying that $\text{vol}(K_A) = O(\text{vol}(K_0))$. Also, because $K_H = K_A^*$, by Lemma 3.2, $\text{vol}(K_A) \cdot \text{vol}(K_H) = \Omega(1)$. Thus, $\text{vol}(K_H) = \Omega(1/\text{vol}(K_0))$. To upper bound $\mu(K_H)$, note that by polarity, $K_H \subseteq K_0^*$, and thus

$$\mu(K_H) = \text{vol}(K_A) \cdot \text{vol}(K_H) = O(\text{vol}(K_0) \cdot \text{vol}(K_0^*)) = O(\mu(K_0)) = O(1),$$

where in the last step, we have used Lemma 3.1 and our assumption that the origin coincides with the centroid of K_0 .

To simplify notation, for the remainder of the proof we assume that ray distances, Macbeath regions, and volumes are defined relative to K_H , that is, $\text{ray} \equiv \text{ray}_{K_H}$, $M \equiv M_{K_H}$, and $\text{vol} \equiv \text{vol}_{K_H}$.

For any point $p \in \partial K_1^*$, let p' denote the point of intersection of the ray Op with ∂K_H . We first establish a bound on the relative ray distance $\text{ray}(p)$. Observe that since p and p' lie on ∂K_1^* and ∂K_H , respectively, their polar hyperplanes, p^* and p'^* , are supporting hyperplanes for K_1 and $K_H^* = K_A$, respectively. Letting r denote the distance between p'^* and the origin, it follows from the definition of K_A that the distance between p^* and the origin is $r + \frac{\varepsilon}{2}$. The distance of p' and p from the origin are the reciprocals of these. Therefore, we have

$$\text{ray}(p) = \frac{\|pp'\|}{\|Op'\|} = \frac{\|Op'\| - \|Op\|}{\|Op'\|} = \frac{\frac{1}{r} - \frac{1}{r+(\varepsilon/2)}}{\frac{1}{r}} = 1 - \frac{r}{r+(\varepsilon/2)} = \frac{\varepsilon/2}{r+(\varepsilon/2)}.$$

Since $\frac{1}{\|Op'\|} = r = \Omega(\varepsilon)$, we have $\text{ray}(p) = \Theta(\varepsilon/r) = \Theta(\varepsilon\|Op'\|)$. (It is noteworthy and somewhat surprising that this relative ray distance is not a dimensionless quantity, since it depends linearly on $\|Op'\|$.)

To analyze $|X|$, we partition it into groups based on $\|Ox'\|$ for each $x \in X$. Define $R_0 = (\text{vol}(K_H))^{1/d}$. By our earlier remarks, $\text{vol}(K_H) = \Omega(1/\text{vol}(K_0))$, and so $R_0 = \Omega(1/\Delta_d(K_0))$. For any integer i (possibly negative), define $R_i = 2^i R_0$ and $\varepsilon_i = \varepsilon R_i$. We can express X as the disjoint union of sets X_i , where X_i consists of points x such that $R_i \leq \|Ox'\| < 2R_i$. Recall that for any $x \in X_i$, we have $\text{ray}(x) = \Theta(\varepsilon\|Ox'\|) = \Theta(\varepsilon R_i) = \Theta(\varepsilon_i)$.

We will bound the contributions of the $|X_i|$ to $|X|$ based on the sign of i . Let us first consider the nonnegative values of i . We remark that $|X_i| = 0$ for large i (specifically, for $i = \omega(\log(1/\varepsilon R_0))$) because a ball of radius $\Omega(\varepsilon)$ centered at the origin is contained within K_0 ,

9:14 **Optimal Volume-Sensitive Bounds for Polytope Approximation**

and so by polarity K_0^* , and hence K_1^* , is contained within a ball of radius $O(1/\varepsilon)$. Recalling that K_H is well-centered and applying Lemma 3.7(i), we have (up to constant factors)

$$\begin{aligned} \sum_{i \geq 0} |X_i| &\leq \sum_{i \geq 0} \left(\frac{1}{\varepsilon_i}\right)^{\frac{d-1}{2}} = \sum_{i \geq 0} \left(\frac{1}{\varepsilon 2^i R_0}\right)^{\frac{d-1}{2}} \leq \sum_{i \geq 0} \left(\frac{\Delta_d(K_0)}{\varepsilon 2^i}\right)^{\frac{d-1}{2}} \\ &= \left(\frac{\Delta_d(K_0)}{\varepsilon}\right)^{\frac{d-1}{2}} \sum_{i \geq 0} \left(\frac{1}{2}\right)^{i \frac{(d-1)}{2}} \leq \left(\frac{\Delta_d(K_0)}{\varepsilon}\right)^{\frac{d-1}{2}} = O(C(K_0, \varepsilon)). \end{aligned}$$

In order to bound the contributions to $|X|$ for negative values of i , we need a more sophisticated strategy. Our approach is to first bound the total relative volume of the Macbeath regions of $\mathcal{M}^{1/4c}(X_i)$, which we assert to be $O(\varepsilon_i 2^{id})$. Assuming this assertion for now, we complete the proof as follows. By applying Lemma 3.7(ii) with $f = O(2^{id})$ and recalling that $\varepsilon_i = \varepsilon R_i = 2^i \varepsilon R_0$, we have (up to constant factors)

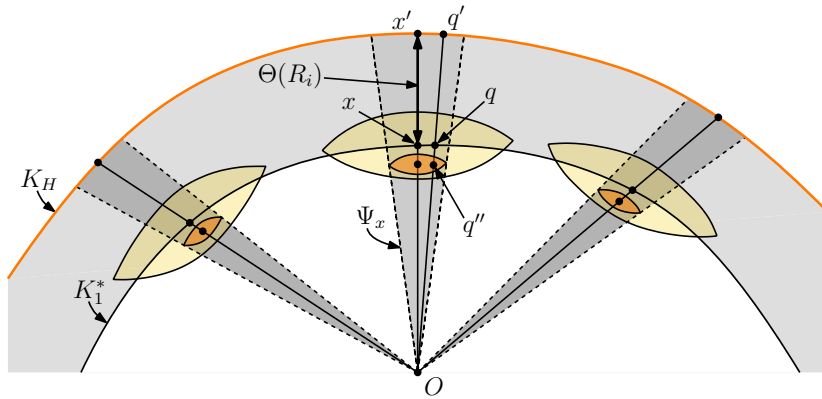
$$\begin{aligned} \sum_{i < 0} |X_i| &\leq \sum_{i < 0} \frac{\sqrt{f}}{\varepsilon_i^{(d-1)/2}} = \sum_{i < 0} \frac{2^{id/2}}{(2^i \varepsilon R_0)^{(d-1)/2}} = \sum_{i < 0} \frac{2^{i(d-(d-1))/2}}{(\varepsilon R_0)^{(d-1)/2}} \\ &= \sum_{i < 0} \frac{2^{i/2}}{(\varepsilon R_0)^{(d-1)/2}} = \sum_{i < 0} 2^{i/2} C(K_0, \varepsilon) = C(K_0, \varepsilon) \sum_{i > 0} \left(\frac{1}{2}\right)^{\frac{i}{2}} \\ &= O(C(K_0, \varepsilon)). \end{aligned}$$

It remains only to prove the assertion on the total relative volume of $\mathcal{M}^{1/4c}(X_i)$. Let $x \in X_i$ and let $M_x = \mathcal{M}^{1/4c}(x)$. By Lemma 4.4 (with x , K_1^* , and K_H playing the roles of c , K_0 , and K_H , respectively), there is an associated convex body M'_x such that

- (i) $\text{vol}(M'_x) = \Omega(\text{vol}(M_x))$,
- (ii) $M'_x \subseteq M_x \cap K_1^*$,
- and (iii) $\text{shadow}_{K_1^*}(M'_x) \subseteq M_x$.

We will use S_x as a shorthand for $\text{shadow}_{K_1^*}(M'_x)$. Since $\text{vol}(M_x) = O(\text{vol}(M'_x)) = O(\text{vol}(S_x))$, it suffices to show that the total relative volume of the shadows $\{S_x : x \in X_i\}$ is $O(\varepsilon_i 2^{id})$.

For $x \in X_i$, we define cone Ψ_x to be the intersection of K_H with the infinite cone consisting of rays emanating from the origin that contain a point of S_x (see Figure 9). Since the Macbeath regions of $\mathcal{M}^{1/4c}(X_i)$ are disjoint, it follows from (iii) that the associated shadows intersect ∂K_1^* in patches that are also disjoint. Thus the set of cones $\Psi = \{\Psi_x : x \in X_i\}$ are disjoint.



■ **Figure 9** Proof of Lemma 5.1.

Consider a ray emanating from the origin that is contained in any cone Ψ_x . Let q and q' be the points of intersection of this ray with ∂K_1^* and ∂K_H , respectively. Let q'' be any point on this ray that lies inside shadow S_x . Since $q'' \in M_x$, by Lemma 3.3, we

have $\text{ray}(q'') = \Theta(\text{ray}(x)) = \Theta(\varepsilon_i)$. By the same reasoning, $\text{ray}(q) = \Theta(\varepsilon_i) = \Theta(\varepsilon R_i)$. Also, recalling our earlier bounds on the relative ray distance of points on ∂K_1^* , we have $\text{ray}(q) = \Theta(\varepsilon \|Oq'\|)$. Equating the two expressions for $\text{ray}(q)$, we obtain $\|Oq'\| = \Theta(R_i)$.

Since the cones of Ψ are disjoint and any ray emanating from the origin and contained in a cone of Ψ has length $\Theta(R_i)$, it follows that the total volume of these cones is $O(R_i^d)$. Further, since only a fraction ε_i of any such ray is contained in the associated shadow, it follows that the total volume of all the shadows $\{S_x : x \in X_i\}$ is $O(\varepsilon_i R_i^d)$. Recalling that $\text{vol}(K_H) = R_0^d$ and $R_i = 2^i R_0$, it follows that the total relative volume of these shadows is $O(\varepsilon_i R_i^d / R_0^d) = O(\varepsilon_i 2^{id})$. This establishes the assertion on the total relative volume of $\mathcal{M}^{1/4c}(X_i)$ and completes the proof. \blacktriangleleft

References

- 1 P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets. In J. E. Goodman, J. Pach, and E. Welzl, editors, *Combinatorial and Computational Geometry*. MSRI Publications, 2005.
- 2 R. Arya, S. Arya, G. D. da Fonseca, and D. M. Mount. Optimal bound on the combinatorial complexity of approximating polytopes. *ACM Trans. Algorithms*, 18:1–29, 2022. doi:10.1145/3559106.
- 3 S. Arya and T. M. Chan. Better ε -dependencies for offline approximate nearest neighbor search, Euclidean minimum spanning trees, and ε -kernels. In *Proc. 30th Annu. Sympos. Comput. Geom.*, pages 416–425, 2014. doi:10.1145/2582112.2582161.
- 4 S. Arya, G. D. da Fonseca, and D. M. Mount. Optimal area-sensitive bounds for polytope approximation. In *Proc. 28th Annu. Sympos. Comput. Geom.*, pages 363–372, 2012. doi:10.1145/2261250.2261305.
- 5 S. Arya, G. D. da Fonseca, and D. M. Mount. Near-optimal ε -kernel construction and related problems. In *Proc. 33rd Internat. Sympos. Comput. Geom.*, pages 10:1–15, 2017. doi:10.4230/LIPIcs.SocG.2017.10.
- 6 S. Arya, G. D. da Fonseca, and D. M. Mount. Economical convex coverings and applications. In *Proc. 34th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1834–1861, 2023. doi:10.1137/1.9781611977554.ch70.
- 7 S. Arya, G. D. da Fonseca, and D. M. Mount. Economical convex coverings and applications, 2023. arXiv:2303.08349.
- 8 K. Böröczky Jr. Approximation of general smooth convex bodies. *Adv. Math.*, 153:325–341, 2000. doi:10.1006/aima.1999.1904.
- 9 J. Bourgain and V. D. Milman. New volume ratio properties for convex symmetric bodies. *Invent. Math.*, 88:319–340, 1987. doi:10.1007/BF01388911.
- 10 E. M. Bronshteyn and L. D. Ivanov. The approximation of convex sets by polyhedra. *Siberian Math. J.*, 16:852–853, 1976. doi:10.1007/BF00967115.
- 11 E. M. Bronstein. Approximation of convex sets by polytopes. *J. Math. Sci.*, 153(6):727–762, 2008. doi:10.1007/s10958-008-9144-x.
- 12 K. L. Clarkson. Building triangulations using ε -nets. In *Proc. 38th Annu. ACM Sympos. Theory Comput.*, pages 326–335, 2006. doi:10.1145/1132516.1132564.
- 13 R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974. doi:10.1016/0021-9045(74)90120-8.
- 14 H. G. Eggleston. *Convexity*. Cambridge University Press, 1958. doi:10.1017/CB09780511566172.
- 15 W. J. Firey. Polar means of convex bodies and a dual to the Brunn-Minkowski theorem. *Canad. J. Math.*, 13:444–453, 1961. doi:10.4153/CJM-1961-037-0.

- 16 P. M. Gruber. Aspects of approximation of convex bodies. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, chapter 1.10, pages 319–345. North-Holland, 1993. doi:10.1016/B978-0-444-89596-7.50015-8.
- 17 B. Grünbaum. Measures of symmetry for convex sets. In *Proc. Sympos. Pure Math.*, volume VII, pages 233–270, 1963. doi:10.1090/pspum/007/0156259.
- 18 G. Kuperberg. From the Mahler conjecture to Gauss linking integrals. *Geom. Funct. Anal.*, 18:870–892, 2008. doi:10.1007/s00039-008-0669-4.
- 19 D. E. McClure and R. A. Vitalie. Polygonal approximation of plane convex bodies. *J. Math. Anal. Appl.*, 51:326–358, 1975. doi:10.1016/0022-247X(75)90125-0.
- 20 P. McMullen. Non-linear angle-sum relations for polyhedral cones and polytopes. *Math. Proc. Cambridge Philos. Soc.*, 78(2):247–261, 1975. doi:10.1017/S0305004100051665.
- 21 P. McMullen. Inequalities between intrinsic volumes. *Monatshefte für Mathematik*, 111:47–53, 1991. doi:10.1007/BF01299276.
- 22 M. Meyer and A. Pajor. On the Blaschke-Santaló inequality. *Arch. Math.*, 55:82–93, 1990. doi:10.1007/BF01199119.
- 23 V. D. Milman and A. Pajor. Entropy and asymptotic geometry of non-symmetric convex bodies. *Adv. Math.*, 152:314–335, 2000. doi:10.1006/aima.1999.1903.
- 24 H. Minkowski. Allgemeine Lehrsätze über die konvexen Polyeder. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Math-Phys. Kl.*, pages 198–219, 1897. URL: <http://eudml.org/doc/58391>.
- 25 F. Nazarov. The Hörmander proof of the Bourgain-Milman theorem. In *Geometric Aspects of Functional Analysis*, pages 335–343. Springer, 2012. doi:10.1007/978-3-642-29849-3_20.
- 26 J. Radon. Über eine Erweiterung des Begriffs der konvexen Functionen mit einer Anwendung auf die Theorie der konvexen Körper. *S.-B. Akad. Wiss. Wien*, 125:241–258, 1916.
- 27 J. Richter-Gebert. *Perspectives on Projective Geometry: A Guided Tour Through Real and Complex Geometry*. Springer, 2011. doi:10.1007/978-3-642-17286-1.
- 28 L. A. Santaló. An affine invariant for convex bodies of n -dimensional space. *Port. Math.*, 8:155–161, 1949. (In Spanish).
- 29 R. Schneider. Polyhedral approximation of smooth convex bodies. *J. Math. Anal. Appl.*, 128:470–474, 1987. doi:10.1016/0022-247X(87)90197-1.
- 30 R. Schneider. *Convex bodies: The Brunn-Minkowski theory*. Cambridge University Press, 1993. doi:10.1017/CB09780511526282.
- 31 G. Toth. *Measures of symmetry for convex sets and stability*. Springer, 2015. doi:10.1007/978-3-319-23733-6.
- 32 L. F. Toth. Approximation by polygons and polyhedra. *Bull. Amer. Math. Soc.*, 54:431–438, 1948.

Coresets for Clustering in Geometric Intersection Graphs

Sayan Bandyapadhyay ✉ 
Portland State University, OR, USA

Fedor V. Fomin ✉ 
University of Bergen, Norway

Tanmay Inamdar ✉ 
University of Bergen, Norway

Abstract

Designing coresets – small-space sketches of the data preserving cost of the solutions within $(1 \pm \epsilon)$ -approximate factor – is an important research direction in the study of center-based k -clustering problems, such as k -means or k -median. Feldman and Langberg [STOC’11] have shown that for k -clustering of n points in general metrics, it is possible to obtain coresets whose size depends logarithmically in n . Moreover, such a dependency in n is inevitable in general metrics. A significant amount of recent work in the area is devoted to obtaining coresets whose sizes are independent of n for special metrics, like d -dimensional Euclidean space [Huang, Vishnoi, STOC’20], doubling metrics [Huang, Jiang, Li, Wu, FOCS’18], metrics of graphs of bounded treewidth [Baker, Braverman, Huang, Jiang, Krauthgamer, Wu, ICML’20], or graphs excluding a fixed minor [Braverman, Jiang, Krauthgamer, Wu, SODA’21].

In this paper, we provide the first constructions of coresets whose size does not depend on n for k -clustering in the metrics induced by *geometric intersection graphs*. For example, we obtain $\frac{k \log^2 k}{\epsilon^{O(1)}}$ size coresets for k -clustering in Euclidean-weighted unit-disk graphs (UDGs) and unit-square graphs (USGs). These constructions follow from a general theorem that identifies two canonical properties of a graph metric sufficient for obtaining coresets whose size is independent of n . The proof of our theorem builds on the recent work of Cohen-Addad, Saulpic, and Schwiegelshohn [STOC’21], which ensures small-sized coresets conditioned on the existence of an interesting set of centers, called *centroid set*. The main technical contribution of our work is the proof of the existence of such a small-sized centroid set for graphs that satisfy the two canonical properties. Loosely speaking, the metrics of geometric intersection graphs are “similar” to the Euclidean metrics for points that are close, and to the shortest path metrics of planar graphs for points that are far apart. The main technical challenge in constructing centroid sets of small sizes is in combining these two very different metrics.

The new coreset construction helps to design the first $(1 + \epsilon)$ -approximation for center-based clustering problems in UDGs and USGs, that is fixed-parameter tractable in k and ϵ (FPT-AS).

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Facility location and clustering; Theory of computation → Sparsification and spanners

Keywords and phrases k -median, k -means, clustering, coresets, geometric graphs

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.10

Related Version *Full Version*: <https://arxiv.org/abs/2303.01400>

Funding The research leading to these results has received funding from the Research Council of Norway via the project BWCA (grant no. 314528), and the European Research Council (ERC) via grant LOPPRE, reference 819416.



© Sayan Bandyapadhyay, Fedor V. Fomin, and Tanmay Inamdar;
licensed under Creative Commons License CC-BY 4.0

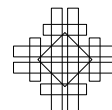
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 10; pp. 10:1–10:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Clustering is one of the most important data analysis techniques where the goal is to partition a dataset into a number of groups such that each group contains similar set of data points. The notion of similarity is captured by a distance function between the data points, and the goal of retrieving the best natural clustering of the data points is achieved by minimizing a proxy cost function. In this work, we study the popular (k, z) -clustering problem.

(k, z) -clustering. Given a set of points P in a metric space (Ω, d) and two positive integers k and z , find a set C of k points (or centers) in Ω that minimizes the following cost function:

$$\text{cost}(C) = \sum_{p \in P} \text{cost}(p, C)$$

where $\text{cost}(p, C) = (d(p, C))^z$ and $d(p, C) = \min_{c \in C} d(p, c)$.

Two widely studied clustering problems, k -means, and k -median clustering, are special versions of (k, z) -clustering with $z = 2$ and $z = 1$, respectively. A popular way of dealing with large data for the purpose of the analysis is to apply a data reduction scheme as a preprocessing step. In the context of clustering, one such way of preprocessing the data is to construct an object known as *coresets*.

Coresets. Informally, an ϵ -coreset for (k, z) -clustering is a small-sized summary of the data that approximately (within $(1 \pm \epsilon)$ factor) preserves the cost of clustering with respect to any set of k centers (we will often shorten “ ϵ -coreset” to simply “coreset”). Thus, any solution set of centers computed for the coreset points can be readily used as a solution for the original dataset. A formal definition follows.

► **Definition 1 (ϵ -Coreset).** A coreset for (k, z) -clustering problem on a set P of points in a metric space (Ω, d) is a weighted subset Y of Ω with weights $\omega : Y \rightarrow \mathbb{R}^+$ such that for any set $\mathcal{S} \subseteq \Omega$ with $|\mathcal{S}| = k$,

$$\left| \sum_{p \in P} \text{cost}(p, \mathcal{S}) - \sum_{p \in Y} \omega(p) \text{cost}(p, \mathcal{S}) \right| \leq \epsilon \cdot \sum_{p \in P} \text{cost}(p, \mathcal{S}).$$

Feldman and Langberg [19] showed that for n points in any general metric, a coreset of size $\mathcal{O}(\epsilon^{-2z} k \log k \log n)$ can be constructed in time $\tilde{\mathcal{O}}(nk)$, where $\tilde{\mathcal{O}}()$ notation hides a poly-logarithmic factor. Also, it is known that the dependency on $\log n$ in the above bound cannot be avoided [3, 17]. However, for several special metrics, it is possible to construct coresets whose size does not depend on the data size. There has been a large pool of work for Euclidean spaces, culminating in a bound of $\mathcal{O}(k \cdot (\log k)^{\mathcal{O}(1)} \cdot 2^{\mathcal{O}(z \log z)} \epsilon^{-2} \cdot \min\{\epsilon^{-z}, k\})$ [17], which is independent of the data size and dimension of the space. Similarly, coresets of size independent of n are also obtained in other specialized settings such as doubling metrics [27], shortest-path metrics in the graphs of bounded-treewidth [3], and graphs excluding a fixed minor [8]. A recent result by Cohen-Addad et al. [17] gives a unified framework that encompasses all these results. We note that, since the number of distinct (weighted) points in coresets is usually much smaller (and sometimes independent of) n , they naturally find applications in non-sequential settings such as streaming [26, 13].

Let us remark that all known results about small coresets in graph metrics strongly exploit the sparse nature of graphs such as bounded treewidth [3] or excluding a fixed minor [8]. There is a very good reason for that. In a complete graph, by setting suitable weights on the edges one can represent *any* general metric. Thus if a graph family contains large cliques, clustering in such graphs is as difficult as in general metrics.

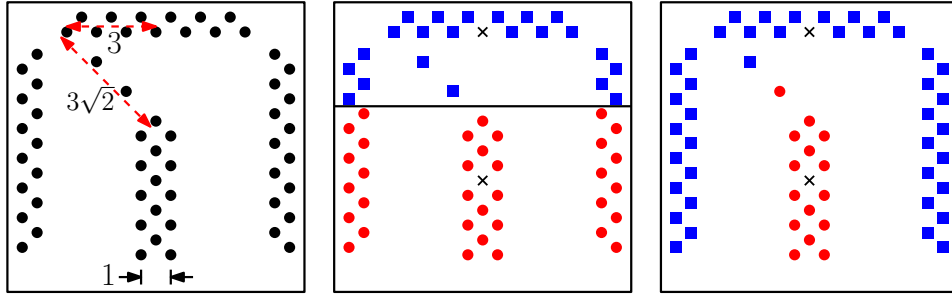
In this work, we are interested in coresets construction for edge-weighted geometric intersection graphs with shortest-path metric. A geometric intersection graph of a set of geometric objects contains a vertex for each object and an edge corresponding to each pair of objects that have non-empty intersection. (We note that for our purpose of designing algorithms, we do not explicitly need the objects or their geometric representation. It is sufficient to work with the graph representation as long as the edge-weights are given.) In particular, geometric intersection graphs are a widely studied model for ad-hoc communication and wireless sensor networks [38, 4, 34, 37, 31]. Notably, clustering is a common topology management method in such networks. Grouping nodes are used as subroutines for executing various tasks in a distributed manner and for resource management, see the survey [39] for an overview of different clustering methods for wireless sensor networks.

Our work is motivated by the following question: “*Is it possible to exploit the properties of geometric intersection graphs for obtaining coresets whose size does not depend on the data size?*” In general, the answer to this question is *no*. This is because geometric intersection graphs can contain large cliques. Even for objects as simple as unit squares, the corresponding intersection graph could be a clique, and, as we already noted, by setting suitable weights on the edges of the clique, one can represent any metric. Hence constructing coresets in geometric intersection graphs with arbitrary edge weights is as difficult as in general metrics. Thus, we need to restrict edge weights in some manner in order to obtain non-trivial coresets for geometric intersection graphs. As an illustrative example, let us take a look at Euclidean-weighted UDGs, a well-studied class of geometric intersection graphs.

Euclidean-weighted unit-disk graph metric. A unit-disk graph (UDG) is defined in the following way – there is a configuration of closed disks of radii 1 in the plane and a one-to-one correspondence between the vertices and the centers of the disks such that there is an edge between two vertices if and only if the disks having the two corresponding centers intersect. The weight of an edge is equal to the Euclidean distance between the two corresponding centers. Euclidean-weighted UDGs have been well-studied in computational geometry [10, 24]. Apart from practical motivation, UDGs are interesting from theoretical perspectives as well. On the one hand, being embedded on the plane they resemble planar graphs when “zoomed out”, but could contain large cliques locally. On the other hand, the metric induced by them is an amalgamation of geometric and graphic settings, as it is locally Euclidean but globally a graph metric. Due to the latter property, UDG metric can be used for fine-tuned clustering, as with pure Euclidean distances one can only retrieve clusters induced by convex partitions of the space (see Figure 1).

1.1 Our Results

We now formalize our intuition about the “hybrid” nature about the Euclidean weighted UDGs, by identifying two canonical geometric properties of a graph G that are sufficient for constructing small-sized coresets. For better exposition, we fix a few notations. For any subgraph H of G , we denote its set of vertices and set of edges by $V(H)$ and $E(H)$, respectively. For vertex set $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by V' . For a subgraph H of G , and $u, v \in V(H)$, let $\pi_H(u, v)$ denote a shortest path between u and v (according to the edge-weights in G restricted to H) that uses the edges of H , and let $d_H(u, v)$ denote the weight of $\pi_H(u, v)$, i.e., the sum of the weights of the edges along $\pi_H(u, v)$. For any path π in a graph, let $|\pi|$ denote the number of edges on π . Note that d_H is the so-called *shortest path metric* on H . Finally, for any pair of points $p, q \in \mathbb{R}^2$, let $|pq|$ denote the euclidean (i.e., ℓ_2 -norm) distance between p and q .



■ **Figure 1** (Left.) A point set in 2D. (Middle.) 2-means clustering with Euclidean distances. Centers are shown by crosses. Points of two clusters are shown by disks (red) and squares (blue). (Right.) 2-means clustering on UDG – due to the $3\sqrt{2}$ diagonal path distance, all the blue points (squares) are closer to the upper center.

Canonical geometric properties.

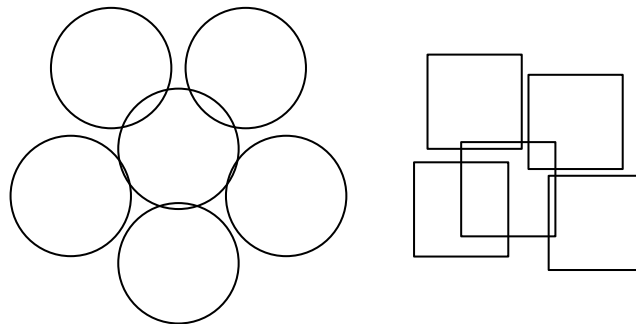
- (1) *Locally Euclidean:* There exist (not necessarily distinct) constants $c_1, c_2, c_3, c_4 \geq 0$, such that the following holds. G has an embedding $\lambda : V(G) \rightarrow \mathbb{R}^2$ in the plane such that the vertices of G are mapped to points in the plane, with the following two properties.
 1. For any two $u, v \in V(G)$, if $|\lambda(u)\lambda(v)| \leq c_1$ then $uv \in E(G)$, and for any $u', v' \in V(G)$, if $|\lambda(u')\lambda(v')| > c_2$, then $u'v' \notin E(G)$.
 2. For any $u, v \in V(G)$ such that $uv \in E(G)$, let $w(uv)$ denote the weight of the edge uv . Then, the edge uv is a shortest path between u and v in G . Furthermore, $c_3 \cdot |\lambda(u)\lambda(v)| \leq w(uv) \leq c_4 \cdot |\lambda(u)\lambda(v)|$.
- (2) *Planar Spanner:* For any induced subgraph $G' = G[V']$ with $V' \subseteq V(G)$, there exists a planar α -spanner $H' = (V', E(H'))$ for some fixed $\alpha \geq 1$, i.e., (i) H' is a subgraph of G' (and hence of G) – $E(H') \subseteq E(G')$, and (ii) for any $u, v \in V'$, $d_{G'}(u, v) \leq d_{H'}(u, v) \leq \alpha \cdot d_{G'}(u, v)$.

Our main result is the following theorem.

► **Theorem 2 (Informal).** *Consider the metric space (V, d_G) induced by any graph G satisfying the two canonical geometric properties (1) and (2), and a set $P \subseteq V(G)$. Then there exists a polynomial time algorithm that constructs a coreset for (k, z) -clustering on P of size $\mathcal{O}(\epsilon^{-\beta} k \log^2 k)$, where $\beta = \mathcal{O}(z \log z)$.*

Theorem 2 is a handy tool to construct coresets for several interesting geometric intersection graphs coupled with suitable metrics. First, let us observe that our initial example, namely, a metric induced by a Euclidean-weighted UDG G satisfies the two canonical properties. Consider an embedding of G in \mathbb{R}^2 . Note that there is an edge between any two points iff the Euclidean distance between the two points is at most 2, and the weight of such an edge is exactly the euclidean distance. Thus, G is *Locally Euclidean* with $c_1 = c_2 = 2$, and $c_3 = c_4 = 1$. Furthermore, due to a result of Li, Calinescu, and Wan [36], any Euclidean-weighted UDG admits a constant-stretch planar spanner. Thus, G also satisfies the *Planar Spanner* property. Therefore, due to Theorem 2, we can obtain $\mathcal{O}(\epsilon^{-\beta} k \log^2 k)$ -size coresets for (k, z) -clustering on Euclidean-weighted UDGs. In the following, we discuss further applications of our framework.

ℓ_∞ -weighted unit-square graph metric. Unit-square graphs (USGs) are similar to UDGs except they are defined as intersection graphs of (axis-parallel) unit squares instead of unit disks¹. Indeed, these two graph classes are distinct. For example, the $K_{1,5}$ claw can be realized by a UDG, but not by any USG. (See Figure 2.) Since a unit square is a “unit ball” in ℓ_∞ -norm, it is more natural to consider ℓ_∞ weights on the edges. It is not too difficult to see that the *Locally Euclidean* property holds for ℓ_∞ -weighted USGs – we give a formal proof in the full version given in the appendix. On the other hand, in order to establish the second property, we have to prove the existence of a constant-stretch planar spanner for USGs. To the best of our knowledge this result was previously not known and is of independent interest. We give a proof of this result in the full version. Thus, ℓ_∞ weighted USGs also satisfy the two properties required to apply Theorem 2 in order to obtain a small-sized coresets.



■ **Figure 2** A set of disks realizing $K_{1,5}$ (left) and a set of squares realizing $K_{1,4}$ (right). A pair of intersecting unit squares must contain a corner of each other, and so the central square can intersect with at most four other unit squares that are pairwise disjoint.

Other extensions. In \mathbb{R}^2 , all ℓ_p distances ($1 \leq p \leq \infty$) are within a $\sqrt{2}$ factor from each other. Thus, our arguments can be easily extended to any ℓ_p weights on UDGs/USGs for any $1 \leq p \leq \infty$ without any changes on the bounds (a formal argument can be found in the full version). Lastly, we consider shortest-path metrics in unweighted (i.e., hop-distance) unit-disk graphs of bounded degree. Notably, these graphs satisfy the *Planar Spanner* property due to a result by [7], but not the *Locally Euclidean* property. Nevertheless, we can modify our approach to construct a small-sized coresets for such metrics. To summarize, we obtain coresets for (k, z) -clustering with size independent of n for the following graph metrics.

- ℓ_p -distance weighted UDGs for any $1 \leq p \leq \infty$,
- ℓ_p -distance weighted USGs for any $1 \leq p \leq \infty$,
- Bounded-degree unweighted UDGs.

FPT Approximation Schemes. As a corollary to Theorem 2, we obtain $(1 + \epsilon)$ -approximations for (k, z) -clustering in geometric intersection graphs that are fixed-parameter tractable (FPT) in k and ϵ . Note that such a $(1 + \epsilon)$ -approximation was not known before even for UDGs, as it does not follow from previously known bound on coresets sizes. Prior to our work, the best known bound for coresets on UDGs – as in general metrics – was $O(k \log n \cdot \epsilon^{-\max(2, z)})$ [18]. Even though a coresets reduces the number of distinct points

¹ Although it might seem unnatural at first, it is convenient to define a *unit square* as a square of sidelength 2. This is analogous to a unit disk being a disk of diameter 2. In either case, the class of USGs remains unaffected by scaling.

(or clients) to be clustered, the number of potential centers (or facilities) still remains the same, i.e., n . Hence, a coreset does not directly help us enumerate all possible sets of k centers from which we could pick the best set. An alternative way to enumerate these sets of centers is to enumerate all possible partitions (or clusterings) of the coreset points. Note that each clustering of coreset points corresponds to a clustering of the original points, and the cost of clustering is preserved to within a $(1 \pm \epsilon)$ factor. With our coreset bound of $\mathcal{O}(\epsilon^{-\beta} k \log^2 k)$, the number of distinct clusterings is only $k^{\mathcal{O}(\epsilon^{-\beta} k \log^2 k)}$. Overall the algorithm takes $k^{\mathcal{O}(\epsilon^{-\beta} k \log^2 k)} n^{\mathcal{O}(1)}$ time.

► **Corollary 3.** *For each of the metrics listed in the above, there exists a $(1 + \epsilon)$ -approximation for (k, z) -clustering with $z \geq 1$ that runs in time $2^{\mathcal{O}(\epsilon^{-\beta} k \log^3 k)} n^{\mathcal{O}(1)}$, where $\beta = \mathcal{O}(z \log z)$.*

1.2 General overview of the methods

Our coreset construction is based on a recent work due to Cohen-Addad, Saulpic, and Schwiegelshohn [18], which gives a framework for constructing coresets in various settings. The essence of the framework is that it translates the problem of coreset construction to showing the existence of an interesting set of centers or centroid set. In particular, consider any set \mathcal{S} of k centers and any subset $X \subseteq P$ of points that are sufficiently close to \mathcal{S} compared to an existing solution \mathcal{A} . Then a subset $\mathbb{C} \subseteq \Omega$ is a *centroid set* for P if it contains centers that well-approximates \mathcal{S} , i.e., there exists $\tilde{\mathcal{S}} \subseteq \mathbb{C}$, such that for every $p \in X$, it holds that $|\text{cost}(p, \mathcal{S}) - \text{cost}(p, \tilde{\mathcal{S}})| \leq \epsilon(\text{cost}(p, \mathcal{S}) + \text{cost}(p, \mathcal{A}))$. The framework informally states that if there is a centroid set \mathbb{C} , then a coreset can be constructed whose size depends logarithmically on $|\mathbb{C}|$. Such a dependency arises in their randomized construction in order to prove a union bound over all possible interesting solutions, which can be at most $|\mathbb{C}|^k$. By showing the existence of small-sized centroid sets, they obtain improved coreset size bounds for a wide range of spaces.

We use the framework of Cohen-Addad et al. for our coreset construction. Our main technical result shows if for a graph G with metric d_G , the two canonical geometric properties are satisfied, then there exists a small-sized centroid set for G . This is the most challenging part of the proof and it requires a novel combination of tools and techniques from computational geometry. As soon as we establish the existence of the centroid set, the construction of coresets follows the steps of [18]. For the sake of exposition, let us consider a concrete example of Euclidean-weighted UDGs.

Consider any cluster of points with cluster center s . The points that are nearby (i.e., within distance $2r$) s behave simply as points in the Euclidean case. But, a point p that is far away from s can have a shortest path distance which may be much larger than the actual Euclidean distance between p and s , see Figure 1. We first show that it is possible to conceptually separate out these two cases – but one has to be careful, as a cluster can potentially contain both types of points. Notably, none of the previous works had to deal with such a hybrid metric. To handle the set of nearby points, we exploit the *Locally Euclidean* property. In particular, by overlaying a grid of appropriately small sidelength, and selecting one representative point from each cell of the grid, we can compute a centroid set that preserves the distances from the nearby points.

In the other case, a shortest path between a point p and a center s consists of more than one edge, and we need to deal with a graphical metric. This case is much more interesting. All other works establishing small-sized centroid sets in certain graph metrics exploit the fact that certain graph classes admit small or well-behaved separators. For example, bounded treewidth graph admit separators bounded by treewidth; whereas graphs excluding a fixed

minor admit shortest path separators. However, UDGs may contain arbitrarily large cliques, and therefore do not admit such separators in general. Thus, we reach a technical bottleneck. Note that this is the first work of its kind that handles such a dense graph. To overcome this challenge, we use the other canonical property. Instead of directly working with the UDG, we consider its planar spanner, where distances are preserved up to a constant factor. The existence of such a spanner is guaranteed by the second canonical property, *Planar Spanner*. As planar graphs have shortest path separators, now we can apply the existing techniques. However, if we were to entirely rely on the spanner, some of the distances may be scaled up by a constant (> 2) factor in the spanner, and thus it would not be possible to ensure the $(1 \pm \epsilon)$ -factor bound required to construct a coreset.

Thus, we use the spanner as a supporting graph in the following way. First, we recursively decompose the original UDG by making use of the shortest path separators admitted by the planar spanner. We note that although planar graph decomposition has been used in coreset literature, using such a guided scheme to obtain a decomposition of a much more general graph is novel. Then, we use this recursive decomposition of the UDG, along with the shortest path separators used to find this decomposition, in order to construct the centroid set. In this construction, we use the spanner in a restricted manner, and use it such that error incurred by the use of the spanner is upper bounded by α times the weight of at most one edge along a shortest path from a point p to its corresponding (approximate) center. However, observe that if such a shortest path consists of a single edge, then even this error is too large. To resolve this issue, we rely on the planar spanner, only if the shortest path is “long enough”, i.e., contains $\Omega(z/\epsilon)$ edges. In this case, *Locally Euclidean* property implies that for such a “long path”, the *length* of the path and the *number of edges* on the path are within a constant factor from each other. This implies that the error introduced by rerouting a single edge using the spanner is at most ϵ times the length of the path, i.e., negligible.

Finally, if a shortest path between a point and a center consists of $\mathcal{O}(z/\epsilon)$ edges, then we can use a modified version of the grid-cell argument to obtain a small-sized centroid set.

We note that this is simply an intuitive overview of the challenges faced in each of the three cases. The actual construction of the centroid set, and the analysis of the error incurred in each of the cases is fairly convoluted. While replacing a center $s \in \mathcal{S}$ by another one $\tilde{s} \in \mathbb{C}$, we need to ensure that for a point p having s as its closest center, $d_G(p, \tilde{s})$ is neither too large nor too small compared to $d_G(p, s)$, since we want to bound the error in the absolute difference. In addition, we have to be extremely careful while combining the three centroid (sub)sets constructed for each of the cases, and ensure that a good replacement \tilde{s} found for a center s in one of the cases does not adversely distort the error for a point that is being handled in another case.

Related work. Here we give an overview of the literature on coresets. For a more exhaustive list, we refer to [18, 28]. Coreset construction was popularized by an initial set of works that obtained small-sized coresets in low-dimensional Euclidean spaces [26, 25, 22]. Chen [13] obtained the first coreset for Euclidean spaces with polynomial dependence on the dimension and the first coreset in general metrics, where the size is $\mathcal{O}(k^2 \epsilon^{-2} \log^2 n)$ for k -median. Subsequently, the dependence on the dimension has been further improved [33, 20]. Finally, such dependence was removed in [21, 40]. See also [5, 18, 28, 8, 17] for recent improvements.

Both k -median and k -means admit polynomial-time $\mathcal{O}(1)$ -approximations in general metrics [11, 12, 29, 35, 9, 1]. Moreover, algorithms with improved approximation guarantees can be obtained that is FPT in k and ϵ [15]. Naturally, the problems have also been studied in specialized metrics. Polynomial-time approximation schemes (PTASes) are known for

Euclidean k -median [2] and k -means [16, 23]. See [14, 32] for other improvements. Similar to geometric clustering, clustering in graphic setting is also widely studied. PTASes are known for excluded-minor graphs [16, 8].

Also, FPT approximation schemes are known for graphs of bounded-treewidth [3] and graphs of bounded highway dimension [6, 8].

2 Coresets for Geometric Graphs

To set up the stage, we need the following definition of *centroid set* from [18].

► **Definition (Centroid Set).** *Consider any metric space (Ω, d) , a set of clients $P \subseteq \Omega$, and two positive integers k and z . Let $\epsilon > 0$ be a precision parameter. Given a set of centers \mathcal{A} , a set \mathbb{C} is an \mathcal{A} -approximate centroid set for (k, z) -clustering on P that satisfies the following property.*

For every set of k centers $\mathcal{S} \subseteq \Omega$, there exists $\tilde{\mathcal{S}} \subseteq \mathbb{C}$, such that for every $p \in P$ that satisfies $\text{cost}(p, \mathcal{S}) \leq \left(\frac{10z}{\epsilon}\right)^z \cdot \text{cost}(p, \mathcal{A})$ or $\text{cost}(p, \tilde{\mathcal{S}}) \leq \left(\frac{10z}{\epsilon}\right)^z \cdot \text{cost}(p, \mathcal{A})$, it holds that

$$|\text{cost}(p, \mathcal{S}) - \text{cost}(p, \tilde{\mathcal{S}})| \leq \frac{\epsilon}{z \log(z/\epsilon)} (\text{cost}(p, \mathcal{S}) + \text{cost}(p, \mathcal{A})).$$

Informally, a centroid set \mathbb{C} is a collection of candidate centers, potentially much smaller than Ω , such that the k centers in \mathcal{S} can be replaced by k centers in $\tilde{\mathcal{S}} \subseteq \mathbb{C}$ without changing the cost of points by a large amount, that are much closer to \mathcal{S} or $\tilde{\mathcal{S}}$ compared to \mathcal{A} w.r.t. d . Cohen-Addad et al. [17] proved that one can obtain coresets whose size depends only logarithmically on the size of any such centroid set. More formally, they prove the following.

► **Proposition 4 ([18]).** *Consider any metric space (Ω, d) , a set of points $P \subseteq \Omega$ with n distinct points, and two positive integers k and z . Let $\epsilon > 0$ be a precision parameter. Suppose \mathcal{A} be a given constant-factor approximation for (k, z) -clustering on P .*

Suppose there exists an \mathcal{A} -approximate centroid set for (k, z) -clustering on P . Then there exists a polynomial time algorithm that constructs with probability at least $1 - \delta$ a coreset of size

$$\mathcal{O}\left(\frac{2^{\mathcal{O}(z \log z)} \cdot \log^4(1/\epsilon)}{\min\{\epsilon^2, \epsilon^z\}} (k \log |\mathbb{C}| + \log \log(1/\epsilon) + \log(1/\delta))\right)$$

with positive weights for (k, z) -clustering on P .

First, note that the above coreset framework requires only existence of such a centroid set. It is not necessary to explicitly compute it. Indeed, such a centroid set is only used to bound the size of computed coresets in their analysis. The main contribution of our work is to obtain small-sized centroid sets for geometric graph metrics that satisfy the two canonical geometric properties. In particular, we prove the following theorem.

► **Theorem 5 (Centroid Set Theorem).** *Consider the metric space (V, d_G) induced by any graph $G = (V, E)$ satisfying the Locally Euclidean and Planar Spanner properties defined before. Also consider a set of points $X \subseteq V$ and two positive integers k and $z \geq 1$. Let $\epsilon > 0$ be the precision parameter. Additionally, suppose \mathcal{A} be a solution for (k, z) -clustering on X . Then there exists an \mathcal{A} -approximate centroid set \mathbb{C} for (k, z) -clustering on X of size $\exp(\mathcal{O}(\log^2 |X| + z^{16} \epsilon^{-8} (\log(z/\epsilon))^8 \log |X|))$.*

We give an overview of the proof of Theorem 5 in the following section, and defer a formal proof to the Section 3 of the full version. Then, by combining Proposition 4 and arguments from [8], with some minor changes due to our different bound on coreset-size, we obtain the following theorem. A formal proof can be found in Section 2 of the full version.

► **Theorem 6.** Consider the metric space (V, d_G) induced by any graph $G = (V, E)$ satisfying *Locally Euclidean* and *Planar Spanner* properties, a set $P \subseteq V$ with n distinct points, and two positive integers k and $z \geq 1$. Then there exists a polynomial time algorithm that constructs with probability at least $1 - \delta$ a coreset for (k, z) -clustering on P of size $\mathcal{O}(\epsilon^{-\mathcal{O}(z \log z)} k \log^2 k \log^3(1/\delta))$, where z is a constant, and $\delta < 1/4$.

3 Overview of the Proof of Centroid Set Theorem

In this section, we give an overview of our main result, namely, the existence of a small-sized centroid set. A formal proof can be found in the full version.

Recall that we are given $G = (V, E)$, a connected, undirected, and edge-weighted graph on n vertices. Moreover, G satisfies the two canonical geometric properties, namely *Locally Euclidean*, and *Planar Spanner*.

As (V, d_G) is our metric space, we use the terms points and vertices interchangeably. $X \subseteq V$ is the given set of points. We are also given \mathcal{A} , a solution for (k, z) -clustering on X . We prove that there exists an \mathcal{A} -approximate centroid set of size $\exp(\mathcal{O}(\log^2 |X| + z^{16} \epsilon^{-8} (\log(z/\epsilon))^8 \log |X|))$ for (k, z) -clustering on X , which satisfies the following property.

For every set of k centers $\mathcal{S} \subseteq V$, there exists $\tilde{\mathcal{S}} \subseteq \mathbb{C}$, such that for every $p \in X$ that satisfies $\text{cost}(p, \mathcal{S}) \leq (\frac{10z}{\epsilon})^z \cdot \text{cost}(p, \mathcal{A})$ or $\text{cost}(p, \tilde{\mathcal{S}}) \leq (\frac{10z}{\epsilon})^z \cdot \text{cost}(p, \mathcal{A})$, it holds that

$$|\text{cost}(p, \mathcal{S}) - \text{cost}(p, \tilde{\mathcal{S}})| \leq \frac{\epsilon}{z \log(z/\epsilon)} (\text{cost}(p, \mathcal{S}) + \text{cost}(p, \mathcal{A})).$$

Now we proceed to an overview of the proof of the theorem. This proof can be divided into three steps.

1. Construction of a centroid set \mathbb{C} of a small size.
2. Given a solution $\mathcal{S} \in V^k$, finding for each center $s \in \mathcal{S}$, a *replacement* center $\rho(s) \in \mathbb{C}$, to construct $\tilde{\mathcal{S}} \in \mathbb{C}^k$.
3. Showing that $\tilde{\mathcal{S}}$ approximates \mathcal{S} , i.e., it satisfies the property specified above.

Given the hybrid nature of the metric, each of these three steps is subdivided into multiple cases. Recall that, due to the first canonical property, points in V that are close to each other behave as in the Euclidean case. To take care of the case of **points nearby** to their closest centers, we add a set of points \mathbb{C}_{net} to our centroid set. The case of far away points is further divided into two subcases. In the first subcase, we deal with the points whose shortest paths to closest centers are **short** or $\mathcal{O}(z/\epsilon)$ hops away. To take care of this subcase, we add a set of points $\mathbb{C}_{\text{support}}$ to our centroid set. The last subcase concerns **long paths**, and here we make use of the planar spanner property. In particular, we construct the centroid points in this case based on a recursive decomposition of the graph guided by underlying planar spanners of the decomposed subgraphs. This subcase resembles the centroid set construction in excluded-minor graph metrics from [17, 8]. In this informal overview, it will be convenient to consider each of these three cases separately, and discuss steps 1-3 in each case (overview of step 1 in each case hints at why the size of centroid set is bounded, but a formal proof is given in the full version).

Nearby points case.

Construction of \mathbb{C}_{net} . Let $p_1, p_2, \dots, p_{n'}$ be the points of X such that $d_G(p, \mathcal{A}) < 1$ for all $p = p_i$, where $1 \leq i \leq n'$. Let $B_i = B(p_i, (10z/\epsilon) \cdot d_G(p_i, \mathcal{A}))$ be the Euclidean ball

10:10 Coresets for Clustering in Geometric Intersection Graphs

centered at p_i having radius $(10z/\epsilon) \cdot d_G(p_i, \mathcal{A})$. For each $1 \leq i \leq n'$, compute an $(\epsilon^3/z^3) \cdot d_G(p_i, \mathcal{A})$ -net of the disk B_i that is a subset of V and add that to \mathbb{C}_{net} . That is, this net picks at most one point of V from each gridcell of length $\beta = (\epsilon^3/z^3) \cdot d_G(p_i, \mathcal{A})$ that intersects with the disk B_i . Note that the distance between any two points that belong to the same cell is $\mathcal{O}(\beta)$. In particular, the *Locally Euclidean* nature of G helps us define this net.

Finding a replacement center. For a center $s \in \mathcal{S}$, let $X_s \subseteq X$ denote the set of points that have s as their closest center in \mathcal{S} . If there exists a $p_i \in X_s$ with $d_G(p, \mathcal{A}) < 1$, then we argue that s belongs to the ball B_i as defined above. It follows that we added a point \tilde{s} to \mathbb{C}_{net} , such that $d_G(s, \tilde{s}) = \mathcal{O}(\frac{\epsilon^3}{z^3} d_G(p_i, \mathcal{A}))$. We pick \tilde{s} as the replacement for s , i.e., $\rho(s) = \tilde{s}$, and add it to $\tilde{\mathcal{S}}$.

Error analysis. Using triangle inequality, it follows that for any $p \in X_s$, $|d_G(p, s) - d_G(p, \tilde{s})| = \mathcal{O}(\frac{\epsilon^3}{z^3} d_G(p_i, \mathcal{A}))$. If there are multiple choices for p_i , a careful choice ensures that this error is upper bounded by $\epsilon/z \cdot (d_G(p, s) + d_G(p, \mathcal{A}))$.

Short-path case.

Construction of $\mathbb{C}_{\text{support}}$. For each point $p \in X$, we consider a disk $D(p, r)$ of radius $r = \Theta(z/\epsilon)$ around p . Finally, we consider the union of the area covered by all such disks. We subdivide this area into small gridcells of sidelength $\mathcal{O}(\epsilon^2/z^2)$, and select one representative point from each grid cell, and add it to the set $\mathbb{C}_{\text{support}}$.²

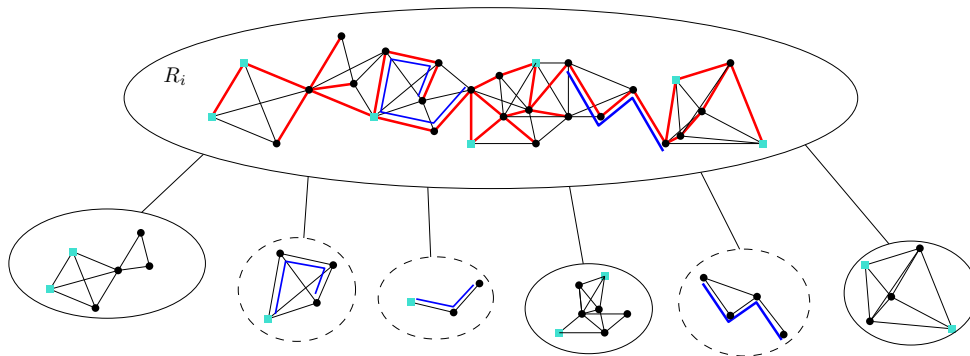
Finding a replacement center. For a center $s \in \mathcal{S}$, suppose we were not able to find a replacement using the previous case. Then, if \tilde{s} from the same cell as s of sidelength $\mathcal{O}(\epsilon^2/z^2)$ was added to $\mathbb{C}_{\text{support}}$, then we set $\rho(s) = \tilde{s}$, and add it to $\tilde{\mathcal{S}}$.

Error analysis. If a shortest path $\pi_G(p, s)$ is *short*, i.e., contains at most ℓ edges, where $\ell = \Theta(z/\epsilon)$, then due to the *Locally Euclidean* property, s belongs to the disk $D(p, r)$. It follows that we select some \tilde{s} in $\mathbb{C}_{\text{support}}$ such that $d_G(s, \tilde{s}) = \mathcal{O}(\epsilon^2/z^2)$. Again, using triangle inequality, it follows that for any $p \in X_s$, it holds that $|d_G(p, s) - d_G(p, \tilde{s})| = \mathcal{O}(\epsilon^2/z^2) \leq \mathcal{O}(\epsilon^2/z^2) \cdot d_G(p, \mathcal{A})$. Here, the second inequality follows since for any point $p \in X_s$, $d_G(p, \mathcal{A}) > 1$; otherwise the previous case would apply. Note that in this simplified argument, precision value of ϵ/z would also suffice; however, in the actual analysis we need the more granular value of ϵ^2/z^2 to ensure that the choices made in **nearby points** and **short path** cases do not affect each other.

Long-path case. This is the most involved case out of the three, and it is here that we rely on the *planar spanner* property. Before going into the details, let us first note the following property. For any $s \in \mathcal{S}$ whose replacement has not been found in the previous two cases, it holds that *for every* $p \in X$, $|\pi(p, s)| > \ell$. That is, the hop-distance of s from every point in X is strictly larger than ℓ . This observation will be crucial in the subsequent error analysis.

Construction of $\mathbb{C}_{\text{landmark}}$. Here, we first use the *Planar Spanner* property to obtain a recursive decomposition of the graph represented by a tree \mathcal{T} (see Figure 3 for an illustration). Each node of \mathcal{T} corresponds to a subset of vertices of G (called *region*). By slightly abusing the notation, we equate a node i with its corresponding region $R_i \subseteq V(G)$. The decomposition ensures that the union of all children regions of R_i is equal to R_i . This decomposition is obtained as follows. The root region R_1 is equal to $V(G)$. Now, consider a region R_i , and the corresponding induced subgraph $G_i = G[R_i]$. Due to *Planar*

² In the actual construction, we define a *support graph* to define the set $\mathbb{C}_{\text{support}}$. But, at a high level, the construction follows the overview given here.



■ **Figure 3** Decomposition of a region R_i into multiple children. Inside R_i , we show the induced subgraph $G_i = G[R_i]$ (which is not planar) and the corresponding planar spanner H_i . The edges of H_i are shown in red, and the edges in $E(G_i) \setminus E(H_i)$ are in black. The two shortest-path separators of H_i are shown in blue, which form a balanced separator \mathcal{P}_i for the vertices of X (shown as light blue squares). Then, we add a child for every connected component of $H \setminus V(\mathcal{P}_i)$ (children regions inside solid ellipses). We also have children corresponding to paths in \mathcal{P}_i (dashed ellipses). One of the paths is broken into two pieces due to a vertex in X .

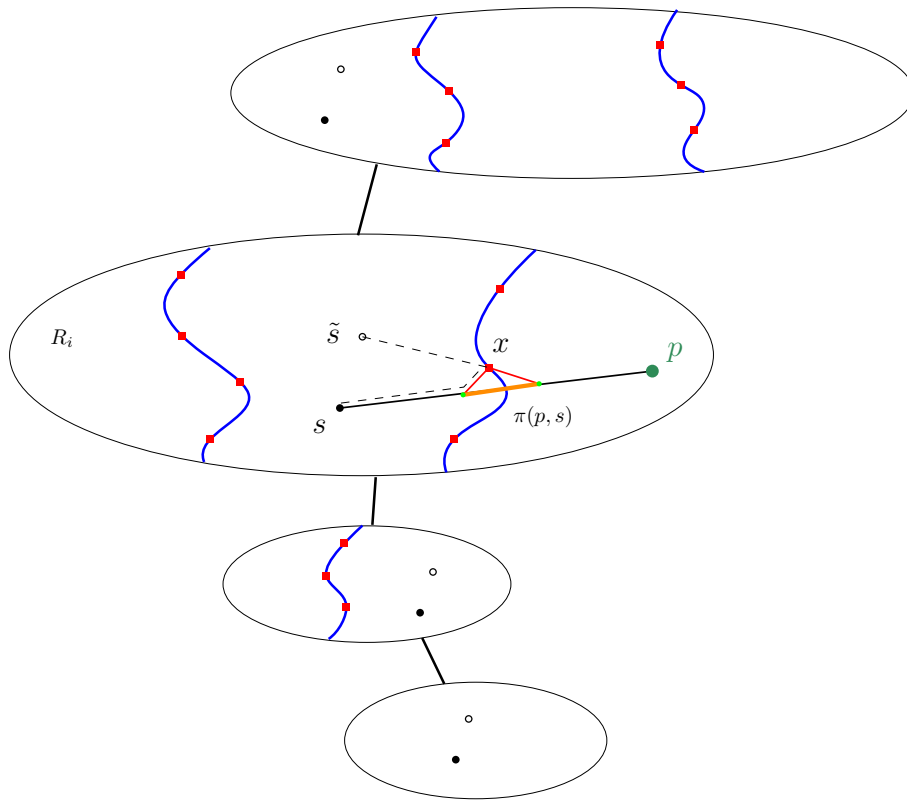
Spanner property, G_i admits a constant stretch planar spanner H_i . We then use the following well-known result.

► **Proposition 7** ([10, 30]). *Given an edge-weighted planar graph H' , with non-negative weights on vertices, there exists a collection of shortest paths $\mathcal{P} = \{P_1, P_2, \dots, P_b\}$ with $b = \mathcal{O}(1)$, such that the set of vertices of every connected component in $H \setminus \bigcup_{P_i \in \mathcal{P}} V(P_i)$ has weight at most half of that of $V(H')$.*

We define the weight of a vertex in H_i as 1 if it belongs to X , and 0 otherwise. Then, by applying Proposition 7, we obtain a collection of shortest paths \mathcal{P}_i in H_i such that each connected component of $H_i \setminus V(\mathcal{P}_i)$ contains at most a constant fraction of vertices of X . Now we add children to R_i in the tree \mathcal{T} , as follows: there is one child of R_i for a subset of R_i corresponding to (1) each connected component in $H_i \setminus V(\mathcal{P}_i)$, and (2) each shortest path $P^{i,j} \in \mathcal{P}_i$ (if a path $P^{i,j}$ contains vertices of X , then we break it at each such vertex and add the pieces as multiple children). We recursively subdivide the current R_i into multiple children in this manner as long as $|R_i \cap X| \geq 2$. Thus the height of the tree is $\mathcal{O}(\log |X|)$.

Now, consider a leaf node corresponding to a region R_t , and the corresponding root-leaf path $\Pi = (R_1, R_2, \dots, R_t)$, where R_1 is the root region. For each region R_i along the path, we have a set of $\mathcal{O}(1)$ shortest paths that are separators for the corresponding spanner H_i of $G_i = G[R_i]$. Thus, in total we have a collection of $\mathcal{O}(\log |X|)$ paths corresponding to a region R_t . Now, we select a set of vertices on each of these paths that are at evenly spaced distances, and add these vertices to a set \mathcal{L} of *landmarks*. We also add the vertices in $R_t \cap X$ to the landmark set \mathcal{L} . It can be shown that the overall size of \mathcal{L} is $\mathcal{O}_{\epsilon, z}(\log |X|)$. Now, for every vertex $s \in R_t$, we look at the distance vector obtained by looking at distances of s from each landmark in \mathcal{L} , and discretize each distance entry by rounding it. Thus, we can partition R_t into a bounded number of equivalence classes based on the *discretized distance vectors* to \mathcal{L} . For each leaf R_t , and each equivalence class of R_t , we add one representative point in $\mathbb{C}_{\text{landmark}}$.³

³ In the actual construction, we consider different spacings along each path $P^{i,j}$, which results in multiple



■ **Figure 4** Using landmarks to reroute paths. Here, we show a root-leaf path in \mathcal{T} , such that each region along the path contains s and its replacement \tilde{s} . This \tilde{s} is chosen such that distances of s and \tilde{s} has same discretized distances w.r.t. all landmarks in \mathcal{L} (shown as red squares), which are spaced evenly along the shortest path separators (shown in blue). Suppose R_i is the lowest region containing s and a point p , which means that a separator path in \mathcal{P}_i separates them. We reroute a single edge (shown in orange) on the shortest path $\pi(p, s)$ via a landmark x (rerouted path shown in red) using the edges of H_i , to obtain an approximate shortest path $\tilde{\pi}(p, s)$. Since s and \tilde{s} have approximately the same distance to x (dashed black paths), we can use this to show that $d_G(p, s) \approx d_G(p, \tilde{s})$.

Finding a replacement center. Consider a center $s \in \mathcal{S}$, and suppose it belongs to a leaf region R_t . By construction, we added a point $\tilde{s} \in R_t$ to $\mathbb{C}_{\text{landmark}}$, such that \tilde{s} and s have same discretized distance vector w.r.t. all landmarks in \mathcal{L} . We set $\rho(s) = \tilde{s}$, and add it to $\tilde{\mathcal{S}}$.

Error analysis. Now consider a point $p \in X_s$. If $p \in R_t$, then $p \in \mathcal{L}$. Therefore, s and \tilde{s} have same discretized distances w.r.t. p . This ensures that $d_G(p, s)$ and $d_G(p, \tilde{s})$ are within the required error bound.

Otherwise, $p \notin R_t$. This means that during the decomposition process, s and p must have been separated when we split a region $R_i \in \Pi$ into its multiple children. If $G_i = G[R_i]$ were planar, then we could conclude that the shortest path $\pi(p, s)$ must intersect some

collections of landmarks for each path. We then consider all possible choices of spacings for each of the $\mathcal{O}(\log |X|)$ paths $P^{i,j}$, which results in multiple landmark sets corresponding to each set of choices. For each landmark set thus obtained, we then define the equivalence classes of R_t in the manner described above, and add one representative of each class to the set $\mathbb{C}_{\text{landmark}}$. However, for the current overview, let us continue with this simplified (although inaccurate) construction.

shortest-path separator $P^{i,j} \in \mathcal{P}_i$. Unfortunately, in our case, G_i is not planar, and $P^{i,j}$ is not a separator for G_i , but for its planar spanner H_i . Here we recall the property that the hop-length of the shortest path $|\pi(p, s)|$ is strictly larger than ℓ , i.e., is $\Omega(z/\epsilon)$, which implies that the weight of a single edge is negligible as compared to the total length of the path. Then, we show that it is possible to reroute a single edge of the path $\pi(p, s)$ using the spanner H_i (which results in a constant factor increase in the distance, but *only for a single edge*) to obtain an *approximately shortest path* $\tilde{\pi}(p, s)$. Furthermore, $\tilde{\pi}(p, s)$ intersects a separator path $P^{i,j}$, and the intersecting vertex x is a landmark. This is the most intricate part of the argument, since we have to argue about shortest paths in G , G_i and H_i . Once we construct $\tilde{\pi}(p, s)$, we can use a subpath of $\tilde{\pi}(p, s)$ to go from p to x , and then use the fact that $d_G(s, x) \approx d_G(\tilde{s}, x)$, since x is a landmark, and s and \tilde{s} have same discretized distances w.r.t. x . Thus, we can show that $d_G(p, s)$ and $d_G(p, \tilde{s})$ are within the required bound. See Figure 4 for an illustration.

Note that the novelty of our work compared to previous works (Braverman et al. [8] and Cohen-Addad et al. [17]) lies in the ability of utilizing an underlying planar spanner instead of the original graph and still achieve a similar error bound sufficient for the analysis.

4 Conclusion

We obtain the first coresets for k -clustering problems whose size is independent of n , on a variety of geometric graph metrics, such as weighted intersection graphs of unit disks and squares. A UDG (or a USG) can contain arbitrarily large cliques, i.e., they can be (locally) dense. Therefore, to the best of our knowledge, ours is the first small-sized (i.e., independent of n) coreset construction for a shortest-path metric on a dense family of graphs. Due to the inherently “hybrid” nature of such metrics, our coreset construction has to carefully navigate the locally-Euclidean and globally-sparse nature of the metric.

We believe the contribution of our work is also conceptual, in that we “abstract out” the geometric properties of such metrics that are sufficient to obtain small-sized coresets via the versatile framework of Cohen-Addad et al. [18]. These structural properties are also satisfied by ℓ_p -norm weighted UDGs and USGs. Furthermore, by suitably modifying the construction, we can also handle hop metrics (i.e., unweighted edges) induced by UDGs of bounded degree. Thus, we obtain small-sized coresets, and thus FPT-approximation schemes, for k -clustering problems for all of these graph families. In order to obtain the result on USGs, we prove that these graphs admit a 3-stretch planar spanner, a result that may be of independent interest.

The most natural question is to find more graph families that satisfy the geometric properties (or some suitable modifications thereof) identified in this work. Disk graphs in \mathbb{R}^2 and Unit Ball Graphs in \mathbb{R}^d (for constant $d \geq 3$) are two orthogonal generalizations of UDGs, and thus may be the most obvious candidates. However, these graphs are not known to admit a constant stretch planar spanner. As an intermediate step, it might be interesting to consider “unit disk graphs” that are embedded on a surface Σ of bounded genus. Here, it might be more natural to require that such a graph admit constant stretch spanner that is also embeddable on Σ (which is a relaxation of planarity). It might be possible to extend our framework with this relaxed setting, also yielding smaller coresets for such geometric intersection graph families, which we leave as an interesting open question.

References

- 1 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k -means and euclidean k -median by primal-dual algorithms. In Chris Umans, editor, *58th*

- IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 61–72. IEEE Computer Society, 2017.
- 2 Sanjeev Arora, Prabhakar Raghavan, and Satish Rao. Approximation schemes for euclidean k -medians and related problems. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, STOC '98*, pages 106–113, New York, NY, USA, 1998. ACM. doi:10.1145/276698.276718.
 - 3 Daniel N. Baker, Vladimir Braverman, Lingxiao Huang, Shaofeng H.-C. Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in graphs of bounded treewidth. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 569–579. PMLR, 2020. URL: <http://proceedings.mlr.press/v119/baker20a.html>.
 - 4 Hari Balakrishnan, Christopher L Barrett, VS Anil Kumar, Madhav V Marathe, and Shripad Thite. The distance-2 matching problem and its relationship to the mac-layer capacity of ad hoc wireless networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1069–1079, 2004.
 - 5 Luca Becchetti, Marc Bury, Vincent Cohen-Addad, Fabrizio Grandoni, and Chris Schwiegelshohn. Oblivious dimension reduction for k -means: beyond subspaces and the johnson-lindenstrauss lemma. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1039–1050, 2019.
 - 6 Amariah Becker, Philip N Klein, and David Saulpic. Polynomial-time approximation schemes for k -center, k -median, and capacitated vehicle routing in bounded highway dimension. In *26th Annual European Symposium on Algorithms (ESA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
 - 7 Ahmad Biniaz. Plane hop spanners for unit disk graphs: Simpler and better. *Comput. Geom.*, 89:101622, 2020. doi:10.1016/j.comgeo.2020.101622.
 - 8 Vladimir Braverman, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in excluded-minor graphs and beyond. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2679–2696. SIAM, 2021.
 - 9 Jaroslaw Byrka, Thomas W. Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k -median and positive correlation in budgeted optimization. *ACM Trans. Algorithms*, 13(2):23:1–23:31, 2017.
 - 10 Timothy M. Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *J. Comput. Geom.*, 10(2):3–20, 2019. doi:10.20382/jocg.v10i2a2.
 - 11 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k -median problem (extended abstract). In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 1–10, 1999.
 - 12 Moses Charikar and Shi Li. A dependent lp-rounding approach for the k -median problem. In Artur Czumaj, Kurt Mehlhorn, Andrew M. Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part I*, volume 7391 of *Lecture Notes in Computer Science*, pages 194–205. Springer, 2012.
 - 13 Ke Chen. On coresets for k -median and k -means clustering in metric and euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009.
 - 14 Vincent Cohen-Addad, Andreas Emil Feldmann, and David Saulpic. Near-linear time approximation schemes for clustering in doubling metrics. *Journal of the ACM (JACM)*, 68(6):1–34, 2021.
 - 15 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight FPT approximations for k -median and k -means. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 42:1–42:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.42.

- 16 Vincent Cohen-Addad, Philip N. Klein, and Claire Mathieu. Local search yields approximation schemes for k-means and k-median in euclidean and minor-free metrics. *SIAM J. Comput.*, 48(2):644–667, 2019. doi:10.1137/17M112717X.
- 17 Vincent Cohen-Addad, Kasper Green Larsen, David Saulpic, and Chris Schwiegelshohn. Towards optimal lower bounds for k-median and k-means coresets. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1038–1051. ACM, 2022. doi:10.1145/3519935.3519946.
- 18 Vincent Cohen-Addad, David Saulpic, and Chris Schwiegelshohn. A new coreset framework for clustering. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 169–182. ACM, 2021. doi:10.1145/3406325.3451022.
- 19 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578. ACM, 2011. doi:10.1145/1993636.1993712.
- 20 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 569–578, 2011.
- 21 Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020.
- 22 Gereon Frahling and Christian Sohler. Coresets in dynamic geometric data streams. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*, pages 209–217, 2005.
- 23 Zachary Friggstad, Mohsen Rezapour, and Mohammad R. Salavatipour. Local search yields a PTAS for k-means in doubling metrics. *SIAM J. Comput.*, 48(2):452–480, 2019. doi:10.1137/17M1127181.
- 24 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM Journal on Computing*, 35(1):151–169, 2005.
- 25 Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. *Discret. Comput. Geom.*, 37(1):3–19, 2007.
- 26 Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 291–300. ACM, 2004.
- 27 Lingxiao Huang, Shaofeng H-C Jiang, Jian Li, and Xuan Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 814–825. IEEE, 2018.
- 28 Lingxiao Huang and Nisheeth K. Vishnoi. Coresets for clustering in euclidean spaces: importance sampling is nearly optimal. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1416–1429. ACM, 2020. doi:10.1145/3357713.3384296.
- 29 Kamal Jain and Vijay V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *J. ACM*, 48(2):274–296, 2001.
- 30 Ken-ichi Kawarabayashi, Christian Sommer, and Mikkel Thorup. More compact oracles for approximate distances in undirected planar graphs. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 550–563. SIAM, 2013. doi:10.1137/1.9781611973105.40.

10:16 Coresets for Clustering in Geometric Intersection Graphs

- 31 Fabian Kuhn, Tim Nieberg, Thomas Moscibroda, and Rogert Wattenhofer. Local approximation schemes for ad hoc and sensor networks. In *Proceedings of the 2005 joint workshop on Foundations of mobile computing*, pages 97–103, 2005.
- 32 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. Linear-time approximation schemes for clustering problems in any dimensions. *J. ACM*, 57(2):5:1–5:32, 2010.
- 33 Michael Langberg and Leonard J Schulman. Universal ε -approximators for integrals. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 598–607. SIAM, 2010.
- 34 Emmanuelle Lebhar and Zvi Lotker. Unit disk graph and physical interference model: Putting pieces together. In *2009 IEEE International Symposium on Parallel & Distributed Processing*, pages 1–8. IEEE, 2009.
- 35 Shi Li and Ola Svensson. Approximating k-median via pseudo-approximation. *SIAM J. Comput.*, 45(2):530–547, 2016.
- 36 Xiang-Yang Li, Gruia Calinescu, and Peng-Jun Wan. Distributed construction of a planar spanner and routing for ad hoc wireless networks. In *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 1268–1277. IEEE, 2002.
- 37 Xiang-Yang Li, Wen-Zhan Song, and Yu Wang. Efficient topology control for ad-hoc wireless networks with non-uniform transmission ranges. *Wireless Networks*, 11(3):255–264, 2005.
- 38 Frank Schulz. Modeling sensor and ad hoc networks. In *Algorithms for Sensor and Ad Hoc Networks*, pages 21–36. Springer, 2007.
- 39 Amin Shahraki, Amir Taherkordi, Øystein Haugen, and Frank Eliassen. Clustering objectives in wireless sensor networks: A survey and research direction analysis. *Computer Networks*, 180:107376, 2020.
- 40 Christian Sohler and David P Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 802–813. IEEE, 2018.

Minimum-Membership Geometric Set Cover, Revisited

Sayan Bandyapadhyay ✉

Portland State University, OR, USA

William Lochet ✉

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Saket Saurabh ✉

Institute of Mathematical Sciences, Chennai, India

Jie Xue ✉ 

New York University Shanghai, China

Abstract

We revisit a natural variant of the geometric set cover problem, called *minimum-membership geometric set cover* (MMGSC). In this problem, the input consists of a set S of points and a set \mathcal{R} of geometric objects, and the goal is to find a subset $\mathcal{R}^* \subseteq \mathcal{R}$ to cover all points in S such that the *membership* of S with respect to \mathcal{R}^* , denoted by $\text{memb}(S, \mathcal{R}^*)$, is minimized, where $\text{memb}(S, \mathcal{R}^*) = \max_{p \in S} |\{R \in \mathcal{R}^* : p \in R\}|$. We give the first polynomial-time approximation algorithms for MMGSC in \mathbb{R}^2 . Specifically, we achieve the following two main results.

- We give the first polynomial-time constant-approximation algorithm for MMGSC with unit squares. This answers a question left open since the work of Erlebach and Leeuwen [SODA'08], who gave a constant-approximation algorithm with running time $n^{O(\text{opt})}$ where opt is the optimum of the problem (i.e., the minimum membership).
- We give the first polynomial-time approximation scheme (PTAS) for MMGSC with halfplanes. Prior to this work, it was even unknown whether the problem can be approximated with a factor of $o(\log n)$ in polynomial time, while it is well-known that the minimum-size set cover problem with halfplanes can be solved in polynomial time.

We also consider a problem closely related to MMGSC, called *minimum-ply geometric set cover* (MPGSC), in which the goal is to find $\mathcal{R}^* \subseteq \mathcal{R}$ to cover S such that the *ply* of \mathcal{R}^* is minimized, where the *ply* is defined as the maximum number of objects in \mathcal{R}^* which have a nonempty common intersection. Very recently, Durocher et al. gave the first constant-approximation algorithm for MPGSC with unit squares which runs in $O(n^{12})$ time. We give a significantly simpler constant-approximation algorithm with near-linear running time.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases geometric set cover, geometric optimization, approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.11

Related Version *Full Version*: <https://arxiv.org/abs/2305.03985>

Funding *Saket Saurabh*: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819416), and Swarnajayanti Fellowship (No. DST/SJF/MSA01/2017-18).

Acknowledgements The authors would like to thank Qizheng He, Daniel Lokshtanov, Rahul Saladi, Subhash Suri, and Haitao Wang for helpful discussions about the problems, and thank the anonymous reviewers for their detailed comments, which help significantly improve the writing of the paper.



© Sayan Bandyapadhyay, William Lochet, Saket Saurabh, and Jie Xue; licensed under Creative Commons License CC-BY 4.0

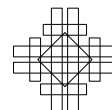
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 11; pp. 11:1–11:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Geometric set cover is one of the most fundamental problems in computational geometry. In the problem, we are given a set S of points and a set \mathcal{R} of geometric objects, and our goal is to cover all the points in S using fewest objects in \mathcal{R} . Motivated by applications, several variants of the geometric set cover problem have been studied in literature. In this paper, we study a natural variant of the geometric set cover problem, called *minimum-membership geometric set cover* (MMGSC).

In the MMGSC problem, the input also consists of a set S of points and a set \mathcal{R} of geometric objects. Similar to the geometric set cover problem our goal is still to cover all the points in S using the objects in \mathcal{R} . However, we do not care about how many geometric objects we use. Instead, we want to guarantee that any point in S is not “over covered”. More precisely, the goal is to find a subset $\mathcal{R}^* \subseteq \mathcal{R}$ to cover all points in S such that the *membership* of S with respect to \mathcal{R}^* , denoted by $\text{memb}(S, \mathcal{R}^*)$, is minimized, where $\text{memb}(S, \mathcal{R}^*) = \max_{p \in S} |\{R \in \mathcal{R}^* : p \in R\}|$.

Kuhn et al. [7], motivated by applications in cellular networks, had introduced the non-geometric version of the MMGSC problem, say *minimum-membership set cover* (MMSC). That is, S is an arbitrary universe with n elements and \mathcal{R} is a collection of subsets of S . They showed that the MMSC problem admits an $O(\log n)$ -approximation algorithm, where $n = |S|$. Furthermore, they complimented the upper bound result by showing, that unless $P=NP$, the problem cannot be approximated, in polynomial time, by a ratio less than $\ln n$. Erlebach and van Leeuwen [6], in their seminal work on geometric coverage problem, considered the geometric version of MMSC, namely MMGSC, from the view of approximation algorithms. They showed NP-hardness for approximating the problem with ratio less than 2 on unit disks and unit squares, and gave a 5-approximation algorithm for unit squares provided that the optimal objective value is bounded by a constant. More precisely, their algorithm runs in time $n^{O(\text{opt})}$ where opt is the optimum of the problem (i.e., the minimum membership). It has remained open that whether MMGSC with unit squares admits a (truly) polynomial-time constant-approximation algorithm.

As our first result, we settle this open question by giving a polynomial-time algorithm for MMGSC with unit squares which achieves a constant approximation ratio. In fact, our algorithm works for a generalized version of the problem, in which the point set to be covered can be different from the point set whose membership is considered.

► **Definition 1** (generalized MMGSC). *In the generalized minimum-membership geometric set cover (MMGSC) problem, the input consists of two sets S, S' of points in \mathbb{R}^d and a set \mathcal{R} of geometric objects in \mathbb{R}^d , and the goal is to find a subset $\mathcal{R}^* \subseteq \mathcal{R}$ to cover all points in S such that $\text{memb}(S', \mathcal{R}^*)$ is minimized. We denote by $\text{opt}(S, S', \mathcal{R})$ the optimum of the problem instance (S, S', \mathcal{R}) , i.e., $\text{opt}(S, S', \mathcal{R}) = \text{memb}(S', \mathcal{R}^*)$ where $\mathcal{R}^* \subseteq \mathcal{R}$ is an optimal solution.*

► **Theorem 2.** *The generalized MMGSC problem with unit squares admits a polynomial-time constant-approximation algorithm.*

As our second result, we gave the first polynomial-time approximation scheme (PTAS) for MMGSC with halfplanes. Prior to this work, it was even unknown whether the problem can be approximated in polynomial time with a factor of $o(\log n)$, while the minimum-size set cover problem with halfplanes can be solved in polynomial time. Again, our PTAS works for the generalized version.

► **Theorem 3.** *The generalized MMGSC problem with halfplanes admits a PTAS.*

The generalized version of MMGSC is interesting because it also generalizes another closely related problem studied in the literature, called *minimum-ply geometric set cover* (MPGSC). The MPGSC problem was introduced by Biedl, Biniaz and Lubiw [3] as a variant of MMGSC. They observed that in some applications, e.g. interference reduction in cellular networks, it is desirable to minimize the membership of every point in the plane, not only points of S . Therefore, in MPGSC, the goal is to find $\mathcal{R}^* \subseteq \mathcal{R}$ to cover S such that the *ply* of \mathcal{R}^* is minimized, where the ply is defined as the maximum number of objects in \mathcal{R}^* which have a nonempty common intersection. Observe that MPGSC is a special case of the generalized MMGSC (by letting S' include a point in every face of the arrangement induced by \mathcal{R}). As such, Theorems 2 and 3 both apply to MPGSC.

Prior to our work, Biedl, Biniaz and Lubiw [3] showed that solving the MPGSC with a set of axis-parallel unit squares is NP-hard, and gave a polynomial-time 2-approximation algorithm for instances in which the optimum (i.e., the minimum ply) is a constant. Very recently, Durocher, Keil and Mondal [5] gave the first constant-approximation algorithm for MPGSC with unit squares, which runs in $O(n^{12})$ time. This algorithm does not extend to other related settings, such as similarly sized squares or unit disks. Our algorithm derived from Theorem 2 is already much more efficient than the one of [5] (while also not extending to similarly sized squares or unit disks). However, we observe that for (only) MPGSC with unit squares, there exists a very simple constant-approximation algorithm which runs in $\tilde{O}(n)$ time; here \tilde{O} hides logarithmic factors. This simple algorithm directly extends to any similarly sized fat objects for which a constant-approximation solution for *minimum-size set cover* can be computed in polynomial time. Therefore, we obtain the following result.

► **Theorem 4.** *The MPGSC problem with unit (or similarly sized) squares/disks admits constant-approximation algorithms with running time $\tilde{O}(n)$.*

A common ingredient appearing in all of our results is to establish connections between MMGSC (or MPGSC) and the standard minimum-size geometric set cover. We show that in certain situations, a minimum-size set cover (satisfying certain conditions) can be a good approximation in terms of MMGSC. This reveals the underlying relations between different variants of geometric set cover problems, and might be of independent interest.

Other related work

Very recently, Mitchell and Pandit [8] proved that MMGSC with rectangles intersecting a horizontal line or anchored on two horizontal lines is NP-hard (among other algorithmic and hardness results).

Organization

The rest of the paper is organized as follows. In Section 2, we present our result for MMGSC with unit squares. In Section 3, we present our result for MMGSC with halfplanes. The result for MPGSC is given in Section 4. Due to the limited space, some (less important) proofs are omitted and can be found in the full version of the paper.

2 Constant approximation for unit squares

Let S, S' be two sets of points in \mathbb{R}^2 and \mathcal{Q} be a set of (axis-parallel) unit squares. We want to solve the generalized MMGSC instance (S, S', \mathcal{Q}) .

2.1 Restricting S to a grid cell

First of all, we construct a grid Γ consisting of square cells of side-length 1. For each grid cell \square , we write $S_\square = S \cap \square$ and $\mathcal{Q}_\square = \{Q \in \mathcal{Q} : Q \cap \square \neq \emptyset\}$.

► **Lemma 5.** *Suppose that, for every $\square \in \Gamma$, $\mathcal{Q}_\square^* \subseteq \mathcal{Q}_\square$ is a c -approximation solution of the generalized MMGSC instance $(S_\square, S', \mathcal{Q}_\square)$. Then $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ is an $O(c)$ -approximation solution of the instance (S, S', \mathcal{Q}) .*

Proof. First notice that $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ is a set cover of S , because any point $p \in S$ is contained in a grid cell \square and thus \mathcal{Q}_\square^* covers p . Then we show that for any point $p' \in S'$, the number of unit squares in $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ containing p' is at most $9c \cdot \text{opt}(S, S', \mathcal{Q})$. Suppose the grid cell containing p' is \square' . Note that a unit square $Q \in \bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ contains p' only if $Q \in \mathcal{Q}_\square^*$ for a grid cell \square that is either \square' or one of the eight grid cells around \square' . For each such cell \square , the number of unit squares in \mathcal{Q}_\square^* containing p' is at most $c \cdot \text{opt}(S_\square, S', \mathcal{Q}_\square)$, since \mathcal{Q}_\square^* is a c -approximation solution of $(S_\square, S', \mathcal{Q}_\square)$. It is clear that $\text{opt}(S_\square, S', \mathcal{Q}_\square) \leq \text{opt}(S, S', \mathcal{Q})$. Therefore, there can be at most $9c \cdot \text{opt}(S, S', \mathcal{Q})$ unit squares in $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ containing p' , which implies that $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ is a $9c$ -approximation solution of (S, S', \mathcal{Q}) . ◀

2.2 Partition the instance using LP

Based on the previous discussion, we will now assume that S is contained in a grid cell \square and all unit squares in \mathcal{Q} intersect \square . Note that the points in S' can be everywhere in the plane. We shall formulate an LP relaxation of the generalized MMGSC instance (S, S', \mathcal{Q}) . To this end, we first introduce the notion of fractional set cover. A *fractional set cover* of a set A of points is a set $\{x_B\}_{B \in \mathcal{B}}$ of numbers in $[0, 1]$ indexed by a collection \mathcal{B} of geometric ranges such that $\sum_{B \in \mathcal{B}, a \in B} x_B \geq 1$ for all $a \in A$. For another set A' of points, we can define the *membership* of A' with respect to this fractional set cover $\{x_B\}_{B \in \mathcal{B}}$ as $\text{memb}(A', \{x_B\}_{B \in \mathcal{B}}) = \max_{a' \in A'} \sum_{B \in \mathcal{B}, a' \in B} x_B$. The LP relaxation of the instance (S, S', \mathcal{Q}) simply asks for a fractional set cover of S using the unit squares in \mathcal{Q} that minimizes the membership of S' with respect to it. Specifically, for each unit square $Q \in \mathcal{Q}$, we create a variable x_Q . In addition, we create another variable y , which indicates the upper bound for the membership of S with respect to $\{x_Q\}_{Q \in \mathcal{Q}}$. We consider the following linear program.

$$\begin{aligned} & \min y \\ \text{s.t.} \quad & 0 \leq x_Q \leq 1 \text{ for all } Q \in \mathcal{Q}, \\ & \sum_{Q \in \mathcal{Q}, p \in Q} x_Q \geq 1 \text{ for all } p \in S, \\ & \sum_{Q \in \mathcal{Q}, p' \in Q} x_Q \leq y \text{ for all } p' \in S'. \end{aligned}$$

We compute an optimal solution $(\{x_Q^*\}_{Q \in \mathcal{Q}}, y^*)$ of the above linear program using a polynomial-time LP solver. We have the following observation about the solution.

► **Fact 6.** $y^* \leq \text{opt}(S, S', \mathcal{Q})$.

Proof. Let $\mathcal{Q}^* \subseteq \mathcal{Q}$ be an optimal solution. We have $S \subseteq \bigcup_{Q \in \mathcal{Q}^*} Q$ and $\text{memb}(S', \mathcal{Q}^*) = \text{opt}(S, S', \mathcal{Q})$. Set $x_Q = 1$ for $Q \in \mathcal{Q}^*$, $x_Q = 0$ for $Q \notin \mathcal{Q}^*$, and $y = \text{memb}(S', \mathcal{Q}^*)$. These values satisfy the LP constraints. Therefore, $y^* \leq y = \text{opt}(S, S', \mathcal{Q})$. ◀

Next, we shall partition the instance (S, S', \mathcal{Q}) into four sub-instances according to the LP solution $(\{x_Q^*\}_{Q \in \mathcal{Q}}, y^*)$. Recall that all points in S are inside the grid cell \square and all unit squares in \mathcal{Q} intersect \square . Let c_1, c_2, c_3, c_4 be the four corners of \square . We can partition \mathcal{Q} into

$\mathcal{Q}_1, \mathcal{Q}_2, \mathcal{Q}_3, \mathcal{Q}_4$, where \mathcal{Q}_i consists of the unit squares containing c_i for $i \in \{1, 2, 3, 4\}$. Also, we partition S into S_1, S_2, S_3, S_4 in the following way. For a point $p \in \mathbb{R}^2$ and $i \in \{1, 2, 3, 4\}$, define $\delta_{p,i}$ as the sum of x_Q^* for all $Q \in \mathcal{Q}_i$ satisfying $p \in Q$. Then we assign each point $p \in S$ to S_i , where $i \in \{1, 2, 3, 4\}$ is the index that maximizes $\delta_{p,i}$. Observe the following fact.

► **Fact 7.** For each $i \in \{1, 2, 3, 4\}$, we have $\sum_{Q \in \mathcal{Q}_i, p \in Q} x_Q^* \geq \frac{1}{4}$ for all $p \in S_i$.

Proof. We have $\sum_{Q \in \mathcal{Q}_i, p \in Q} x_Q^* = \delta_{p,i}$ and $\sum_{i=1}^4 \delta_{p,i} = \sum_{Q \in \mathcal{Q}, p \in Q} x_Q^* \geq 1$, because of the LP constraints. Furthermore, $\delta_{p,i} \geq \delta_{p,j}$ for all $j \in \{1, 2, 3, 4\}$, as $p \in S_i$. Thus, $\delta_{p,i} \geq \frac{1}{4}$. ◀

We now partition the original instance into $(S_1, S', \mathcal{Q}_1), \dots, (S_4, S', \mathcal{Q}_4)$. Consider an index $i \in \{1, 2, 3, 4\}$. If we define $\tilde{x}_Q^* = 4x_Q^*$ for all $Q \in \mathcal{Q}_i$, then the above fact implies $\sum_{Q \in \mathcal{Q}_i, p \in Q} \tilde{x}_Q^* \geq 1$ for all $p \in S_i$. In other words, $\{\tilde{x}_Q^*\}_{Q \in \mathcal{Q}_i}$ is a fractional set cover of S_i . Note that $\text{memb}(S', \{\tilde{x}_Q^*\}_{Q \in \mathcal{Q}_i}) \leq 4y^*$, because

$$4y^* \geq \sum_{Q \in \mathcal{Q}, p' \in Q} 4x_Q^* \geq \sum_{Q \in \mathcal{Q}_i, p' \in Q} \tilde{x}_Q^*$$

for all $p' \in S$, due to the constraints of the LP. With this observation, it now suffices to compute a solution for each instance (S_i, S', \mathcal{Q}_i) that is a constant-factor approximation even with respect to the fractional solutions. The union of these solutions is a set cover of $S = \bigcup_{i=1}^4 S_i$, the membership of S' with respect to it is $O(y^*)$. A nice property of the instances (S_i, S', \mathcal{Q}_i) is that all unit squares in \mathcal{Q}_i contain the same corner c_i of \square . In the next section, we show how to compute the desired approximation solution for such instances.

2.3 The one-corner case

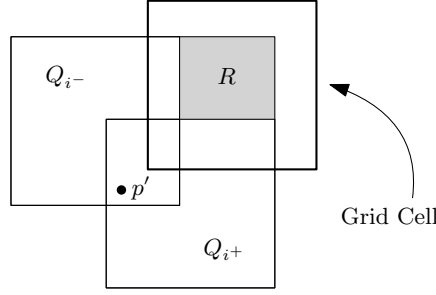
Now consider an instance (S, S', \mathcal{Q}) , where all points in S lie in a grid cell \square and all unit squares contain the same corner (say the bottom-left corner) of \square . For a point $p \in \mathbb{R}^2$, denote by $x(p)$ and $y(p)$ the x -coordinate and y -coordinate of p , respectively. Also, for a unit square $Q \in \mathcal{Q}$, denote by $x(Q)$ and $y(Q)$ the x -coordinate and y -coordinate of the top-right corner of Q , respectively. We make two simple observations. The first one shows that the integral gap of the minimum-size set cover problem in this setting is equal to 1 (the proof is omitted and can be found in the full paper). The second one gives a useful geometric property for unit squares containing the same corner of \square .

► **Fact 8.** Let $S_0 \subseteq S$ be a subset and $\mathcal{Q}_0 \subseteq \mathcal{Q}$ be a minimum-size set cover of S_0 . For any fractional set cover $\{\hat{x}_Q\}_{Q \in \mathcal{Q}}$ of S_0 , we have $\sum_{Q \in \mathcal{Q}_0} \hat{x}_Q \geq |\mathcal{Q}_0|$.

► **Fact 9.** Let Q^-, Q, Q^+ be three unit squares all containing the bottom-left corner of \square which satisfy $x(Q^-) \leq x(Q) \leq x(Q^+)$ and $y(Q^-) \geq y(Q) \geq y(Q^+)$. Then $Q^- \cap Q^+ \subseteq Q$.

Proof. Let $p \in Q^- \cap Q^+$. The fact $p \in Q^-$ implies $x(p) \leq x(Q^-)$ and $y(Q^-) - 1 \leq y(p)$. So we have $x(p) \leq x(Q)$ and $y(Q) - 1 \leq y(p)$. On the other hand, the fact $p \in Q^+$ implies $x(Q^+) - 1 \leq x(p)$ and $y(p) \leq y(Q^+)$. So we have $x(Q) - 1 \leq x(p)$ and $y(p) \leq y(Q)$. Therefore, $x(Q) - 1 \leq x(p) \leq x(Q)$ and $y(Q) - 1 \leq y(p) \leq y(Q)$, which implies that $p \in Q$. ◀

We say a unit square $Q \in \mathcal{Q}$ is *dominated* by another unit square $Q' \in \mathcal{Q}$ if $Q \cap \square \subseteq Q' \cap \square$. A unit square in \mathcal{Q} is *maximal* if it is not dominated by any other unit squares in \mathcal{Q} . We denote by $\mathcal{Q}_{\max} \subseteq \mathcal{Q}$ the set of maximal unit squares in \mathcal{Q} . The following lemma shows that any minimum-size set cover of S that only uses the unit squares in \mathcal{Q}_{\max} is also a good approximation for the minimum-membership set cover.



■ **Figure 1** Illustrating the rectangle R .

► **Lemma 10.** *Let $\mathcal{Q}^* \subseteq \mathcal{Q}$ be a minimum-size set cover of S such that $\mathcal{Q}^* \subseteq \mathcal{Q}_{\max}$, and $\{x_Q\}_{Q \in \mathcal{Q}}$ be a fractional set cover of S . Then $\text{memb}(S', \mathcal{Q}^*) \leq \text{memb}(S', \{x_Q\}_{Q \in \mathcal{Q}}) + 2$.*

Proof. Suppose $\mathcal{Q}^* = \{Q_1, \dots, Q_r\}$ where $x(Q_1) \leq \dots \leq x(Q_r)$. As $\mathcal{Q}^* \subseteq \mathcal{Q}_{\max}$, we must have $x(Q_1) < \dots < x(Q_r)$ and $y(Q_1) > \dots > y(Q_r)$. Consider a point $p' \in S'$. Let $i^- \in [r]$ (resp., $i^+ \in [r]$) be the smallest (resp., largest) index such that $p' \in Q_{i^-}$ (resp., $p' \in Q_{i^+}$). By Fact 9, we have $p' \in Q_{i^-} \cap Q_{i^+} \subseteq Q_i$ for all $i \in \{i^-, \dots, i^+\}$ and thus $|\{Q \in \mathcal{Q}^* : p' \in Q\}| = i^+ - i^- + 1$. It suffices to show that $\sum_{Q \in \mathcal{Q}, p' \in Q} x_Q \geq i^+ - i^- - 1$.

Consider the rectangle $R = (x(Q_{i^-}), x(Q_{i^+})) \times (y(Q_{i^+}), y(Q_{i^-}))$; see Figure 1. Set $S_0 = S \cap R$ and $\mathcal{Q}_0 = \{Q_{i^-+1}, \dots, Q_{i^+-1}\}$. Observe that \mathcal{Q}_0 covers S_0 , since no unit square in $\mathcal{Q}^* \setminus \mathcal{Q}_0$ contains any point in S_0 . We claim that $\mathcal{Q}_0 \subseteq \mathcal{Q}$ is a minimum-size set cover of S_0 . Indeed, since $x(Q_{i^-}) < \dots < x(Q_{i^+})$ and $y(Q_{i^-}) > \dots > y(Q_{i^+})$, the points in $S \setminus S_0$ are all covered by the unit squares Q_1, \dots, Q_{i^-} and Q_{i^+}, \dots, Q_r . If there exists a set cover $\mathcal{Q}'_0 \subseteq \mathcal{Q}$ of S_0 such that $|\mathcal{Q}'_0| < |\mathcal{Q}_0|$, then \mathcal{Q}'_0 together with $Q_1, \dots, Q_{i^-}, Q_{i^+}, \dots, Q_r$ form a set cover of S whose size is smaller than \mathcal{Q}^* , contradicting with the fact that \mathcal{Q}^* is a minimum-size set cover of S . Therefore, $\mathcal{Q}_0 \subseteq \mathcal{Q}$ is a minimum-size set cover of S_0 . Now for each $Q \in \mathcal{Q}$, define $\hat{x}_Q = x_Q$ if $Q \cap R \neq \emptyset$ and $\hat{x}_Q = 0$ if $Q \cap R = \emptyset$. As $\{x_Q\}_{Q \in \mathcal{Q}}$ is a fractional set cover of S , for each $p \in S_0$, we have $\sum_{Q \in \mathcal{Q}, p \in Q} x_Q \geq 1$, which implies $\sum_{Q \in \mathcal{Q}, p \in Q} \hat{x}_Q \geq 1$ because $p \in R$ and thus $\hat{x}_Q = x_Q$ for all $Q \in \mathcal{Q}$ such that $p \in Q$. So $\{\hat{x}_Q\}_{Q \in \mathcal{Q}}$ is a fractional set cover of S_0 . By Fact 8, we then have

$$\sum_{Q \in \mathcal{Q}} \hat{x}_Q \geq |\mathcal{Q}_0| = i^+ - i^- - 1. \quad (1)$$

Next, we observe that $x(Q_{i^-}) \leq x(Q) \leq x(Q_{i^+})$ and $y(Q_{i^-}) \geq y(Q) \geq y(Q_{i^+})$ for any unit square $Q \in \mathcal{Q}$ such that $Q \cap R \neq \emptyset$. Let $Q \in \mathcal{Q}$ and assume $Q \cap R \neq \emptyset$. The inequalities $x(Q_{i^-}) \leq x(Q)$ and $y(Q) \geq y(Q_{i^+})$ follow directly from the fact $Q \cap R \neq \emptyset$. If $x(Q) > x(Q_{i^+})$, then Q dominates Q_{i^+} , contradicting the fact $Q_{i^+} \in \mathcal{Q}_{\max}$. Similarly, if $y(Q_{i^-}) < y(Q)$, then Q dominates Q_{i^-} , contradicting the fact $Q_{i^-} \in \mathcal{Q}_{\max}$. Thus, $x(Q_{i^-}) \leq x(Q)$ and $y(Q_{i^-}) \geq y(Q)$. By Fact 9, we have $p' \in Q_{i^-} \cap Q_{i^+} \subseteq Q$ for all $Q \in \mathcal{Q}$ such that $Q \cap R \neq \emptyset$. Thus, $\hat{x}_Q = 0$ for all $Q \in \mathcal{Q}$ such that $p' \notin Q$, which implies

$$\sum_{Q \in \mathcal{Q}, p' \in Q} x_Q \geq \sum_{Q \in \mathcal{Q}, p' \in Q} \hat{x}_Q = \sum_{Q \in \mathcal{Q}} \hat{x}_Q. \quad (2)$$

Combining Equations 1 and 2, we have $\sum_{Q \in \mathcal{Q}, p' \in Q} x_Q \geq i^+ - i^- - 1$. ◀

Using the above lemma, now it suffices to compute a minimum-size set cover of S using the unit squares in \mathcal{Q}_{\max} . It is well-known that in this setting, the minimum-size set

cover problem can be solved in polynomial time (or even near-linear time) using a greedy algorithm, because the unit squares in \mathcal{Q} are in fact equivalent to southwest quadrants; see for example [1]. Thus, we can compute in polynomial time a set cover $\mathcal{Q}^* \subseteq \mathcal{Q}$ of S such that $\text{memb}(S', \mathcal{Q}^*) \leq \text{memb}(S', \{x_Q\}_{Q \in \mathcal{Q}}) + 2$ for any fractional set cover $\{x_Q\}_{Q \in \mathcal{Q}}$ of S .

2.4 Putting everything together

Recall that at the end of Section 2.2, we have four generalized MMGSC instances $(S_1, S', \mathcal{Q}_1), \dots, (S_4, S', \mathcal{Q}_4)$. Also, for each $i \in \{1, \dots, 4\}$, we have a fractional set cover $\{\tilde{x}_Q^*\}_{Q \in \mathcal{Q}_i}$ of S_i such that $\text{memb}(S', \{\tilde{x}_Q^*\}_{Q \in \mathcal{Q}_i}) \leq 4y^* \leq 4 \cdot \text{opt}(S, S', \mathcal{Q})$. By the discussion in Section 2.3, for each $i \in \{1, \dots, 4\}$, we can compute in polynomial time a set cover $\mathcal{Q}_i^* \subseteq \mathcal{Q}_i$ of S_i satisfying that $\text{memb}(S', \mathcal{Q}_i^*) \leq \text{memb}(S', \{\tilde{x}_Q^*\}_{Q \in \mathcal{Q}_i}) + 2$. Set $\mathcal{Q}^* = \bigcup_{i=1}^4 \mathcal{Q}_i^*$. As $S = \bigcup_{i=1}^4 S_i$, \mathcal{Q}^* is a set cover of S . Furthermore, we have

$$\begin{aligned} \text{memb}(S', \mathcal{Q}^*) &\leq \sum_{i=1}^4 \text{memb}(S', \mathcal{Q}_i^*) \\ &\leq \sum_{i=1}^4 \text{memb}(S', \{\tilde{x}_Q^*\}_{Q \in \mathcal{Q}_i}) + 8 \\ &\leq 16y^* + 8 \\ &\leq 16 \cdot \text{opt}(S, S', \mathcal{Q}) + 8. \end{aligned}$$

If $\text{opt}(S, S', \mathcal{Q}) > 0$, then \mathcal{Q}^* is a constant-approximation solution. The case $\text{opt}(S, S', \mathcal{Q}) = 0$ can be easily solved by picking all unit squares in \mathcal{Q} that do not contain any points in S' . Therefore, we obtain a constant-approximation algorithm for the case where S is contained in a grid cell. Further combining this with Lemma 5, we conclude the following.

► **Theorem 2.** *The generalized MMGSC problem with unit squares admits a polynomial-time constant-approximation algorithm.*

3 Polynomial-time approximation scheme for halfplanes

Let S, S' be two sets of points in \mathbb{R}^2 and \mathcal{H} be a set of halfplanes. We want to solve the generalized MMGSC instance (S, S', \mathcal{H}) . Set $n = |S| + |S'| + |\mathcal{H}|$.

In order to describe our algorithm, we first need to introduce some basic notions about halfplanes. The *normal vector* (or *normal* for short) of a halfplane H is the unit vector perpendicular to the bounding line of H whose direction is to the interior of H , that is, if the equation of H is $ax + by + c \geq 0$ where $a^2 + b^2 = 1$, then its normal is $\vec{v} = (a, b)$. For two nonzero vectors \vec{u} and \vec{v} in the plane, we denote by $\text{ang}(\vec{u}, \vec{v})$ the *clockwise ordered angle* from \vec{u} to \vec{v} , i.e., the angle between \vec{u} and \vec{v} that is to the clockwise of \vec{u} and to the counter-clockwise of \vec{v} . For two halfplanes H and J , we write $\text{ang}(H, J) = \text{ang}(\vec{u}, \vec{v})$ where \vec{u} (resp., \vec{v}) is the normal of H (resp., J). For a set \mathcal{R} of halfplanes, we use $\bigcap \mathcal{R}$ and $\bigcup \mathcal{R}$ to denote the intersection and the union of all halfplanes in \mathcal{R} , respectively. We say a halfplane $H \in \mathcal{R}$ is *redundant* in \mathcal{R} if $\bigcap \mathcal{R} = \bigcap (\mathcal{R} \setminus \{H\})$. We say \mathcal{R} is *irreducible* if every halfplane in \mathcal{R} is *not* redundant. The *complement region* of \mathcal{R} refers to the closure of $\mathbb{R}^2 \setminus \bigcup \mathcal{R}$, which is always a convex polygon (possibly unbounded). The following simple facts about halfplanes will be used throughout the section, and their proofs can be found in the full paper.

► **Fact 11.** *Let \mathcal{R} be an irreducible set of halfplanes such that $\bigcup \mathcal{R} \neq \mathbb{R}^2$. Then the following two properties hold.*

- (i) *For any halfplane $H \in \mathcal{R}$ and another halfplane H' different from H , we have that $\bigcup \mathcal{R} \neq \bigcup \mathcal{R}'$, where $\mathcal{R}' = (\mathcal{R} \setminus \{H\}) \cup \{H'\}$.*

(ii) If the halfplanes in \mathcal{R} has a nonempty intersection, i.e., $\bigcap \mathcal{R} \neq \emptyset$, then we can write $\mathcal{R} = \{H_1, \dots, H_t\}$ such that $0 < \text{ang}(H_1, H_2) < \text{ang}(H_1, H_3) < \dots < \text{ang}(H_1, H_t) < \pi$.

► **Fact 12.** Let H_1, \dots, H_t be halfplanes such that $0 < \text{ang}(H_1, H_2) < \text{ang}(H_1, H_3) < \dots < \text{ang}(H_1, H_t) \leq \pi$. Then the following two properties hold.

- (i) If $H_1 \cup H_t \neq \mathbb{R}^2$, then neither H_1 nor H_t is redundant in $\{H_1, \dots, H_t\}$.
- (ii) If $\{H_1, \dots, H_t\}$ is irreducible, then $\bigcap_{i=1}^t H_i = H_1 \cap H_t$.

3.1 An $n^{O(\text{opt})}$ -time exact algorithm

In this section, we show how to compute an (exact) optimal solution of the instance (S, S', \mathcal{H}) in $n^{O(\text{opt})}$ time. It suffices to solve a decision problem: given an integer $k \geq 0$, find a subset $\mathcal{Z} \subseteq \mathcal{H}$ which covers S and satisfies $\text{memb}(S', \mathcal{Z}) \leq k$ or decide that such a subset does not exist. As long as this problem can be solved in $n^{O(k)}$ time, by trying $k = 1, \dots, |\mathcal{H}|$, we can finally compute an optimal solution of (S, S', \mathcal{H}) in $n^{O(\text{opt})}$ time. In what follows, a *valid* solution of (S, S', \mathcal{H}) refers to a subset $\mathcal{Z} \subseteq \mathcal{H}$ which covers S and satisfies $\text{memb}(S', \mathcal{Z}) \leq k$.

Let Δ be a sufficiently large number such that $S \cup S' \subseteq [-\Delta, \Delta]^2$. For convenience, we add to \mathcal{H} four dummy halfplanes with equations $y \leq -\Delta$, $y \geq \Delta$, $x \leq -\Delta$, and $x \geq \Delta$. As these dummy halfplanes does not contain any points in $S \cup S'$, including them in \mathcal{H} does not change the problem. We say a set of halfplanes is *regular* if it is irreducible and its complement region is nonempty and bounded. We have the following simple observation, whose proof can be found in the full paper.

► **Fact 13.** If (S, S', \mathcal{H}) has a valid solution, then either it has a regular valid solution or it has a valid solution that covers the entire plane \mathbb{R}^2 .

If (S, S', \mathcal{H}) has a valid solution that covers \mathbb{R}^2 , then it also has an irreducible valid solution that covers \mathbb{R}^2 , which is of size at most 3 by Helly's theorem. Therefore, in this case, we can solve the problem in $n^{O(1)}$ time by simply enumerating all subsets of \mathcal{H} of size at most 3. Otherwise, by the above fact, it suffices to check whether there exists a regular valid solution of (S, S', \mathcal{H}) . In what follows, we assume (S, S', \mathcal{H}) has a regular valid solution and show how to find such a solution in $n^{O(k)}$ time. If our algorithm does not find a regular valid solution at the end, we can conclude its non-existence. Let $\mathcal{Z} \subseteq \mathcal{H}$ be a (unknown) regular valid solution of (S, S', \mathcal{H}) . By definition, the complement region of \mathcal{Z} is nonempty, and is a (bounded) convex polygon. Consider the arrangement \mathcal{A} of the boundary lines of the halfplanes in \mathcal{H} . This arrangement has $O(n^2)$ faces, among which at least one face is contained in the complement region of \mathcal{Z} . We simply guess such a face. By making $O(n^2)$ guesses, we can assume that we know a face F in the complement region of \mathcal{Z} . Then we take a point p in the interior of F , which is also in the interior of the complement region of \mathcal{Z} .

Now the problem becomes finding a regular valid solution of (S, S', \mathcal{H}) whose complement region contains p . Therefore, we can remove from \mathcal{H} all halfplanes that contain p . Now the complement region of any subset of \mathcal{H} contains p . We say a convex polygon Γ is \mathcal{H} -compatible if each edge e of Γ is a portion of the boundary line of some halfplane $H \in \mathcal{H}$ such that $\Gamma \cap H = e$ (or equivalently H does not contain Γ). Note that the complement region of a regular valid solution is an \mathcal{H} -compatible convex polygon Γ which satisfies (i) no point in S lies in the interior of Γ and (ii) for any $k+1$ edges e_1, \dots, e_{k+1} of Γ , the intersection $\bigcap_{i=1}^{k+1} H(e_i)$ does not contain any point in S' ; here $H(e) \in \mathcal{H}$ denotes the halfplane whose boundary line containing e and $\Gamma \cap H = e$. On the other hand, every \mathcal{H} -compatible convex polygon satisfying conditions (i) and (ii) is the complement region of a regular valid solution of (S, S', \mathcal{H}) , which is just the set of halfplanes corresponding to the edges of Γ . With

this observation, it suffices to find an \mathcal{H} -compatible convex polygon Γ satisfying the two conditions. We notice the follow fact which can be used to simplify condition (ii).

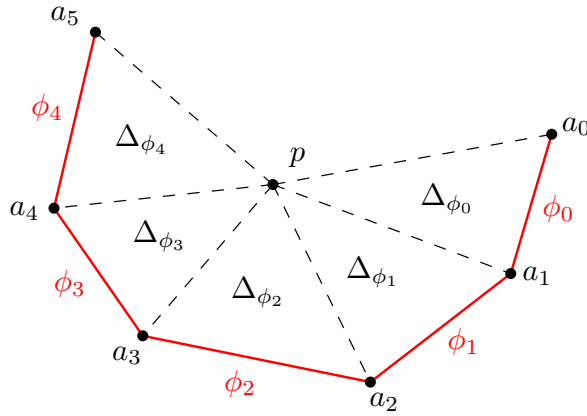
► **Fact 14.** *If there exist t edges e_1, \dots, e_t of an \mathcal{H} -compatible convex polygon Γ such that $(\bigcap_{i=1}^t H(e_i)) \cap S' \neq \emptyset$, then there exist t consecutive edges f_1, \dots, f_t of Γ such that $(\bigcap_{i=1}^t H(f_i)) \cap S' \neq \emptyset$.*

Proof. Set $\mathcal{R} = \{H(e_1), \dots, H(e_t)\}$ and assume $(\bigcap \mathcal{R}) \cap S' \neq \emptyset$, which implies $\bigcap \mathcal{R} \neq \emptyset$. Note that the set of halfplanes corresponding to the edges of Γ are irreducible, because Γ is \mathcal{H} -compatible and thus the interior of each edge e of Γ can only be covered by the halfplane $H(e)$. In particular, \mathcal{R} is irreducible. Also, $\bigcup \mathcal{R} \neq \mathbb{R}^2$ by our assumption $\bigcup \mathcal{H} \neq \mathbb{R}^2$. Therefore, by (ii) of Fact 11, there exist $e^-, e^+ \in \{e_1, \dots, e_t\}$ such that $\text{ang}(H(e^-), H(e_i)) \leq \text{ang}(H(e^-), H(e^+)) < \pi$ for all $i \in [t]$. Now we go clockwise along the boundary of Γ from e^- to e^+ , and let E be the set of edges of Γ we visit (including e^- and e^+). Clearly, $e_i \in E$ for all $i \in [t]$ and thus $|E| \geq t$. Furthermore, $0 < \text{ang}(H(e^-), H(e)) < \text{ang}(H(e^-), H(e^+)) < \pi$ for all $e \in E \setminus \{e^-, e^+\}$. Define $\mathcal{R}' = \{H(e) : e \in E\}$. Since \mathcal{R} and \mathcal{R}' are both irreducible, we can apply (ii) of Fact 12 to deduce $\bigcap \mathcal{R} = H(e^-) \cap H(e^+) = \bigcap \mathcal{R}'$, which implies $(\bigcap \mathcal{R}') \cap S' \neq \emptyset$. Finally, because E consists of consecutive edges of Γ and $|E| \geq t$, there exist $f_1, \dots, f_t \in E$ which are t consecutive edges of Γ . We have $\bigcap \mathcal{R}' \subseteq \bigcap_{i=1}^t H(f_i)$ and thus $(\bigcap_{i=1}^t H(f_i)) \cap S' \neq \emptyset$. ◀

By the above fact, we only need to find an \mathcal{H} -compatible convex polygon Γ which satisfies (i) no point in S lies in the interior of Γ and (ii) $(\bigcap_{i=1}^{k+1} H(e_i)) \cap S' \neq \emptyset$ for any $k + 1$ consecutive edges e_1, \dots, e_{k+1} of Γ . For convenience, we say Γ is *well-behaved* if it satisfies conditions (i) and (ii). Next, we reduce this problem to a shortest-cycle problem in a (weighted) directed graph G as follows. Let \mathcal{L} denote the set of boundary lines of halfplanes in \mathcal{H} . We consider every segment s in the plane which is on some line $\ell \in \mathcal{L}$ and satisfies that each endpoint of s is the intersection point of ℓ and another line in \mathcal{L} . We use Φ to denote the set of these segments. Note that $|\Phi| = O(n^3)$, as Φ contains $O(n^2)$ segments on each line $\ell \in \mathcal{L}$. Clearly, the edges of an \mathcal{H} -compatible convex polygon are all segments in Φ . Consider a segment $\phi \in \Phi$. Recall that the point p is the interior of F , which is a face of the arrangement \mathcal{A} . Thus, no line in \mathcal{L} goes through p . It follows that for every segment $\phi \in \Phi$, the two endpoints of ϕ and p form a triangle Δ_ϕ . If $p \rightarrow a \rightarrow b$ is the clockwise ordering of the three vertices of Δ_ϕ from p , then we call a the *left* endpoint of ϕ and call b the *right* endpoint of ϕ . Clearly, $\text{ang}(\vec{pa}, \vec{pb}) < \pi$. The vertices of the graph G to be constructed are one-to-one corresponding to the $(k + 1)$ -tuples $(\phi_0, \phi_1, \dots, \phi_k) \in \Phi^{k+1}$ which satisfy the following three conditions.

1. The left endpoint of ϕ_i is the right endpoint of ϕ_{i-1} for all $i \in [k]$. Below we use a_i to denote the left endpoint of ϕ_i (i.e., the right endpoint of ϕ_{i-1}). This condition guarantees that the segments $\phi_0, \phi_1, \dots, \phi_k$ form a polygonal chain of $k + 1$ pieces.
2. $\text{ang}(\vec{a_{i-1}a_i}, \vec{a_i a_{i+1}}) \leq \pi$ for all $i \in [k]$. This condition guarantees that the chain formed by $\phi_0, \phi_1, \dots, \phi_k$ is *clockwise convex*, in the sense that when we go along the chain from a_0 to a_{k+1} , we always turn *right* at the vertices of the chain. Figure 2 shows a chain satisfying this condition (and also condition 1).
3. $S \cap (\bigcup_{i=0}^k \Delta_{\phi_i}) \subseteq \bigcup_{i=0}^k \phi_i$ and $(\bigcap_{i=0}^k H(\phi_i)) \cap S' = \emptyset$.

Intuitively, the $(k + 1)$ -tuple corresponding to each vertex of G represents a possible choice of $k + 1$ consecutive edges of the \mathcal{H} -compatible convex polygon we are looking for. For two vertices $v = (\phi_0, \phi_1, \dots, \phi_k)$ and $v' = (\phi'_0, \phi'_1, \dots, \phi'_k)$ such that $(\phi_1, \dots, \phi_k) = (\phi'_0, \phi'_1, \dots, \phi'_{k-1})$, we add a directed edge from v to v' with weight $\text{ang}(\vec{pa}, \vec{pb})$, where a is



■ **Figure 2** Illustrating the conditions for a vertex of G .

the left endpoint of ϕ_0 and b is the left endpoint of $\phi'_0 = \phi_1$ (which is also the right endpoint of ϕ_0 by condition 1 above). Note that the weight of every edge of G is positive, since the two endpoints of every $\phi \in \Phi$ and p form a triangle Δ_ϕ . The key observation is the following lemma, whose proof (which is simple but tedious) can be found in the full paper.

► **Lemma 15.** *There exists a well-behaved \mathcal{H} -compatible convex polygon containing p iff the (weighted) length of a shortest cycle in G is exactly 2π .*

Based on the above lemma, it suffices to compute a shortest cycle in G , which can be done by standard algorithms (e.g., Dijkstra) in polynomial time in the size of G . Note that G has $n^{O(k)}$ vertices. Therefore, we obtain an $n^{O(k)}$ -time algorithm for computing a set cover $\mathcal{Z} \subseteq \mathcal{H}$ of S satisfying $\text{memb}(S', \mathcal{Z}) \leq k$, if such a set cover exists. By iteratively trying $k = 1, \dots, |\mathcal{H}|$, we can solve the MMGSC problem with halfplanes in $n^{O(\text{opt})}$ time.

3.2 An algorithm with constant additive error

In this section, we show how to compute in polynomial time an approximation solution $\mathcal{Z} \subseteq \mathcal{H}$ of the instance (S, S', \mathcal{H}) with constant additive error, that is, $\text{memb}(S', \mathcal{Z}) = \text{opt}(S, S', \mathcal{H}) + O(1)$. If $\bigcup \mathcal{H} = \mathbb{R}^2$, then by Helly's theorem, there exist $H_1, H_2, H_3 \in \mathcal{H}$ such that $H_1 \cup H_2 \cup H_3 = \mathbb{R}^2$. In this case, we can take $\{H_1, H_2, H_3\}$ as our solution, which clearly has constant additive error. So assume $\bigcup \mathcal{H} \neq \mathbb{R}^2$. Our algorithm is in the spirit of local search. However, different from most local-search algorithms which improve the “quality” of the solution in each step (via local modifications), our algorithm does not care about the quality (i.e., membership), and instead focuses on shrinking the complement region of the solution. Formally, for two sets \mathcal{Z} and \mathcal{Z}' , we write $\mathcal{Z} \prec \mathcal{Z}'$ if $\bigcup \mathcal{Z} \subsetneq \bigcup \mathcal{Z}'$, and $\mathcal{Z} \preceq \mathcal{Z}'$ if $\bigcup \mathcal{Z} \subseteq \bigcup \mathcal{Z}'$. We define the following notion of “locally (non-)improvable” solutions.

► **Definition 16.** *A subset $\mathcal{Z} \subseteq \mathcal{H}$ is k -expandable if there exists $\mathcal{Z}' \subseteq \mathcal{H}$ such that $|\mathcal{Z} \setminus \mathcal{Z}'| = |\mathcal{Z}' \setminus \mathcal{Z}| \leq k$ and $\mathcal{Z} \prec \mathcal{Z}'$. A subset of \mathcal{H} is k -stable if it is not k -expandable.*

In other words, $\mathcal{Z} \subseteq \mathcal{H}$ is k -expandable (resp., k -stable) if we can (resp., cannot) replace k halfplanes in \mathcal{Z} with other k halfplanes in \mathcal{H} to shrink the complement region of \mathcal{Z} . We are interested in subsets $\mathcal{Z} \subseteq \mathcal{H}$ that are *minimum-size* set covers of S and are k -stable. Such a set can be constructed via the standard local-search procedure.

► **Lemma 17.** *A minimum-size set cover $\mathcal{Z} \subseteq \mathcal{H}$ of S that is k -stable can be computed in $n^{O(k)}$ time.*

Proof. The standard set cover problem for halfplanes is polynomial-time solvable. So we can compute a minimum-size set cover $\mathcal{Z} \subseteq \mathcal{H}$ of S in $n^{O(1)}$ time. To further obtain a k -stable one, we keep doing the following procedure. Whenever there exists $\mathcal{Z}' \subseteq \mathcal{H}$ such that $|\mathcal{Z} \setminus \mathcal{Z}'| = |\mathcal{Z}' \setminus \mathcal{Z}| \leq k$ and $\mathcal{Z} \prec \mathcal{Z}'$, we update \mathcal{Z} to \mathcal{Z}' . During this procedure, the size of \mathcal{Z} does not change and the complement region of \mathcal{Z} shrinks. So \mathcal{Z} is always a minimum-size set cover of S . Furthermore, as the complement region of \mathcal{Z} shrinks in every step, the procedure will finally terminate. At the end, \mathcal{Z} is not k -expandable and is thus k -stable. This proves the correctness of our algorithm. To see it takes $n^{O(k)}$ time, we show that (i) we terminate in $O(n)$ steps and (ii) each step can be implemented in $n^{O(k)}$ time.

For (i), the key observation is that every halfplane $H \in \mathcal{H}$ can be removed from \mathcal{Z} at most once. Formally, we denote by \mathcal{Z}_i the set \mathcal{Z} after the i -th step of the procedure, and thus the original \mathcal{Z} is \mathcal{Z}_0 . Let P_i be the complement region of \mathcal{Z}_i . Suppose $H \in \mathcal{Z}_{i-1}$ and $H \notin \mathcal{Z}_i$. We claim that $H \notin \mathcal{Z}_j$ for all $j > i$. Assume $H \in \mathcal{Z}_j$ for some $j > i$. Since \mathcal{Z}_j is a minimum-size set cover of S , H is not redundant in \mathcal{Z}_j and thus one edge e of P_j is defined by H , i.e., e is a segment on the boundary line of H . Note that e is also a portion of the boundary of P_{i-1} , because $H \in \mathcal{Z}_{i-1}$ and $P_j \subseteq P_{i-1}$. It follows that e is a portion of the boundary of P_i , since $P_j \subseteq P_i \subseteq P_{i-1}$. But this cannot be the case, as $H \notin \mathcal{Z}_i$. Thus, $H \notin \mathcal{Z}_j$ for all $j > i$. Now for every index $i \geq 1$, there exists at least one halfplane $H \in \mathcal{H}$ such that $H \in \mathcal{Z}_{i-1}$ and $H \notin \mathcal{Z}_i$, simply because $|\mathcal{Z}_{i-1}| = |\mathcal{Z}_i|$ and $\mathcal{Z}_{i-1} \neq \mathcal{Z}_i$. We then charge the i -th step to this halfplane H . By the above observation, each halfplane is charged at most once. Therefore, the procedure terminates in at most n steps. To see (ii), observe that in each step, the number of $\mathcal{Z}' \subseteq \mathcal{H}$ satisfying $|\mathcal{Z} \setminus \mathcal{Z}'| = |\mathcal{Z}' \setminus \mathcal{Z}| \leq k$ is bounded by $n^{O(k)}$, and these sets can be enumerated in $n^{O(k)}$ time. So each step can be implemented in $n^{O(k)}$ time. As a result, the entire algorithm terminates in $n^{O(k)}$ time. ◀

Our key observation is that any minimum-size set cover of S that is k -stable has additive error at most 2 in terms of MMGSC, even for $k = 1$.

► **Lemma 18.** *If $\mathcal{Z} \subseteq \mathcal{H}$ is a minimum-size set cover of S that is 1-stable, then we have $|\mathcal{Z}| \leq \text{opt}(S, S', \mathcal{H}) + 2$.*

Proof. Consider a point $p \in S'$. We show that $\text{memb}(p, \mathcal{Z}') \geq \text{memb}(p, \mathcal{Z}) - 2$ for any set cover $\mathcal{Z}' \subseteq \mathcal{H}$ of S . Let $\mathcal{Z}(p) \subseteq \mathcal{Z}$ consist of all halfplanes in \mathcal{Z} that contain p . As $\bigcap \mathcal{Z}(p) \neq \emptyset$, by (ii) of Fact 11 and the assumption $\bigcup \mathcal{H} \neq \mathbb{R}^2$, we have $\mathcal{Z}(p) = \{H_1, \dots, H_r\}$ such that $0 < \text{ang}(H_1, H_2) < \text{ang}(H_1, H_3) < \dots < \text{ang}(H_1, H_r) < \pi$. Let $S_0 \subseteq S$ consist of points contained in $\bigcup_{i=2}^{r-1} H_i$ but not contained in any other halfplanes in \mathcal{Z} , and $\mathcal{Z}'_0 \subseteq \mathcal{Z}'$ consist of halfplanes that contain at least one point in S_0 . Note that $|\mathcal{Z}'_0| \geq r - 2$, for otherwise $(\mathcal{Z} \setminus \{H_2, \dots, H_{r-1}\}) \cup \mathcal{Z}'_0$ is a set cover of S of size strictly smaller than \mathcal{Z} , which contradicts the fact that \mathcal{Z} is a minimum-size set cover of S . We shall show that every halfplane in \mathcal{Z}'_0 contains p and thus

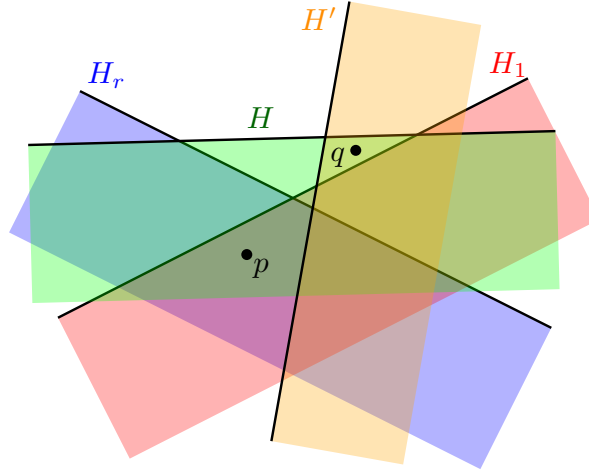
$$\text{memb}(p, \mathcal{Z}') \geq \text{memb}(p, \mathcal{Z}'_0) = |\mathcal{Z}'_0| \geq r - 2 = \text{memb}(p, \mathcal{Z}) - 2.$$

Consider a halfplane $H' \in \mathcal{Z}'_0$. We want to show $p \in H'$. By the construction of \mathcal{Z}'_0 , H' contains a point $q \in S_0$. Furthermore, by the construction of S_0 , q is contained in $\bigcup_{i=2}^{r-1} H_i$ but not contained in any halfplane in $\mathcal{Z} \setminus \{H_2, \dots, H_{r-1}\}$. In particular, $q \notin H_1$ and $q \notin H_r$, which implies $H' \neq H_1$ and $H' \neq H_r$. We observe that $\{H_1, H_r, H'\}$ is irreducible. Clearly, H' is not redundant in $\{H_1, H_r, H'\}$, as it contains q while H_1 and H_r do not contain q . If H_1 is redundant in $\{H_1, H_r, H'\}$, then $\mathcal{Z} \preceq (\mathcal{Z} \setminus \{H_1\}) \cup \{H'\}$. Since \mathcal{Z} is irreducible and $H' \neq H_1$, by (i) of Fact 11, this implies $\mathcal{Z} \prec (\mathcal{Z} \setminus \{H_1\}) \cup \{H'\}$, which contradicts the fact

11:12 Minimum-Membership Geometric Set Cover

that \mathcal{Z} is 1-stable. So H_1 is not redundant in $\{H_1, H_r, H'\}$. For the same reason, H_r is also not redundant in $\{H_1, H_r, H'\}$. Thus, $\{H_1, H_r, H'\}$ is irreducible.

In what follows, we complete the proof by showing that either $p \in H'$ or \mathcal{Z} is 1-expandable. As the latter is false (for \mathcal{Z} is 1-stable), this implies $p \in H'$. If $\text{ang}(H_1, H') < \text{ang}(H_1, H_r)$, by the irreducibility of $\{H_1, H_r, H'\}$ and (ii) of Fact 12, we have $H_1 \cap H' \cap H_r = H_1 \cap H_r$, which implies $H_1 \cap H_r \subseteq H'$ and thus $p \in H_1 \cap H_r \subseteq H'$. If $\text{ang}(H_1, H') = \text{ang}(H_1, H_r)$, then either $H' \subseteq H_r$ or $H_r \subseteq H'$. Note that the former is not true as $q \in H'$ but $q \notin H_r$. Thus, we have $p \in H_r \subseteq H'$. It suffices to consider the case $\text{ang}(H_1, H') > \text{ang}(H_1, H_r)$. In this case, we show that \mathcal{Z} is 1-expandable. Since $q \in \bigcup_{i=2}^{r-1} H_i$, there exists $H \in \{H_2, \dots, H_{r-1}\}$ which contains q . Now $\text{ang}(H_1, H) < \text{ang}(H_1, H_r) < \text{ang}(H_1, H')$, which implies $\text{ang}(H, H_r) < \text{ang}(H, H')$ and $\text{ang}(H', H_1) < \text{ang}(H', H)$. We further distinguish two cases, $\text{ang}(H, H') \leq \pi$ and $\text{ang}(H, H') \geq \pi$ (which are in fact symmetric). Assume $\text{ang}(H, H') \leq \pi$. Figure 3 shows the situation of the points p, q and the halfplanes H, H', H_1, H_r this case. As $\text{ang}(H, H_r) < \text{ang}(H, H')$, by (ii) of Fact 12, if $\{H, H_r, H'\}$ is irreducible, then $H \cap H_r \cap H' = H \cap H'$. But $H \cap H_r \cap H' \neq H \cap H'$, because $q \in H \cap H'$ and $q \notin H_r$. Thus, $\{H, H_r, H'\}$ is reducible. Note that $H \cup H' \neq \mathbb{R}^2$, since $\bigcup \mathcal{H} \neq \mathbb{R}^2$ by our assumption. So by (i) of Fact 12, neither H nor H' is redundant in $\{H, H_r, H'\}$. It follows that H_r is redundant in $\{H, H_r, H'\}$, because $\{H, H_r, H'\}$ is reducible. Therefore, $\mathcal{Z} \preceq (\mathcal{Z} \setminus \{H_r\}) \cup \{H'\}$. Since \mathcal{Z} is irreducible and $H' \neq H_r$, by (i) of Fact 11, we have $\mathcal{Z} \prec (\mathcal{Z} \setminus \{H_r\}) \cup \{H'\}$, i.e., \mathcal{Z} is 1-expandable. The other case $\text{ang}(H, H') \geq \pi$ is similar. In this case, $\text{ang}(H', H) \leq \pi$. Using the fact $\text{ang}(H', H_1) < \text{ang}(H', H)$ and the same argument as above, we can show that $\mathcal{Z} \prec (\mathcal{Z} \setminus \{H_1\}) \cup \{H'\}$, i.e., \mathcal{Z} is 1-expandable. ◀



■ **Figure 3** Illustration of the proof of Lemma 18.

Using Lemma 17, we can compute a 1-stable minimum-size set cover $\mathcal{Z} \subseteq \mathcal{H}$ of S in $n^{O(1)}$ time. Then by Lemma 18, \mathcal{Z} is an approximation solution for the MMGSC instance (S, S', \mathcal{H}) with additive error 2. This gives us a polynomial-time approximation algorithm for MMGSC with halfplanes with $O(1)$ additive error.

3.3 Putting everything together

Our PTAS can be obtained by directly combining the algorithms in Sections 3.1 and 3.2. Let $c = O(1)$ be the additive error of the algorithm in Section 3.2. We first run the algorithm in

Section 3.2 to obtain a solution $\mathcal{Z} \subseteq \mathcal{H}$. If $|\mathcal{Z}| \geq \frac{1+\varepsilon}{\varepsilon} \cdot c$, then

$$\frac{|\mathcal{Z}|}{\text{opt}(S, S', \mathcal{H})} \leq \frac{|\mathcal{Z}|}{|\mathcal{Z}| - c} \leq 1 + \varepsilon.$$

In this case, \mathcal{Z} is already a $(1 + \varepsilon)$ -approximation solution. Otherwise, $|\mathcal{Z}| < \frac{1+\varepsilon}{\varepsilon} \cdot c$ and thus $\text{opt}(S, S', \mathcal{H}) < \frac{1+\varepsilon}{\varepsilon} \cdot c$. We can then run the algorithm in Section 3.1 to compute an optimal solution in $n^{O(1/\varepsilon)}$ time. So we conclude the following.

► **Theorem 3.** *The generalized MMGSC problem with halfplanes admits a PTAS.*

4 Minimum-ply geometric set cover

In this section, we give a very simple constant-approximation algorithm for minimum-ply geometric set cover with unit squares. The technique can be applied to the problem with any similarly-sized geometric objects in \mathbb{R}^2 .

Let (S, \mathcal{Q}) be an MPGSC instance. As in Section 2, we first apply the grid technique. We construct a grid Γ consisting of square cells of side-length 1. For each grid cell \square , write $S_\square = S \cap \square$ and $\mathcal{Q}_\square = \{Q \in \mathcal{Q} : Q \cap \square \neq \emptyset\}$. The key observation is the following.

► **Lemma 19.** *Suppose that, for every $\square \in \Gamma$, $\mathcal{Q}_\square^* \subseteq \mathcal{Q}_\square$ is a c -approximation solution of the minimum-size geometric set cover instance $(S_\square, \mathcal{Q}_\square)$. Then $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ is an $O(c)$ -approximation solution of the MPGSC instance (S, \mathcal{Q}) .*

Proof. Let $\gamma = \text{ply}(\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*)$ and $p \in \mathbb{R}^2$ be a point contained in γ unit squares in $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$. Consider the grid cell \square_p containing p and define \mathcal{C} as the set of 3×3 grid cells centered at \square_p . Note that all unit squares containing p belong to $\bigcup_{\square \in \mathcal{C}} \mathcal{Q}_\square^*$. So we have $|\bigcup_{\square \in \mathcal{C}} \mathcal{Q}_\square^*| \geq \gamma$ and $|\max_{\square \in \mathcal{C}} \mathcal{Q}_\square^*| \geq \gamma/9$. Therefore, there exists $\square \in \Gamma$ such that $|\mathcal{Q}_\square^*| \geq \gamma/9$. As \mathcal{Q}_\square^* is a c -approximation solution of the minimum-size set cover instance $(S_\square, \mathcal{Q}_\square)$, we know that any subset of \mathcal{Q}_\square that covers S_\square has size at least $\gamma/(9c)$. It follows that any subset of \mathcal{Q} that covers S must include at least $\gamma/(9c)$ unit squares in \mathcal{Q}_\square . Note that each of these unit squares contains a corner of \square . Thus, at least one corner of \square is contained in $\gamma/(36c)$ such unit squares, which implies that the ply of any solution is at least $\gamma/(36c)$. As a result, $\bigcup_{\square \in \Gamma} \mathcal{Q}_\square^*$ is an $O(c)$ -approximation solution of the MPGSC instance (S, \mathcal{Q}) . ◀

Note that the argument in the above proof can be extended to any similarly-size fat objects in any fixed dimension. Here a set of geometric objects are *similarly-size fat objects* if there exist constants $\alpha, \beta > 0$ such that every object in the set contains a ball of radius α and is contained in a ball of radius β .

► **Theorem 20.** *For any class \mathcal{C} of similarly sized fat objects in \mathbb{R}^d , if the minimum-size geometric set cover problem with \mathcal{C} admits a constant-approximation algorithm with running time $T(n)$ for a function T satisfying $T(a + b) \geq T(a) + T(b)$, then the MPGSC problem with \mathcal{C} also admits a constant-approximation algorithm with running time $T(n)$.*

Proof. Let (S, \mathcal{R}) be an MPGSC instance where $\mathcal{R} \subseteq \mathcal{C}$. We use the above grid technique to decompose the input instance (S, \mathcal{R}) into a set $\{(S_\square, \mathcal{R}_\square)\}$ of instances. Then apply the algorithm for minimum-size geometric set cover problem with \mathcal{C} to compute constant-approximation (with respect to size) solutions $\mathcal{R}_\square^* \subseteq \mathcal{R}_\square$. By Lemma 19, $\bigcup_{\square \in \Gamma} \mathcal{R}_\square^*$ is a constant-approximation solution of the MPGSC instance (S, \mathcal{R}) . If the algorithm for minimum-size set cover runs in $T(n)$ time, then our algorithm also takes $T(n)$ time, as long as the function T satisfies $T(a + b) \geq T(a) + T(b)$. ◀

► **Theorem 4.** *The MPGSC problem with unit (or similarly sized) squares/disks admits constant-approximation algorithms with running time $\tilde{O}(n)$.*

Proof. The $\tilde{O}(n)$ -time constant-approximation algorithms for minimum-size set cover with similarly sized squares/disks are well-known. For similarly sized squares, see for example [1]. For similarly sized disks, see for example [2, 4]. Applying Theorem 20 directly yields $\tilde{O}(n)$ -time constant-approximation algorithms for MPGSC with unit squares and unit disks. ◀

References

- 1 Pankaj Agarwal, Hsien-Chih Chang, Subhash Suri, Allen Xiao, and Jie Xue. Dynamic geometric set cover and hitting set. *ACM Transactions on Algorithms (TALG)*, 18(4):1–37, 2022.
- 2 Pankaj K Agarwal and Jiangwei Pan. Near-linear algorithms for geometric hitting sets and set covers. In *Proceedings of the thirtieth annual symposium on Computational geometry*, pages 271–279, 2014.
- 3 Therese Biedl, Ahmad Biniiaz, and Anna Lubiw. Minimum ply covering of points with disks and squares. *Comput. Geom.*, 94:101712, 2021. doi:10.1016/j.comgeo.2020.101712.
- 4 Timothy M Chan and Qizheng He. Faster approximation algorithms for geometric set cover. In *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164, page 27. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2020.
- 5 Stephane Durocher, J Mark Keil, and Debajyoti Mondal. Minimum ply covering of points with unit squares. In *WALCOM: Algorithms and Computation: 17th International Conference and Workshops, WALCOM 2023, Hsinchu, Taiwan, March 22–24, 2023, Proceedings*, pages 23–35. Springer, 2023.
- 6 Thomas Erlebach and Erik Jan van Leeuwen. Approximating geometric coverage problems. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2008, San Francisco, California, USA, January 20–22, 2008*, pages 1267–1276. SIAM, 2008. URL: <http://dl.acm.org/citation.cfm?id=1347082.1347220>.
- 7 Fabian Kuhn, Pascal von Rickenbach, Roger Wattenhofer, Emo Welzl, and Aaron Zollinger. Interference in cellular networks: The minimum membership set cover problem. In Lusheng Wang, editor, *Computing and Combinatorics, 11th Annual International Conference, COCOON 2005, Kunming, China, August 16–29, 2005, Proceedings*, volume 3595 of *Lecture Notes in Computer Science*, pages 188–198. Springer, 2005. doi:10.1007/11533719_21.
- 8 Joseph SB Mitchell and Supantha Pandit. Minimum membership covering and hitting. *Theoretical Computer Science*, 876:1–11, 2021.

FPT Constant-Approximations for Capacitated Clustering to Minimize the Sum of Cluster Radii

Sayan Bandyapadhyay

Department of Computer Science, Portland State University, OR, USA

William Lochet

LIRMM, Université de Montpellier, CNRS, Montpellier, France

Saket Saurabh

The Institute of Mathematical Sciences, HBNI, Chennai, India

Abstract

Clustering with capacity constraints is a fundamental problem that attracted significant attention throughout the years. In this paper, we give the first FPT constant-factor approximation algorithm for the problem of clustering points in a general metric into k clusters to minimize the sum of cluster radii, subject to non-uniform hard capacity constraints (CAPACITATED SUM OF RADII). In particular, we give a $(15 + \epsilon)$ -approximation algorithm that runs in $2^{\mathcal{O}(k^2 \log k)} \cdot n^3$ time.

When capacities are uniform, we obtain the following improved approximation bounds.

- A $(4 + \epsilon)$ -approximation with running time $2^{\mathcal{O}(k \log(k/\epsilon))} n^3$, which significantly improves over the FPT 28-approximation of Inamdar and Varadarajan [ESA 2020].
- A $(2 + \epsilon)$ -approximation with running time $2^{\mathcal{O}(k/\epsilon^2 \cdot \log(k/\epsilon))} dn^3$ and a $(1 + \epsilon)$ -approximation with running time $2^{\mathcal{O}(kd \log((k/\epsilon)))} n^3$ in the Euclidean space. Here d is the dimension.
- A $(1 + \epsilon)$ -approximation in the Euclidean space with running time $2^{\mathcal{O}(k/\epsilon^2 \cdot \log(k/\epsilon))} dn^3$ if we are allowed to violate the capacities by $(1 + \epsilon)$ -factor. We complement this result by showing that there is no $(1 + \epsilon)$ -approximation algorithm running in time $f(k) \cdot n^{\mathcal{O}(1)}$, if any capacity violation is not allowed.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Clustering, FPT-approximation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.12

Related Version *Full Version*: <https://arxiv.org/abs/2303.07923>

Funding *Saket Saurabh*: Supported by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement no. 819416), and Swarnajayanti Fellowship (no. DST/SJF/MSA01/2017-18).

1 Introduction

The SUM OF RADII (clustering) problem is among the most popular and well-studied clustering models in the literature, together with k -center, k -means, and k -median [12, 25, 4, 30]. In SUM OF RADII, we are given a set P of n points in a metric space with distance dist , and a non-negative integer k specifying the number of clusters sought. We would like to find: (i) a subset C of P containing k points (called centers) and a non-negative integer r_q (called radius) for each $q \in C$, and (ii) a function assigning each point $p \in P$ to a center $q \in C$ such that $\text{dist}(p, q) \leq r_q$. The goal is to minimize the sum of the radii $\sum_{q \in C} r_q$. Alternatively, the objective is to select k balls in the metric space centered at k distinct points of P , such that each point $p \in P$ is contained in at least one of those k balls and the sum of the radii of the balls is minimized.



© Sayan Bandyapadhyay, William Lochet, and Saket Saurabh;
licensed under Creative Commons License CC-BY 4.0

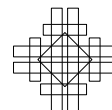
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 12; pp. 12:1–12:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



In a seminal work, Charikar and Panigrahy [12] studied the SUM OF RADII problem. As mentioned in their paper, sum of radii objective can be used as an alternative to the k -center objective to reduce the so called *dissection effect*. The k -center objective is similar to sum of radii, except here one would want to minimize the maximum radius. As in k -center all balls are assumed to have the same maximum radius, the balls can have huge overlap. Consequently, points that should have been assigned to the same cluster might end up in different clusters. This phenomenon is called the dissection effect which can be reduced by using the sum of radii objective instead.

Considering the sum of radii objective, Charikar and Panigrahy [12] obtained a 3.504-approximation running in polynomial time, which is the best known approximation factor for this problem in polynomial time to date. Their algorithm is based on a primal-dual scheme coupled with an application of Lagrangean relaxation. Subsequently, Gibson et al. [23] obtained a $(1 + \epsilon)$ -approximation in quasi-polynomial time. It follows from the standard complexity theoretic assumptions that the problem cannot be APX-hard. We note that the problem is known to be NP-hard even in weighted planar metrics and metrics of constant doubling dimension [23]. Surprisingly, the problem can be solved in polynomial time in Euclidean spaces when the dimension is fixed [24]. When the dimension is arbitrary, one can obtain a $(1 + \epsilon)$ -approximation in $2^{\mathcal{O}((k \log k)/\epsilon^2)} \cdot n^{\mathcal{O}(1)}$ time, extending the coresset based algorithm for k -center [5].

1.1 Our Problem and Results

In this work, we are interested in the capacitated version of SUM OF RADII. Clustering with capacity constraints is a fundamental problem and has attracted significant attention recently [10, 8, 11, 13, 21, 33, 34, 7, 22, 1, 37, 19]. Indeed, capacitated clustering is relevant in many real-life applications, such as load balancing where the representative of each cluster can handle the load of only a bounded number of objects. It is widely known that clustering problems become much harder in the presence of capacity constraints.

Formally, in the CAPACITATED SUM OF RADII problem, along with the points of P in a metric space, we are also given a non-negative integer η_q for each $q \in P$, which denotes the capacity of q . The goal is similar to the goal of SUM OF RADII except here each chosen center $q \in C$ can be assigned at most η_q points of P . Alternatively, each cluster contains a bounded number of points specified with respect to the center of the cluster. In the uniform-capacitated version of the problem, $\eta_p = \eta_q$ for all $p, q \in P$, and we denote the capacity by U . We note that in this work, we only consider *hard* capacities, i.e., each point can be chosen at most once to be a cluster center. In this setting, a major open question is to determine whether there is a polynomial time $\mathcal{O}(1)$ -approximation algorithm for CAPACITATED SUM OF RADII, even in the uniform-capacitated case.

Question 1: Does CAPACITATED SUM OF RADII admit a polynomial time constant-approximation algorithm, even with uniform capacities?

Designing polynomial time constant-approximations for capacitated clustering problems are notoriously hard. In fact such algorithms exist only for the k -center objective out of the four objectives mentioned before. For uniform capacitated k -center, Khuller and Sussmann [31] designed a 6-approximation improving a 10-approximation of Bar-Ilan et al. [6] who introduced the problem. The first constant-approximation in the non-uniform case [20] was designed after 12 years, which was subsequently improved to a 9-approximation by An et al. [3]. The capacitated problems with k -means and k -median objectives have

attracted a lot of attention over the years. But, despite a recent progress for the uniform version in \mathbb{R}^2 [14], where a PTAS is achieved, even in \mathbb{R}^3 , the problem of finding a polynomial time constant-approximation remains open. The best-known polynomial time approximation factor in general metrics is $\mathcal{O}(\log k)$ [19], which is based on a folklore tree embedding scheme.

Technical Barriers for Sum of Radii. The problem with sum of radii objective also appears to be fairly challenging. The main difficulty in achieving a polynomial time $\mathcal{O}(1)$ -approximation for CAPACITATED SUM OF RADII is obviously the presence of the capacity bounds even if they are uniform, which makes the problem resilient to the techniques used for solving SUM OF RADII. The only polynomial time $\mathcal{O}(1)$ -approximation known for SUM OF RADII is via a primal-dual scheme. However, it is not clear how to interpret the capacity constraints in the primal, in the realm of dual. Also, while the algorithms for capacitated k -center use LP-relaxation of the natural LP, the standard LP relaxation for CAPACITATED SUM OF RADII has a large integrality gap [29]. Needless to say, the situation becomes much more intractable in the non-uniform capacitated case.

Hardness of Approximation. The lower bounds known for capacitated clustering are equally frustrating as their upper bounds. Surprisingly, the only known lower bounds are the ones for the uncapacitated versions, and hence trivially translated to the capacitated case. Due to the 20-year old work of Guha and Khuller [26], k -median and k -means are known to be NP-hard to approximate within factors of 1.735 and 3.943, respectively. In a recent series of papers, Cohen-Addad, Karthik and Lee [16, 17, 18] have obtained improved constant lower bounds for various clustering problems in different metrics and settings. In particular, in the last work, they introduced an interesting Johnson Coverage Hypothesis [18] which helped them obtain improved bounds in various metrics. As mentioned before, SUM OF RADII cannot be APX-hard, and hence there is no known inapproximability results that can be translated to the capacitated version.

In the light of the above discussions, one may conclude that the rather benign capacity constraints have played a bigger role compared to the choice of objective function, in our current lack of understanding of practical clustering models. Therefore, it seems that making any intermediate progress towards understanding capacitated clustering, irrespective of the objective function, is significant and timely.

Coping with Capacitated Clustering. In order to improve the understanding of these challenging open questions, researchers have mainly studied two types of relaxations to obtain constant-approximation algorithms. The more traditional approach taken for k -means and k -median is to design bi-criteria approximation where we are allowed to violate either capacity or the bound on the number of clusters by a small amount [10, 8, 11, 13, 21, 33, 34]. The other (relatively newer) approach is to design *fixed-parameter tractable* (FPT) approximation, thus allowing an extra factor $f(k)$ in the running time. We note that, in recent years, FPT approximations are designed for classic problems improving the best known approximation factors in polynomial time, e.g., k -vertex separator [32], k -cut [27] and k -treewidth deletion [28].

FPT Approximation for Clustering. In the context of clustering problems, the number of clusters k is a natural choice for the parameter, as the value of k is typically small in practice, e.g., $k \leq 10$ in [35, 36]. Consequently, the approach of designing FPT approximation have become fairly successful for clustering problems and have led to interesting results which

12:4 FPT Approx. for Capacitated Sum of Radii

are not known or impossible in polynomial time. For example, constant-approximations are obtained for the capacitated version of k -median and k -means [1, 37, 19], which almost match the polynomial time constant approximation factors in the uncapacitated case. In the uncapacitated case of k -median and k -means, tight $(1.735 + \epsilon)$ and $(3.943 + \epsilon)$ -factor FPT approximations are recently obtained [15, 1, 37], whereas the best known factors in polynomial time are only 2.611 [9] and 6.357 [2]. These results are interesting in particular, as a popular belief in the clustering community is that there is no algorithmic separation between FPT and polynomial time in general metrics (for example, see the comment in [17] after Theorem 1.3). We note that it is possible to obtain $(1 + \epsilon)$ -approximations in high-dimensional Euclidean spaces [7, 22], which is impossible in polynomial time, assuming standard complexity theoretic conjectures.

Inamdar and Varadarajan [29] adapted the approach of designing FPT approximation to study the CAPACITATED SUM OF RADII problem with uniform capacities. They make the first substantial progress in understanding this problem through the lens of fixed-parameter tractability. In particular, they obtained a 28-approximation algorithm for this problem that runs in time $2^{\mathcal{O}(k^2)} n^{\mathcal{O}(1)}$. Unfortunately, their algorithm does not work in the presence of non-uniform capacities. Based on their result, the following natural questions arise.

Question 2: Does CAPACITATED SUM OF RADII admit a constant-approximation algorithm, in FPT time, even when capacity constraints are non-uniform?

Question 3: Does CAPACITATED SUM OF RADII admit a $(1 + \epsilon)$ -approximation algorithm, in FPT time, when the points are in \mathbb{R}^d (Euclidean Metric)?

We make significant advances towards answering Questions 2 and 3. Our first result completely answers Question 2.

► **Theorem 1.** *For any constant $\epsilon > 0$, the CAPACITATED SUM OF RADII problem admits a $(15 + \epsilon)$ -approximation algorithm with running time $2^{\mathcal{O}(k^2 \log k)} \cdot n^3$.*

Next, we consider the uniform-capacitated version and prove the following theorem significantly improving over the approximation factor of 28 in [29].

► **Theorem 2.** *For any constant $\epsilon > 0$, there exists a randomized algorithm for the CAPACITATED SUM OF RADII problem with uniform capacities that outputs with constant probability a $(4 + \epsilon)$ -approximate solution in time $2^{\mathcal{O}(k \log(k/\epsilon))} \cdot n^3$.*

The approximation factor in the above result is interesting in particular, as it almost matches the approximation factor of 3.504 in the uncapacitated case and keeps the avenue of obtaining a matching approximation in polynomial time open.

Finally, we mention the Euclidean version of the problem where we show that adapting the standard coresset argument for regular k -clustering allows us to obtain the following two results.

► **Theorem 3.** *For any constant $\epsilon > 0$, there exists a randomized algorithm for the Euclidean version of CAPACITATED SUM OF RADII with uniform capacities that outputs with constant probability a $(2 + \epsilon)$ -approximate solution in time $2^{\mathcal{O}((k/\epsilon^2) \log(k/\epsilon))} \cdot dn^3$, where d is the dimension.*

► **Theorem 4.** *For any constant $\epsilon > 0$, the Euclidean version of CAPACITATED SUM OF RADII admits an $(1 + \epsilon)$ -approximation algorithm with running time $2^{\mathcal{O}(kd \log(k/\epsilon))} n^3$.*

We also complement our approximability results by hardness bounds. The NP-hardness of CAPACITATED SUM OF RADII trivially follows from the NP-hardness of SUM OF RADII. We strengthen this bound by showing the following result.

► **Theorem 5.** *CAPACITATED SUM OF RADII with uniform capacities cannot be solved in $f(k)n^{o(k)}$ time for any computable function f , unless ETH is false. Moreover, it does not admit any FPTAS, unless $P=NP$.*

We also show an inapproximability bound in the Euclidean case even with uniform capacities.

► **Theorem 6.** *The Euclidean version of CAPACITATED SUM OF RADII with uniform capacities does not admit any FPTAS even if $k = 2$, unless $P=NP$.*

Although the above bound does not eradicate the possibility of obtaining a $(1 + \epsilon)$ -approximation in the Euclidean case, it shows that to obtain such an approximation, even when $k = 2$, one needs $n^{f(\epsilon)}$ time for some non-constant function f that depends on ϵ . This is in contrast to the uncapacitated version of the problem, where one can get $(1 + \epsilon)$ -approximation in $2^{\mathcal{O}((k \log k)/\epsilon^2)} \cdot n^{\mathcal{O}(1)}$ time as mentioned before.

As by products of our techniques we have also obtained improved bi-criteria approximations for the uniform-capacitated version of the problem where we are allowed to use $(1 + \epsilon)U$ capacity.

► **Theorem 7.** *There is a randomized algorithm for CAPACITATED SUM OF RADII with uniform capacities that runs in time $2^{\mathcal{O}(k \log(k/\epsilon))} \cdot n^{\mathcal{O}(1)}$ and returns a solution with constant probability, such that each ball in the solution uses at most $(1 + \epsilon)U$ capacity and the cost of the solution is at most $(2 + \epsilon) \cdot \text{OPT}$, where OPT is the cost of any optimal solution in which the balls use at most U capacity.*

The above theorem improves the approximation factor in Theorem 2. In the Euclidean case, we obtain a similar algorithm.

► **Theorem 8.** *There is a randomized algorithm for the Euclidean version of CAPACITATED SUM OF RADII with uniform capacities that runs in time $2^{\mathcal{O}((k/\epsilon^2) \log k)} \cdot dn^3$ and returns a solution with constant probability, such that each ball in the solution uses at most $(1 + \epsilon)U$ capacity and the cost of the solution is at most $(1 + \epsilon) \cdot \text{OPT}$, where OPT is the cost of any optimal solution in which the balls use at most U capacity.*

Note that, by Theorem 6, a result as in the above theorem is not possible if we are not allowed to violate the capacity.

1.2 Preliminaries

Capacitated Sum of Radii. We are given a set P of n points in a metric space with distance dist , a non-negative integer η_q for each $q \in P$, and a non-negative integer k . The goal is to find: (i) a subset C of P containing k points and a non-negative integer r_q for each $q \in C$, and (ii) a function $\mu : P \rightarrow C$, such that for each $p \in P$, $\text{dist}(p, \mu(p)) \leq r_{\mu(p)}$, for each $q \in C$, $|\mu^{-1}(q)| \leq \eta_q$, and $\sum_{q \in C} r_q$ is minimized. We will sometimes use OPT to denote the value of an optimal solution.

In the uniform-capacitated case, we denote the common capacity of all centers by U . In the general metric version of our problem, we assume that we are given the pairwise distances dist between the points in P . In the Euclidean version, P is a set of points in \mathbb{R}^d for some $d \geq 1$, dist is the Euclidean distance and any point in \mathbb{R}^d can be selected as a center. Moreover, the capacities of all these centers are uniform.

12:6 FPT Approx. for Capacitated Sum of Radii

We denote the ball with center c and radius r by $B(c, r)$. For any ball $B = B(c, r)$, we will use $\text{ext}(B, r')$ to denote the ball $B(c, r + r')$. Sometimes we will also use $\text{rad}(B)$ to denote the radius of B . Let S be a set of points in \mathbb{R}^d . The *minimum enclosing ball* of S , noted $\text{MEB}(S)$ is the smallest ball in \mathbb{R}^d containing all points of S . We say a ball B *covers* a point p if p is in B .

One important remark regarding solving our capacitated clustering problem on P is that, given a set of k balls \mathcal{B} , the problem of deciding whether there is a valid assignment $\mu : P \rightarrow \mathcal{B}$ satisfying the capacities can easily be modeled as a bipartite matching problem. This implies in particular that if such an assignment exists, it can also be found in $\mathcal{O}(\sqrt{kn}^{3/2})$ time. Therefore, in all our descriptions of the algorithms, we will focus on finding the solution balls while ensuring that a valid assignment exists.

Another remark is that, in the case where every ball has capacity U , we can assume that $|P| \leq k \cdot U$, or the instance is a trivial NO instance.

Organization. Due to limited place, we chose to only present the proof of Theorem 1. The rest of the proofs can be found in the extended version.

2 Capacitated Sum of Radii : General Metric

In this section, we study the case of non uniform capacities. In this setting, for every point x of P , there is an associated integer η_x and any ball centered at x can be assigned at most η_x points. The uniform case correspond to the case where $\eta_x = U$ for all $x \in P$. For convenience, we restate the theorem statement.

► **Theorem 1.** *For any constant $\epsilon > 0$, the CAPACITATED SUM OF RADII problem admits a $(15 + \epsilon)$ -approximation algorithm with running time $2^{\mathcal{O}(k^2 \log k)} \cdot n^3$.*

From now on, let $\mathcal{B}^* := \{B_1^*, \dots, B_k^*\}$ denote the set of balls of a hypothetical optimal solution, $\mu^* : P \rightarrow \mathcal{B}^*$ be the associated assignment and for all $i \in [k]$, let r_i^* and c_i^* denote the radius and the center of the ball B_i^* , respectively. By Lemma 9, just below, we can assume that the algorithm knows an approximate radius r_i for each r_i^* . For a ball $B_i^* \in \mathcal{B}^*$, we say that a ball B_i is an *approximate* ball of B_i^* if $B_i^* \subseteq B_i$, and if x_i denotes the center of B_i , then $\eta_{x_i} \geq \eta_{c_i^*}$. Note that because of the capacity constraints, we can associate $(\mu^*)^{-1}(B_i^*)$ to B_i .

Let us first mention that it is possible to guess an approximation to each of the radii, r_i , in polynomial time. The proof can be found in the extended version.

► **Lemma 9.** *For every $0 < \epsilon < 1$, there exists a randomized algorithm, running in linear time that finds with probability at least $\frac{\epsilon^k}{k^k \cdot n^2}$ a set of reals $\{r_1, \dots, r_k\}$ such that for every $i \in [k]$, $r_i^* \leq r_i$ and $\sum_{i \in [k]} r_i \leq (1 + \epsilon) \sum_{i \in [k]} r_i^*$.*

From now on we assume for simplicity that the algorithm knows an approximate value r_i of r_i^* for all $i \in [k]$. Let us give some informal ideas about how the algorithm of Theorem 1 works. Some technicalities, especially about making sure we don't pick the same center twice, will be left out to the more formal description of the algorithm.

Informal sketch

Ideally we would like to find for each optimal ball B_i^* an approximate ball B_i having the same center as B_i^* and a radius $r_i \leq C \cdot r_i^*$, for some constant C . Indeed, if we have such a set of balls, then the obvious assignment μ defined as $\mu(x) = B_i$ whenever $\mu^*(x) = B_i^*$ would give a solution. While this is not possible in general, the algorithm will start with a greedy procedure to get a set of approximate balls \mathcal{B}_1 for some indices I_1 . The procedure is quite simple: start with $\mathcal{B}_1 := \emptyset$, $I_1 = \emptyset$ and as long as the union of balls in \mathcal{B}_1 does not cover P , pick a point x of P not in the union, guess the index i such that $\mu^*(x) = B_i^*$ and pick c the point at distance at most r_i of x which maximises the value of η_c . Since c_i^* is at distance at most r_i of x , we have that $\eta_{c_i^*} \leq \eta_c$ and that $\text{dist}(c, c_i^*) \leq 2r_i$, which means that the ball B_i of radius $5r_i$ centered around c is an approximate ball of B_i^* . Therefore, the algorithm will add B_i to \mathcal{B}_1 and the index i to I_1 . This procedure stops when the union of \mathcal{B}_1 covers P . At the end of that first step, we have that \mathcal{B}_1 contains an approximate ball for each ball B_i^* of radius $5r_i$ with $i \in I_1$. And while the union of \mathcal{B}_1 covers all P , we are far from being done as the capacity constraints have not been taken into account.

Consider now a ball B_j^* such that $j \notin I_1$, which means that no approximate ball of B_j^* is in \mathcal{B}_1 . In the best case (Lemma 13 below), there is a ball $B_i \in \mathcal{B}_1$ of center x_i approximating B_i^* such that $5r_i \leq r_j$ and $B_i \cap B_j^*$ is non empty. Indeed, in that case the ball of radius $5r_i + r_j$ around x_i contains c_j^* , the center of B_j^* . This means that if x is the point in that ball maximizing η_x , then the ball of center x and radius $2 \cdot (5r_i + 2r_j) \leq 4r_j$ contains x_j^* and thus the ball B_j of center x and radius $2 \cdot (5r_i + 2r_j) + r_j \leq 5r_j$ contains B_j^* and is an approximate ball of B_j^* . Therefore, if such indices j and i exist, the algorithm can guess them and add a new approximate ball to \mathcal{B}_1 .

After this second step, we reach a point where, if $j \notin I_1$ and $i \in I_1$ are such that $B_j^* \cap B_i \neq \emptyset$, then $r_j \leq 5r_i$. In particular, incurring an extra $5r_i$ as we just did to get a replacement for x_j^* is too costly. For this reason, at this step of the algorithm we stop trying to find approximate balls and instead focus on finding balls to “fix” the assignment. Since the balls in \mathcal{B}_1 are approximate balls, it means that we can replace B_i^* with $B_i \in \mathcal{B}_1$ for any $i \in I_1$ (and take the other B_j^*), and still have a solution to our problem with slightly bigger balls. Abusing notation we can still use μ^* for the valid assignment. Now for an index $j \notin I_1$, the ball B_j^* intersects a subset, say T_j , of balls in \mathcal{B}_1 . Ideally we would like to find a ball B_j of center x and radius r_j such that $\eta_x \geq \eta_{x_j}$ and $|B_j \cap (\mu^*)^{-1}(B_i)| \geq |(\mu^*)^{-1}(B_j^*) \cap B_i|$ for all $i \in T_j$. Indeed, in that case we could replace B_j^* by B_j and the condition on $B_j \cap B_i^*$ ensures that we could adapt the assignment μ^* to be a valid assignment by assigning $(\mu^*)^{-1}(B_j^*) \cap B_i$ to B_i and a set of the same size in $B_j \cap (\mu^*)^{-1}(B_i)$ to B_j .

The main difficulty here is that even if we guess the set T_j , picking B_j greedily is not possible as there might be some competitions between the sizes of the intersection with the different balls in B_i for $i \in T_j$ (we cannot afford to guess the $|(\mu^*)^{-1}(B_j^*) \cap B_i|$). The way to avoid this problem is to expand all the balls B_i of \mathcal{B}_1 by $10r_i$. Indeed, since we have assumed that $r_j \leq 5r_i$ for every $i \in T_j$, it means now that B_j^* is **entirely contained** in the $\text{ext}(B_i, 10r_i)$ (expanded ball) for $i \in T_j$. So now, denoting P_j to be the intersection of all the $\text{ext}(B_i, 10r_i)$ for $i \in T_j$ where we removed all the other $B_{i'}$ for $i' \in I_1 \setminus T_j$, we have that $(\mu^*)^{-1}(B_j^*)$ is a subset of P_j , and we can take B_j as the ball of center x and radius r_j which maximizes $s_x = \min\{\eta_x, |B(x, r_j) \cap P_j|\}$. Here there are some technicalities if B_j intersects some ball $B_{j'}$ for $j' \notin I_1$ (that includes $j = j'$), but assume for the moment that it is not the case and let us hint why we can actually replace B_j^* by B_j in our solution if all the balls $B_i \in \mathcal{B}_1$ are replaced by $\text{ext}(B_i, 10r_i)$. Indeed, by choice of B_j , we can assign s_x points of P_j to B_j . By our assumptions, all these points were assigned to B_i for $i \in T_j$ in μ^* , which means that by assigning these points to B_j , there is now a new budget s_x of available

points in the union of $\text{ext}(B_i, 10r_i)$ for $i \in T_j$. However, since B_j^* is entirely contained in the $\text{ext}(B_i, 10r_i)$ for $i \in T_j$, and by choice of s_x , we can assign all the elements of $(\mu^*)^{-1}(B_j^*)$ to the balls $\text{ext}(B_i, 10r_i)$ for $i \in T_j$ using this new budget.

Therefore, the last phase of the algorithm consists in building a set of “replacement” balls \mathcal{B}_2 for the balls B_j^* with $j \notin I_1$ by guessing T_j and building the intersection P_j to take greedily a ball of radius r_j inside that set (see Lemma 14). This is done sequentially, and the set I_2 will contain all indices j for which \mathcal{B}_2 contains a replacement ball B_j for B_j^* . An important remark here is that the properties required for balls in \mathcal{B}_2 are dependent on the balls in \mathcal{B}_1 and not just the optimal balls. For technical reasons, we might have to add a new ball in \mathcal{B}_1 during the process of building \mathcal{B}_2 , in which case we cannot guarantee that the properties of balls in \mathcal{B}_2 hold anymore. If this happens, the algorithm will then erase all the choices of \mathcal{B}_2 and I_2 and start the second phase again. However, since we only do this when I_1 gets bigger, this is done at most k times before $I_1 = [k]$. The algorithm stops when the sets I_1 and I_2 contains all indices of $[k]$ which means that each ball B_j^* either has an approximate ball in \mathcal{B}_1 or a replacement ball in \mathcal{B}_2 .

The algorithm

As explained previously, the algorithm maintains two disjoint sets of indices I_1 and I_2 , initially set to \emptyset , as well as two sets of balls \mathcal{B}_1 and \mathcal{B}_2 , also initially set to \emptyset . \mathcal{B}_1 and \mathcal{B}_2 will eventually contain a representative ball B_i for every ball B_i^* in the optimal solution. Moreover, we will argue that there exists a valid assignment of the points to the balls (with an expansion) in the union of these two sets.

For every $i \in I_2$, let T_i denote the set of indices j of I_1 , such that $B_i^* \cap B_j$ is not empty, and P_i denote the intersection of the extensions $\text{ext}(B_j, 10r_j)$ over all $j \in T_i$ after removing the points of B_s for $s \in I_1 \setminus T_i$.

We say that the sets $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ form a *valid configuration* if the following properties are satisfied.

- For every $i \in I_1$, there is an approximate ball $B_i \in \mathcal{B}_1$ of B_i^* of radius at most $5r_i$.
- For every $i \in I_2$, T_i is non-empty, $B_i^* \subseteq P_i$ and there exists a ball $B_i \in \mathcal{B}_2$ of center x_i and radius r_i , such that both η_{x_i} and $B_i \cap P_i$ have size at least $|(\mu^*)^{-1}(B_i^*)|$.
- For $i, j \in I_2$, B_i and B_j do not intersect.
- For every $i \in I_2$ and $s \notin I_1$, B_s^* and B_i do not intersect.
- For every $j \in [k]$, if c_j^* is a center of a ball in \mathcal{B}_1 (respectively, \mathcal{B}_2), then $j \in I_1$ (respectively, I_2) and c_j^* is the center of B_j .

Before describing the algorithm to construct a valid configuration $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ such that $I_1 \cup I_2 = [k]$, let us show that such a configuration would indeed yield an approximate solution.

► **Lemma 10.** *Let $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ be a valid configuration such that $I_1 \cup I_2 = [k]$, then the set of balls \mathcal{B} containing the balls in \mathcal{B}_2 , as well as for every $i \in I_1$ the ball $\text{ext}(B_i, 10r_i)$ is a 15-approximate solution.*

Proof. The fact that the sum of radii of the balls in \mathcal{B} is at most 15 times the optimal solution follows from the definition. To prove the lemma, we have to show that this is a valid solution by giving a valid assignment. Recall that μ^* is the assignment for \mathcal{B}^* .

For every $i \in I_2$, recall that T_i denotes the set of indices j of I_1 such that B_i^* intersects B_j and P_i denotes the intersection of all the $\text{ext}(B_j, 10r_j)$ for $j \in T_i$ where we removed the points in B_s for $s \in I_1 \setminus T_i$ for $j \in I_1$. By definition of a valid configuration, if we use Y_i to denote the set $(\mu^*)^{-1}(B_i^*)$, then there exists a set X_i of size $|Y_i|$ in $B_i \cap P_i$. The following claim is a crucial part of the proof.

▷ Claim 11. Any point $x \in X_i$ is such that $\mu^*(x) = B_j^*$ for some $j \in T_i$

Proof. Indeed, by definition, the only balls $B_j \in \mathcal{B}_1$ containing an element x of P_i are such that $j \in T_i$, so in particular if $j \in I_1 \setminus T_i$, $B_j^* \subseteq B_j$ doesn't contain x . Moreover, if $j \in I_2$, then we know by definition that $B_j^* \cap B_i$ is empty (that includes B_i^*). Since $x \in B_i$, this concludes the proof of our claim. ◁

The previous claim implies that, if we define for every $j \in T_i$ the set $X_{i,j} := X_i \cap (\mu^*)^{-1}(B_j^*)$, then the $X_{i,j}$ for $j \in T_i$ actually defines a partition of X_i . As $|X_i| = |Y_i|$, we can partition the set Y_i into sets $Y_{i,j}$ for $j \in T_i$ such that $|X_{i,j}| = |Y_{i,j}|$ for all $j \in T_i$. Remember that, for $j \in T_i$, $Y_i \subseteq \text{ext}(B_j, 10r_j)$. By convention, if $j \in T_1 \setminus T_i$, then $X_{i,j}$ and $Y_{i,j}$ are defined as the empty set.

For every $j \in I_1$, we can now define $L_j = ((\mu^*)^{-1}(B_j^*) \setminus (\cup_{i \in I_2} X_{i,j})) \cup_{i \in I_2} Y_{i,j}$. Since the sets B_i , for $i \in I_2$, are pair-wise disjoint and $|X_{i,j}| = |Y_{i,j}|$ for all elements $j \in I_1$ and $i \in I_2$, we have that $|L_j| = |(\mu^*)^{-1}(B_j^*)|$. Moreover, since $Y_{i,j}$ is non empty only if $Y_i \subseteq \text{ext}(B_j, 10r_j)$, it means that $L_j \subseteq \text{ext}(B_j, 10r_j)$ and because B_j is an approximate ball of B_j^* , it means that the center x_j of B_j is such that $\eta_{x_j} \geq |(\mu^*)^{-1}(B_j^*)| = |L_j|$.

Finally this means that the function μ such that $\mu^{-1}(\text{ext}(B_j, 10r_j)) = L_j$ for all $j \in I_1$ and $\mu^{-1}(B_i) = X_i$ for all $i \in I_2$ is a valid assignment from P to \mathcal{B} , which ends the proof. ◀

Now, we describe the algorithm that constructs the desired configuration. The first phase of the algorithm will consist of a greedy selection of elements in I_1 , such that the union of \mathcal{B}_1 covers P (Lemma 12). As said previously, this will not imply that we can assign points to these balls, without violating capacity constraints. The following two other lemmas (Lemma 13 and 14) will be used to achieve that.

As we deal with hard capacities, we cannot reuse any center. We need the following definition to enforce that. We call a point $p \in P$ an *available center*, if p has not already been selected as a center of a ball in \mathcal{B}_1 or \mathcal{B}_2 .

► **Lemma 12.** *If $(I_1, I_2 = \emptyset, \mathcal{B}_1, \mathcal{B}_2 = \emptyset)$ is a valid configuration such that the union of the balls in \mathcal{B}_1 do not cover P , then there exists a randomized algorithm, running in linear time and with probability at least $1/2k^2$, that finds an index s and a ball B_s such that adding s to I_1 and B_s to \mathcal{B} still yields a valid configuration.*

Proof. Let x be any point in P not covered by the union of the balls in \mathcal{B}_1 and i be the index such that $\mu^*(x) = B_i^*$. Let c be the available potential center in P at distance at most r_i from x which maximises the value of η_c . If c is not a center of some B_j^* for $j \notin (I_1 \cup I_2)$, then the ball B_i of center c and radius $3r_i$ is an approximate ball of B_i^* and thus adding B_i to \mathcal{B}_1 and i to I_1 yields a valid configuration. If c is a center of some B_j^* for $j \notin (I_1 \cup I_2)$, then adding the ball B_j of center c and radius r_j to \mathcal{B}_1 and j to I_1 also yields a valid configuration.

The algorithm will then pick uniformly at random an index $i' \in [k]$, then decide uniformly at random whether the available center c' at distance at most $r_{i'}$ is a center of some B_j^* for some $j \notin (I_1 \cup I_2)$. If it decides negatively, then the algorithm will output $s := i'$ and B_s the ball of center c' and radius $3r_{i'}$. If it decides positively, then the algorithm will then also pick uniformly at random and index $j' \in [k]$ and output $s := j'$ as well as B_s the ball of center c' and radius $r_{j'}$.

The algorithm then succeeds if $i' = i$, if it decides correctly if c' is a center of some B_j^* and if $j' = j$ in the case where it is. Overall this is true with probability at least $\frac{1}{k \cdot 2 \cdot k}$, which ends the proof. ◀

12:10 FPT Approx. for Capacitated Sum of Radii

The first phase of the algorithm consists of applying the algorithm from Lemma 12 until the union of the balls in \mathcal{B}_1 covers all the points in P . The next two lemmas are used in the next phase of the algorithm.

► **Lemma 13.** *If $(I_1, I_2 = \emptyset, \mathcal{B}_1, \mathcal{B}_2 = \emptyset)$ is a valid configuration such that the balls in \mathcal{B}_1 cover the points of P and there exist two indices $i \in I_1$ and $j \in [k] \setminus (I_1 \cup I_2)$ such that B_i and B_j^* intersect and $r_j \geq 5r_i$, then there exists a randomized algorithm that in linear time and with probability at least $1/2k$, finds an index $t \in [k] \setminus (I_1 \cup I_2)$ and a ball B_t such that $(I_1 \cup \{t\}, I_2, \mathcal{B}_1 \cup \{B_t\}, \mathcal{B}_2)$ is a valid configuration.*

Proof. Let x_i denote the center of B_i , and B' be the ball of center x_i and radius $5r_i + r_j$. Because B_i is an approximate ball of B_i^* , and B_j^* and B_i intersect, we have that B' contains c_j^* . Let x be the potential center of B' which maximises η_x . If there exists an index $j' \in [k] \setminus (I_1 \cup I_2)$, such that the ball $B_{j'}$ is centered at x , then $t := j'$ and the ball B_t of center x and radius $r_{j'}$, satisfy the property of the lemma (remember that $\mathcal{B}_2 = \emptyset$). If not, then the ball B_j at center x and of radius $2 \cdot (5r_i + r_j) + r_j$ is an approximate ball of B_j^* of radius at most $5r_j$. Indeed, because the ball at center x_i and of radius $(5r_i + r_j)$ contains both x and c_j^* , it means that the ball at center x and of radius $2(5r_i + r_j)$ contains c_j^* and thus $B_j^* \subseteq B_j$. Again, as $\mathcal{B}_2 = \emptyset$, then $t := j$ and $B_t := B_j$ satisfy the properties of the lemma.

The algorithm therefore consists of choosing uniformly at random which of the two cases is true. In the first case it also chooses uniformly at random an index $j_1 \in [k]$ and outputs $t := j_1$ as well as the ball B_t of center x and radius r_{j_1} . In the second case it outputs $t := j$ as well as the ball B_t of center x and radius $2 \cdot (5r_i + r_j) + r_j$. The previous discussion implies that the algorithm succeeds if it chooses correctly between the two cases, and in the first case if $j_1 = j'$. Overall, the probability of success is at least $1/2k$. ◀

► **Lemma 14.** *Suppose $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ is a valid configuration with the property that the balls in \mathcal{B}_1 cover the points of P and for every $i \in [k] \setminus (I_1 \cup I_2)$ and $j \in I_1$, such that B_j and B_i^* intersect, $r_i \leq 5r_j$. Then there exists a randomized algorithm that in linear time and with probability at least $1/4k^2$, either finds an index $t \in [k] \setminus (I_1 \cup I_2)$ and a ball B_t such that $(I_1 \cup \{t\}, I_2 = \emptyset, \mathcal{B}_1 \cup \{B_t\}, \mathcal{B}_2 = \emptyset)$ is a valid configuration, or finds an index $s \in [k] \setminus (I_1 \cup I_2)$ and a ball B_s such that $(I_1, I_2 \cup \{s\}, \mathcal{B}_1, \mathcal{B}_2 \cup \{B_s\})$ is a valid configuration.*

Proof. Let i be the element of $[k] \setminus (I_1 \cup I_2)$ minimizing r_i , and let T_i denote the set of indices $j \in I_1$ such that $B_j \cap B_i^*$ is non-empty. By the hypothesis of the lemma, we have that $r_i \leq 5r_j$ for each element $j \in T_i$. In particular, it means that $B_i^* \subseteq \text{ext}(B_j, 10r_j)$ for every $j \in T_i$. Let P_i denotes the intersection of all those sets $\text{ext}(B_j, 10r_j)$ where we removed B_s for all $s \in I_1 \setminus T_i$. Let x be the available center in P_i such that, denoting B_x the ball at center x and of radius r_i , B_x is disjoint from all the elements in \mathcal{B}_2 and $s_x = \min\{\eta_x, |B_x \cap P_i|\}$ is maximized. Note that because $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ is a valid configuration, x_i^* is an available center in P_i and B_i^* does not intersect any of the balls in \mathcal{B}_2 . This implies that $s_x \geq |(\mu^*)^{-1}(B_i^*)|$. If B_x does not intersect any ball $B_{i'}$ with $i' \in [k] \setminus (I_1 \cup I_2)$, then by the above discussion we have that setting $B_i = B_x$, $(I_1, I_2 \cup \{i\}, \mathcal{B}_1, \mathcal{B}_2 \cup \{B_i\})$ is a valid configuration.

Suppose now that B_x intersects some $B_{i'}$ with $i' \in [k] \setminus (I_1 \cup I_2)$. (i' can also be i .) In that case, the ball at center x and of radius $r_i + r_{i'}$ contains $c_{i'}^*$. Since $r_i \leq r_{i'}$ by the choice of i , this means that the ball $B_{i'}$ of center x and radius $r_i + 2r_{i'} \leq 3r_{i'}$ is an approximate ball of $B_{i'}^*$. There might be several options for i' , but we can just make an arbitrary choice. The only thing to be careful about is if $x = c_{i''}^*$ for some $i'' \in [k] \setminus (I_1 \cup I_2)$, then the algorithm will pick that index and add the ball $B_{i''}$ of center x and radius $r_{i''}$ to \mathcal{B}_1 . In any case, $(I_1 \cup \{i'\}, I_2 = \emptyset, \mathcal{B}_1 \cup \{B_{i'}\}, \mathcal{B}_2 = \emptyset)$ or $(I_1 \cup \{i''\}, I_2 = \emptyset, \mathcal{B}_1 \cup \{B_{i''}\}, \mathcal{B}_2 = \emptyset)$, depending on that last condition, is a valid configuration, in particular as $\mathcal{B}_2 = \emptyset$.

Finally, the algorithm just decides between these two cases randomly, and in the second case picks uniformly the index i' and then outputs the described ball and index. In the second case, it also needs to decide if there exists i'' such that $x = c_{i''}^*$, and in which case pick that index uniformly at random as well. Overall, the probability of success of this algorithm is at least $1/4k^2$. ◀

We are now ready to prove our main theorem.

Proof of Theorem 1. Let us describe the algorithm. First, it applies Lemma 9 to obtain an approximation r_i of each r_i^* with probability at least $\frac{\epsilon^k}{k^k \cdot \eta^2}$. Then the algorithm initialize a valid configuration $(I_1 = \emptyset, I_2 = \emptyset, \mathcal{B}_1 = \emptyset, \mathcal{B}_2 = \emptyset)$ and run the algorithm of Lemma 12 at most k times, until \mathcal{B}_1 covers all the points of P . At each step, the probability of success is at least $1/k^2$, so in total at least $1/k^{2k}$. Then the algorithm enters into the second phase. This phase is divided into multiple steps.

In the beginning of each step, we maintain the invariant that the current configuration is valid with $I_2 = \emptyset$ and $\mathcal{B}_2 = \emptyset$. Each step then consists of a series of applications of Lemma 13 followed by a series of applications of Lemma 14. The current step ends when an index is added to I_1 by the application of Lemma 14, and hence at that point $I_2 = \emptyset$ and $\mathcal{B}_2 = \emptyset$, or $I_1 \cup I_2 = [k]$. We go to the next step (maintaining the invariant), unless $I_1 \cup I_2 = [k]$ in which case the algorithm terminates.

In a step, the algorithm decides which lemma to apply as long as $I_2 = \emptyset$. Otherwise, it applies only Lemma 14. If $I_2 = \emptyset$, it randomly decides if there exists indices $i \in I_1$ and $j \in [k] \setminus (I_1 \cup I_2)$ such that B_i^* and B_j^* intersect and $r_j \geq 5r_i$. In which case, the algorithm applies Lemma 13 to increase the size of $|I_1|$ in linear time and with probability at least $1/2^2k$. If no such pair of indices exists, then the algorithm applies Lemma 14 to increase $|I_1 \cup I_2|$ or $|I_1|$ in linear time and with probability at least $1/2k^2$.

▷ **Claim 15.** Assuming the algorithm made all the correct random choices, it terminates with a valid configuration $(I_1, I_2, \mathcal{B}_1, \mathcal{B}_2)$ such that $I_1 \cup I_2 = [k]$ after $O(k^2)$ applications of Lemma 13 and 14.

Proof. First, we argue about the maximum number of applications of the two lemmas. Note that in each step, we add at least one index to I_1 and then only go to the next step. Also, once an index is added to I_1 it is never removed. Thus, the total number of steps is at most k . Also, in each step, everytime we apply a lemma, the size of $I_1 \cup I_2$ gets increased, which can be at most k . Thus, in each step we will apply the lemmas at most k times in total. Hence, the total number of applications of both lemmas is $O(k^2)$.

Next, we prove that the algorithm terminates with the desired configuration. Fix a step. Note that if $I_2 = \emptyset$ and we make correct choices, we can correctly apply a lemma and make progress. This is true, as the conditions in the two lemmas are complementary. Now, if $I_2 \neq \emptyset$, then we have applied Lemma 14 at least once. This implies for every $i \in [k] \setminus (I_1 \cup I_2)$ and $j \in I_1$, such that B_j and B_i^* intersect, $r_i \leq 5r_j$. Hence, the condition of Lemma 13 is false for the current set I_1 . Now, we do not change I_1 throughout a step once we apply Lemma 14, except at the last time, in which case we go to the next step emptying I_2 . Thus, once $I_2 \neq \emptyset$, throughout the step, it holds that for every $i \in [k] \setminus (I_1 \cup I_2)$ and $j \in I_1$, such that B_j and B_i^* intersect, $r_i \leq 5r_j$. Hence, we can always apply Lemma 14 and make progress.

Now, consider the last step, we prove that at the end of this step $I_1 \cup I_2 = [k]$. By the above discussion, this step ends either if $I_1 \cup I_2$ becomes $[k]$ or an index is added to I_1 . In the latter case, we go to the next step. However, as the current step is the last one, it must be the case that $I_1 \cup I_2 = [k]$. This completes the proof of the claim, as we always maintain a valid configuration. ◀

If the algorithm made all the correct random choices, by the above claim together with Lemma 10, we can get a 15 -approximate solution from \mathcal{B}_1 and \mathcal{B}_2 . The algorithm runs in linear time and succeeds with probability at least $\frac{\epsilon^k}{k^k \cdot n^2} \cdot \frac{1}{k^{2k}} \cdot \frac{1}{(2k^3)^{k^2}}$. This means that running the previous algorithm $2^{\mathcal{O}(k^2 \log k)} \cdot n^2$ times, we obtain a $(15 + \epsilon)$ -approximation algorithm with constant probability. Lastly, it is not hard to derandomize this algorithm by performing exhaustive searches in each step instead of making decisions randomly. The time bound still remains the same. ◀

3 Conclusions

In this paper, considering the CAPACITATED SUM OF RADII problem, we obtained the first constant-factor $(15 + \epsilon)$ approximation algorithm that runs in FPT time, making significant progress towards understanding the barriers of capacitated clustering. While our techniques are tailor-made for FPT type results, we hope some of the ideas will also be useful in obtaining a similar approximation in polynomial time. We leave this as an open question.

Question 1: Does CAPACITATED SUM OF RADII admit a polynomial time constant-factor approximation algorithm, even with uniform capacities?

For the problem with uniform capacities, we obtained improved approximation bounds of $4 + \epsilon$ and $2 + \epsilon$ in general and Euclidean metric spaces, respectively. We also showed hardness bounds in both general and Euclidean metric spaces complementing our approximation results. The following two natural open questions are left by our work.

Question 2: What is the best constant-factor approximation possible for CAPACITATED SUM OF RADII or uniform CAPACITATED SUM OF RADII in FPT time?

Question 3: Does Euclidean CAPACITATED SUM OF RADII admit an $(1 + \epsilon)$ -approximation algorithm, in $f(k, \epsilon) \cdot n^{g(\epsilon)}$ time for some functions f and g ?

References

- 1 Marek Adamczyk, Jaroslaw Byrka, Jan Marcinkowski, Syed Mohammad Meesum, and Michal Wlodarczyk. Constant-factor FPT approximation for capacitated k-median. In Michael A. Bender, Ola Svensson, and Grzegorz Herman, editors, *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany*, volume 144 of *LIPICs*, pages 1:1–1:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ESA.2019.1.
- 2 Sara Ahmadian, Ashkan Norouzi-Fard, Ola Svensson, and Justin Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. *SIAM Journal on Computing*, 49(4):FOCS17–97, 2019.
- 3 Hyung-Chan An, Aditya Bhaskara, Chandra Chekuri, Shalmoli Gupta, Vivek Madan, and Ola Svensson. Centrality of trees for capacitated k-center. *Math. Program.*, 154(1-2):29–53, 2015. doi:10.1007/s10107-014-0857-y.
- 4 Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.
- 5 Mihai Badoui, Sarel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002*,

- Montréal, Québec, Canada, STOC '02, pages 250–257, New York, NY, USA, 2002. Association for Computing Machinery. doi:10.1145/509907.509947.
- 6 Judit Barilan, Guy Kortsarz, and David Peleg. How to allocate network centers. *Journal of Algorithms*, 15(3):385–415, 1993.
 - 7 Anup Bhattacharya, Ragesh Jaiswal, and Amit Kumar. Faster algorithms for the constrained k -means problem. *Theory Comput. Syst.*, 62(1):93–115, 2018. doi:10.1007/s00224-017-9820-7.
 - 8 Jaroslav Byrka, Krzysztof Fleszar, Bartosz Rybicki, and Joachim Spoerhase. Bi-factor approximation algorithms for hard capacitated k -median problems. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 722–736. SIAM, 2015.
 - 9 Jarosław Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k -median, and positive correlation in budgeted optimization. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 737–756. SIAM, 2014.
 - 10 Jaroslav Byrka, Bartosz Rybicki, and Sumedha Uniyal. An approximation algorithm for uniform capacitated k -median problem with $1+\epsilon$ capacity violation. In Quentin Louveaux and Martin Skutella, editors, *Integer Programming and Combinatorial Optimization - 18th International Conference, IPCO 2016, Liège, Belgium, June 1-3, 2016, Proceedings*, volume 9682 of *Lecture Notes in Computer Science*, pages 262–274. Springer, 2016.
 - 11 Moses Charikar, Sudipto Guha, Éva Tardos, and David B. Shmoys. A constant-factor approximation algorithm for the k -median problem. *J. Comput. Syst. Sci.*, 65(1):129–149, 2002.
 - 12 Moses Charikar and Rina Panigrahy. Clustering to minimize the sum of cluster diameters. *J. Comput. Syst. Sci.*, 68(2):417–441, 2004. doi:10.1016/j.jcss.2003.07.014.
 - 13 Julia Chuzhoy and Yuval Rabani. Approximating k -median with non-uniform capacities. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2005, Vancouver, British Columbia, Canada, January 23-25, 2005*, pages 952–958. SIAM, 2005.
 - 14 Vincent Cohen-Addad. Approximation schemes for capacitated clustering in doubling metrics. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2241–2259. SIAM, 2020. doi:10.1137/1.9781611975994.138.
 - 15 Vincent Cohen-Addad, Anupam Gupta, Amit Kumar, Euiwoong Lee, and Jason Li. Tight fpt approximations for k -median and k -means. *arXiv preprint arXiv:1904.12334*, 2019.
 - 16 Vincent Cohen-Addad and CS Karthik. Inapproximability of clustering in l_p metrics. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 519–539. IEEE, 2019.
 - 17 Vincent Cohen-Addad, CS Karthik, and Euiwoong Lee. On approximability of clustering problems without candidate centers. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2635–2648. SIAM, 2021.
 - 18 Vincent Cohen-Addad and Euiwoong Lee. Johnson coverage hypothesis: Inapproximability of k -means and k -median in l_p -metrics. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1493–1530. SIAM, 2022.
 - 19 Vincent Cohen-Addad and Jason Li. On the fixed-parameter tractability of capacitated clustering. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming, ICALP 2019, July 9-12, 2019, Patras, Greece*, volume 132 of *LIPICs*, pages 41:1–41:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.ICALP.2019.41.
 - 20 Marek Cygan, MohammadTaghi Hajiaghayi, and Samir Khuller. LP rounding for k -centers with non-uniform hard capacities. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 273–282. IEEE Computer Society, 2012. doi:10.1109/FOCS.2012.63.

- 21 H. Gökalp Demirci and Shi Li. Constant approximation for capacitated k -median with $(1+\epsilon)$ -capacity violation. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 73:1–73:14, 2016.
- 22 Hu Ding and Jinhui Xu. A unified framework for clustering constrained data without locality property. *Algorithmica*, 82(4):808–852, 2020. doi:10.1007/s00453-019-00616-2.
- 23 Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi R. Varadarajan. On metric clustering to minimize the sum of radii. *Algorithmica*, 57(3):484–498, 2010. doi:10.1007/s00453-009-9282-7.
- 24 Matt Gibson, Gaurav Kanade, Erik Krohn, Imran A. Pirwani, and Kasturi R. Varadarajan. On clustering to minimize the sum of radii. *SIAM J. Comput.*, 41(1):47–60, 2012. doi:10.1137/100798144.
- 25 Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.
- 26 Sudipto Guha and Samir Khuller. Greedy strikes back: Improved facility location algorithms. *Journal of algorithms*, 31(1):228–248, 1999.
- 27 Anupam Gupta, Euiwoong Lee, and Jason Li. An fpt algorithm beating 2-approximation for k -cut. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2821–2837. SIAM, 2018.
- 28 Anupam Gupta, Euiwoong Lee, Jason Li, Pasin Manurangsi, and Michał Włodarczyk. Losing treewidth by separating subsets. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1731–1749. SIAM, 2019.
- 29 Tanmay Inamdar and Kasturi R. Varadarajan. Capacitated sum-of-radii clustering: An FPT approximation. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPICs*, pages 62:1–62:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ESA.2020.62.
- 30 Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k -means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.
- 31 Samir Khuller and Yoram J. Sussmann. The capacitated K -center problem. *SIAM J. Discret. Math.*, 13(3):403–418, 2000. doi:10.1137/S0895480197329776.
- 32 Euiwoong Lee. Partitioning a graph into small pieces with applications to path transversal. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1546–1558. SIAM, 2017.
- 33 Shi Li. On uniform capacitated k -median beyond the natural LP relaxation. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 696–707, 2015.
- 34 Shi Li. On uniform capacitated k -median beyond the natural LP relaxation. *ACM Trans. Algorithms*, 13(2):22:1–22:18, 2017.
- 35 Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- 36 Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining Workshop on Text Mining*, pages 525–526, 2000.
- 37 Yicheng Xu, Yong Zhang, and Yifei Zou. A constant parameterized approximation for hard-capacitated k -means. *CoRR*, abs/1901.04628, 2019. arXiv:1901.04628.

Multilevel Skeletonization Using Local Separators

J. Andreas Bærentzen  

Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

Rasmus Emil Christensen

Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

Emil Toftegaard Gæde  

Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

Eva Rotenberg  

Department of Applied Mathematics and Computer Science, Technical University of Denmark, Lyngby, Denmark

Abstract

In this paper we give a new, efficient algorithm for computing curve skeletons, based on local separators. Our efficiency stems from a multilevel approach, where we solve small problems across levels of detail and combine these in order to quickly obtain a skeleton. We do this in a highly modular fashion, ensuring complete flexibility in adapting the algorithm for specific types of input or for otherwise targeting specific applications.

Separator based skeletonization was first proposed by Bærentzen and Rotenberg in [ACM Transactions on Graphics '21], showing high quality output at the cost of running times which become prohibitive for large inputs. Our new approach retains the high quality output, and applicability to any spatially embedded graph, while being orders of magnitude faster for all practical purposes.

We test our skeletonization algorithm for efficiency and quality in practice, comparing it to local separator skeletonization on the University of Groningen Skeletonization Benchmark [Telea'16].

2012 ACM Subject Classification Computing methodologies → Computer graphics; Theory of computation → Computational geometry; Software and its engineering → Software design engineering

Keywords and phrases Algorithm engineering, experimentation and implementation, shape skeletonization, curve skeletons, multilevel algorithm

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.13

Related Version *Full Version*: <https://arxiv.org/abs/2303.07210>

Supplementary Material

Software: <https://github.com/janba/GEL> [10]

archived at sw.h1.dir:91f125aa9a06d7adf992cdb11ca2108d86acdefe

Software (Supplementary repo for testing and additional variations): <https://github.com/Sgelet/GEL>, archived at sw.h1.dir:09aadd7e8f82cc8d8dfdbbc72ea71e56e65a6e0cfc

Funding *Emil Toftegaard Gæde* and *Eva Rotenberg*: supported by Eva Rotenberg's Carlsberg Foundation Young Researcher Fellowship CF21-0302 - "Graph Algorithms with Geometric Applications". *J. Andreas Bærentzen*: Partially supported by the Villum Foundation through Villum Investigator Project InnoTop.

1 Introduction

A curve skeleton is a compact simplified representation of a shape, consisting only of curves. The act of *skeletonization*, in this context, is the computation of such a curve skeleton for a given input. For the remainder of this paper *skeleton* refers exclusively to curve skeletons. Various fields, including feature extraction, visualisation and medical imaging, care not only



© J. Andreas Bærentzen, Rasmus Emil Christensen, Emil Toftegaard Gæde, and Eva Rotenberg;

licensed under Creative Commons License CC-BY 4.0

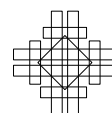
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 13; pp. 13:1–13:18

Leibniz International Proceedings in Informatics

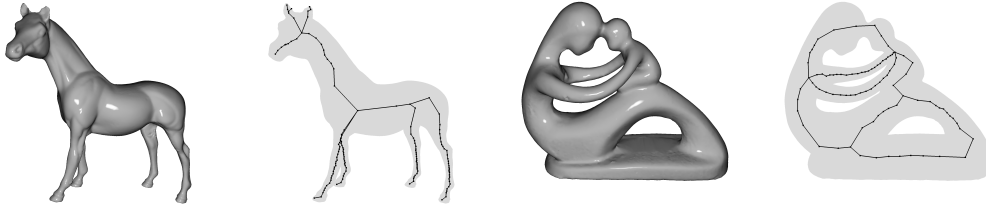


Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



about shapes and objects, but also about their structures and features. In applications such as shape matching, the skeleton acts as a simplified representation of an object, allowing for reduced computation cost [8], whereas in virtual navigation the curve skeleton can act as a collision free navigational structure [24, 30].

The broad areas of application, and the different roles that skeletons play, lead to differing interpretations of exactly what the skeleton is. Although no widely agreed upon definition of skeletons exist, work has been done on narrowing down desirable properties of skeletons in the general case [12].



■ **Figure 1** Shaded renders of triangle meshes and skeletons obtained by our algorithm.

Instead of giving a formal definition, we will base our work on the evocative if imprecise definition of skeletons as simplified curve representations of the underlying structure and topology. In Figure 1 we show skeletons of various input, to exemplify our definition.

Many different approaches to skeletonization exist [27], such as computing and pruning the medial surface [13], computing mean curvature flow [26] or contracting meshes [20]. In a recent paper A. Bærentzen and E. Rotenberg present a new algorithm that bases itself on computing *local separators* [5]. We refer to this algorithm as the local separator skeletonization algorithm, *LSS*. This approach has the benefit that it requires only that the input be given as a spatially embedded graph, rather than a specific shape representation. This makes the method applicable to a wide variety of inputs, such as meshes, voxel grids or even input that does not necessarily represent a shape. In addition, the skeletons that it generates are of high quality, capturing features that contractive methods tend to miss. However, the algorithm is also computationally expensive.

In this paper we present a multilevel algorithm for computing curve skeletons that we obtain by adapting *LSS* to a multilevel framework. Below, we start with some preliminaries and then present an overview of our contributions. Next, after a discussion of related work, we describe our approach to graph coarsening, projecting separators onto finer level graphs, and, finally, the multilevel skeletonization algorithm that builds on these components. We provide analyses of the algorithms in the paper and we test our work on a skeletonization benchmark. Our results show that our algorithm is orders of magnitude faster than that proposed by Bærentzen and Rotenberg while producing skeletons of comparable quality.

1.1 Preliminaries

We consider the discrete skeletonization problem, where both the input and output is represented by spatially embedded undirected graphs. Formally, we consider skeletonization of a graph $G = (V, E)$ where each vertex is associated with a geometric position $p_{v \in V} \in \mathbb{R}^3$. Note that we make no other assumptions about the graph, such as whether it is sampled from the surface of a manifold, created from a point cloud, or otherwise.

In graph theory a *vertex separator* is a set of vertices whose removal disconnects the graph. In [5], this notion is extended to *local separators*, defined as a subset of vertices, $S \subset V$, that is a vertex separator of the subgraph induced by the closed neighbourhood of S . Likewise, the notion of a minimal local separator is defined as a local separator that is a minimal vertex separator of the subgraph induced by the closed neighbourhood. Intuitively, we cannot remove a vertex from a minimal local separator without the remaining set ceasing to be a local separator. For the rest of this paper, the term *separator* means local separator.

1.2 Contributions

The LSS algorithm computes skeletons through a three-phased approach. A large number of minimal local separators is computed, the minimal separators are selected using a greedy packing method, and, lastly, the skeleton is extracted from the packed set of minimal separators. A visualisation of these phases can be seen in Figure 2.



■ **Figure 2** Visualisation of the three phases of the LSS algorithm. From left to right: A shaded render of the input, a number of computed minimal separators, a non-overlapping subset of the separators, and the resulting skeleton after extraction.

As the algorithms for the first two phases play an intrinsic role in our algorithm, we give a brief description of these.

Computing local separators is done through a two-step process. First a region growing approach is used to find a local separator. A vertex is picked, and we iteratively add to the separator an adjacent vertex and check if the neighbourhood is disconnected. We refer to this as *growing* a separator. Once a local separator has been found, it is heuristically minimised by removing vertices that would not destroy the separator. We refer to this as *shrinking* a separator.

Because the running time of LSS is often dominated by the search for local separators, a sampling scheme is used to reduce computation. According to the scheme, vertices are selected for separator computation with probability 2^{-x} , where x is the number of previously computed separators that contain that vertex.

Unfortunately, sampling only addresses the number of separators that need to be computed and not the time it takes to compute each separator. In this paper we address the latter issue using a multilevel approach. Specifically, we find separators on coarser versions of the graph and project them back up onto the original graph. Importantly, this allows us to set a *patience threshold* for the amount of computation that should be used to find a separator from a given vertex. When the threshold is exceeded we stop the search relying on a separator containing the given vertex to be found on a coarser level.

1.3 Related Work

Skeletonization, in terms of computing curve skeletons, is a diverse field not only in terms of interpretations of skeletons, but also in the algorithmic approaches. Several classifications of algorithms exist [12, 27], based on underlying traits of the algorithms.

13:4 Multilevel Skeletonization Using Local Separators

The interpretation that the curve skeleton should lie on the medial surface, gives rise to methods that, in a sense, extract a curve skeleton from the medial surface of the input [13, 29, 19, 25, 32]. Since the medial surface is highly sensitive to noise, so are the skeletons generated by these methods.

A class of algorithms that are resilient to noise are the contractive methods, based on the concept of reducing the volume and surface area of the input until a skeleton is found [31, 4, 26]. In their simplest forms these algorithms require that the input be manifold, however it is possible to extend to other types of input [11, 18].

A related notion for shape analysis is that of Reeb graphs [7, 6]. These can be used for skeletonization, lending themselves to a topologically driven class of algorithms [22, 23, 14]. The resulting skeletons depend on a parameter, giving some flexibility in targeting specific properties of the output, but also requiring great care in the choice of the parameter.

In addition there are algorithms that fit into classifications not presented here [20, 15, 2].

A very successful heuristic approach to the NP-complete problem of graph partitioning is that of multilevel algorithms [9]. Although the problem considered is different, we employ a similar multilevel scheme for vastly improving practical performance. Such multilevel schemes have been extensively studied [16, 21, 1].

2 The Multilevel Framework

In its most general sense, the multilevel framework is a heuristic approach that aims to solve a problem by obtaining a solution to a smaller problem.

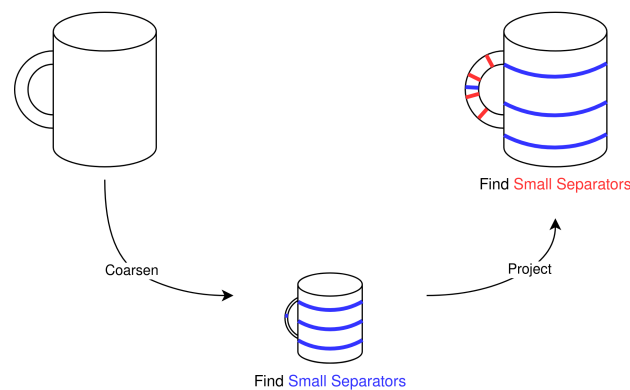
Initially, a series of increasingly simplified approximations of the input is generated. We call this the *coarsening* phase, and the series of simplifications we call *levels*. Since the last level is small, computing a solution is much faster. In graph partitioning literature, this is called the *partitioning* phase; however, we will consider it in terms of solving a *restricted* problem. Then, the solution found on the last level is transformed into a solution on the input through *uncoarsening*. This process is also sometimes called *projection and refinement*, since uncoarsening from one level to the previous is often done by projecting onto the previous level, and then employing some refinement process to improve the solution according to some heuristic.

By design, the multilevel framework is highly flexible. Various coarsening schemes can be used, that may prioritise preserving different properties of the input when simplifying. The restricted problem can be solved by any reasonable approach, and the refinement strategies can be adapted to suit the application.

Our algorithm works by first coarsening the input into several levels of decreasing resolution. The details of this coarsening is described in Section 2.1. Once the hierarchy of graphs has been generated, we do a restricted search for local separators on each level of resolution. The details are covered in Section 2.2, but the intuition is that searching for large local separators is slow in practice, and by restricting our search we save computation. Since the separators found are small, this does however also mean that we are only able to capture small features of the structure.

Since small features obtained on low resolution can represent large features on the original input, we obtain separators capturing features of varying sizes by searching for separators across every level.

By projection and refinement, see Section 2.3, we then transform the minimal local separators found across the levels into minimal local separators on the input. These can then be packed and extracted by the approach of LSS. The procedure is visualised in Figure 3.



■ **Figure 3** Visualisation of the multilevel skeletonization approach. A solid cylinder with a handle is coarsened until it is of small size. A number of small local separators are found (shown in blue), and then projected back to the original input. Searching for small local separators again yields the separators around the handle (shown in red), but separators are too large at this level to be discovered around the cylinder. We combine the separators to obtain a general solution.

2.1 Coarsening

Given as input a graph, $G = (V, E)$, we construct a sequence of increasingly simplified graphs, G_0, G_1, \dots, G_l s.t. $G_0 \succ G_1 \succ \dots \succ G_l$ where $l = O(\log n)$ and $G_i \succ G_j$ denotes that G_j is a minor of G_i , and $G_i = (V_i, E_i)$. Moreover $G_0 = G$ and $\forall i \in [0, l), |V_i| \geq 2|V_{i+1}|$.

We do this by a matching contraction scheme, in which we repeatedly construct and contract maximal matchings. Various approaches to such coarsening schemes exist in literature [21], and from these, we choose to consider *light edge matching*.

To construct G_{i+1} from G_i , greedily find a maximal matching and contract it. Such a matching can be constructed in $O(|E_i|)$ time by visiting vertices in a random order, matching them to an unmatched neighbour of smallest euclidean distance. We repeat this procedure until the number of vertices has been at least halved.

In Figure 4 we show some of the graphs obtained during coarsening of a triangle mesh resembling a statue of Neptune.



■ **Figure 4** A series of increasingly simplified approximations of `neptune.ply`, from the Groningen Skeletonization Benchmark, obtained through light edge matching contraction.

Note that by contraction we always preserve the number of connected components. This is one of the homotopy-preserving properties of the algorithm.

Theoretical Analysis

In the worst case, we may spend $O(|V_i|)$ rounds of contraction in order to reach the desired number of vertices. This is a well known problem of matching contraction schemes on general graphs, but graphs obtained from the world of geometry tend to take a small number of rounds to contract [1].

A general bound on the time spent on the coarsening phase is then $\sum_{i=0}^l (|V_i||E_i|) = |E| \sum_{i=0}^l |V_i| = O(|V||E|)$. For graphs that are contracted in a constant number of rounds we get $\sum_{i=0}^l |E_i|$ and if we furthermore have $|E_i| = O(|V_i|)$, as is the case for triangle meshes and voxel grids, the bound becomes $O(|V|)$.

2.2 Restricted Separator Search

For a given connected set of vertices, V' , we refer to the subgraph induced by vertices adjacent to V' , that are not in V' themselves, as the *front* of V' and denote it $F(V')$.

A region growing based approach to computing local separators is given in [5], where a separator, Σ , is iteratively grown until $F(\Sigma)$ is disconnected. The approach uses an enclosing ball around the vertices of Σ to guide what vertex of $F(\Sigma)$ is added next, and the connectivity of $F(\Sigma)$ is then checked by traversal. As noted by the authors, it is possible to improve performance of the search by using a dynamic connectivity data structure to maintain the front, so that a traversal in every iteration is avoided.

In addition to adapting the algorithm to use a dynamic connectivity data structure, we will also restrict the number of iterations the search performs. Given a vertex, v , set $\Sigma_0 = \emptyset, F_0 = \{v\}$, and then iteratively construct $\Sigma_i = \Sigma_{i-1} \cup \{v_i \in F(\Sigma_{i-1})\}$, where v_i is the closest neighbour of the front to an enclosing sphere around Σ_{i-1} . Maintain $F(\Sigma_i)$, update the enclosing sphere and repeat until $F(\Sigma_i)$ is disconnected or empty, or $|\Sigma_i|$ exceeds a threshold value. Pseudocode for this restricted separator search is shown in Algorithm 1.

Theoretical Analysis

We analyse the complexity of this restricted separator search in terms of the graph $G' = \Sigma \cup F(\Sigma)$ with n' vertices and m' edges, using the dynamic connectivity data structure of Holm, de Lichtenberg and Thorup with updates in amortized $O(\log^2 n')$ time [17]. Since the size of the separator is at most α and we add one vertex each iteration, we use at most α iterations selecting the closest vertex from the front, updating the bounding sphere and maintaining the dynamic connectivity structure. Selecting the closest vertex is done naively with a scan through the front, taking $O(n')$ time and updating the bounding sphere takes $O(1)$ time each iteration. This gives a running time of $O(\alpha n')$. Each edge in the dynamic connectivity structure is inserted and removed at most once, with each operation taking $O(\log^2 n')$ amortized time, totalling $O(m' \log^2 n')$. The total running time is then $O(\alpha n' + m' \log^2 n')$.

In the general case we give no better worst case bound than $O(\alpha|V| + |E| \log^2 |V|)$. For graphs of bounded maximum degree we can bound the size of the front. Let Δ be the maximum degree of G , then $n' = O(|\Sigma|\Delta) = O(\alpha\Delta)$ and $m' = O(\alpha\Delta^2)$. This gives a time of $O(\alpha^2\Delta + \alpha\Delta^2 \log^2(\alpha\Delta))$. In addition if we choose α to be a small constant, the bound is further improved to $O(\Delta^2 \log^2 \Delta)$. For graphs where $\Delta = O(1)$ as for voxel grids or knn-graphs, the search then becomes $O(1)$. Note that for this bound to be applicable across the entirety of the algorithm, the degree needs to remain bounded through coarsening.

■ **Algorithm 1** Restricted Separator Search

Given a spatially embedded graph, G , a starting vertex, v_0 , and a thresholding value, α , search for a separator of size at most α and return it, or \emptyset if failure. Here ϵ is a small constant to prevent division by zero.

```

RESTRICTED-SEPARATOR-SEARCH( $G, v_0, \alpha$ ):
   $\Sigma = \emptyset$ 
   $F = (\{v_0\}, \emptyset)$ 
   $\mathbf{c} = \mathbf{p}_{v_0}$ 
   $i = 0$ 
   $r = 0$ 
  repeat
     $v = \arg \min_{f \in V(F)} \|\mathbf{c} - \mathbf{p}_f\|$            // Scan front for closest vertex
    if  $\|\mathbf{c} - \mathbf{p}_v\| > r$  then
       $r = \frac{1}{2}(r + \|\mathbf{c} - \mathbf{p}_v\|)$            // Update the sphere
       $\mathbf{c} = \mathbf{p}_v + \frac{r}{\epsilon + \|\mathbf{c} - \mathbf{p}_v\|}(\mathbf{c} - \mathbf{p}_v)$ 
       $\Sigma = \Sigma \cup \{v\}$ 
      REMOVE( $F, v$ )
    for  $(x, y) \in E(\text{NEIGHBOURHOOD}(G, v) - \Sigma)$  do
      CONNECT( $F, x, y$ )           // Maintain the front of  $\Sigma$ 
     $i = i + 1$ 
  until NUMBER-OF-COMPONENTS( $F$ ) > 1 or  $i = \alpha$ 
  if NUMBER-OF-COMPONENTS( $F$ ) = 1 then
    return  $\emptyset$ 
  return  $\Sigma$ 

```

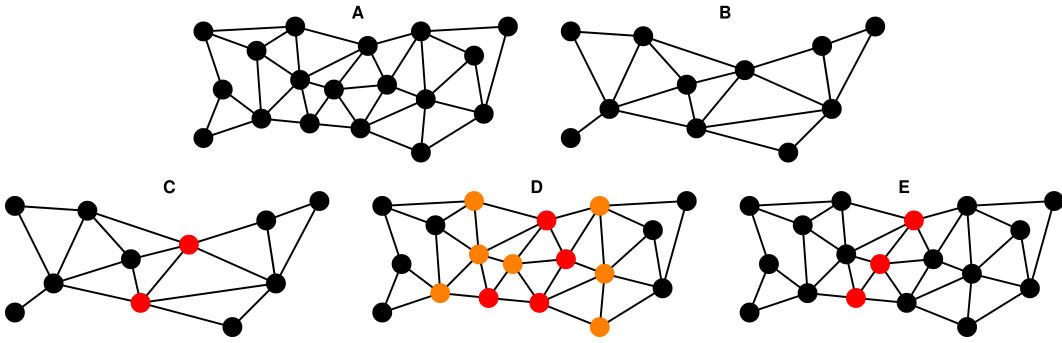
2.3 Projection and Refinement

For projecting separators to graphs of higher levels of detail, we employ a simple uncoarsening technique. By storing information about what vertices were contracted during coarsening, we can reverse the contractions that gave rise to the vertices of a given separator. Note however that a separator that has been projected in such a way is not guaranteed to be minimal.

The simplest refinement scheme is thus one that uses the algorithm for minimising separators as in LSS. The minimising algorithm is a heuristic approach that seeks to minimise a separator such that the structure becomes that of a thin band. When used on separators that are obtained through projection, there is not necessarily much room for choice. Therefore we consider a variation of our refinement scheme, we thus choose to “thicken” the separators after projection, by adding the adjacent vertices if it would not destroy the separator. This gives the heuristic minimisation more options for creating separators of shorter length, as visualised in Figure 5.

Projecting a separator can be done in linear time proportional to the size of the resulting separator, while the minimisation in worst case takes quadratic time (see the full version).

After processing each separator in this way, we obtain a set of minimal separators for the current level. If we accumulate separators indiscriminately, we will spend time projecting and refining separators that will ultimately be discarded due to overlapping. If we perform set packing on every level, we are going to be too eager in our efforts, discarding things that might not overlap once projected further. Intuitively, we would like to only discard separators if there is a large overlap.



■ **Figure 5** A separator undergoing expansion as part of refinement. (A) shows an input, (B) a coarsened representation, (C) a computed separator denoted by red vertices, (D) the projected separator denoted by red vertices and the added vertices denoted by orange, (E) shows the separator obtained by minimising the thickened separator.

To do this, we associate with each vertex, v , of every graph, G_i , a capacity, c_v^i , equal to the sum of capacities of vertices contracted to obtain it. For vertices of G_0 we define the capacities as 1, formally $\forall v \in V(G_0), c_v^0 = 1$. In this way, the capacity of a given vertex is the number of vertices of the original input that it represents.

We then modify the greedy set packing algorithm of LSS, so that we include a separator iff it would not cause any vertex to exceed its capacity. Since the capacities of G_0 are 1, this packing is equivalent to the original when applied to the highest level of resolution, and thus we will still have a non-overlapping set of separators at the end.

Note however that this packing allows for duplicates to persist through packing, essentially reducing the capacities of vertices while providing no valuable information. To counteract this, we perform a filtering step using hashing to rid duplicates prior to the packing procedure. Filtering and packing in this way takes time linear in the sum of sizes of separators in the set.

2.4 The Multilevel Skeletonization Algorithm

With the details of the phases in place, we can then combine these to construct the multilevel skeletonization algorithm. Given a spatially embedded input graph, $G = (V, E)$, and a threshold value α , we construct a curve skeleton by the following:

Generate G_i from G_{i-1} by coarsening, until $|G_l| \leq \alpha$ for some l . This generates the sequence of graphs of decreasing resolution G_0, G_1, \dots, G_l where $l = O(\log |V|)$ and $\sum_{i=0}^l |V_i| = O(|V|)$.

Then, starting at the lowest resolution, G_l , find restricted separators. We do this by the restricted separator search, starting at each vertex with probability 2^{-x} , where x is the number of currently computed separators containing that vertex, using α as the restriction on the size of the search. After computing the separators for a level, we perform capacity packing, and then we project the computed separators to the next level and refine them. This process is repeated for every level until we arrive at the original graph. At this point, after performing capacity packing, we obtain a non-overlapping set of minimal separators from which we extract the skeleton, using the extraction procedure of LSS [5]. Pseudocode for this algorithm can be seen in Algorithm 2.

■ **Algorithm 2** Multilevel Skeletonization

Given a spatially embedded graph, G , and a thresholding value, α , compute a curve skeleton.

```

MULTILEVEL-SKELETONIZATION( $G, \alpha$ ):
   $G_0 = G$ 
   $l = 0$ 
  repeat                                     // Coarsening phase
     $l = l + 1$ 
     $G_l = \text{COARSEN}(G_{l-1})$ 
  until  $|V(G_l)| \leq \alpha$ 
   $S = \emptyset$                                // Maintain set of minimal separators
  for  $i = l$  to 0 do                          // From low to high resolution
     $S' = \emptyset$ 
    for  $s \in S$  do                            // Project and refine from previous levels
       $S' = S' \cup \text{PROJECT-REFINE}(s)$ 
     $S = S'$ 
    for  $v \in V(G_i)$  with probability  $2^{-x(v)}$  do // Search on this level
       $s = \text{RESTRICTED-SEPARATOR-SEARCH}(G_i, v, \alpha)$ 
       $S = S \cup \{\text{MINIMISE-SEPARATOR}(s)\}$ 
     $S = \text{CAPACITY-PACK}(S)$ 
  return EXTRACT-SKELETON( $G, S$ )

```

Theoretical Analysis

For completeness' sake we consider then the complexity of the algorithm. Recall that the coarsening phase in the general worst case takes $O(|V||E|)$ time, but for not too irregular input takes $O(|V|)$ time. We then perform a restricted separator search from each vertex across every level, which is $O(\alpha|V|^2 + |V||E|\log^2|V|)$ in the general worst case, but $O(\alpha|V|)$ for graphs that retain constant bounded degree through coarsening. We consider then the time a single separator contributes to the total when expanding, filtering and packing. These operations are linear in the size of the separator on each level, which is worst case $O(|V_i|)$ on level i . This totals $\sum_{i=0}^l O(|V_i|) = O(|V|)$ for a single separator across all levels. Minimizing a single separator takes worst case $O(|V_i|^2)$ on level i , which for a single separator contributes $\sum_{i=0}^l O(|V_i|^2) = O(|V|^2)$ across all levels. We also perform packing on each level, linear in the sum of sizes of separators, which in total takes $\sum_{i=0}^l O(|V_i|^2) = O(|V|^2)$ time. The general worst case bound then becomes $O(|V|^3 + |V||E|\log^2|V|)$, which is an improvement over LSS. It is worth mentioning however, that in practice the running time for both LSS and our algorithm is heavily dominated by the search for separators, and that theoretically expensive procedures, such as minimisation, make up only a small fraction of the running time.

3 Experiments

In this work, our main objective was to make an algorithm that produces the same quality of skeletons as LSS [5], only with improved running times, using new algorithmic ideas and algorithm engineering. As we will show in this section, the improvements to practical running times are very satisfactory.

As for quality, it is our overall assessment that the quality has not been compromised by the speed-up.

13:10 Multilevel Skeletonization Using Local Separators

There is, however, no standard for how skeletons should be compared. In [27] it is remarked that quantifying the quality of skeletons is an open challenge, but we shall instead compare ourselves only to skeletons obtained by LSS, to quantify the deviation obtained by employing the multilevel approach.

To do this, we will measure a number of metrics, namely the number of vertices in the skeleton, the number of leaf nodes, branch nodes, chordless cycles which estimates the genus of the input, and the directed Hausdorff distance in both directions. For our comparisons, it is the relationship between directed Hausdorff distances that matter, rather than the magnitudes. A high distance from an LSS skeleton to our skeleton, with a low distance the other way, could indicate that LSS captures a feature that our skeleton does not. Likewise for the inverse, which might indicate that we are capturing a feature that LSS does not deem to exist. We give our Hausdorff distances divided by the radius of a bounding sphere, to reduce influence from the differing scales of input.

We run our tests on the Groningen Skeletonization Benchmark [28], consisting of several triangle meshes of varying structure. The tests are executed on HPC Cluster nodes with Xeon Gold 6226R (2.90GHz) CPUs, using 8 cores of a single CPU for each test. For running time measurements, tests are run three times, and the median value is reported.

For comparisons we examine three algorithms, namely the local separator skeletonization algorithm (LSS) [5], our multilevel algorithm using light edge matchings, described in Section 2.1, as contraction scheme (LEM) as well as with light edge matchings and thickened separators, described in Section 2.3, as the refinement scheme (LEMTS).

We note that the variation of LSS with which we compare our algorithms also includes usage of a dynamic connectivity structure, so that the search procedures are identical up to the threshold parameter.

Implementation

Our implementation is written in C++, built into the GEL library, and made publicly available [10]. This is the same library that contains LSS, and as such our algorithms use the same underlying data structures and subroutines. All programs are compiled using `-O3` optimisation flags. Details regarding the dynamic connectivity structure are given in the full version. We run the restricted searches in parallel internally on each level, using a simple fork-join pattern, identical to that of LSS. To account for the multilevel structure of our algorithm, we then pack using a single thread, project using at most two threads and repeat the pattern for the next level. The decision to use only two threads for projection stems from the fact that the overhead associated with spinning up threads quickly outweighs the benefits of parallel projection since there are often few separators after packing.

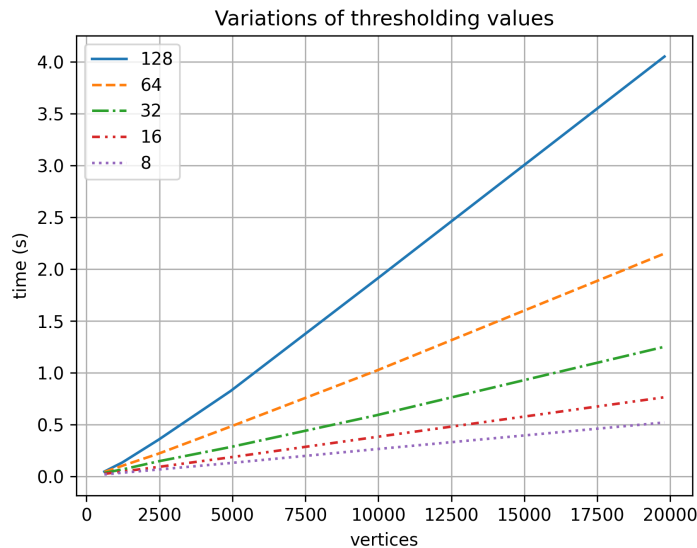
3.1 Results

Here we present our results in terms of measurements on the Groningen Skeletonization Benchmark. Initially we argue for our choice of α , showing how the threshold impacts both skeleton quality as well as running time. We then present a number of results relating to the skeletons themselves to showcase the quality of output. We then present our measurements of running times, as well as discuss interesting observations.

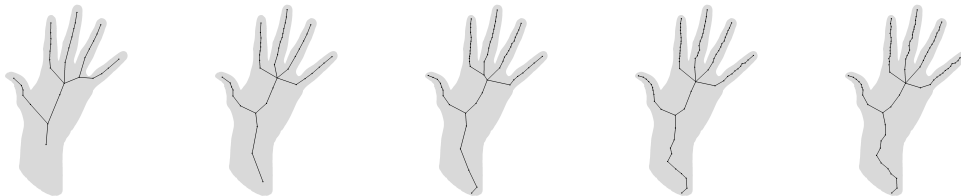
Additional measurements are presented in the full version.

The Right Amount of Patience – Determining a Threshold Value α

To show the effects of α on the running time, we consider a small test suite using subdivisions of a mesh to generate various sizes of input. This is done to ensure a similar underlying structure throughout the test. We test our multilevel algorithm for $\alpha = 8, 16, 32, 64, 128$ (see Figure 6). Not surprisingly, a lower threshold leads to a lower running time.



■ **Figure 6** Running time measurements of varying values of α on subdivisions of a mesh.



■ **Figure 7** Skeletons found on `human_hand.ply` for increasing patience thresholds. From left to right: $\alpha = 8, 16, 32, 64, 128$.

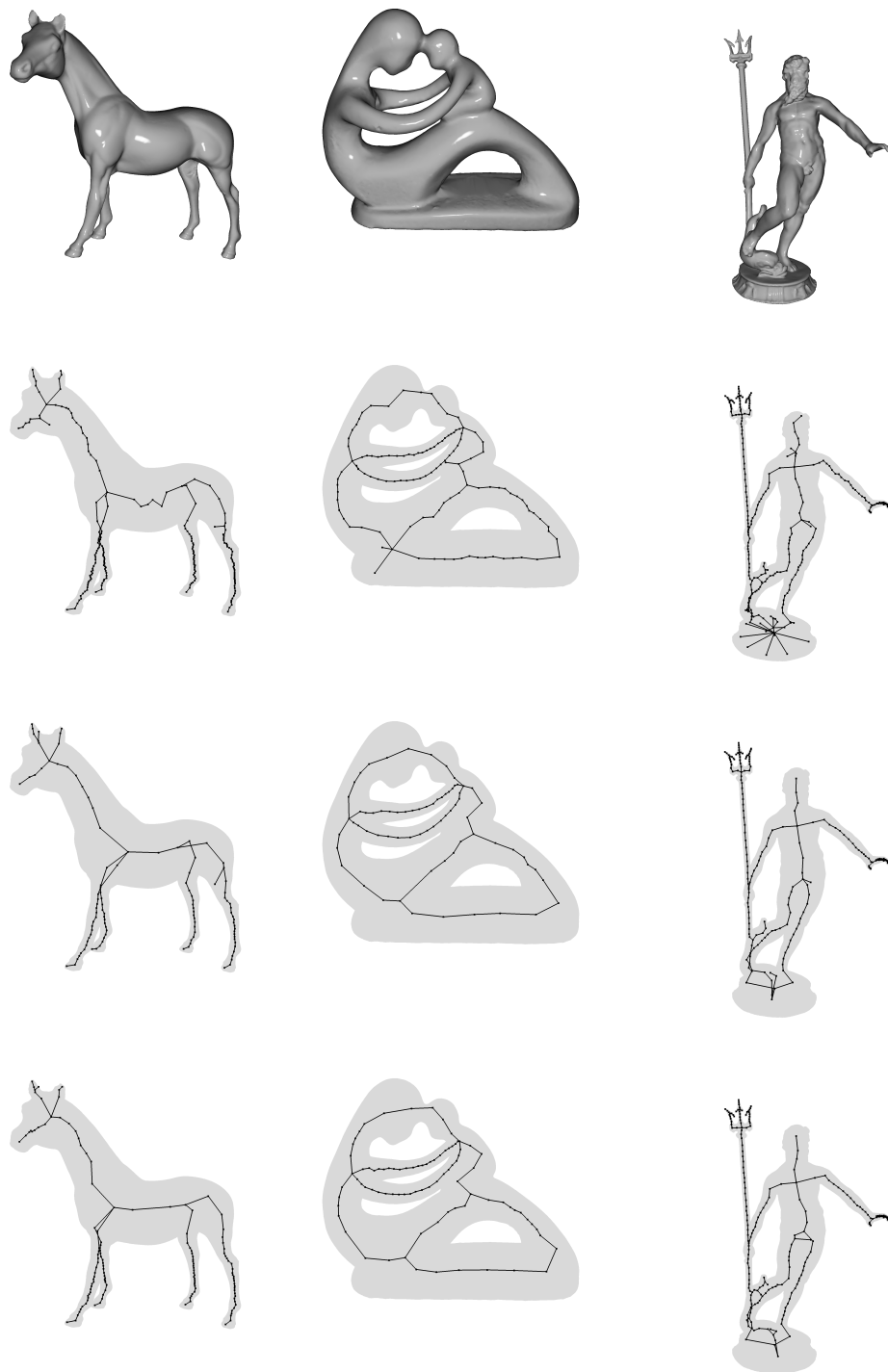
However, there may be a trade-off between skeleton quality and threshold value, which we explore qualitatively.

We visually examine the skeletons generated for our choices of α on `human_hand.ply`, as seen in Figure 7. For very low values of α , the skeletons have few curves in areas that are relatively thick. With a low threshold, separators must be found on lower resolutions, which in turn means that few separators can be found. As we progress to higher values of α , the level of detail of the skeleton rises, up to a certain point. Intuitively, if the threshold is high enough that the details can be captured on the higher levels of detail, then we gain nothing from the lower resolution levels.

It could be argued that $\alpha = 16$ or $\alpha = 32$ generates the most visually appealing skeletons for this particular input, however we find that $\alpha = 64$ offers the best trade-off for running time on other examined input such as that shown in Figure 8.

Thus, we run the remainder of our tests using $\alpha = 64$.

13:12 Multilevel Skeletonization Using Local Separators



■ **Figure 8** Each column indicates a different input, with each row showcasing a different method. From top to bottom: shaded renders of the input, skeletons obtained by LSS, skeletons obtained by LEM, skeletons obtained by LEMTS.

Skeleton Quality

In Figure 8 we show some of the skeletons obtained by LSS, LEM and LEMTS. Note that both LEM and LEMTS appear smoother, while also reaching the features that LSS finds in many cases. In addition, it seems that for these inputs our methods find less spurious features, giving a cleaner result. When comparing LEM and LEMTS the differences are subtle. On `horse.ply` it can be seen that the vertex where the front legs meet the body is positioned further to the left. This is due to LEMTS having a denser skeleton. On the other hand, LEMTS seems to not capture the structure of the groin area of `neptune.ply` as well as LEM. Although it appears as if LEMTS finds a cycle, this is not actually the case. The skeletal branch is, however, not positioned as one would expect.

■ **Table 1** Excerpt of measurements on skeletons. The metrics denoted by Δ are relative to the skeletons of LSS, with negative values implying that LSS has more vertices, leafs, branches etc. Here $H(A, B)$ denotes the directed Hausdorff distance between A and B , divided by the radius of a bounding sphere, and $*$ denotes skeletons generated by our multilevel algorithms.

input	algorithm	Δ vertices	Δ leafs	Δ branches	Δ genus	$H(LSS, *)$	$H(*, LSS)$
19465	LEM	-563	-3	-35	2	0.0437028	0.0415616
	LEMTS	-416	11	-24	3	0.0355444	0.0382796
fertility	LEM	-46	-3	-1	0	0.273804	0.0874419
	LEMTS	-30	-3	-1	0	0.240033	0.165239
happy4	LEM	-589	-409	-21	-100	0.199295	0.0887277
	LEMTS	-560	-384	-22	-99	0.160472	0.100934
horse	LEM	-95	-1	-1	0	0.112222	0.111066
	LEMTS	-66	-2	-2	0	0.0883287	0.0851157
neptune	LEM	-73	-16	-5	0	0.200924	0.0508065
	LEMTS	-51	-15	-1	0	0.225756	0.0516766

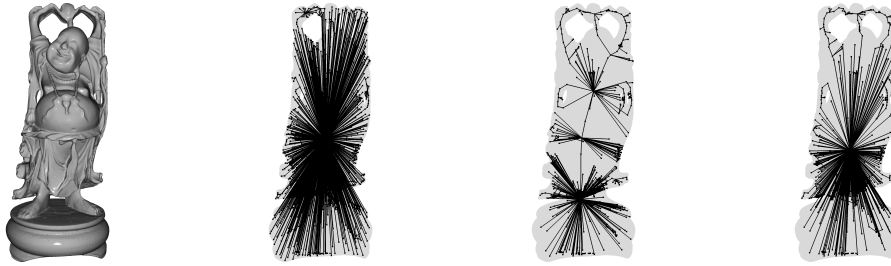
In addition, we also showcase a small excerpt of measurements from the full version, which can be seen in Table 1. Here it is clear that LEM and LEMTS produce slightly simpler skeletons with fewer vertices, leaves, and branches. However, from visual inspection of the models it is clear that (at least for the models in the table) the missing details in the skeleton correspond to features which are so subtle that the skeletal details might be considered spurious. For all inputs of the benchmark except `happy4.ply`, there is little deviation in the genus compared to LSS.

For context on the strange genus found on `happy4.ply`, we show the generated skeletons in Figure 9. Of note is that the mesh has several missing patches, which seems to cause spurious small separators to be found on all of the local separator based methods. We consider this an error case for all of the methods examined.

Running Time

Although the running time of local separator skeletonization methods depends very much on the search for separators, which in turn depends on the structure of the input, we give the running times of the examined methods in Figure 10 as a function of the number of vertices in the input, over the entirety of the Groningen Skeletonization Benchmark [28].

Remarkably, we find that the multilevel algorithm not only outperforms LSS by several orders of magnitude, but also that it seems to be less dependant on the underlying structure of the triangle meshes, giving what appears to be a slightly superlinear curve. This effect is even



■ **Figure 9** A triangle mesh with a large number of missing patches on the surface, `happy4.ply`, resulting in erroneous output for LSS, LEM, and LEMTS.

more pronounced when considering only the time to search for separators. Under assumptions about the degree of the graphs, we showed that searching was $O(1)$ for a single separator and $O(|V|)$ in total. This experiment seems to confirm that this assumption is fitting for classical input, as is the case with the triangle meshes of the Groningen Skeletonization Benchmark.

In Table 2 we show an excerpt of the running time measurements, including measurements of the phases of the algorithm. Here the vast gap in performance is clear, especially for `dragon.ply`, which is the largest input for which we have been able to run LSS, given a time frame of 20 hours. For this particular instance we achieve a running time that is almost a thousand times faster.

Of note is that LEMTS spends more time on projection, as expected, but less time on packing than LEM. As stated previously, the search for separators is often the dominating phase, however there are types of input for which this is not the case, as evidenced by `19465.ply`. The mesh consists of flat sheets with small details engraved, as can be seen in Figure 11. For both LSS and the multilevel algorithms, a large portion of the time is spent on packing and projection. This can occur if the separators are generally small and plentiful, so that many of them may quickly be found. For `19465.ply` these are particularly present around the imprinted text on the top sheets. It is worth noting that this would likely also be an example for which the structure of the input matters greatly for the running time of our multilevel algorithms.

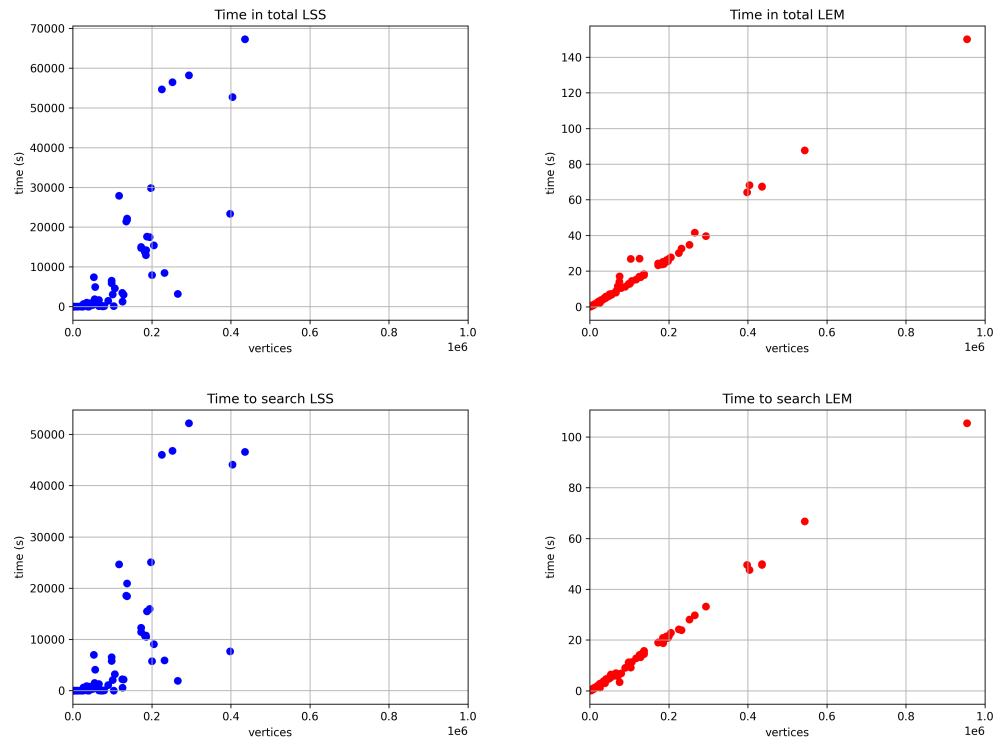
4 Conclusions and Future Work

We have proposed a multilevel algorithm for computing local separator-based curve skeletons, and shown that the approach is very efficient. We obtain a practical running time that appears near linear in the number of vertices of the input (see Figure 10) with up to thousandfold improvement in running time while not deteriorating the quality of the output substantially, if at all.

This type of running time improvement makes separator-based skeletonization applicable as a tool in biomedical image analysis, including frame-by-frame skeletonization of videos [3].

The application to video skeletonization motivates an unexplored line of related work, namely that of efficiently dynamically updating skeletons in a series of related shapes.

The multilevel approach offers great flexibility that has yet to be explored. It is easy to imagine coarsening schemes targeting specific structures of input, such as contracting clusters, rather than edges, on voxel input. These contraction schemes may provide new trade-offs between practical performance and skeleton quality.



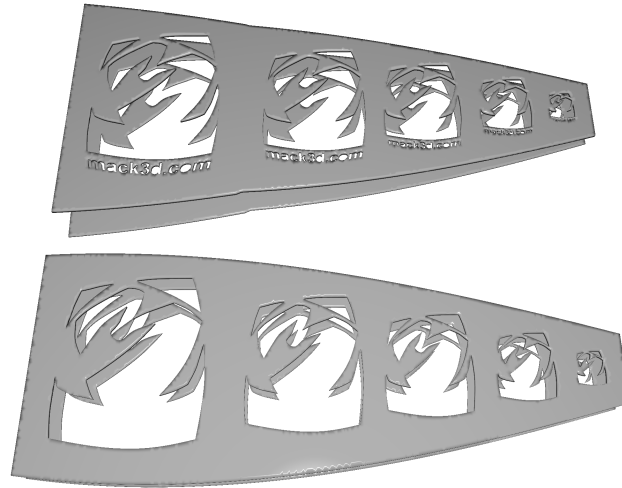
(a) Running times of LSS.

(b) Running times of LEM.

■ **Figure 10** Running times in total (top) and for searching (bottom) as function of the number of vertices on the Groningen Skeletonization Benchmark. Values over 20 hours omitted.

■ **Table 2** Excerpt of running time measurements. In addition to measuring the total time, we also measure the time spent on each phase of the algorithm.

input	algorithm	coarsen (s)	search (s)	project (s)	pack (s)	total (s)
19465	LSS	-	25.8787	-	138.836	164.714
	LEM	1.20002	9.18989	9.410914	12.4392	26.8306
	LEMETS	1.21294	9.41457	14.851733	9.8696	26.5518
dragon	LSS	-	46615.4	-	20637.9	67255.9
	LEM	6.25851	49.6567	18.753684	3.42042	67.4554
	LEMETS	6.19801	50.2347	24.6420235	3.22507	69.4672
fertility	LSS	-	95.8574	-	42.1409	136.664
	LEM	0.234681	2.38572	0.7753682	0.0600592	3.03186
	LEMETS	0.244822	2.4305	1.8177949	0.0559833	3.44687
happy4	LSS	-	7646.21	-	15731.9	23379.2
	LEM	6.89638	49.6684	12.7328477	2.22244	64.2245
	LEMETS	6.93433	50.1269	16.0581559	1.75872	65.4557
horse	LSS	-	544.182	-	85.6915	628.3
	LEM	0.504898	5.10887	1.2762728	0.0669417	6.27845
	LEMETS	0.502791	5.17779	2.7604771	0.0542464	6.86374
neptune	LSS	-	56.296	-	62.8555	119.232
	LEM	0.312728	2.7382	1.0849813	0.230939	3.77785
	LEMETS	0.308124	2.76549	2.4149765	0.182575	4.22093



■ **Figure 11** The triangle mesh 19465.ply, where packing makes up a large portion of the running time for (all) local separator based skeletonization algorithms.

When applying coarsening to scale-free graphs, as might be the case for data visualisation or areas of application that are not classical for skeletonization, we move into a domain known from the field of graph partitioning to cause trouble for matching contraction schemes [1]. It is interesting to see if the improved practical performance, and the applicability to any spatially embedded graph, opens up for new areas of application of skeletons.

References

- 1 Amine Abou-Rjeili and George Karypis. Multilevel algorithms for partitioning power-law graphs. In *20th International Parallel and Distributed Processing Symposium (IPDPS 2006), Proceedings, 25-29 April 2006, Rhodes Island, Greece*. IEEE, 2006. doi:10.1109/IPDPS.2006.1639360.
- 2 Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust. In David C. Anderson and Kunwoo Lee, editors, *Sixth ACM Symposium on Solid Modeling and Applications, Sheraton Inn, Ann Arbor, Michigan, USA, June 4-8, 2001*, pages 249–266. ACM, 2001. doi:10.1145/376957.376986.
- 3 Kasra Arnavaz, Oswin Krause, Kilian Zepf, Jakob Andreas Bærentzen, Jelena M. Krivokapic, Silja Heilmann, Pia Nyeng, and Aasa Feragen. Quantifying topology in pancreatic tubular networks from live imaging 3d microscopy. *Machine Learning for Biomedical Imaging*, 1, 2022. URL: <https://melba-journal.org/papers/2022:015.html>.
- 4 Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee. Skeleton extraction by mesh contraction. *ACM Trans. Graph.*, 27(3):44, 2008. doi:10.1145/1360612.1360643.
- 5 Andreas Bærentzen and Eva Rotenberg. Skeletonization via local separators. *ACM Trans. Graph.*, 40(5):187:1–187:18, 2021. doi:10.1145/3459233.


- 6 Silvia Biasotti, Leila De Floriani, Bianca Falcidieno, Patrizio Frosini, Daniela Giorgi, Claudia Landi, Laura Papaleo, and Michela Spagnuolo. Describing shapes by geometrical-topological properties of real functions. *ACM Comput. Surv.*, 40(4):12:1–12:87, 2008. doi:10.1145/1391729.1391731.
- 7 Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. Reeb graphs for shape analysis and applications. *Theor. Comput. Sci.*, 392(1-3):5–22, 2008. doi:10.1016/j.tcs.2007.10.018.
- 8 Angela Brennecke and Tobias Isenberg. 3d shape matching using skeleton graphs. In Thomas Schulze, Stefan Schlechtweg, and Volkmar Hinz, editors, *Simulation und Visualisierung 2004 (SimVis 2004) 4-5 März 2004, Magdeburg*, pages 299–310. SCS Publishing House e.V., 2004. URL: http://www.isg.cs.uni-magdeburg.de/graphik/pub/files/Brennecke_2004_3SM.pdf.
- 9 Aydin Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In Lasse Kliemann and Peter Sanders, editors, *Algorithm Engineering - Selected Results and Surveys*, volume 9220 of *Lecture Notes in Computer Science*, pages 117–158. Springer International Publishing, 2016. doi:10.1007/978-3-319-49487-6_4.
- 10 J. Andreas Bærentzen. Gel. <https://github.com/janba/GEL>, 2022.
- 11 Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhixun Su. Point cloud skeletons via laplacian based contraction. In *SMI 2010, Shape Modeling International Conference, Aix en Provence, France, June 21-23 2010*, pages 187–197. IEEE Computer Society, 2010. doi:10.1109/SMI.2010.25.
- 12 Nicu D. Cornea, Deborah Silver, and Patrick Min. Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Vis. Comput. Graph.*, 13(3):530–548, 2007. doi:10.1109/TVCG.2007.1002.
- 13 Tamal K. Dey and Jian Sun. Defining and computing curve-skeletons with medial geodesic function. In Alla Sheffer and Konrad Polthier, editors, *Proceedings of the Fourth Eurographics Symposium on Geometry Processing, Cagliari, Sardinia, Italy, June 26-28, 2006*, volume 256 of *ACM International Conference Proceeding Series*, pages 143–152. Eurographics Association, 2006. doi:10.2312/SGP/SGP06/143-152.
- 14 William Harvey, Yusu Wang, and Rephael Wenger. A randomized $O(m \log m)$ time algorithm for computing reeb graphs of arbitrary simplicial complexes. In David G. Kirkpatrick and Joseph S. B. Mitchell, editors, *Proceedings of the 26th ACM Symposium on Computational Geometry, Snowbird, Utah, USA, June 13-16, 2010*, pages 267–276. ACM, 2010. doi:10.1145/1810959.1811005.
- 15 M. Sabry Hassouna and Aly A. Farag. Variational curve skeletons using gradient vector flow. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2257–2274, 2009. doi:10.1109/TPAMI.2008.271.
- 16 Bruce Hendrickson and Robert W. Leland. A multi-level algorithm for partitioning graphs. In Sidney Karin, editor, *Proceedings Supercomputing '95, San Diego, CA, USA, December 4-8, 1995*, page 28. ACM, 1995. doi:10.1145/224170.224228.
- 17 Jacob Holm, Kristian de Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *J. ACM*, 48(4):723–760, 2001. doi:10.1145/502090.502095.
- 18 Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Hao Zhang, Guiqing Li, and Baoquan Chen. L_1 -medial skeleton of point cloud. *ACM Trans. Graph.*, 32(4):65:1–65:8, 2013. doi:10.1145/2461912.2461913.
- 19 Andrei C. Jalba, Jacek Kustra, and Alexandru C. Telea. Surface and curve skeletonization of large 3d models on the GPU. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(6):1495–1508, 2013. doi:10.1109/TPAMI.2012.212.
- 20 Wei Jiang, Kai Xu, Zhi-Quan Cheng, Ralph R. Martin, and Gang Dang. Curve skeleton extraction by coupled graph contraction and surface clustering. *Graph. Model.*, 75(3):137–148, 2013. doi:10.1016/j.gmod.2012.10.005.

- 21 George Karypis and Vipin Kumar. Analysis of multilevel graph partitioning. In Sidney Karin, editor, *Proceedings Supercomputing '95, San Diego, CA, USA, December 4-8, 1995*, page 29. ACM, 1995. doi:10.1145/224170.224229.
- 22 Mattia Natali, Silvia Biasotti, Giuseppe Patanè, and Bianca Falcidieno. Graph-based representations of point clouds. *Graph. Model.*, 73(5):151–164, 2011. doi:10.1016/j.gmod.2011.03.002.
- 23 Valerio Pascucci, Giorgio Scorzelli, Peer-Timo Bremer, and Ajith Mascarenhas. Robust on-line computation of reeb graphs: simplicity and speed. *ACM Trans. Graph.*, 26(3):58, 2007. doi:10.1145/1276377.1276449.
- 24 Diane Perchet, Catalin I. Fetita, and Françoise J. Prêteux. Advanced navigation tools for virtual bronchoscopy. In Edward R. Dougherty, Jaakko Astola, and Karen O. Egiazarian, editors, *Image Processing: Algorithms and Systems III, San Jose, California, USA, January 18, 2004*, volume 5298 of *SPIE Proceedings*, pages 147–158. SPIE, 2004. doi:10.1117/12.533096.
- 25 Dennie Reniers, Jarke J. van Wijk, and Alexandru C. Telea. Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure. *IEEE Trans. Vis. Comput. Graph.*, 14(2):355–368, 2008. doi:10.1109/TVCG.2008.23.
- 26 Andrea Tagliasacchi, Ibraheem Alhashim, Matt Olson, and Hao Zhang. Mean curvature skeletons. *Comput. Graph. Forum*, 31(5):1735–1744, 2012. doi:10.1111/j.1467-8659.2012.03178.x.
- 27 Andrea Tagliasacchi, Thomas Delamé, Michela Spagnuolo, Nina Amenta, and Alexandru C. Telea. 3d skeletons: A state-of-the-art report. *Comput. Graph. Forum*, 35(2):573–597, 2016. doi:10.1111/cgf.12865.
- 28 Alexandru C. Telea. 3d skeletonization benchmark. <https://webpace.science.uu.nl/~telea001/Shapes/SkelBenchmark>, 2016. Accessed: 2022-10-31.
- 29 Alexandru C. Telea and Andrei C. Jalba. Computing curve skeletons from medial surfaces of 3d shapes. In Hamish A. Carr and Silvester Czanner, editors, *Theory and Practice of Computer Graphics, Rutherford, United Kingdom, 2012. Proceedings*, pages 99–106. Eurographics Association, 2012. doi:10.2312/LocalChapterEvents/TPCG/TPCG12/099-106.
- 30 Ming Wan, Frank Dacheille, and Arie E. Kaufman. Distance-field-based skeletons for virtual navigation. In Thomas Ertl, Kenneth I. Joy, and Amitabh Varshney, editors, *12th IEEE Visualization Conference, IEEE Vis 2001, San Diego, CA, USA, October 24-26, 2001, Proceedings*, pages 239–246. IEEE Computer Society, 2001. doi:10.1109/VISUAL.2001.964517.
- 31 Yu-Shuen Wang and Tong-Yee Lee. Curve-skeleton extraction using iterative least squares optimization. *IEEE Trans. Vis. Comput. Graph.*, 14(4):926–936, 2008. doi:10.1109/TVCG.2008.38.
- 32 Yajie Yan, Kyle Sykes, Erin W. Chambers, David Letscher, and Tao Ju. Erosion thickness on medial axes of 3d shapes. *ACM Trans. Graph.*, 35(4):38:1–38:12, 2016. doi:10.1145/2897824.2925938.

Efficient Computation of Image Persistence

Ulrich Bauer   

Department of Mathematics, TUM School of Computation, Information and Technology, and Munich Data Science Institute, Technical University of Munich, Germany

Maximilian Schmahl 

Universität Heidelberg, Germany

Abstract

We present an algorithm for computing the barcode of the image of a morphism in persistent homology induced by an inclusion of filtered finite-dimensional chain complexes. The algorithm makes use of the clearing optimization and can be applied to inclusion-induced maps in persistent absolute homology and persistent relative cohomology for filtrations of pairs of simplicial complexes. The clearing optimization works particularly well in the context of relative cohomology, and using previous duality results we can translate the barcodes of images in relative cohomology to those in absolute homology. This forms the basis for an implementation of image persistence computations for inclusions of filtrations of Vietoris–Rips complexes in the framework of the software Ripser.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Theory of computation → Computational geometry

Keywords and phrases Persistent homology, image persistence, barcode computation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.14

Related Version *Full Version*: [arXiv:2201.04170](https://arxiv.org/abs/2201.04170) [4]

Supplementary Material

Software (Source Code): <https://github.com/Ripser/ripser/tree/image-persistence-simple> [6] archived at [swh:1:dir:135b51a39778c4004326b548ff59998f19a8cf74](https://www.swh.io/dir/135b51a39778c4004326b548ff59998f19a8cf74)

Funding *Ulrich Bauer*: Supported by the German Research Foundation (DFG) through the Collaborative Research Center SFB/TRR 109 *Discretization in Geometry and Dynamics* – 195170736.

Maximilian Schmahl: Supported by the German Research Foundation (DFG) through the Collaborative Research Center SFB/TRR 191 *Symplectic Structures in Geometry, Algebra and Dynamics* – 281071066, the Cluster of Excellence EXC 2181 *STRUCTURES* – 390900948, and the Research Training Group RTG 2229 *Asymptotic Invariants and Limits of Groups and Spaces* – 281869850.

1 Introduction

Over the last few decades, *persistent homology* has established its role as an important tool in data science, with numerous applications in a variety of disciplines, including computer vision, neuroscience, materials science, and evolutionary biology [18, 10, 15, 20, 7]. Recently, there has also been renewed interest in *image persistence*, which is a natural extension of persistent homology [21, 22, 17]. Persistent homology starts with a filtration of simplicial complexes K_\bullet and concerns the *barcode*, which encodes the algebraic structure of the *persistence module* $H_*(K_\bullet)$. In contrast, *image persistence* starts with two filtrations L_\bullet and K_\bullet that are related by a map of filtrations $f_\bullet: L_\bullet \rightarrow K_\bullet$. This map induces a morphism $H_*(f_\bullet): H_*(L_\bullet) \rightarrow H_*(K_\bullet)$. The image of this morphism, $\text{im } H_*(f_\bullet)$, is again a persistence module, and image persistence concerns the barcode of this persistence module.

A key part of the appeal of image persistence is that it enables the construction of meaningful *matchings*, i.e., partial bijections, between the barcodes of the domain and codomain of the morphism one starts with. The first such construction that appeared in the literature was the *induced matching* construction, which was introduced by Bauer and Lesnick [2] to give a proof of the famous *stability theorem* of Cohen-Steiner et al. [11]. Such



© Ulrich Bauer and Maximilian Schmahl;

licensed under Creative Commons License CC-BY 4.0

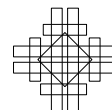
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 14; pp. 14:1–14:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



constructions have now also appeared in work with a more practical focus, with Reani and Bobrowski [21] and García-Redondo et al. [17] proposing and applying general schemes for matching cycles in different filtrations using image persistence, as well as Stucki et al. [22] applying such a method in the context of medical image data analysis.

The first algorithm for computing image persistence in a special case was proposed by Cohen-Steiner et al. [12] for maps f_\bullet of the form $L_\bullet = K_\bullet \cap L \hookrightarrow K_\bullet$ for some fixed subcomplex $L \subseteq K$ (*one-filtration setting*). An implementation for the algorithm from [12] exists in the framework of the software Dionysus by Dmitriy Morozov [19]. The algorithm described by Cohen-Steiner et al. is similar to the standard algorithm for a single filtration and naturally does not make use of many important speed-ups that have been developed for the computation of the barcode of a single filtration since the publication of [12]. Cohen-Steiner et al. also propose an adaption of their method to the general (*two-filtration*) setting using a mapping cylinder construction, which however has never been implemented and might not be computationally feasible. The goal of the present work is to adapt some of the speed-ups for a single filtration to the computation of image persistence, and to show that the resulting algorithm also works for general injective maps f_\bullet without the intersection assumption and without the need for the mapping cylinder construction.

The basic algorithm for computing persistent homology is based on performing *matrix reduction*, a variant of column-wise Gaussian elimination, on a *boundary matrix* associated to the given filtration of simplicial complexes. This algorithm can be made faster using the *clearing* optimization, introduced by Chen and Kerber in [9], and also used implicitly in the cohomology algorithm by de Silva et al. [16]. In short, this optimization makes use of the homological grading of the boundary matrix to disregard certain unnecessary columns in the reduction process. The basic algorithm for image persistence additionally requires the reduction of a permuted boundary matrix, to which clearing cannot be straightforwardly applied. We will remedy this by showing that one can delete the columns in the permuted boundary matrix that were already reduced to 0 in the boundary matrix corresponding to the codomain filtration.

The clearing optimization works particularly well in conjunction with cohomology based algorithms. These were first studied by de Silva et al. in [16] for the single filtration case and justified by certain duality results that provide a translation between barcodes for persistent homology and for *persistent cohomology*, as well as the barcodes for *persistent relative homology* $H_*(K, K_\bullet): H_*(K, K_0) \rightarrow \cdots \rightarrow H_*(K, K)$ and similarly for *persistent relative cohomology*. These duality results were recently extended by Bauer and Schmalz in [5] in order to also provide translations for images of $H_*(f_\bullet)$ and $H^*(f_\bullet)$, as well as their relative counterparts $H_*(f, f_\bullet)$ and $H^*(f, f_\bullet)$. This allows us to perform cohomology based computations and still obtain the desired barcodes in homology.

To apply clearing in the relative cohomology setting for image persistence, we will reformulate the algorithm for image persistence by Cohen-Steiner et al. [12] in the purely algebraic setting of filtered chain complexes of vector spaces. More precisely, we will consider two filtrations of (co)chain complexes C_\bullet and C'_\bullet and a monomorphism $\varphi_\bullet: C_\bullet \rightarrow C'_\bullet$. This setup includes both the absolute homology case $C_*(L_\bullet) \hookrightarrow C_*(K_\bullet)$ and the relative cohomology case $C^*(K, K_\bullet) \hookrightarrow C^*(L, L_\bullet)$ from before. The general idea for computing the image of $H_*(\varphi_\bullet)$ is to first write it as a subquotient of C'_\bullet :

$$\operatorname{im} H_*(\varphi_\bullet) \cong \frac{\varphi_\bullet(Z_*(C_\bullet))}{\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)},$$

where the intersection of persistence modules is to be interpreted indexwise, meaning that $(\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet))_t = \varphi_t(Z_t(C_t)) \cap B_t(C'_t)$.

Performing matrix reductions that make use of the clearing optimization, we will find a pair of inclusion-related *filtration compatible bases* for the filtrations appearing in the equation above. Filtration compatible bases provide a formal framework for many standard arguments for barcode computations via matrix reduction, and they can be interpreted as special cases of *matching diagrams*, which are equivalent to barcodes [3]. Using the general theory of matching diagrams, the data we compute can easily be shown to determine the barcode of $\text{im } H_*(\varphi_\bullet)$.

Applying these general considerations in the relative cohomology setting and combining this with the translation between relative cohomology and absolute homology from [5] yields an algorithm for computing the absolute homology image of $f_\bullet: L_\bullet \rightarrow K_\bullet$ by reducing two coboundary matrices that can be reduced with clearing as summarized in our main result Theorem 22. An implementation of this method based on Ripser [1] is publicly available [6] and we provide some computational benchmarks. Our software works under the assumption that $L_\bullet = \text{Rips}_\bullet(X, d)$ and $K_\bullet = \text{Rips}_\bullet(X, d')$ are filtrations of Vietoris–Rips complexes corresponding to two metrics d and d' on a finite set X that satisfy $d(x, y) \geq d'(x, y)$ for all $x, y \in X$. This ensures that $L_t = \text{Rips}_t(X, d)$ is a subcomplex of $K_t = \text{Rips}_t(X, d')$ for all t , with the maps $f_t: L_t \rightarrow K_t$ being given by inclusion. The implementation also makes use of a version of the emergent and apparent pairs optimizations, which shortcuts the construction of the coboundary matrix and reduces the memory requirements for storing persistence pairings [1].

Contributions

- We propose the first algorithm for the general problem of computing the image of a map in persistent homology induced by an inclusion of filtrations of simplicial complexes, without imposing any restrictions on the subfiltration (called the “two function setting” in [12]) and without the inefficient use of a mapping cylinder (Theorem 22).
- We show that our general method can be augmented by the most important optimizations in persistence computations, including clearing (Corollary 20), cohomology based computations (Proposition 21), and apparent pairs (Section 3.4).
- We provide an implementation in the framework of Ripser [6] and experiments on data sets of varying difficulty (Section 3.5).
- This enables the use of image persistence and consequently induced matchings in computational settings, such as supervised learning [22, 17].

Notation. Throughout the paper, we fix a totally ordered set (T, \leq) to be $\{0, \dots, n\}$ with the obvious order and a field \mathbb{F} over which all vector spaces are considered.

2 Linear Algebra for Filtrations

In this section, we develop some machinery based on filtration compatible bases, which forms the foundation for our constructions of image persistence barcodes. First, we need to recall some basic theory for persistence modules and barcodes. We write \mathbf{Vec} for the category of vector spaces over our fixed field \mathbb{F} . We fix $T = \{0, \dots, n\}$ as a finite totally ordered index set, and we write \mathbf{T} for T considered as a poset category.

► **Definition 1.** *The category of persistence modules indexed by \mathbf{T} is defined as the category $\mathbf{Vec}^{\mathbf{T}}$ whose objects are functors $\mathbf{T} \rightarrow \mathbf{Vec}$ and whose morphisms are natural transformations.*

14:4 Efficient Computation of Image Persistence

Since \mathbf{T} is a small category and \mathbf{Vec} is an abelian category, the functor category $\mathbf{Vec}^{\mathbf{T}}$ is again abelian, with kernels, cokernels, images, direct sums, and more generally, all limits and colimits given pointwise. The prime example for a persistence module is the persistent homology of a filtration of spaces. Other examples are given by *interval modules*. If $I \subseteq T$ is an interval, the corresponding interval module $C(I)_\bullet$ is defined by

$$C(I)_t = \begin{cases} \mathbb{F} & \text{if } t \in I, \\ 0 & \text{otherwise,} \end{cases} \quad \text{with structure maps} \quad C(I)_{t,u} = \begin{cases} \text{id}_{\mathbb{F}} & \text{if } t, u \in I, \\ 0 & \text{otherwise.} \end{cases}$$

These interval modules are of particular interest because they lead to a structure theory for persistence modules.

► **Definition 2.** *If there is a family of intervals $(I_\alpha)_{\alpha \in A}$ such that for a persistence module M_\bullet we have $M_\bullet \cong \bigoplus_{\alpha \in A} C(I_\alpha)_\bullet$, then M_\bullet is said to have a barcode given by $(I_\alpha)_{\alpha \in A}$.*

If a persistence module has a barcode, then it is unique, by a version of the Krull–Remak–Schmidt–Azumaya Theorem [8, Theorem 2.7]. In this paper, we will only consider persistence modules consisting of finite dimensional vector spaces, which are guaranteed to have a barcode by Crawley-Boevey’s Theorem [14].

Persistent homology is the homology of a chain complex of persistence modules. In practice, the persistence modules forming these chain complexes arise from filtrations of simplicial complexes, so their structure maps are all inclusions. We will now study this kind of persistence module more closely, as our later considerations will mostly happen in terms of chain complexes rather than in terms of homology.

► **Definition 3.** *We say that a persistence module M_\bullet is a filtration of the vector space $M = M_n$ if for all $t \leq u$ the structure map $M_{t,u}$ is a subspace inclusion $M_t \hookrightarrow M_u$. For any $m \in M$, we define its support in M_\bullet as $\text{supp}_{M_\bullet}(m) = \{t \in T \mid m \in M_t\}$. A basis \mathfrak{M} of M is said to be filtration compatible if $\mathfrak{M}_t = \mathfrak{M} \cap M_t$ is a basis for M_t for all $t \in T$. An ordered basis (\mathfrak{M}, \leq) for M is said to be a filtration compatible ordered basis if it is filtration compatible and $m \leq m' \in \mathfrak{M}$ implies $\text{supp } m' \subseteq \text{supp } m$.*

If M_\bullet and M'_\bullet are filtrations of vector spaces, we write $M_\bullet \subseteq M'_\bullet$ if $M_t \subseteq M'_t$. We write M'_\bullet/M_\bullet for the persistence module given by $(M'_\bullet/M_\bullet)_t = M'_t/M_t$. Similarly, if M''_\bullet is another filtration with $M''_\bullet \subseteq M'_\bullet$, we write $M_\bullet \cap M''_\bullet$ for the persistence module given by $(M_\bullet \cap M''_\bullet)_t = M_t \cap M''_t$.

Observe that if M_\bullet is a filtration of vector spaces and \mathfrak{M} is a filtration compatible basis, then $(\text{supp}(m))_{m \in \mathfrak{M}}$ is a barcode of M_\bullet . By interpreting \mathfrak{M} as a so-called matching diagram, this may be seen as a special case of the general equivalence of matching diagrams and barcodes [3]. This theory also yields the following result that forms the basis for our computational results.

► **Proposition 4.** *Let $M_\bullet \subseteq M'_\bullet$ be filtrations of vector spaces with respective filtration compatible bases \mathfrak{M} and \mathfrak{M}' related by an inclusion $\mathfrak{M} \subseteq \mathfrak{M}'$. Then M'_\bullet/M_\bullet has the barcode*

$$(\text{supp}_{M'_\bullet}(m) \setminus \text{supp}_{M_\bullet}(m))_{m \in \mathfrak{M}} \cup (\text{supp}_{M'_\bullet}(m))_{m \in \mathfrak{M}' \setminus \mathfrak{M}}.$$

We now state some helpful facts about filtration compatible bases. We refer to the full version of this paper [4] for the proofs. We start with a lemma relating supports of basis elements with filtration compatibility.

► **Lemma 5.** *Let M_\bullet be a filtration of the vector space M with filtration compatible basis \mathfrak{M} . Let \mathfrak{M}' be another basis for M such that there exists a bijection $g: \mathfrak{M} \rightarrow \mathfrak{M}'$ with $\text{supp}_{M_\bullet}(m) = \text{supp}_{M_\bullet}(g(m))$ for all $m \in \mathfrak{M}$. Then \mathfrak{M}' is a filtration compatible basis for M_\bullet .*

Next, we extend a standard fact about intersections of vector spaces to filtrations.

► **Lemma 6.** *Let $M'_\bullet, M''_\bullet \subseteq M_\bullet$ be filtrations of vector spaces and let \mathfrak{M}' and \mathfrak{M}'' be filtration compatible bases for M'_\bullet and M''_\bullet , respectively, such that $\mathfrak{M}' \cup \mathfrak{M}''$ is linearly independent. Then $\mathfrak{M}' \cap \mathfrak{M}''$ is a filtration compatible basis for $M'_\bullet \cap M''_\bullet$. Moreover, for all $m \in \mathfrak{M}' \cap \mathfrak{M}''$*

$$\text{supp}_{M'_\bullet \cap M''_\bullet}(m) = \text{supp}_{M'_\bullet}(m) \cap \text{supp}_{M''_\bullet}(m).$$

We will use the special case where M'_\bullet is included in M''_\bullet at the last filtration step (but not necessarily before):

► **Corollary 7.** *Let $M'_\bullet, M''_\bullet \subseteq M_\bullet$ be filtrations of vector spaces $M' \subseteq M'' \subseteq M$, respectively. Moreover, let $\mathfrak{M}' \subseteq \mathfrak{M}''$ be filtration compatible bases for M'_\bullet and M''_\bullet , respectively. Then \mathfrak{M}' is a filtration compatible basis for $M'_\bullet \cap M''_\bullet$. Moreover, for all $m \in \mathfrak{M}'$*

$$\text{supp}_{M'_\bullet \cap M''_\bullet}(m) = \text{supp}_{M'_\bullet}(m) \cap \text{supp}_{M''_\bullet}(m).$$

Finally, we state a version of the rank-nullity-theorem for filtrations.

► **Lemma 8.** *Let $\phi_\bullet: M_\bullet \rightarrow P_\bullet$ be a morphism of filtrations of vector spaces and consider the linear map $\phi = \phi_n: M \rightarrow P$. Let \mathfrak{M} be a filtration compatible basis for M_\bullet , let $\mathfrak{M}' = \mathfrak{M} \cap \ker \phi$, and assume that $\mathfrak{M}'' = (\phi(m))_{m \in \mathfrak{M} \setminus \mathfrak{M}'}$ is a linearly independent family of vectors. Then*

- \mathfrak{M}' is a filtration compatible basis for $\ker \phi_\bullet$,
- \mathfrak{M}'' is a filtration compatible basis for $\text{im } \phi_\bullet$,
- $\text{supp}_{\ker \phi_\bullet}(m') = \text{supp}_{M_\bullet}(m')$ for all $m' \in \mathfrak{M}'$, and
- $\text{supp}_{\text{im } \phi_\bullet}(\phi(m)) = \text{supp}_{M_\bullet}(m)$ for all $m \in \mathfrak{M} \setminus \mathfrak{M}'$.

Note that if one drops the assumption of the above lemma that P_\bullet , and hence the image $\text{im } \phi_\bullet$, is a filtration, then it may happen that $\mathfrak{M}' = \mathfrak{M} \cap \ker \phi$ is a basis for the vector space $\ker \phi$ but not a filtration compatible basis for the filtration $\ker \phi_\bullet$.

3 Computing Image Persistence Barcodes

Recall that we fixed a finite totally ordered index set $T = \{0, \dots, n\}$ and a field \mathbb{F} over which we consider vector spaces. For our purposes, a chain (resp. cochain) complex is a graded finite dimensional vector space with a differential of degree -1 (resp. 1) that squares to 0. A chain complex of persistence modules C_\bullet with differential ∂_\bullet is called a filtration of a chain complex of vector spaces C with differential ∂ if C_\bullet is a filtration of C as a vector space and $\partial_n = \partial$. Recall that a basis for the final vector space in a filtration is called filtration compatible if it yields bases for the constituent vector spaces of the filtration by intersecting. Further, recall that if the basis is ordered, we say that it is a filtration compatible ordered basis if its order refines the order in which the basis elements appear in the filtration.

► **Definition 9.** *If C_\bullet is a filtration of the (co)chain complex C with a filtration compatible ordered basis \mathfrak{C} , then the matrix D representing the (co)boundary operator on C with respect to \mathfrak{C} is called filtration (co)boundary matrix.*

► **Example 10.** If $K_\bullet: \emptyset = K_0 \subseteq K_1 \subseteq \dots \subseteq K_n = K$ is a filtration of finite simplicial complexes, we get a filtration of chain complexes $C_*(K_\bullet)$. A filtration compatible ordered basis is given by the simplices of K , ordered by a linear refinement of the order in which they appear in the filtration. If D^K is a corresponding filtration boundary matrix, then one can check (see [16]) that $(D^K)^\perp$ is a filtration coboundary matrix for the filtration of relative

cochains $0 = C^*(K, K) = C^*(K, K_n) \subseteq \cdots \subseteq C^*(K, K_0) = C^*(K, \emptyset) = C^*(K)$. The matrix represents the coboundary operator on $C^*(K)$ with respect to the dual basis corresponding to the simplices of K , ordered by the opposite of the filtration order. Here, $(-)^{\perp}$ denotes taking the transpose of a matrix along its anti-diagonal.

To avoid notational clutter, we will from now on only talk about chain complexes in the general setting, but everything also straightforwardly applies to cochain complexes.

► **Definition 11.** *If X is a matrix, we write x_i for the i -th column of the matrix X . For a non-zero column vector x_i , we define $\text{Pivot } x_i$ as the largest index where the column has a non-zero entry. We write $\text{Pivots } X$ for the set of all indices which occur as pivots of non-zero columns of X . A matrix is called reduced if no two non-zero columns have the same pivot.*

Note that any set of non-zero vectors with unique pivots is linearly independent. In particular, the non-zero columns of a reduced matrix are linearly independent.

Computing the barcode for the homology of a filtration of a chain complex is done by *reducing* a filtration boundary matrix D , i.e., performing a variant of Gaussian elimination on the columns of this matrix until one obtains a reduced matrix. This can be expressed as finding a reduced matrix R and a full-rank upper-triangular matrix V such that $R = DV$. The columns of these matrices naturally represent elements of C by interpreting them as coordinate vectors with respect to the ordered basis \mathfrak{C} . The barcode for persistent homology may then be obtained from this data as follows.

► **Theorem 12** (Cohen-Steiner et al. [13]). *Let D be a filtration boundary matrix of a filtration of chain complexes C_{\bullet} and assume we have a full-rank and upper-triangular matrix V such that $R = DV$ is reduced. Then $H_*(C_{\bullet})$ has a barcode given by the multiset*

$$\{\text{supp}_{C_{\bullet}}(r_j) \setminus \text{supp}_{C_{\bullet}}(v_j) \mid r_j \neq 0\} \cup \{\text{supp}_{C_{\bullet}}(v_i) \mid r_i = 0 \text{ and } i \notin \text{Pivots } R\}.$$

The supports of column vectors that appear in the theorem can easily be determined from the initial data via pivots: If \mathfrak{M} is a filtration compatible ordered basis for a filtration M_{\bullet} of a vector space M , then we can consider elements of M via their coordinate vectors with respect to \mathfrak{M} . Because \mathfrak{M} is a filtration compatible ordered basis, we then have $\text{supp}_{M_{\bullet}}(v) = \text{supp}_{M_{\bullet}}(v')$ if and only if $\text{Pivot } v = \text{Pivot } v'$ for any two such coordinate vectors v and v' . In particular, this means that in the setting of simplicial complexes the support of a column vector is the same as the support of its pivot simplex.

The theorem is formulated in a different language by Cohen-Steiner et al., but the version above also follows as a special case from Theorem 14. Note that the theorem is also compatible with the homological grading: Assume that $C_{\bullet} = \bigoplus_d C_{\bullet,d}$ is graded with ∂ mapping $C_{\bullet,d}$ to $C_{\bullet,d-1}$. If the filtration compatible ordered basis \mathfrak{C} used to build D is chosen such that its intersection with each grading summand is a filtration compatible ordered basis for that summand, then one gets a barcode for $H_d(C_{\bullet})$ by restricting the barcode given in Theorem 12 to those intervals coming from columns that represent d -dimensional cycles.

3.1 Image Barcodes via Matrix Reduction

We now turn to the setting of image persistence. Let C_{\bullet} and C'_{\bullet} be filtrations of the chain complexes C and C' with corresponding filtration compatible ordered bases \mathfrak{C} and \mathfrak{C}' . Let D and D' be the corresponding filtration boundary matrices. Assume that we are given an injection of filtrations $\varphi_{\bullet}: C_{\bullet} \rightarrow C'_{\bullet}$ such that the map $\varphi: C \rightarrow C'$ on the final filtration step is an isomorphism. Note that this is not a restriction, as any injection of filtrations can be

extended to one satisfying this assumption, and subsequently restricting the barcodes to the original indexing set provides the desired result for this more general setting as well. Let F be the matrix representing φ with respect to \mathfrak{C} and \mathfrak{C}' and define the *mixed basis boundary matrix* $D^\varphi = DF^{-1} = F^{-1}D'$. The columns of D^φ thus correspond to \mathfrak{C}' , while the rows correspond to \mathfrak{C} .

► **Example 13.** If $K_\bullet : \emptyset = K_0 \subseteq \dots \subseteq K_n = K$ and $L_\bullet : \emptyset = L_0 \subseteq \dots \subseteq L_n = L$ are filtrations of finite simplicial complexes, we get filtrations of chain complexes $C_*(K_\bullet)$ and $C_*(L_\bullet)$. If we are given a monomorphism $f_\bullet : L_\bullet \rightarrow K_\bullet$ that induces an isomorphism $L \rightarrow K$ (i.e., assuming that $L_i \subseteq K_i$ for all i and $L = K$), then we are in the setting above. Filtration compatible ordered bases are given by the simplices of K and L , ordered by a linear refinement of the order in which they appear in the respective filtrations. Let D^L and D^K denote the corresponding filtration boundary matrices, and let D^f denote the mixed basis boundary matrix for the induced map $C_*(L_\bullet) \rightarrow C_*(K_\bullet)$. Then, analogously to Example 10, we obtain that $(D^L)^\perp$, $(D^K)^\perp$ and $(D^f)^\perp$ are the filtration and mixed basis coboundary matrices for the relative cohomology counterpart $C^*(K, K_\bullet) \rightarrow C^*(L, L_\bullet)$. In the mixed matrix $(D^f)^\perp$, the columns thus correspond to L_\bullet , while the rows correspond to K_\bullet .

Our goal is to determine a barcode for $\text{im } H_*(\varphi_\bullet)$ by reducing the matrices D and D^φ . Assume that we have $R = DV$ and $R^\varphi = D^\varphi V^\varphi$ reduced with V and V^φ full-rank and upper-triangular. The columns of the matrices R, D, V, R^φ , and D^φ naturally represent elements of C by interpreting them as coordinate vectors with respect to \mathfrak{C} . Similarly, the columns of V^φ naturally represent elements of C' by interpreting them as coordinate vectors with respect to \mathfrak{C}' . Recall that if X is a matrix, we denote its j th column by x_j . The main result can then be stated as follows.

► **Theorem 14.** *The image of $H_*(\varphi_\bullet)$ has a barcode given by the multisets*

$$\{\text{supp}_{C_\bullet}(r_j^\varphi) \setminus \text{supp}_{C'_\bullet}(v_j^\varphi) \neq \emptyset \mid r_j^\varphi \neq 0\} \cup \{\text{supp}_{C_\bullet}(v_i) \mid r_i = 0 \text{ and } i \notin \text{Pivots } R\}.$$

Note that the intervals $\text{supp}_{C_\bullet}(v_i)$ in the barcode of $\text{im } H_*(\varphi_\bullet)$ that are not bounded above are precisely the same as those in the barcode of $H_*(C_\bullet)$ as given in Theorem 12.

The proof of Theorem 14 will be based on a sequence of intermediate results. As mentioned in the introduction, the general idea is to write

$$\text{im } H_*(\varphi_\bullet) \cong \frac{\varphi(Z_*(C_\bullet))}{\varphi(Z_*(C_\bullet)) \cap B_*(C'_\bullet)},$$

and to find filtration compatible bases \mathfrak{Z} and \mathfrak{B} for $\varphi(Z_*(C_\bullet))$ and $\varphi(Z_*(C_\bullet)) \cap B_*(C'_\bullet)$, respectively, such that $\mathfrak{B} \subseteq \mathfrak{Z}$ holds so that we can apply Proposition 4.

If X is a matrix, we will write $\text{cols } X$ for the family of all its non-zero column vectors.

► **Lemma 15.** *The family $\text{cols } V^\varphi$ is a filtration compatible basis for C'_\bullet , $\mathfrak{B} = \text{cols } FR^\varphi$ is a filtration compatible basis for $B_*(C'_\bullet)$, and for all j with $r_j^\varphi \neq 0$ we have*

$$\text{supp}_{B_*(C'_\bullet)}(Fr_j^\varphi) = \text{supp}_{C'_\bullet}(v_j^\varphi).$$

Proof. We start by showing that $\text{cols } V^\varphi$ is a filtration compatible basis for C'_\bullet : We have $\text{Pivot } v_j^\varphi = j$ since V^φ is full-rank and upper-triangular. It follows that v_j^φ has the same support in C'_\bullet as the j th element of \mathfrak{C}' . Thus, $\text{cols } V^\varphi$ is a filtration compatible basis for C'_\bullet by Lemma 5.

14:8 Efficient Computation of Image Persistence

Next, note that $(\partial(v))_{v \in \text{cols } V^\varphi \setminus \ker \partial} = \text{cols } FR^\varphi$ is linearly independent since R^φ is reduced and F has full rank. Thus, we can apply Lemma 8 to the map of filtrations $\partial_\bullet: C'_\bullet \rightarrow C'_\bullet$ and the filtration compatible basis $\text{cols } V^\varphi$ to obtain that $\text{cols } FR^\varphi$ is a filtration compatible basis for $B_*(C'_\bullet) = \text{im } \partial_\bullet$. The assertion on the supports follows from the support formula in Lemma 8. \blacktriangleleft

Now that we have a filtration compatible basis for $B_*(C'_\bullet)$, we want to extend it to a filtration compatible basis for $\varphi_\bullet(Z_*(C_\bullet))$.

► **Lemma 16.** *Let $\mathfrak{X} = \text{cols } R^\varphi \cup \{v_j \mid j \notin \text{Pivots } R^\varphi\}$ and $\mathfrak{X}' = \mathfrak{X} \cap \ker \partial$. Then \mathfrak{X} is a filtration compatible basis for C_\bullet , $\mathfrak{Z} = F\mathfrak{X}' = \mathfrak{B} \cup \{Fv_j \mid j \notin \text{Pivots } R^\varphi\}$ is a filtration compatible basis for $\varphi_\bullet(Z_*(C_\bullet))$, and for all $x \in \mathfrak{X}'$ we have*

$$\text{supp}_{\varphi_\bullet(Z_*(C_\bullet))}(Fx) = \text{supp}_{C_\bullet}(x).$$

Proof. We start by showing that \mathfrak{X} is a filtration compatible basis for C_\bullet . The same argument as in the beginning of the proof of Lemma 15 yields that $\text{cols } V$ is a filtration compatible basis for C_\bullet . Next, note that \mathfrak{X} is linearly independent since all elements have unique pivots: R^φ is reduced and we only consider those v_j with $\text{Pivot } v_j = j \notin \text{Pivots } R^\varphi$. Moreover, we have a bijection $\mathfrak{X} \rightarrow \text{cols } V$ given by mapping v_j to itself and mapping r_j^φ to v_i for $i = \text{Pivot } r_j^\varphi$. Recall that $\text{Pivot } v_i = i = \text{Pivot } r_j^\varphi$ implies $\text{supp}_{C_\bullet}(r_j^\varphi) = \text{supp}_{C_\bullet}(v_i)$. Since $\text{cols } V$ is a filtration compatible basis for C_\bullet , Lemma 5 now implies that \mathfrak{X} is also a filtration compatible basis for C_\bullet .

Since R is reduced and thus $(\partial(v))_{v \in \mathfrak{X} \setminus \mathfrak{X}'} \subseteq \text{cols } R$ is linearly independent, we can apply Lemma 8 to the boundary operator $\partial_\bullet: C_\bullet \rightarrow C_\bullet$ and the filtration compatible basis \mathfrak{X} . We obtain that $\mathfrak{X}' = F^{-1}\mathfrak{Z}$ is a filtration compatible basis for $\ker \partial_\bullet = Z_*(C_\bullet)$ with $\text{supp}_{Z_*(C_\bullet)}(x) = \text{supp}_{C_\bullet}(x)$ for all $x \in \mathfrak{X}'$. The claim now follows from the fact that φ_\bullet is mono, so that its restriction is an isomorphism $Z_*(C_\bullet) \rightarrow \varphi_\bullet(Z_*(C_\bullet))$ represented by F . \blacktriangleleft

Since the filtration compatible basis \mathfrak{B} for $B_*(C'_\bullet)$ extends to a basis \mathfrak{Z} for $\varphi_\bullet(Z_*(C_\bullet))$, we can conclude that \mathfrak{B} is also a filtration compatible for $\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)$.

► **Lemma 17.** *The family $\mathfrak{B} = \text{cols } FR^\varphi$ is a filtration compatible basis for $\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)$, and for all j with $r_j^\varphi \neq 0$ we have*

$$\text{supp}_{\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)}(Fr_j^\varphi) = \text{supp}_{C_\bullet}(r_j^\varphi) \cap \text{supp}_{C'_\bullet}(v_j^\varphi).$$

Proof. Recall that \mathfrak{B} is a filtration compatible basis for $B_*(C'_\bullet)$, and \mathfrak{Z} extends \mathfrak{B} to one for $\varphi_\bullet(Z_*(C_\bullet))$. Now Corollary 7 together with the support equalities from Lemmas 15 and 16 yield the claim. \blacktriangleleft

► **Lemma 18.** $\text{Pivots } R = \text{Pivots } R^\varphi$.

Proof. The matrices D and $D^\varphi = DF^{-1}$ have the same column space. Matrix reduction does not change column spaces, so R and R^φ also have the same column space. In particular, every non-zero column of R is a non-trivial linear combination of non-zero columns of R^φ and vice versa. The pivots of a linear combination of a reduced set of column vectors must be the same as the pivot of one of these vectors, so we indeed obtain $\text{Pivots } R = \text{Pivots } R^\varphi$. \blacktriangleleft

We are now ready to prove the main result of this section.

Proof of Theorem 14. By definition of the induced map in homology, we have

$$\text{im } H_*(\varphi_\bullet) \cong \frac{\varphi_\bullet(Z_*(C_\bullet))}{\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet)}.$$

The claim follows by applying Proposition 4 to the inclusion $\varphi_\bullet(Z_*(C_\bullet)) \cap B_*(C'_\bullet) \subseteq B_*(C'_\bullet)$ with the filtration compatible bases $\mathfrak{B} \subseteq \mathfrak{Z}$, with supports as previously determined in Lemmas 16 and 17. Note that in the basis \mathfrak{Z} we choose columns Fv_i with $i \notin \text{Pivots } R^\varphi$, while the formula in Theorem 14 requires $i \notin \text{Pivots } R$. These conditions are, however, equivalent by Lemma 18. \blacktriangleleft

3.2 Clearing

The *clearing* optimization [9] is a key ingredient of efficient persistence computation. We first recall the basic idea of clearing, which applies to the computation of persistent homology of a filtration of chain complexes C_\bullet by reducing the boundary matrix D to $R = DV$. We keep the notation from the beginning of this section, and we assume that our filtration compatible basis \mathfrak{C} is compatible with the homological grading in the sense that the restriction of this basis to each grading summand is again a basis of that summand. Our discussion focuses on chain complexes, but of course the findings naturally apply to cochain complexes with the appropriate adjustments to the grading.

If a column r_j of the reduced matrix R is nonzero, then necessarily $r_i = 0$ for $i = \text{Pivot } r_j$. The homological degree of the i -th element of \mathfrak{C} is one less than that of the j -th element. This leads to the clearing procedure: Instead of reducing D by column operations from left to right, we reduce columns in decreasing order of their homological degree (increasing in the case of cohomology). Before reducing the columns in dimension d , we set $r_j = 0$ for all j which appear as pivots of the already reduced columns in dimension $d + 1$.

Turning to the image setting, we also assume that the basis \mathfrak{C}' and the map $\varphi_\bullet : C_\bullet \rightarrow C'_\bullet$ are compatible with the grading. Here, there is no direct analogue to the procedure outlined above, as the mixed basis boundary matrix D^φ fails to have the property described above; $r_j^\varphi \neq 0$ does not imply $r_i^\varphi = 0$ for $i = \text{Pivot } r_j^\varphi$. In order to obtain a useful condition for columns of R^φ to be zero, we need to additionally consider a reduction $R' = D'V'$ of the boundary matrix $D' = FD^\varphi$.

► Proposition 19. *Let $R' = D'V'$ and $R^\varphi = D^\varphi V^\varphi$ be reduced. For all indices j we have $r_j^\varphi = 0$ if and only if $r'_j = 0$.*

Proof. First, note that $r_j^\varphi = 0$ if and only if $Fr_j^\varphi = 0$ because F is invertible. Moreover, FR^φ and R' have the same column space, since $FR^\varphi = R'(V')^{-1}V^\varphi$. Thus, the number of zero columns of R^φ is the same as the number of zero columns of R' since their ranks are equal and their non-zero columns are linearly independent. Now, it suffices to show that $r_j^\varphi = 0$ implies $r'_j = 0$, so assume $r_j^\varphi = 0$. Then $Fr_j^\varphi = 0$, but Fr_j^φ is also the same as the j -th column of $R'(V')^{-1}V^\varphi$. This is a linear combination of columns of R' with non-zero coefficient for r'_j since $(V')^{-1}V^\varphi$ is full-rank and upper-triangular. Non-zero columns of R' are linearly independent, so this linear combination can only be zero if $r'_j = 0$. \blacktriangleleft

In order to apply clearing to the reduction of D^φ , one can now reduce D' with clearing as usual, and clear the columns with the same indices in D^φ . Even more than that, one can not only clear the columns of D^φ whose index appears as a pivot in R' , but rather *every* column with the same index as a zero column in R' , meaning also those that have been reduced to zero via column operations on D' . Thus, with this optimization, the reduction of D^φ only establishes unique pivots among the non-zero columns, but no columns are reduced to zero.

► **Corollary 20.** *If D' has already been reduced to R' , one can initialize the reduction R^φ of D^φ by setting $r_j^\varphi = 0$ for all j with $r'_j = 0$, and no further columns of R^φ will reduce to 0.*

3.3 Assembling Barcodes from (Co)homology Computations

Recalling our concrete setting of persistent homology for simplicial complexes, assume that we are given filtrations L_\bullet and K_\bullet of two isomorphic simplicial complexes $L \cong K$ and a monomorphism $f_\bullet: L_\bullet \rightarrow K_\bullet$, inducing an isomorphism $f: L \rightarrow K$. Following the notation from Example 13 and applying the previous results with $\varphi_\bullet = C_*(f_\bullet)$, we see that the barcode of $\text{im } H_*(f_\bullet)$ can be determined via reductions of D^L and D^f and that the reduction of D^f may be performed with clearing if D^K has already been reduced before.

As known from the single filtration case, clearing requires a full persistence computation in the first homological degree for which persistence is computed. As persistence computations are often only feasible in low dimensions and practitioners are often only interested in barcodes in low degrees, it is much more powerful to apply clearing for cohomological grading, allowing for the initialization to be performed in degree 0. Thus, our goal is to perform cohomological computations and still recover the image $\text{im } H_*(f_\bullet)$ in homology.

As a first step towards that goal, we recall that $\text{im } H^*(f_\bullet)$ and $\text{im } H_*(f_\bullet)$ have the same barcodes [5]. However, the persistent cochain complex giving rise to persistent cohomology is not a filtration, so the basic matrix reduction algorithm does not directly apply there. Instead, we perform computations in the relative cohomology setting given by the map $H^*(f, f_\bullet)$. Its image no longer has the same barcode as $\text{im } H_*(f_\bullet)$, but there are some correspondence results [5, Section 6.2], which we will summarize next. To state the result, for a barcode B we write B^\dagger for the intervals in B that do not extend to any of the endpoints of our index set T and B^∞ for those intervals that do.

► **Proposition 21** (Bauer, Schmahl [5]). *For all degrees d , we have*

$$B(\text{im } H_{d-1}(f_\bullet))^\dagger = B(\text{im } H^d(f, f_\bullet))^\dagger,$$

and the map $I \mapsto T \setminus I$ defines bijections

$$B(\text{im } H_d(f_\bullet))^\infty \leftrightarrow B(H^d(L, L_\bullet))^\infty \quad \text{and} \quad B(\text{im } H^d(f, f_\bullet))^\infty \leftrightarrow B(H_d(K_\bullet))^\infty.$$

Note that none of the intervals in the barcodes considered here span the whole index set T , since we assume that our filtrations start with $L_0 = K_0 = \emptyset$.

Proposition 21 implies that in order to determine the barcode of $\text{im } H_*(f_\bullet)$, it suffices to compute $B(H^*(L, L_\bullet))^\infty$ and $B(\text{im } H^*(f, f_\bullet))^\dagger$. Following Example 10 and Theorem 12, we observe that $B(H^*(L, L_\bullet))^\infty$ may be determined from a reduction of the coboundary matrix $(D^L)^\perp$, and following Example 13 and Theorem 14, we know that $B(\text{im } H^*(f, f_\bullet))^\dagger$ may be determined from a reduction of the coboundary matrix $(D^f)^\perp$. In the relative cohomology setting, the matrices $(D^L)^\perp$ and $(D^f)^\perp$ play the roles of D' and D^φ in the general setting, so by Corollary 20 we can simultaneously reduce these matrices with clearing.

We summarize the discussion in the following theorem. To simplify notation, we will assume that we are given functions k and l on $K \cong L$ that induce the filtrations K_\bullet and L_\bullet , respectively, via their sublevel set filtrations. For example, if K_\bullet and L_\bullet are Vietoris–Rips filtrations for different metrics on the same set of points, the functions l and k would be given by the corresponding diameter functions. Recall that the column and row indices of the matrices $(D^f)^\perp$ and $(D^L)^\perp$ correspond to the simplices of $K \cong L$ in different orders. We denote the column of a matrix X corresponding to a simplex σ by x_σ . Combining Theorem 14, Corollary 20, and Proposition 21, we can now determine barcodes from reductions of boundary matrices as follows.

► **Theorem 22.** *The matrices $(D^f)^\perp$ and $(D^L)^\perp$ can be reduced with clearing, and given reductions $S = (D^f)^\perp W$ and $R = (D^L)^\perp V$, the barcode of $\text{im } H_*(f_\bullet)$ is the multiset*

$$\{[l(\sigma), k(\text{Pivot } s_\sigma)) \neq \emptyset \mid s_\sigma \neq 0\} \cup \{[l(\tau), \infty) \mid r_\tau = 0 \text{ and } \tau \notin \text{Pivots } R\}.$$

Recall that the column and row indices of the coboundary matrices indicated by $(-)^\perp$ correspond to the simplices of $K \cong L$ in reverse filtration order. Hence, the pivot simplex of a column vector appearing in the theorem will be the *first* simplex appearing in the filtration among those that correspond to a non-zero entry of the column, while for the usual boundary matrices D^L , D^K , etc., the pivot simplex of a column would be the one that appears *last* in the filtration. We summarize the algorithm resulting from Theorem 22 in pseudocode in Algorithm 1. To do so, we keep the notation from Section 3.3. In addition, for a column vector c , we write $\text{PivotEntry } c$ to denote the entry of c at its pivot index.

■ **Algorithm 1** Algorithm to compute image persistence via two matrix reductions with clearing in cohomological grading.

Input: Filtration boundary matrix D^L with n columns, mixed basis boundary matrix D^f , maximum homological degree p for persistence to be computed

Result: Barcode of $\text{im } H_*(f)$

$R \leftarrow (D^L)^\perp; S \leftarrow (D^f)^\perp; B \leftarrow \emptyset$

for $m = 0, \dots, p$ **do**

while $\exists \sigma <_L \tau$ with $r_\sigma \neq 0$, $\text{Pivot } r_\sigma = \text{Pivot } r_\tau$, and $\dim \sigma = m$ **do**

$r_\tau \leftarrow r_\tau - \frac{\text{PivotEntry } r_\tau}{\text{PivotEntry } r_\sigma} r_\sigma$

for σ with $\dim \sigma = m$ **do**

if $r_\sigma = 0$ **then**

$s_\sigma \leftarrow 0$

$B \leftarrow B \sqcup \{[l(\sigma), \infty)\}$

else if $\sigma \notin \text{Pivots } R$ **then**

$r_{\text{Pivot } r_\sigma} \leftarrow s_{\text{Pivot } r_\sigma} \leftarrow 0$

while $\exists \sigma <_L \tau$ with $s_\sigma \neq 0$ and $\text{Pivot } s_\sigma = \text{Pivot } s_\tau$ **do**

$s_\tau \leftarrow s_\tau - \frac{\text{PivotEntry } s_\tau}{\text{PivotEntry } s_\sigma} s_\sigma$

for σ with $\dim \sigma = m$, $s_\sigma \neq 0$, and $l(\text{Pivot } w_\sigma) < k(\text{Pivot } s_\sigma)$ **do**

$B \leftarrow B \sqcup \{[l(\sigma), k(\text{Pivot } s_\sigma))\}$

return B

3.4 Apparent and Emergent Pairs in Image Matrix Reduction

An important optimization in persistence computation leading to significant computational improvements is given by utilizing the *apparent pairs* in the filtration, which are pairs (σ, τ) in the filtration such that σ is the latest facet of τ in the filtration and τ is the earliest cofacet of σ . Apparent pairs always form persistence pairs, since the corresponding columns are reduced already in the (co)boundary matrix. More generally, if (σ, τ) is a persistence pair and τ is the earliest cofacet of σ , we say that (σ, τ) is an *emergent cofacet pair*. The special case where such a pair (σ, τ) has persistence 0 can be identified in Ripser [1] during the construction of the columns of the coboundary matrix, terminating this construction early without constructing the entire column.

This strategy turns out to carry over to the image setting as well. The criterion used in Ripser for identifying the pivot index early is that its corresponding simplex appears in the filtration simultaneously with the simplex corresponding to the column. When reducing the

14:12 Efficient Computation of Image Persistence

■ **Table 1** Running time and memory usage for image barcode and standard barcode (of the codomain filtration) for different data sets. The filtrations are defined by two different metrics on the point cloud. The maximum homological degree for persistence to be computed is specified by p , the number of points in the data set is specified by $|X|$.

Data Set	p	$ X $	image barcode	standard barcode
S^2 intrinsic \rightarrow extrinsic	2	128	0.56 s, 45 MB	0.26 s, 47 MB
		256	5.7 s, 315 MB	2.97 s, 316 MB
		512	155 s, 5.7 GB	65.6 s, 5.7 GB
SO(3) intrinsic \rightarrow extrinsic	3	64	0.71 s, 51.7 MB	0.39 s, 52.7 MB
		128	13.7 s, 735 MB	7.3 s, 743 MB
		256	317 s, 13.1 GB	160 s, 13.1 GB
Möbius strip $\rightarrow \mathbb{R}P^2$	1	256	0.34 s, 24.0 MB	0.11 s, 25.0 MB
		512	2.66 s, 159 MB	0.73 s, 159 MB
		1024	25.6 s, 1.06 GB	7.21 s, 1.06 GB
$S^2 \rightarrow \mathbb{R}P^2$	2	32	0.37 s, 11.7 MB	0.00 s, 2.3 MB
		64	10.9 s, 27.1 MB	0.02 s, 7.5 MB
		128	574 s, 608 MB	0.24 s, 31.2 MB

mixed basis coboundary matrix $(D^f)^\perp$ for $f_\bullet: L_\bullet \rightarrow K_\bullet$, we apply the criterion with respect to the filtration K_\bullet , which determines the row order and hence the pivot of a column. Note that the apparent or emergent pairs (σ, τ) identified this way thus have the same filtration value for the filtration K_\bullet .

3.5 Computational Experiments

We provide an implementation [6] of the algorithm resulting from Theorem 22 including the clearing optimization, based on the `simple` branch of Ripser [1], for the special case where $L_\bullet = \text{Rips}_\bullet(X, d)$ and $K_\bullet = \text{Rips}_\bullet(X, d')$ are filtrations of Vietoris–Rips complexes corresponding to two metrics d and d' on a finite set X that satisfy $d(x, y) \geq d'(x, y)$ for all $x, y \in X$, with the map between filtrations given by the inclusions of L_t into K_t . Recall that the inequality $d \geq d'$ ensures that L_t is in fact a subcomplex of K_t . We did not include a comparison with Dionysus [19], as the general two-filtration setting considered in this paper is not supported. We further note that computation of image persistence is no longer supported in the current version of Dionysus.

Our computations were done on a notebook computer with an Apple M2 processor and 24 GB memory. The first example is given by X being $\{128, 256, 512\}$ points sampled uniformly from the unit sphere in \mathbb{R}^3 , with the distance d being given by the geodesic distance on the sphere and the distance d' being given by the Euclidean distance in \mathbb{R}^3 . The second example consists of $\{64, 128, 256\}$ points sampled uniformly at random from SO(3), with d given by the geodesic distance on $\text{SO}(3) \cong \mathbb{R}P^3$ and d' given by the Frobenius norm distance on $\mathbb{R}^{3 \times 3}$ (scaled by a factor of $1/\sqrt{2}$ to ensure that $d \geq d'$ holds). The third example is constructed by sampling $\{256, 512, 1024\}$ points uniformly from a cylinder with height π over a unit circle, equipped with the quotient metric that identifies antipodal points, resulting in a Möbius strip; the canonical map from the cylinder to the unit sphere given by $(\phi, \psi) \mapsto (\sin \phi \cos \psi, \cos \phi \cos \psi, \sin \psi)$ is nonexpanding, and it induces a nonexpanding map from the Möbius strip to the projective plane, both with the intrinsic metric. The fourth example is constructed by sampling $\{32, 64, 128\}$ points from the unit sphere, and considering the canonical quotient map to the projective plane. Running times and memory

usage are summarized in Table 1. Note that the examples differ significantly in terms of difficulty: while the first two examples comparing intrinsic and extrinsic metrics take only roughly twice as long as a standard persistence computation, the other two examples are more demanding, with the last one showing a huge difference in running time and memory usage. We attribute this to the vastly different total orders of simplices for the two filtrations.

References

- 1 Ulrich Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *J. Appl. Comput. Topol.*, 5(3):391–423, 2021. doi:10.1007/s41468-021-00071-5.
- 2 Ulrich Bauer and Michael Lesnick. Induced matchings and the algebraic stability of persistence barcodes. *J. Comput. Geom.*, 6(2):162–191, 2015. doi:10.20382/jocg.v6i2a9.
- 3 Ulrich Bauer and Michael Lesnick. Persistence diagrams as diagrams: A categorification of the stability theorem. In Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thauale, editors, *Topological Data Analysis*, pages 67–96, Cham, 2020. Springer. doi:10.1007/978-3-030-43408-3_3.
- 4 Ulrich Bauer and Maximilian Schmahl. Efficient computation of image persistence. Preprint, 2022. arXiv:2201.04170.
- 5 Ulrich Bauer and Maximilian Schmahl. Lifespan functors and natural dualities in persistent homology. *Homology Homotopy Appl.*, 2023. To appear. arXiv:2012.12881.
- 6 Ulrich Bauer and Maximilian Schmahl. Ripser for image persistence, 2023. GitHub. URL: <https://github.com/Ripser/ripser/tree/image-persistence-simple>.
- 7 Michael Bleher, Lukas Hahn, Juan Angel Patino-Galindo, Mathieu Carriere, Ulrich Bauer, Raul Rabadan, and Andreas Ott. Topology identifies emerging adaptive mutations in sars-cov-2. Preprint, 2021. arXiv:2106.07292.
- 8 Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The structure and stability of persistence modules*. SpringerBriefs in Mathematics. Springer, 2016. doi:10.1007/978-3-319-42545-0.
- 9 Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *Proceedings of the 27th European Workshop on Computational Geometry*, 2011. URL: <https://eurocg11.inf.ethz.ch/abstracts/22.pdf>.
- 10 James Clough, Nicholas Byrne, Ilkay Oksuz, Veronika Zimmer, Julia Schnabel, and Andrew King. A topological loss function for deep-learning based image segmentation using persistent homology. *IEEE transactions on pattern analysis and machine intelligence*, PP, September 2020. doi:10.1109/TPAMI.2020.3013679.
- 11 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete Comput. Geom.*, 37(1):103–120, 2007. doi:10.1007/s00454-006-1276-5.
- 12 David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Dmitriy Morozov. Persistent homology for kernels, images, and cokernels. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1011–1020. SIAM, Philadelphia, PA, 2009. doi:10.1137/1.9781611973068.110.
- 13 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Computational geometry (SCG’06)*, pages 119–126. ACM, New York, 2006. doi:10.1145/1137856.1137877.
- 14 William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *J. Algebra Appl.*, 14(5):1550066, 8, 2015. doi:10.1142/S0219498815500668.
- 15 Y. Dabaghian, F. Mémoli, L. Frank, and G. Carlsson. A topological paradigm for hippocampal spatial map formation using persistent homology. *PLoS Computational Biology*, 8(8):1–14, August 2012. doi:10.1371/journal.pcbi.1002581.
- 16 Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27(12):124003, 17, 2011. doi:10.1088/0266-5611/27/12/124003.

14:14 Efficient Computation of Image Persistence

- 17 Inés García-Redondo, Anthea Monod, and Anna Song. Fast topological signal identification and persistent cohomological cycle matching. Preprint, 2022. [arXiv:2209.15446](https://arxiv.org/abs/2209.15446).
- 18 Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL: <https://proceedings.neurips.cc/paper/2019/file/2d95666e2649fcfc6e3af75e09f5adb9-Paper.pdf>.
- 19 Dmitriy Morozov. Dionysus. URL: <https://mrzv.org/software/dionysus/>.
- 20 Takenobu Nakamura, Yasuaki Hiraoka, Akihiko Hirata, Emerson G Escobar, and Yasumasa Nishiura. Persistent homology and many-body atomic structure for medium-range order in the glass. *Nanotechnology*, 26(30):304001, 2015. doi:10.1088/0957-4484/26/30/304001.
- 21 Y. Reani and O. Bobrowski. Cycle registration in persistent homology with applications in topological bootstrap. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2022. doi:10.1109/TPAMI.2022.3217443.
- 22 Nico Stucki, Johannes C. Paetzold, Suprosanna Shit, Bjoern Menze, and Ulrich Bauer. Topologically faithful image segmentation via induced matching of persistence barcodes. To appear in the proceedings of ICML 2023. Preprint, 2022. [arXiv:2211.15272](https://arxiv.org/abs/2211.15272).

Efficient Two-Parameter Persistence Computation via Cohomology

Ulrich Bauer   

Department of Mathematics, TUM School of Computation, Information and Technology, and Munich Data Science Institute, Technical University of Munich, Germany

Fabian Lenzen  

Department of Mathematics, TUM School of Computation, Information and Technology, Technical University of Munich, Germany

Michael Lesnick   

Department of Mathematics, SUNY Albany, NY, USA

Abstract

Clearing is a simple but effective optimization for the standard algorithm of persistent homology (PH), which dramatically improves the speed and scalability of PH computations for Vietoris–Rips filtrations. Due to the quick growth of the boundary matrices of a Vietoris–Rips filtration with increasing dimension, clearing is only effective when used in conjunction with a dual (cohomological) variant of the standard algorithm. This approach has not previously been applied successfully to the computation of two-parameter PH.

We introduce a cohomological algorithm for computing minimal free resolutions of two-parameter PH that allows for clearing. To derive our algorithm, we extend the duality principles which underlie the one-parameter approach to the two-parameter setting. We provide an implementation and report experimental run times for function-Rips filtrations. Our method is faster than the current state-of-the-art by a factor of up to 20.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Theory of computation → Computational geometry

Keywords and phrases Persistent homology, persistent cohomology, two-parameter persistence, clearing

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.15

Related Version *Full Version*: <https://arxiv.org/abs/2303.11193> [8]

Supplementary Material *Software (Source Code)*: <https://gitlab.com/flenzen/2-parameter-persistent-cohomology> [33]

archived at [swh:1:dir:7ef1c55bee7980f9a68507241eeb461e0af7d99e](https://sw.hawthorn.ai/dir/7ef1c55bee7980f9a68507241eeb461e0af7d99e)

Funding *Ulrich Bauer*: Supported by the German Research Foundation (DFG) through the Collaborative Research Center SFB/TRR 109 *Discretization in Geometry and Dynamics* – 195170736.

1 Introduction

Motivation. Persistent homology [19, 50, 37] analyzes how the homology of a filtered topological space changes as the filtration parameter increases. By assigning filtered spaces (e.g., Vietoris–Rips filtrations) to data sets, it provides simple signatures of the data called *barcodes*, which encode multi-scale information about the shape of the data. Thanks to recent advances in PH computation and software [4, 49, 1, 39, 45, 26, 27], PH has become popular for practical data applications [25]. A well-known stability result [16, 9, 12] guarantees that small perturbations (in the Gromov–Hausdorff distance) of the input data lead to small perturbations (in the bottleneck distance) of the barcodes of the PH of Vietoris–Rips filtrations. However, Vietoris–Rips PH is notoriously unstable to outliers. Besides other strategies [12, Section 1.7], a commonly proposed remedy for this is the introduction of a



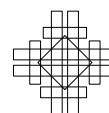
© Ulrich Bauer, Fabian Lenzen, and Michael Lesnick;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 15; pp. 15:1–15:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



second filtration parameter controlling the local density of the point cloud, which leads to the notions of *two-* and *multi-parameter* PH [14, 43, 17, 34, 41, 13]. A central computational problem of two-parameter PH, somewhat akin to the computation of a barcode, is the computation of a *minimal free presentation* or a *minimal free resolution* (MFR) of the PH module. While a resolution contains more information than a presentation, the underlying algorithmic problems are essentially the same, and we focus on the computation of a MFR in this work. Such a resolution is often quite small in practice [23], and computing it is a natural first step in computing invariants or metrics in the multi-parameter setting [13].

Computing a MFR of two-parameter PH is more involved than in the one parameter case. The problem can be solved by classical Gröbner basis algorithms, which work in much greater generality but do not scale well enough for practical TDA applications [35]. Recently, a specialized algorithm was introduced [35, 32], which is far more efficient than the Gröbner basis approach, both in theory and in practice. This has substantially lowered the barrier to practical data analysis with two-parameter persistence [30]. For recent applications, see, e.g., [11, 46].

Nevertheless, existing software for two-parameter PH is much slower and less scalable than one-parameter PH implementations such as [4, 1, 39, 49]. The approach of [35, 32] boils down to a matrix reduction scheme similar to the standard algorithm of one-parameter PH [50], and has the same asymptotic run time, cubic in the size of the complex. However, modern one-parameter PH algorithms incorporate several critical optimizations. In particular, it is known that for Vietoris–Rips filtrations, *clearing* [5, 15] leads to major performance gains when combined with a dual (cohomological) variant of the standard persistence algorithm [18, 5]. All state-of-the-art software for computing Vietoris–Rips PH employ this strategy. In two parameters, however, working with persistent cohomology (PC) is more challenging, essentially because, in contrast to one parameter, relative simplicial cochains of filtered complexes do not form free modules.

Contributions. In order to compute MFRs of PH of function-Rips bifiltrations more efficiently, we introduce a cohomological variant of the algorithm of [35, 32], which we outline now.

Let K_* be a finite simplicial \mathbf{Z}^n -filtration with $K = \bigcup_{z \in \mathbf{Z}^n} K_z$. Assume that K_* is one-critical, i.e., the set $\{z \mid \sigma \in K_z\}$ has a unique minimal element $g(\sigma)$ for every $\sigma \in K$. We define a certain cochain complex $N^\bullet(K_*)$ of free persistence modules. In this paper, $H_\bullet(K)$ always denotes the reduced simplicial homology of K . If $H_d(K_z) = 0$ for all d but finitely many indices $z \in \mathbf{Z}^n$ (which can easily be ensured by adding additional simplices to K_*), then $H^{d+n}(N^\bullet(K_*))$ is isomorphic to the dual module of $H_d(K_*)$ for all d (see Proposition 6). This can be seen as a generalization of a corresponding statement for one-parameter persistence, in which case $N^\bullet(K_*)$ equals the relative cochain complex $C^\bullet(K, K_*)$ [18, Theorem 2.4]; see also [10].

Given a (minimal) free resolution F_\bullet of an n -parameter persistence module M and a choice of basis for each module of F_\bullet , we show that the matrices representing F_\bullet also represent a (minimal) injective resolution of M ; see Theorem 10. In particular, this allows us to easily convert a (minimal) free resolution of a module (e.g., $H^{d+n}(N^\bullet(K_*))$) to a (minimal) free resolution of its dual (i.e., $H_d(K_*)$); see Corollary 14.

For $n = 2$, we propose a method to compute a MFR of $H^{d+2}(N^\bullet(K_*))$ (and thus $H_d(K_*)$) solely from the coboundary map $\delta^{d+1}: N^d(K_*) \rightarrow N^{d+1}(K_*)$; see Section 3.4. At the core of this method is an algorithm for the following problem: given a morphism $f: F \rightarrow F'$ of free persistence modules and a basis of the vector space $\text{colim im } f$, compute a basis of the free persistence module $\ker f$; see Theorem 19. The algorithm is compatible with the clearing optimization, which improves its performance considerably.

We have implemented our approach [33] and report timing results from computational experiments with function-Rips bifiltrations. On most instances considered, our approach is significantly faster than the approach [23] used in `mpfree`, and on certain instances, our implementation is able to outperform the approach of by a factor of up to 20.

A number of recent methods for computation of multiparameter persistence focus on decreasing the size of the input complex without changing its homology [22, 42, 23, 2]. These methods can be used as a preprocessing step to the computation of a minimal free resolution. In our computational experiments, we explore the effect of the chunk preprocessing method of [23] on the efficiency of our method. We find that in our experiments, our method generally performs better without this preprocessing. In contrast, we observe that the preprocessing is very helpful for the the approach of [35, 32], as previously reported [23]. We also observe that applying the chunk algorithm on cochain complexes instead of chain complexes may significantly increase the performance even for homology computation.

2 Background

2.1 Persistence modules

Let k be a field, let $n \in \mathbf{N}$, let \mathbf{vec} denote the category of finite dimensional k -vector spaces, and consider \mathbf{Z}^n as a poset with the usual product partial order. A (pointwise finite dimensional) \mathbf{Z}^n -persistence module, also called an n -parameter persistence module, is a functor $M: \mathbf{Z}^n \rightarrow \mathbf{vec}$. The maps $M_{z \leq z'}: M_z \rightarrow M_{z'}$ are called the *structure maps* of M . If $m \in M_z$, we call z the *grade* of m , denoted by $g(m)$. The *total dimension* of M is $\sum_{z \in \mathbf{Z}^n} \dim M_z$. We write \mathbf{Z}^n -pers for the abelian category of pointwise finite dimensional \mathbf{Z}^n -persistence modules. Its morphisms are natural transformations. \mathbf{Z}^n -pers is equivalent to a full subcategory of the category of multigraded modules over the ring $k[x_1, \dots, x_n]$; see [14]. The algebra $k[x_1, \dots, x_n]$ is not a principal ideal domain unless $n = 1$; therefore, \mathbf{Z}^n -persistence modules cannot be described by a barcode for $n > 1$.

Let $V^* = \text{Hom}_k(V, k)$ denote the dual of a k -vector space V . The *dual* of a \mathbf{Z}^n -persistence module M is the \mathbf{Z}^n -persistence module M^* with $(M^*)_z = (M_{-z})^*$ and $(M^*)_{z \leq z'} = (M_{-z' \leq -z})^*$. An object M of an abelian category \mathcal{C} is *projective* (respectively *injective*) if the functor $\text{Hom}_{\mathcal{C}}(M, -)$ (respectively $\text{Hom}_{\mathcal{C}}(-, M)$) is exact. The duality $M \mapsto M^*$ is an exact contravariant equivalence of categories and thus maps projective to injective modules and vice versa.

For $z \in \mathbf{Z}^n$, let $F(z)$ be the module with $F(z)_w = \begin{cases} k & \text{if } z \leq w, \\ 0 & \text{otherwise,} \end{cases}$ and $F(z)_{w \leq w'} = \text{id}_k$ if $z \leq w \leq w'$. A module F is *free* if there are elements $(z_i)_{i \in I} \subseteq \mathbf{Z}^n$, for some indexing set I , such that there is an isomorphism $b: \bigoplus_{i \in I} F(z_i) \rightarrow F$. Every finitely generated projective persistence module is free [40, 44, 28]. Let e_i denote the element $1 \in F(z_i)_{z_i}$ of the component of $\bigoplus_{i \in I} F(z_i)$ indexed by i . The set $\{b(e_i) \mid i \in I\}$ is a *basis* of F . The multiset $\text{rk } F := \{z_i \mid i \in I\}$ is uniquely determined by F and called its (*graded*) *rank*. A module M is *finitely generated* if there is a pointwise surjection $F \rightarrow M$ from a free module F of finite rank; the image of a basis of F under such a map is called a *generating system* of M .

A *graded matrix* M is a matrix with entries $M_{ij} \in k$ whose rows and columns are decorated with row grades rg_*^M and column grades cg_*^M . The *graded transpose* of a graded $m \times n$ -matrix M is the graded $n \times m$ -matrix M^T with entries $(M^T)_{ij} = M_{m+1-j, n+1-i}$, row grades $\text{rg}_i^{M^T} = -\text{cg}_{n+1-i}^M$ and column grades $\text{cg}_j^{M^T} = -\text{rg}_{m+1-i}^M$. A morphism $f: F \rightarrow F'$ of finite rank free modules F and F' with respective bases b_1, \dots, b_n and b'_1, \dots, b'_m is uniquely represented by a graded $m \times n$ -matrix M with $\text{cg}_j^M = g(b_j)$, $\text{rg}_i^M = g(b'_i)$, and entries M_{ij} such that $f(b_j) = \sum_i M_{ij} F'_{g(b'_i) \leq g(b_j)}(b'_i)$ for all j .

► **Lemma 1.** *A graded matrix M represents a morphism of finite rank free modules iff $M_{ij} = 0$ whenever $\text{rg}_i^M \not\leq \text{cg}_j^M$.*

Proof. This follows from the fact that $\text{Hom}(F(z), F(z')) = \begin{cases} k & \text{if } z \geq z', \\ 0 & \text{otherwise.} \end{cases}$ ◀

If bases of free modules F, F' are fixed, we identify a morphism $F \rightarrow F'$ with the graded matrix representing it. A *free resolution* (resp., *injective resolution*) of a module M is a chain complex $F_\bullet: \cdots \rightarrow F_1 \rightarrow F_0$ of free modules (resp., cochain complex $I^\bullet: I^0 \rightarrow I^1 \rightarrow \cdots$ of injective modules) concentrated in non-negative degrees that is quasi-isomorphic to M . A (*homological*) *d-ball* is an acyclic chain complex of the form $\cdots \rightarrow 0 \rightarrow F(z) \xrightarrow{\text{id}} F(z) \rightarrow 0 \rightarrow \cdots$ for some $z \in \mathbf{Z}^n$, concentrated in degrees $d, d-1$. A free resolution, and, more generally, a chain complex of free modules, is called *minimal* if it contains no direct summand isomorphic to a ball. An injective resolution is *minimal* if its dual is minimal. A morphism $F_1 \rightarrow F_0$ of free modules is called a (*minimal*) *free presentation* of a module M if it extends to a (minimal) free resolution of M .

► **Theorem 2** (see [21, Theorem 20.2], [38, Theorem 7.5]). *Every finitely generated module has a MFR. Every free resolution is isomorphic to the direct sum of a MFR with a direct sum of homological balls. In particular, a MFR is unique up to isomorphism of chain complexes.*

Thus, letting F_\bullet be a MFR of a finitely generated persistence module M , the graded ranks $\beta_q(M) := \text{rk } F_q$, called the *graded Betti numbers* of M , are independent of the choice of F_\bullet .

► **Theorem 3** (Hilbert's Syzygy theorem [38, Theorem 15.2], [21, Corollary 19.7]). *Every \mathbf{Z}^n -persistence module has a MFR of length at most n .*

2.2 Filtrations

For P any poset, a (simplicial) *P-filtration* is a functor K_* from \mathbf{Z}^n to the category of simplicial complexes such that the simplicial maps $K_{z \leq z'}$ are inclusions. We write $K = \bigcup_{z \in \mathbf{Z}^n} K_z$.

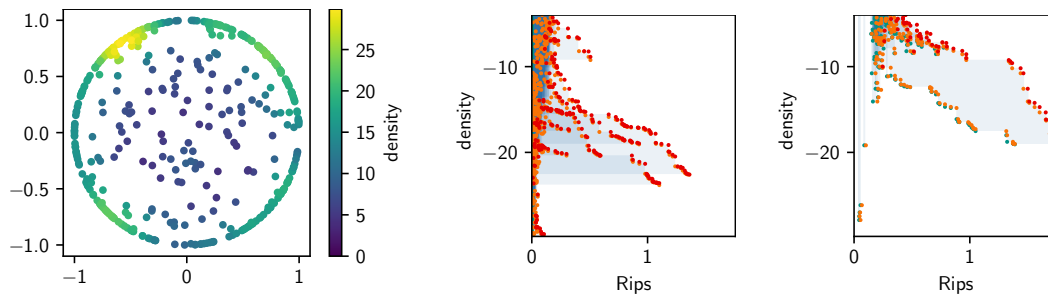
► **Example 4.** Let S be a metric space and let $\text{diam } \sigma = \max_{s,t \in \sigma} d(s,t)$ for every finite, non-empty $\sigma \subseteq S$. The *Vietoris–Rips filtration* $\widehat{VR}_*(S)$ associated to S is the \mathbf{R} -filtration given by $\widehat{VR}_r(S) = \{\sigma \subseteq S \mid 0 < |\sigma| < \infty, \text{diam } \sigma \leq r\}$. If S is finite and non-empty, let $r_1 < r_2 < \cdots < r_n$ be the distinct values $\text{diam } \sigma$ can attain for $\sigma \subseteq S$. By setting

$$VR_i(S) = \begin{cases} \widehat{VR}(S)_{r_1} & \text{if } i \leq 1, \\ \widehat{VR}(S)_{r_i} & \text{if } 1 < i < n, \\ \widehat{VR}(S)_{r_n} & \text{if } n \leq i, \end{cases}$$

we obtain a \mathbf{Z} -filtration $VR_*(S)$, which we also call a Vietoris–Rips filtration.

If K_* is a \mathbf{Z}^n -filtration, its absolute and relative simplicial chains $C_\bullet(K_*)$ and $C_\bullet(K, K_*)$ (with coefficients in k) form chain complexes of \mathbf{Z}^n -persistence modules, and the respective cycles $Z_d(-)$, boundaries $B_d(-)$ and homology $H_d(-)$ are \mathbf{Z}^n -persistence modules for all d . The dual cochain complex $C^\bullet := (C_\bullet)^*$ of a chain complex C_\bullet has components $C^d = (C_d)^*$. Its cocycles $Z^d(C^\bullet)$, coboundaries $B^d(C^\bullet)$ and cohomology $H^d(C^\bullet)$ are \mathbf{Z}^n -persistence modules for all d . Because $(-)^*$ is exact, there is a natural isomorphism $H^d(C^\bullet) \rightarrow H_d(C_\bullet)^*$ for all d . Our indexing convention is that a chain complex C_\bullet has the boundary morphisms $\partial_d: C_d \rightarrow C_{d-1}$ and a cochain complex C^\bullet has coboundary morphisms $\delta^d = (\partial_d)^*: C^{d-1} \rightarrow C^d$. This differs from the standard convention, but is chosen such that $C^\bullet = (C_\bullet)^*$ has $\delta^d = (\partial_d)^*$.

A \mathbf{Z}^n -filtration K_* is *one-critical* if for every $\sigma \in K$ the set $\{z \in \mathbf{Z}^n \mid \sigma \in K_z\}$ has a unique minimal element $g(\sigma)$, called the *grade* of σ . In this case, $C_\bullet(K_*) = \bigoplus_{\sigma \in K_*} F(g(\sigma))$ is free, with a basis $\{e_\sigma \mid \sigma \in K_*\}$ satisfying $g(e_\sigma) = g(\sigma)$. In particular, ∂_\bullet can be represented by a graded matrix $[\partial_\bullet]$.



■ **Figure 1** A point set S (left) with $|S| = 400$, with density function $\rho(p) := \sum_{q \in S \setminus \{p\}} \exp(-\frac{d(p,q)^2}{2\sigma^2})$ for $\sigma = 0.15$; graded Betti numbers (teal: β_0 , red: β_1 , orange: β_2) and Hilbert function (shades of blue increasing from $\dim = 0$ to $\dim \geq 10$) of $H_0(VR_*^2(S))$ (middle) and $H_1(VR_*^2(S))$ (right).

► **Example 5.** The *function-Rips bifiltration* $VR_*^f(S)$ associated to a finite metric space S and a function $f: S \rightarrow \mathbf{Z}$ is the one-critical \mathbf{Z}^2 -filtration with $VR_{x,y}^f = \{\sigma \in VR_y(S) \mid \max_{s \in \sigma} f(s) \leq x\}$. Figure 1 illustrates the Hilbert function and graded Betti-numbers of the PH of a function-Rips bifiltration, where the function is a density function.

2.3 One-parameter persistence and clearing

We next turn attention to persistence modules over \mathbf{Z} . For $-\infty \leq b_i < d_i \leq \infty$, let $I(b, d)$ be the *interval module* with $I(b, d)_z = \begin{cases} k & \text{if } b \leq z < d, \\ 0 & \text{otherwise} \end{cases}$ and $I(b, d)_{z \leq z'} = \text{id}_k$ if $b \leq z \leq z' < d$. Every pointwise finite-dimensional \mathbf{Z} -persistence module is isomorphic to an essentially unique direct sum $\bigoplus_{i \in I} I(b_i, d_i)$ [47, 50]. The collection of the pairs (b_i, d_i) is called the *barcode* of M .

Given a finite \mathbf{Z} -filtered complex K_* , one is usually interested in computing the barcode of $H_\bullet(K_*)$. Since $k[x]$ is a principal ideal domain, the submodules $Z_d(K_*), B_d(K_*) \subseteq C_d(K_*)$ are free for all d . The *standard algorithm* [20, §3] computes bases of $Z_d(K_*)$ and $B_d(K_*)$ and thus the barcode of $H_\bullet(K_*)$ by applying an order-respecting Gaussian column reduction scheme to each graded matrix $[\partial_d]$. Each relative cochain module $C^d(K, K_*)$ is also free, so the same algorithm computes the barcode of $H^\bullet(K, K_*)$ from the graded matrices $[\delta^d] = [\partial_d]^T$ representing the coboundary operators δ^d . The barcodes of $H_\bullet(K_*)$ and $H^\bullet(K, K_*)$ determine each other in a simple way, as is seen by considering the long exact sequence of the pair (K, K_*) [18, 10].

It has been observed that for Vietoris–Rips filtrations, computing $H^\bullet(K, K_*)$ instead of $H_\bullet(K_*)$ is far more efficient. This increase in efficiency hinges on the use of the *clearing* optimization scheme [5, 15, 4], which we now explain. The *pivot* of a matrix column is the largest row index of a non-zero entry in that column. The standard algorithm applies left-to-right column additions to bring $[\delta^{d+1}]$ into *reduced* form R^{d+1} , meaning that all columns of R^{d+1} have pairwise distinct pivots. If a column R_j^d is non-zero with pivot i , then $R_i^{d+1} = 0$. Therefore, if R^d is known from previous computations, the reduction of $[\delta^{d+1}]_j$ to zero can be skipped. As the standard algorithm would typically spend most of its run time on the columns of $[\delta^{d+1}]$ that are reduced to zero, skipping most of these accelerates the algorithm considerably.

If the reduced homology $H_d(K)$ is zero for all d , then the long exact sequence of the pair (K, K_*) shows that $H^{d+1}(K, K_*)^* \cong H_d(K_*)$ for all d , so one would expect that they can be computed from the same data. Indeed, one can use clearing to compute $H^{d+1}(K, K_*)$ (respectively $H_d(K_*)$) from δ^{d+1} (respectively ∂_{d+1}) alone; see Algorithm 3 in the full version.

2.4 Computation of 2-parameter persistence

The LW-Algorithm. Assume that C_\bullet is a chain complex of free \mathbf{Z}^2 -persistence modules of finite rank; e.g., $C_\bullet = C_\bullet(K_*)$ for a one-critical \mathbf{Z}^2 -filtration K_* , and let D_d be the matrix representing $\partial_d: C_d \rightarrow C_{d-1}$ for all d . Theorems 2 and 3 imply that the kernel of a morphism of finitely generated free \mathbf{Z}^2 -modules is free. In particular, $Z_d(C_\bullet)$ is free for all d , so the sequence $0 \rightarrow Z_{d+1}(C_\bullet) \xrightarrow{i_{d+1}} C_{d+1} \xrightarrow{p_{d+1}} Z_d(C_\bullet)$ is a free resolution of $H_d(K_*)$. From D_d , the *LW-Algorithm* [35, 32] (see Algorithm 4 in the full version) computes a graded matrix I_d representing $i_d: Z_d(C_\bullet) \hookrightarrow C_d$. A variant of that algorithm (see Algorithm 5 in the full version) computes from D_{d+1} a graded matrix $D'_{d+1}: C'_{d+1} \rightarrow C_d$, whose columns represent a minimal generating system of $B_d(C_\bullet)$, together with a graded matrix I'_{d+1} representing the kernel Z'_{d+1} of the morphism represented by D'_{d+1} . There is a unique graded matrix P'_{d+1} such that $D'_{d+1} = I_d P'_{d+1}$, which can be obtained by Algorithm 9 in the full version. Then I'_{d+1} and P'_{d+1} represent a free resolution

$$0 \rightarrow Z'_{d+1} \xrightarrow{I'_{d+1}} C'_{d+1} \xrightarrow{P'_{d+1}} Z_d(C_\bullet) \quad (1)$$

of $H_d(C_\bullet)$. To obtain a MFR, it remains to split off summands from (1) that are isomorphic to homological balls. There is an embarrassingly parallel algorithm that computes a minimal chain complex quasi-isomorphic to a given one; see Remark 23 in the full version. In particular, this algorithm can be used to convert a free resolution to a minimal one. It can also be used to split off balls from the input complex C_\bullet . This is known as *chunk preprocessing* and typically improves performance of the LW-algorithm by a considerable amount [22, 32, 23].

3 Cohomology computation

Let K_* be a one-critical \mathbf{Z}^n -filtration. If $n > 1$, then neither $C^\bullet(K_*)$ nor $C^\bullet(K, K_*)$ are complexes of free modules. Since the LW-algorithm assumes that the input complex is a complex of free modules, the strategy from Section 2.4 cannot be used to compute $H^\bullet(K_*)$ or $H^\bullet(K, K_*)$ directly. Instead, we consider a cochain complex $N^\bullet(K_*)$ that can be used to compute $H_d(K_*)$.

3.1 The free cochain complex $N^\bullet(K_*)$

For a module M and $z \in \mathbf{Z}^n$, let $M\langle z \rangle$ be the module with graded components $M\langle z \rangle_w = M_{z+w}$. For $z \geq 0$, the structure maps of M give a morphism $M \rightarrow M\langle z \rangle$. Note that $M\langle z \rangle^* = M^*\langle -z \rangle$. For a graded matrix A , let $A\langle z \rangle$ be the graded matrix with $A\langle z \rangle_{ij} = A_{ij}$, $\text{rg}_i^{A\langle z \rangle} = \text{rg}_i^A + z$ and $\text{cg}_j^{A\langle z \rangle} = \text{cg}_j^A + z$ for all i, j .

Fix a total order on the simplices of K_* , so that the boundary map ∂_\bullet of the chain complex $C_\bullet(K_*) = \bigoplus_{\sigma \in K_*} F(g(\sigma))$ is represented by the graded matrix $[\partial_\bullet]$. Let $\epsilon = (1, \dots, 1) \in \mathbf{Z}^n$. Let

$$N^\bullet(K_*) = \bigoplus_{\sigma} F(-g(\sigma) + \epsilon)$$

be the cochain complex whose coboundary operator δ_N^\bullet is represented by $[\partial_\bullet]^T \langle -\epsilon \rangle$ with respect to the standard basis. It follows from Lemma 1 that this is a well-defined cochain complex. The key property of this chain complex is summarized in the following proposition, whose proof is deferred to the next subsection.

► **Proposition 6.** *If $H_d(K_*)$ has finite total dimension for all d , then there is a natural isomorphism $H_d(K_*) \cong H^{d+n}(N^\bullet(K_*))^*$ for all d .*

► **Corollary 7.** *If $H_\bullet(K_*)$ has finite total dimension and F_\bullet is a free resolution of $H^{d+n}(N^\bullet(K_*))$, then $(F_\bullet)^*$ is an injective resolution of $H_d(K_*)$.*

3.2 The Calabi–Yau-property of persistence modules

Besides proving Proposition 6, we will need to convert the injective resolution of $H_d(K_*)$ from Corollary 7 into a free resolution of $H_d(K_*)$. Both will follow from Theorem 10, which establishes a property of persistence modules known as the *Calabi–Yau property* in some areas of algebra [24]; see [29, Lemma 4.1] for a proof in a more general context. As it turns out, there is a close correspondence between injective and free resolutions that we explore in this section. For $z \in \mathbf{Z}^n$, we define the injective module $I(z) = F(-z)^*$; i.e., $I(z)_w = \begin{cases} k & \text{if } w \leq z, \\ 0 & \text{otherwise,} \end{cases}$ and $I(z)_{w \leq w'} = \text{id}$ if $w \leq w' \leq z$.

► **Definition 8.** *For persistence modules M, N , let $\text{Hom}(M, N)$ be the persistence module with components $\text{Hom}(M, N)_z = \text{Hom}(M, N\langle z \rangle)$.*

Let $\mathcal{P}_{\mathbf{Z}^n\text{-pers}}$ and $\mathcal{I}_{\mathbf{Z}^n\text{-pers}}$ be the full subcategories of $\mathbf{Z}^n\text{-pers}$ consisting of free and injective modules, respectively.

► **Lemma 9** (see [3, Proposition 2.10 in Chapter III]). *The Nakayama functor*

$$\nu := \text{Hom}(-, F(0))^* : \mathcal{P}_{\mathbf{Z}^n\text{-pers}} \rightarrow \mathcal{I}_{\mathbf{Z}^n\text{-pers}}$$

is an equivalence of categories with quasi-inverse $\nu^{-1} = \text{Hom}(I(0)^, -)$.*

One checks that $\nu F(z) = I(z)$ and $\nu^{-1}I(z) = F(z)$. Therefore, $N^\bullet(K_*) = (\nu C_\bullet(K_*)\langle \epsilon \rangle)^*$. For a chain complex C_\bullet and $i \in \mathbf{Z}$, let $C_\bullet[i]$ be the chain complex whose d th module is $(C_\bullet[i])_d = C_{i+d}$. Analogously, for a cochain complex C^\bullet , let $C^\bullet[i]$ be the cochain complex with $(C^\bullet[i])^d = C^{i+d}$. Note that $C_\bullet[i]^* = (C_\bullet)^*[i]$.

► **Theorem 10.** *If F_\bullet is a complex of free \mathbf{Z}^n -persistence modules such that $H_d(F_\bullet)$ has finite total dimension for all d , then F_\bullet and $\nu F_\bullet[n]\langle \epsilon \rangle$ are naturally quasi-isomorphic.*

Proof. For $z \in \mathbf{Z}^n$, we write $z = (z_1, \dots, z_n)$. For $n \in \mathbf{N}$, let $[n] := \{1, \dots, n\}$ and let $\binom{[n]}{k} := \{S \subseteq [n] \mid |S| = k\}$. For $S = \{s_1 < \dots < s_k\} \in \binom{[n]}{k}$ and $z \in \mathbf{Z}^n$, $w \in \mathbf{Z}^k$, we let

$$z|_w^S := (z_1, \dots, z_{s_1-1}, w_1, z_{s_1+1}, \dots, z_{s_k-1}, w_k, z_{s_k+1}, \dots, z_n)$$

be the n -tuple obtained from z by replacing the components indexed by S by the entries of w . For any module $M \in \mathbf{Z}^n\text{-pers}$ and $S \subseteq [n]$, we let $\text{Colim}_S M$ be the module with

$$(\text{Colim}_S M)_z = \text{colim}_{w \in \mathbf{Z}^k} M_{z|_w^S}, \quad (\text{Colim}_S M)_{z \leq z'} = \text{colim}_{w \in \mathbf{Z}^k} M_{z|_w^S \leq z'|_w^S}.$$

For example, for $n = 3$ we get $(\text{Colim}_{\{1,3\}} M)_{(z_1, z_2, z_3)} = \text{colim}_{(w_1, w_2) \in \mathbf{Z}^2} M_{(w_1, z_2, w_2)}$. The module $\text{Colim}_S M$ is constant along the axes specified by S . In particular, $\text{Colim} M = \text{Colim}_{[n]} M$ is the module that is constantly $\text{colim} M$. For a module M , we define the modules $K_k M = \bigoplus_{S \in \binom{[n]}{k}} \text{Colim}_S M$ for each k . If $S \subseteq S'$, then there is a canonical morphism $\text{Colim}_S M \rightarrow \text{Colim}_{S'} M$. For a free module $F(z)$, these assemble to an exact sequence

$$0 \rightarrow F(z) \rightarrow K_1 F(z) \rightarrow \dots \rightarrow K_n F(z) \rightarrow I(z)\langle \epsilon \rangle \rightarrow 0, \tag{2}$$

15:8 Efficient Two-Parameter Persistence Computation via Cohomology

called the *Koszul complex* of $F(z)$. The last morphism is the canonical morphism $K_n F(z) = \text{Colim } F(z) = \text{Lim } I(z)\langle \epsilon \rangle \rightarrow I(z)\langle \epsilon \rangle$. Let F_\bullet be a bounded complex of free modules. Using $\nu F(z) = I(z)$, we get an exact sequence

$$K_\bullet: \quad 0 \rightarrow F_\bullet \rightarrow K_1 F_\bullet \rightarrow \cdots \rightarrow K_n F_\bullet \rightarrow \nu F_\bullet \langle \epsilon \rangle \rightarrow 0 \quad (3)$$

of chain complexes, given by taking a shifted copy of (2) for every $F(z)$ in F_\bullet . We unsplice (3) into short exact sequences

$$\begin{array}{ccccccc}
 & & 0 & & 0 & & 0 \\
 & & \searrow & & \searrow & & \searrow \\
 & & & U_\bullet^{(1)} & & & U_\bullet^{(n-1)} \\
 & & \swarrow & & \swarrow & & \swarrow \\
 0 & \rightarrow & F_\bullet & \rightarrow & K_1 F_\bullet & \rightarrow & K_2 F_\bullet \rightarrow \cdots \rightarrow K_n F_\bullet \rightarrow \nu F_\bullet \langle \epsilon \rangle \rightarrow 0 \\
 & & \parallel & & \nearrow & & \searrow \\
 & & F_\bullet & & & U_\bullet^{(2)} & & \nu F_\bullet \langle \epsilon \rangle \\
 & & \swarrow & & \swarrow & & \swarrow \\
 & & 0 & & & 0 & & 0
 \end{array} \quad (4)$$

with chain complexes $U_\bullet^{(k)}$ for each k . Each of these short exact sequences gives a triangle in the derived category $\mathcal{D}^b(\mathbf{Z}^n\text{-pers})$ [48, §10.4.9]. We obtain connecting homomorphisms

$$\partial^{(1)}: U_\bullet^{(1)}[1] \rightarrow F_\bullet, \quad \partial^{(2)}: U_\bullet^{(2)}[1] \rightarrow U_\bullet^{(1)}, \quad \dots \quad \partial^{(n-1)}: \nu F_\bullet \langle \epsilon \rangle[1] \rightarrow U_\bullet^{(n-1)}$$

in $\mathcal{D}^b(\mathbf{Z}^n\text{-pers})$. These fit into the long exact sequences

$$\begin{aligned}
 \cdots & \rightarrow H_{d+1}(K_1 F_\bullet) \rightarrow H_{d+1}(U_\bullet^{(1)}) \xrightarrow{\partial^{(1)}} H_d(F_\bullet) \rightarrow H_d(K_1 F_\bullet) \rightarrow \cdots, \\
 \cdots & \rightarrow H_{d+2}(K_2 F_\bullet) \rightarrow H_{d+2}(U_\bullet^{(2)}) \xrightarrow{\partial^{(2)}} H_{d+1}(U_\bullet^{(1)}) \rightarrow H_{d+1}(K_2 F_\bullet) \rightarrow \cdots, \\
 & \vdots \\
 \cdots & \rightarrow H_{d+n}(K_n F_\bullet) \rightarrow H_{d+n}(\nu F_\bullet \langle \epsilon \rangle) \xrightarrow{\partial^{(n-1)}} H_{d+n-1}(U_\bullet^{(n-1)}) \rightarrow H_{d+n-1}(K_n F_\bullet) \rightarrow \cdots,
 \end{aligned} \quad (5)$$

induced by the short exact sequences (4). Since $H_d(F_\bullet)$ is of finite total dimension for all d , we have $\text{Colim}_S H_d(F_\bullet) = 0$ if $|S| > 0$. The functor Colim_S is exact for all S because it is a directed colimit. In particular, $H_d(K_k F_\bullet) = H_d(\bigoplus_{|S|=k} \text{Colim}_S F_\bullet) = 0$ for all $k > 0$. Therefore, the long exact sequences (5) show that all connecting homomorphisms $\partial^{(k)}$ are quasi-isomorphisms. Thus, $\partial^{(1)} \circ \cdots \circ \partial^{(n-1)}: \nu F_\bullet \langle \epsilon \rangle \rightarrow F_\bullet$ is a quasi-isomorphism. ◀

► **Corollary 11.** *Let $M \in \mathbf{Z}^n\text{-pers}$ be of finite total dimension.*

1. *If F_\bullet is a free resolution of M , then $\nu F_\bullet[n]\langle \epsilon \rangle$ is an injective resolution of M .*
2. *If I^\bullet is an injective resolution of M , then $\nu^{-1} I^\bullet[-n]\langle -\epsilon \rangle$ is a free resolution of M .*

Proof of Proposition 6. With $N^\bullet(K_*) = (\nu C_\bullet(K_*)\langle \epsilon \rangle)^*$, Theorem 10 gives

$$H^{d+n}(N^\bullet(K_*))^* \cong H_d(N^\bullet(K_*)^*[n]) \cong H_d(\nu C_\bullet(K_*)\langle \epsilon \rangle[n]) \cong H_d(C_\bullet(K_*)) = H_d(K_*). \quad \blacktriangleleft$$

► **Lemma 12.** *A graded matrix M represents a morphism $\bigoplus_j I(\text{cg}_j^M) \rightarrow \bigoplus_i I(\text{rg}_i^M)$ iff $M_{ij} = 0$ whenever $\text{rg}_i^M \not\preceq \text{cg}_j^M$.*

Proof. This follows from $\text{Hom}(I(z), I(z')) = \begin{cases} k & \text{if } z \geq z' \\ 0 & \text{otherwise} \end{cases} \quad \blacktriangleleft$

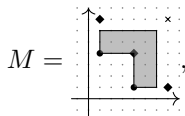
In particular, a graded matrix represents a morphism of free modules iff it represents a morphism of injective modules (cf. Lemma 1).

► **Lemma 13.** Let $f: \bigoplus_{j=1}^n F(z_j) \rightarrow \bigoplus_{i=1}^m F(z'_i)$ be a morphism of free modules represented by the graded matrix $[f]$. Then the morphism $\nu f: \bigoplus_{j=1}^n I(z_j) \rightarrow \bigoplus_{i=1}^m I(z'_i)$ is represented by the same graded matrix $[\nu f] = [f]$.

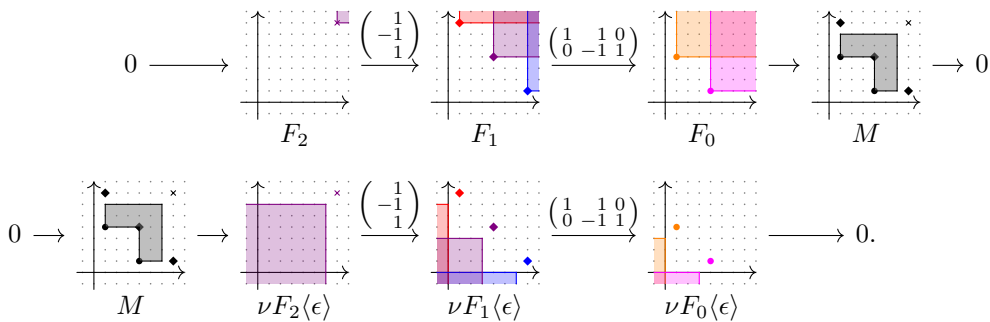
► **Corollary 14.** For $M \in \mathbf{Z}^n\text{-pers}$ of finite total dimension and graded matrices U_1, \dots, U_n , the following are equivalent:

1. U_1, \dots, U_n represent a free resolution $\dots \rightarrow F_n \xrightarrow{U_n} \dots \xrightarrow{U_1} F_0$ of M ,
 2. $U_1\langle\epsilon\rangle, \dots, U_n\langle\epsilon\rangle$ represent an injective resolution $I^0 \xrightarrow{U_n\langle\epsilon\rangle} \dots \xrightarrow{U_1\langle\epsilon\rangle} I^n \rightarrow 0 \dots$ of M ,
 3. $U_1\langle\epsilon\rangle^T, \dots, U_n\langle\epsilon\rangle^T$ represent a free resolution $\dots \rightarrow G_n \xrightarrow{U_n\langle\epsilon\rangle^T} \dots \xrightarrow{U_1\langle\epsilon\rangle^T} G_0$ of M^* .
- In this case $I^q = \nu F_{n-q}\langle\epsilon\rangle = G_q^*$ for all q .

► **Example 15.** Consider the module



where $M_z = k$ if z lies in the shaded region, $M_z = 0$ otherwise, and all structure morphisms between non-zero vector spaces of M being identities. The first line of the following diagram exhibits F_\bullet as a free resolution of M , and the second line exhibits $\nu F_\bullet[2]\langle\epsilon\rangle$ as an injective resolution of M :



3.3 Pulling back modules from the colimit

From now on, we consider \mathbf{Z}^2 -persistence modules only. It remains to explain how we compute $H^\bullet(N^\bullet(K_*))$. In principle, this could be done by a procedure analogous to the LW-Algorithm described in Section 2.4: the horizontal sequence in the commutative diagram

$$\begin{array}{ccccccc}
 & & N^d(K_*) & & & & \\
 & & \downarrow & \searrow^{\delta^{d+1}} & & & \\
 0 & \longrightarrow & Z^{d+1}(N^\bullet(K_*)) & \xrightarrow{i^{d+1}} & N^{d+1}(K_*) & \xrightarrow{p^{d+1}} & Z^{d+2}(N^\bullet(K_*)) \\
 & & & & \searrow^{\delta^{d+2}} & \downarrow_{i^{d+2}} & \\
 & & & & & & N^{d+2}(K_*) \\
 & & & & & & \searrow^{\delta^{d+3}} \\
 & & & & & & N^{d+3}(K_*)
 \end{array} \tag{6}$$

is a free resolution of $H^{d+2}(N^\bullet(K_*))$, and we can obtain matrices representing this resolution as described in Section 2.4. This would, however, involve the coboundary maps δ^{d+2} and

15:10 Efficient Two-Parameter Persistence Computation via Cohomology

δ^{d+3} , leading to a very expensive computation, especially for function-Rips bifiltrations. Instead, we propose a method that computes a free resolution of $H^{d+2}(N^\bullet(K_*))$ from δ^{d+1} only.

For a vector space V , denote by ΔV the persistence modules with components $(\Delta V)_z = V$, such that all structure morphisms of ΔV are the identity. Let $\text{Colim } M = \Delta \text{colim } M$.

► **Definition 16.** For a module $M \in \mathbf{Z}^2\text{-pers}$ and a vector space $V \subseteq \text{colim } M$, we let $[V]_M \in \mathbf{Z}^2\text{-pers}$ be the preimage of ΔV under the canonical map $\eta_M: M \rightarrow \text{Colim } M$.

► **Lemma 17.** If $f: M \rightarrow N$ is a morphism and N is free, then $\ker f = [\text{colim } \ker f]_M$.

Proof. If N is free, then $\eta_N: N \rightarrow \text{Colim } N$ is injective. For every submodule $L \subseteq M$, we have $L \subseteq \eta_M^{-1}(\text{Colim } L) = [\text{colim } L]_M$, so $\ker f \subseteq [\text{colim } \ker f]_M$. It remains to show the other inclusion $[\text{colim } \ker f]_M \subseteq \ker f$. Consider the commutative diagram

$$\begin{array}{ccccc}
 [\text{colim } \ker f]_M & \xleftarrow{j} & & & \\
 \downarrow \eta_{[\text{colim } \ker f]_M} & \swarrow & \ker f & \xleftarrow{i} & M & \xrightarrow{f} & N \\
 & & \downarrow \eta_{\ker f} & & \downarrow \eta_M & & \downarrow \eta_N \\
 \text{Colim } \ker f & \xrightarrow{\text{Colim } i} & \text{Colim } M & \xrightarrow{\text{Colim } f} & \text{Colim } N.
 \end{array}$$

The functor Colim is a directed colimit and thus exact. Therefore, $\text{Colim } \ker f = \ker \text{Colim } f$. This implies $\eta_N \circ f \circ j = \text{Colim } f \circ \text{Colim } i \circ \eta_{[\text{colim } \ker f]_M} = 0$. Since η_N is injective, we obtain $f \circ j = 0$. Therefore, j factors uniquely through $\ker f$. This proves the claim. ◀

The *lexicographic order* \preceq_{lex} and the *colexicographic order* \preceq_{colex} are the total orders on \mathbf{Z}^2 defined as

$$\begin{aligned}
 (x, y) \preceq_{\text{lex}} (x', y') & \text{ iff either } x < x' \text{ or } x = x' \text{ and } y \leq y', \\
 (x, y) \preceq_{\text{colex}} (x', y') & \text{ iff either } y < y' \text{ or } y = y' \text{ and } x \leq x'.
 \end{aligned}$$

Two grades $z_1, z_2 \in \mathbf{Z}^2$ satisfy $z_1 \leq z_2$ iff $z_1 \preceq_{\text{colex}} z_2$ and $z_1 \preceq_{\text{lex}} z_2$.

► **Definition 18.** For $b \in k^m$ and $r \in (\mathbf{Z}^2)^m$, the *lex pivot* of b with respect to r , $\text{l-piv}(b)$, is the smallest index i such that $b_i \neq 0$ and r_i takes its maximum value with respect to \preceq_{lex} . The *colex pivot*, $\text{c-piv}(b)$, is defined analogously. For $0 \in k^m$, we let $\text{l-piv}(0) = \text{c-piv}(0) = 0$.

► **Theorem 19.** Let M be a free \mathbf{Z}^2 -persistence module of finite rank with a fixed basis, let $V \subseteq \text{colim } M$ be a subspace, and let B be a matrix representing a generating set of V . Then $[V]_M$ is free, and Algorithm 1 calculates a graded matrix representing a basis of $[V]_M$.

For a tuple $r \in (\mathbf{Z}^2)^m$ and a matrix M with m rows, let $[M]_r$ be the graded matrix with $\text{rg}_i^{[M]_r} = r_i$ and $\text{cg}_j^{[M]_r} = \bigvee_{M_{ij} \neq 0} r_i$. Then $[M]_r$ has the least possible column grades for which $[M]_r$ represents a map of free modules.

Proof of Theorem 19. Let m_1, \dots, m_s be a basis of M and $r = (g(m_1), \dots, g(m_s))$. Without loss of generality, we assume that B represents a basis of V . The first for-loop in Algorithm 1 is a standard reduction scheme. In each iteration, the pivot index of one column decreases, so the loop terminates. When it does, all columns have distinct colex-pivots. During each iteration of the second for-loop, the lex-pivot of a column decreases. When it terminates, all columns have distinct lex-pivots. During the second loop, line (*) ensures that no column

■ **Algorithm 1** Computes a basis of $[V]_M$, where $M = \bigoplus_{i=1}^m F(r_i)$ and $V \subseteq \text{colim } M$.

Data: An $m \times n$ -matrix B representing a generating set of V , $r = (r_1, \dots, r_m) \in (\mathbf{Z}^2)^m$.

Result: A graded $m \times n$ -matrix whose nonzero columns represent a basis of $[V]_M$.

function Bireduce(B):

```

     $p \leftarrow 0 \in [m]^n$ 
    for  $j = 1, \dots, n$  do
        while  $i \leftarrow \text{c-piv}(B_j) \neq 0$  do
            if  $p_i = 0$  then  $p_i \leftarrow j$ ; break
             $B_j \leftarrow B_j + B_{p_i}$ 
     $p \leftarrow 0 \in [m]^n$ 
    for  $j' = 1, \dots, n$  do
         $j \leftarrow j'$ 
        while  $i \leftarrow \text{l-piv}(B_j) \neq 0$  do
            if  $p_i = 0$  then  $p_i \leftarrow j$ ; break
            if  $\text{c-piv}(B_j) < \text{c-piv}(B_{p_i})$  then swap  $p_i$  and  $j$ 
             $B_j \leftarrow B_j + B_{p_i}$ 
    return  $[B]_r$ 

```

is added to another column with a smaller colex-pivot. Since all columns have distinct colex-pivots after the first for-loop, the colex-pivots of the columns thus do not change during the second for-loop. Therefore, when the algorithm terminates, all columns of B have pairwise distinct lex- and colex-pivots.

Let $A = [B]_r$ for the state of B when the algorithm terminates. Then A represents a basis $\alpha_1, \dots, \alpha_t$ of a free submodule N of M , with $g(\alpha_j) = \text{cg}_j^A$. It remains to show that $N = [V]_M$. Since all column operations performed by Algorithm 1 are invertible, A represents a basis of V . Therefore, $\text{colim } N = V$, which implies $N \subseteq [V]_M$. Let $v \in [V]_M$. Then there are unique coefficients ξ_j such that $\eta_M(v) = \sum_{j=1}^t \xi_j \eta_M(\alpha_j)$. Since η_M is injective, $M_{g(v) \leq z}(v) = \sum_{j=1}^t \xi_j M_{g(\alpha_j) \leq z}(\alpha_j)$ for all $z \geq g(v) \vee \bigvee_{\xi_j \neq 0} \text{cg}_j^A$. Since all columns of A have distinct lex- and colex-pivots, v cannot have smaller grade than $\bigvee_{\xi_j \neq 0} \text{cg}_j^A$, so $v \in N$. This proves the claim. ◀

3.4 The free resolution of cohomology

Assume C^\bullet is a cochain complex of free modules such that $\text{colim } H^\bullet(C^\bullet) = 0$, and recall the commutative diagram (6). A matrix $[\delta^{d+1}]$ representing δ^{d+1} is a generating system for $\text{colim } Z^{d+1}(C^\bullet)$, and Lemma 17 states that $Z^{d+1}(C^\bullet) = [\text{colim } Z^{d+1}(C^\bullet)]_{C^{d+1}}$. Applying Algorithm 1 to $[\delta^{d+1}]$ thus yields a graded matrix $[\zeta^{d+1}]$ representing a basis of $Z^{d+1}(C^\bullet)$.

► **Lemma 20.** *If $0 \rightarrow F_2 \xrightarrow{f_2} F_1 \xrightarrow{f_1} F_0$ is a free resolution of a module of finite total dimension, then $(\nu F_0)^* = \ker(\nu f_2)^*$.*

Proof. The sequence $0 \rightarrow F_2 \xrightarrow{f_2} F_1 \xrightarrow{f_1} F_0$ is exact. By Theorem 10, the sequence $\nu F_2 \xrightarrow{\nu f_2} \nu F_1 \xrightarrow{\nu f_1} \nu F_0 \rightarrow 0$ and therefore also the dual sequence $0 \rightarrow (\nu F_0)^* \xrightarrow{(\nu f_1)^*} (\nu F_1)^* \xrightarrow{(\nu f_2)^*} (\nu F_2)^*$ are exact. ◀

Thus, if a matrix $[f_2]$ representing f_2 is known, then the matrix $[f_1]^T = [(\nu f_1)^*]$ can be computed by applying the LW-Algorithm (Algorithm 4 in the full version) to $[f_2]^T = [(\nu f_2)^*]$. In particular, if $H_d(C^\bullet)$ has finite total dimension, then so has $H^{d+2}(N^\bullet(K_*))$, so Lemma 20 can be applied to the free resolution (6) of $H^{d+2}(N^\bullet(K_*))$. This shows that $[p^{d+1}]^T$ can be computed by applying the LW-Algorithm to $[\zeta^{d+1}]^T$.

■ **Algorithm 2** Computes a minimal free resolution of $H^\bullet(C^\bullet)$ for a cochain complex C^\bullet of free \mathbf{Z}^2 -modules, using clearing.

Input: Graded matrices $[\delta^\bullet]$ representing C^\bullet .
Output: Pairs of graded matrices representing a free resolution of $H^d(C^\bullet)$ for $d = 0, 1, \dots$
 $q \leftarrow \emptyset$ ▷ pivots for clearing
for $d = 0, 1, \dots$ **do**
 for $j \in q$ **do** $[\delta^{d+1}]_j \leftarrow 0$ ▷ clearing
 $[i^{d+1}] \leftarrow \text{Bireduce}([\delta^{d+1}])$
 $n \leftarrow \# \text{columns of } [i^{d+1}]$
 $q \leftarrow \{\text{piv}[i^{d+1}]_1, \dots, \text{piv}[i^{d+1}]_n\}$
 $[i^{d+1}]^T, [p^{d+1}]^T \leftarrow \text{MGsWithKer}([i^{d+1}]^T)$ ▷ See Algorithm 5 in the full version
 yield $\text{MinimizeChainComplex}([i^{d+1}], [p^{d+1}])$ ▷ MFR of $H^{d+2}(\nu C^\bullet \langle \epsilon \rangle) \cong H^d(C^\bullet)$

► **Corollary 21.** Let C^\bullet be a cochain complex of free modules such that $\dim H^\bullet(C^\bullet)$ is finite. Then Algorithm 2 computes free resolutions of $H^d(C^\bullet)$.

► **Remark 22 (Clearing).** In general, $[\delta^{d+1}]$ is not injective. As in one-parameter persistent cohomology, the first loop in Algorithm 1 spends a significant amount of time on reducing the columns of $[\delta^{d+1}]$ that are eventually reduced to zero. The computation can be accelerated considerably by using the pivots of the reduced matrix $[\delta^d]$ to implement a clearing scheme before invoking Algorithm 1. This is implemented in Algorithm 2.

4 Experiments

We have implemented our cohomology algorithm in C++ [33]. We have also implemented the algorithm [23] used in `mpfree` [31], in order to vary the implementation details. Where applicable, the run time of our clone is similar to the one of `mpfree`. We have run our implementation to compute MFRs of the PH of various function-Rips bifiltrations.

4.1 Setup

All computations are done with coefficients in $k = \mathbf{F}_2$. Matrix columns are implemented as binary heaps [7]. Our code also implements an alternative representation of columns as dynamically allocated arrays. We have run our code on a MacBook Pro 2017 with a 2.3 GHz Dual-Core Intel Core i5 and 16GB RAM. The code is compiled using clang++ 15.0.7. Each instance of our program may run four threads in parallel.

The run time of the homology algorithm for minimal presentation computation [23] is dominated by chunk preprocessing and the LW-Algorithm. While it is standard to implement chunk preprocessing in an embarrassingly parallel way, no way is known to parallelize the LW-Algorithm. While the minimization step in Algorithm 2 is parallelized in our implementation, the bigraded reduction (Algorithm 1) is not, although we hypothesize it could be parallelized analogously to [36] or [6]. We found that the minimization is not a performance bottleneck of our algorithm, so one would expect similar performance on a single core.

Datasets. We have generated point clouds S by sampling n -spheres S^n , n -tori $S^1 \times \dots \times S^1$ and orthogonal groups $O(n)$. Additionally, we use some of the point clouds from [37]. To each point $p \in S$, we associate the value $\sum_{q \in S \setminus \{p\}} \exp(-\frac{d(p,q)^2}{2\sigma^2})$ for a manually chosen parameter σ . These values and a distance matrix are written to a file, from which the program generates the coboundary matrices of the associated full function-Rips bifiltration.

Chunk preprocessing. The LW-Algorithm works most efficiently if combined with chunk preprocessing [22, 23]; this is the approach implemented in `mpfree`. Chunk preprocessing (Algorithm 7 in the full version) applies a certain column operation scheme to the matrices representing the chain complex C_\bullet . As an alternative, we propose to manipulate $C_\bullet(K_*)$ by row operations. Equivalently, one can see this procedure as column operations on the matrices representing the cochain complex $\nu C^\bullet(K_*)$; hence, we refer to this approach as *cochain complex chunk preprocessing*; see Algorithm 8 in the full version.

Coning off. To ensure that $H^{d+2}(N^\bullet(K_*)) \cong H_d(K_*)^*$, Corollary 21 requires that C_\bullet has homology of finite total dimension. Therefore, our implementation offers the ability to cone off the complex as follows. Let C_\bullet be a chain complex of free \mathbf{Z}^2 -persistence modules. The assignment $C'_d: y \mapsto \text{colim}_{x \in \mathbf{Z}} (C_d)_{xy}$ defines a chain complex C'_\bullet of free \mathbf{Z} -persistence modules. We compute the barcode of $H_d(C'_\bullet)$ using the cohomological standard algorithm with clearing (Algorithm 3 in the full version). This can be used to implement a clearing mechanism in the homological standard algorithm, which we use to compute representatives for the homology classes in the barcode of $H_d(C'_\bullet)$. Let y_0 such that $y_0 \geq g_y(\sigma)$ for all σ , where $g(\sigma) = (g_x(\sigma), g_y(\sigma))$. Let $\hat{C}_\bullet = C_\bullet$. For every bar (b, d) of $H_d(C'_\bullet)$ of non-zero length represented by a q -cycle $c \in C'_q$ of grade $g(c) = b$, we add a basis element \hat{c} of grade $g(\hat{c}) = (b, y_0)$ to \hat{C}_{q+1} with $\partial_{q+1}(\hat{c}) = c$. If $d < \infty$, then c bounds a chain c' with $g(c') = d$, and we add a basis element \hat{c}' of grade $g(\hat{c}') = (d, y_0)$ to \hat{C}_{q+2} with $\partial_{q+2}(\hat{c}') = c' - \hat{c}$. The resulting chain complex \hat{C}_\bullet satisfies $H_\bullet(\hat{C}_\bullet)_{xy} = 0$ for $y \geq y_0$. If not stated otherwise, cohomology computation is done with this preprocessing applied to the density parameter.

Sparsification. We observe that the second for-loop in Algorithm 1 runs considerably longer than the first. The loop also increases the matrix density, which many incur a high cost on the subsequent steps in Algorithm 2. For an interpretation, see Remark 24 in the full version. As a remedy, we have added a step that decreases the sparsity of the matrix using row operations that are compatible with the column sparse matrix implementation. Specifically, if a row contains only a single entry, any row addition from this row to another affects only a single entry. Therefore, an entry in a row with grade g can be eliminated directly if there is a row with grade $g' \geq g$ containing only a single entry in the same column; see Algorithm 10 in the full version. All cohomology computation run times are reported with this sparsification scheme applied.

4.2 Results

An overview of the results is given in Table 1. We report only the time needed to compute the MFR and, if applicable, to apply the chunk preprocessing. We do not report the time necessary to set up the (co)boundary matrices. In all cases with $d \geq 2$, computing $H^{d+2}(N^\bullet)$ (without chunk preprocessing) was faster than computing H_d with chunk preprocessing. Our cohomology approach does not benefit from chunk preprocessing. The speedup of the cohomology approach increases with dimension. For two instances, computation of $H_d(K_*)$ did not terminate within five minutes, while computing $H^{d+2}(N^\bullet(K_*))$ was no problem. We also observe that the cohomology algorithm uses less memory for almost all instances with $d \geq 2$.

Matrix representations. The efficiency of the LW-Algorithm and of chunk preprocessing does not vary very much depending on the matrix implementation; see Table 1 and Table 2 in the full version. In contrast, our cohomology algorithm runs faster in the implementation

■ **Table 1** Run times (in milliseconds) comparing our implementation of [35, 23] (including chunk preprocessing) and our cohomology algorithm, applied a density-Rips filtration on 300 vertices ($d = 1$), 100 vertices ($d = 2$) and 60 vertices ($d = 3$). RSS is peak resident memory as measured by `time`. Speed up is the run time of the homology computation (including chunk preprocessing), divided by the run time of the cohomology algorithm. The program has been killed after exceeding five minutes of run time.

d	sample	chunk	H_d	sum	RSS	H^{d+2}	RSS	speedup
1	c. elegans	5,457	40,841	46,298	6,423,600	119,444	6,526,976	0.39
	2-torus	11,480	19,875	31,355	6,620,404	5,032	2,827,912	6.23
	4-torus	6,342	28,627	34,969	5,916,816	50,607	3,384,472	0.69
	dragon	9,721	18,489	28,210	5,774,492	5,064	2,829,620	5.57
	2-sphere	8,657	29,180	37,837	6,421,312	25,021	4,098,268	1.51
	4-sphere	7,699	33,619	41,318	6,642,260	47,355	7,154,632	0.87
	$O(3)$	7,023	33,874	40,897	6,315,708	42,702	3,816,124	0.96
2	c. elegans	28,583	7,484	36,067	4,428,440	5,655	2,371,324	6.38
	2-torus	39,630	2,191	41,821	3,216,372	5,054	2,334,712	8.27
	4-torus	33,788	19,875	53,663	5,538,232	5,969	2,425,136	8.99
	dragon	19,023	2,379	21,402	2,557,124	5,188	2,367,488	4.13
	2-sphere	32,611	12,416	45,027	5,099,604	6,417	2,426,924	7.02
	4-sphere	29,272	25,357	54,629	6,039,576	8,637	2,505,664	6.32
	$O(3)$	31,780	29,123	60,903	6,654,692	6,796	2,445,996	8.96
3	c. elegans	38,349	2,393	40,742	3,515,708	8,984	4,227,820	4.53
	2-torus	>300,000	–	–	7,141,648	11,725	5,072,192	>25.59
	4-torus	>300,000	–	–	10,843,356	9,930	4,358,580	>30.21
	dragon	67,463	2,334	69,797	6,666,732	9,782	4,900,800	7.14
	2-sphere	59,385	3,110	62,495	4,280,112	9,051	4,185,036	6.90
	4-sphere	92,365	8,577	100,942	5,526,344	8,818	4,197,636	11.45
	$O(3)$	204,263	25,284	229,547	7,966,600	10,851	4,365,864	21.15

with binary heaps. We observe that the vector based implementations generally use less memory than the heap based ones. This happens because, in contrast to vectors, heaps may contain multiple entries for the same row index.

Cochain chunk preprocessing. For the homology computation, the cochain complex chunk preprocessing described above often is more efficient than chunk preprocessing if combined with the heap implementation of matrix columns, see Table 3 in the full version. This is true in particular for higher homology dimensions. If combined with vector-based matrices, cochain chunk preprocessing is less efficient than conventional chunk preprocessing in almost all cases, and does not terminate at all within five minutes.

References

- 1 Manu Aggarwal and Vipul Periwal. Dory: Overcoming Barriers to Computing Persistent Homology, March 2021. [arXiv:2103.05608](https://arxiv.org/abs/2103.05608).
- 2 Ángel Javier Alonso, Michael Kerber, and Siddharth Pritam. Filtration-domination in bifiltered graphs. In *2023 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 27–38, 2023. doi:10.1137/1.9781611977561.ch3.

- 3 Ibrahim Assem, Daniel Simson, and Andrzej Skowronski. *Elements of the Representation Theory of Associative Algebras*, volume 1 of *London Mathematical Society Student Texts*. Cambridge University Press, Cambridge, 2006.
- 4 Ulrich Bauer. Ripser: efficient computation of Vietoris–Rips persistence barcodes. *J. Appl. Comput. Topol.*, 5(3):391–423, 2021. doi:10.1007/s41468-021-00071-5.
- 5 Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and Compress: Computing Persistent Homology in Chunks. In Peer-Timo Bremer, Ingrid Hotz, Valerio Pascucci, and Ronald Peikert, editors, *Topological Methods in Data Analysis and Visualization III*, pages 103–117. Springer International Publishing, Cham, 2014. doi:10.1007/978-3-319-04099-8_7.
- 6 Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Distributed Computation of Persistent Homology. In Catherine C. McGeoch and Ulrich Meyer, editors, *2014 Proceedings of the Sixteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 31–38. Society for Industrial and Applied Mathematics, Philadelphia, PA, May 2014. doi:10.1137/1.9781611973198.4.
- 7 Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat—persistent homology algorithms toolbox. *J. Symbolic Comput.*, 78:76–90, 2017. doi:10.1016/j.jsc.2016.03.008.
- 8 Ulrich Bauer, Fabian Lenzen, and Michael Lesnick. Efficient two-parameter persistence computation via cohomology, March 2023. arXiv:2303.11193.
- 9 Ulrich Bauer and Michael Lesnick. Induced matchings and the algebraic stability of persistence barcodes. *Journal of Computational Geometry*, 6(2):162–191, March 2015. doi:10.20382/jocg.v6i2a9.
- 10 Ulrich Bauer and Maximilian Schmahl. Lifespan Functors and Natural Dualities in Persistent Homology, October 2021. To appear in *Homology, Homotopy and Applications*. arXiv:2012.12881.
- 11 Katherine Benjamin, Aneesha Bhandari, Zhouchun Shang, Yanan Xing, Yanru An, Nannan Zhang, Yong Hou, Ulrike Tillmann, Katherine R Bull, and Heather A Harrington. Multiscale topology classifies and quantifies cell types in subcellular spatial transcriptomics. *arXiv preprint arXiv:2212.06505*, 2022.
- 12 Andrew J. Blumberg and Michael Lesnick. Stability of 2-Parameter Persistent Homology. *Foundations of Computational Mathematics*, October 2022. doi:10.1007/s10208-022-09576-6.
- 13 Magnus Bakke Botnan and Michael Lesnick. An Introduction to Multiparameter Persistence, March 2022. arXiv:2203.14289.
- 14 Gunnar Carlsson and Afra Zomorodian. The Theory of Multidimensional Persistence. *Discrete & Computational Geometry*, 42(1):71–93, July 2009. doi:10.1007/s00454-009-9176-0.
- 15 Chao Chen and Michael Kerber. Persistent Homology Computation with a Twist. In *27th European Workshop on Computational Geometry*, 2011. URL: <https://eurocg11.inf.ethz.ch/abstracts/22.pdf>.
- 16 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37:103–120, January 2007. doi:10.1007/s00454-006-1276-5.
- 17 René Corbet, Michael Kerber, Michael Lesnick, and Georg Osang. Computing the multicover bifiltration. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2021.27.
- 18 Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27(12):124003, December 2011. doi:10.1088/0266-5611/27/12/124003.
- 19 Edelsbrunner, Letscher, and Zomorodian. Topological Persistence and Simplification. *Discrete & Computational Geometry*, 28(4):511–533, November 2002. doi:10.1007/s00454-002-2885-2.

- 20 Herbert Edelsbrunner and John Harer. Persistent homology—a survey. In Jacob E. Goodman, János Pach, and Richard Pollack, editors, *Contemporary Mathematics*, volume 453, pages 257–282. American Mathematical Society, Providence, Rhode Island, 2008. doi:10.1090/conm/453/08802.
- 21 David Eisenbud. *Commutative Algebra*, volume 150 of *Graduate Texts in Mathematics*. Springer New York, New York, NY, 1995. doi:10.1007/978-1-4612-5350-1.
- 22 Ulderico Fugacci and Michael Kerber. Chunk reduction for multi-parameter persistent homology. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry (SoCG 2019)*, volume 129 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 37:1–37:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.SoCG.2019.37.
- 23 Ulderico Fugacci, Michael Kerber, and Alexander Rolle. Compression for 2-parameter persistent homology. *Computational Geometry*, 109:101940, February 2023. doi:10.1016/j.comgeo.2022.101940.
- 24 Victor Ginzburg. Calabi–Yau algebras, 2007. arXiv:math/0612139.
- 25 Barbara Giunti and Jānis Lazovskis. TDA-Applications (An online database of papers on applications of TDA outside of math), 2021. URL: <https://www.zotero.org/groups/2425412/tda-applications>.
- 26 Gregory Henselman and Robert Ghrist. Matroid Filtrations and Computational Persistent Homology, October 2017. arXiv:1606.00199.
- 27 Gregory Henselman-Petrusek. Eirene, 2021. URL: <https://github.com/Eetion/Eirene.jl>.
- 28 Michael Höppner and Helmut Lenzing. Projective diagrams over partially ordered sets are free. *Journal of Pure and Applied Algebra*, 20(1):7–12, January 1981. doi:10.1016/0022-4049(81)90045-1.
- 29 Bernhard Keller. Calabi–Yau triangulated categories. In Andrzej Skowroński, editor, *Trends in Representation Theory of Algebras and Related Topics*, EMS Series of Congress Reports, pages 467–489. EMS Press, first edition, September 2008. doi:10.4171/062-1/11.
- 30 Michael Kerber. Multi-parameter persistent homology is practical. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020. URL: <https://openreview.net/pdf?id=TDU6UycGYxR>.
- 31 Michael Kerber. mpfree, 2021. URL: <https://bitbucket.org/mkerber/mpfree>.
- 32 Michael Kerber and Alexander Rolle. Fast Minimal Presentations of Bi-graded Persistence Modules. In Martin Farach-Colton and Sabine Storandt, editors, *2021 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, Proceedings, pages 207–220. Society for Industrial and Applied Mathematics, January 2021. doi:10.1137/1.9781611976472.16.
- 33 Fabian Lenzen. 2-parameter persistent cohomology, 2023. URL: <https://gitlab.com/flenzen/2-parameter-persistent-cohomology>.
- 34 Michael Lesnick and Matthew Wright. Interactive Visualization of 2-D Persistence Modules, December 2015. arXiv:1512.00180.
- 35 Michael Lesnick and Matthew Wright. Computing Minimal Presentations and Bigraded Betti Numbers of 2-Parameter Persistent Homology. *SIAM Journal on Applied Algebra and Geometry*, 6(2):267–298, June 2022. doi:10.1137/20M1388425.
- 36 Dmitriy Morozov and Arnur Nigmatov. Towards Lockfree Persistent Homology. In *Proceedings of the 32nd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 555–557, Virtual Event USA, July 2020. ACM. doi:10.1145/3350755.3400244.
- 37 Nina Otter, Mason A Porter, Ulrike Tillmann, Peter Grindrod, and Heather A Harrington. A roadmap for the computation of persistent homology. *EPJ Data Science*, 6(1):17, December 2017. doi:10.1140/epjds/s13688-017-0109-5.
- 38 Irena Peeva. *Graded Syzygies*. Springer London, London, 2011. doi:10.1007/978-0-85729-177-6.

- 39 Julián Burella Pérez, Sydney Hauke, Umberto Lupo, Matteo Caorsi, and Alberto Dassatti. *giotto-ph: A Python Library for High-Performance Computation of Persistent Homology of Vietoris-Rips Filtrations*, August 2021. [arXiv:2107.05412](https://arxiv.org/abs/2107.05412).
- 40 Daniel Quillen. Projective modules over polynomial rings. *Inventiones Mathematicae*, 36(1):167–171, December 1976. doi:10.1007/BF01390008.
- 41 Alexander Rolle. Multi-parameter hierarchical clustering and beyond. In *NeurIPS 2020 Workshop on Topological Data Analysis and Beyond*, 2020. URL: <https://openreview.net/pdf?id=g0-tBxQTPRy>.
- 42 Sara Scaramuccia, Federico Iuricich, Leila De Floriani, and Claudia Landi. Computing multiparameter persistent homology through a discrete Morse-based approach. *Computational Geometry*, 89:101623, August 2020. doi:10.1016/j.comgeo.2020.101623.
- 43 Donald R. Sheehy. A multicover nerve for geometric inference. In *Proceedings of the 24th Canadian Conference on Computational Geometry, CCCG 2012, Charlottetown, Prince Edward Island, Canada, August 8-10, 2012*, pages 309–314, 2012. URL: <http://2012.cccg.ca/papers/paper52.pdf>.
- 44 Andrei Suslin. Projective modules over polynomial rings are free. *Soviet Mathematics*, 17(4):1160–1164, 1976.
- 45 The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.6.0 edition, 2022. URL: <https://gudhi.inria.fr/doc/3.6.0/>.
- 46 Oliver Vipond, Joshua A Bull, Philip S Macklin, Ulrike Tillmann, Christopher W Pugh, Helen M Byrne, and Heather A Harrington. Multiparameter persistent homology landscapes identify immune cell spatial patterns in tumors. *Proceedings of the National Academy of Sciences*, 118(41):e2102166118, 2021.
- 47 Cary Webb. Decomposition of Graded Modules. *Proceedings of the American Mathematical Society*, 94(4):565–571, 1985. doi:10.2307/2044864.
- 48 Charles A. Weibel. *An Introduction to Homological Algebra*. Number 38 in Cambridge Studies in Advanced Mathematics. Cambridge Univ. Press, Cambridge, reprint. 1997, transf. to digital print edition, 2003.
- 49 Simon Zhang, Mengbai Xiao, and Hao Wang. GPU-Accelerated Computation of Vietoris-Rips Persistence Barcodes. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 70:1–70:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2020.70.
- 50 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33:249–274, February 2005. doi:10.1007/s00454-004-1146-y.

The Complexity of Geodesic Spanners

Sarita de Berg 

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Marc van Kreveld 

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Frank Staals 

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

A *geometric t -spanner* for a set S of n point sites is an edge-weighted graph for which the (weighted) distance between any two sites $p, q \in S$ is at most t times the original distance between p and q . We study geometric t -spanners for point sets in a constrained two-dimensional environment P . In such cases, the edges of the spanner may have non-constant complexity. Hence, we introduce a novel spanner property: the spanner *complexity*, that is, the total complexity of all edges in the spanner. Let S be a set of n point sites in a simple polygon P with m vertices. We present an algorithm to construct, for any constant $\varepsilon > 0$ and fixed integer $k \geq 1$, a $(2k + \varepsilon)$ -spanner with complexity $O(mn^{1/k} + n \log^2 n)$ in $O(n \log^2 n + m \log n + K)$ time, where K denotes the output complexity. When we consider sites in a polygonal domain P with holes, we can construct such a $(2k + \varepsilon)$ -spanner of similar complexity in $O(n^2 \log m + nm \log m + K)$ time. Additionally, for any constant $\varepsilon \in (0, 1)$ and integer constant $t \geq 2$, we show a lower bound for the complexity of any $(t - \varepsilon)$ -spanner of $\Omega(mn^{1/(t-1)} + n)$.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases spanner, simple polygon, polygonal domain, geodesic distance, complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.16

Related Version *Full Version*: <https://arxiv.org/abs/2303.02997>

1 Introduction

In the design of networks on a set of nodes, we often consider two criteria: few connections between the nodes, and small distances. Spanners are geometric networks on point sites that replace the small distance criterion by a small detour criterion. Formally, a *geometric t -spanner* for a set S of n point sites is an edge-weighted graph $\mathcal{G} = (S, E)$ for which the (weighted) distance $d_{\mathcal{G}}(p, q)$ between any two sites $p, q \in S$ is at most $t \cdot d(p, q)$, where $d(p, q)$ denotes the distance between p and q in the distance metric we consider [29]. The smallest value t for which a graph \mathcal{G} is a t -spanner is called the *spanning ratio* of \mathcal{G} . The number of edges in the spanner is called the *size* of the spanner.

In the real world, spanners are often constructed in some sort of environment. For example, we might want to connect cities by a railway network, where the tracks should avoid obstacles such as mountains or lakes. One way to model such an environment is by a polygonal domain. In this paper, we study the case where the sites lie in a polygonal domain P with m vertices and h holes, and we measure the distance between two points p, q by their *geodesic distance*: the length of the shortest path between p and q fully contained within P . An example of such a spanner is provided in Figure 1.

The spanning ratio and the size of spanners are not the only properties of interest. Many different properties have been studied, such as total weight (or lightness), maximum degree, (hop) diameter, and fault-tolerance [4, 9, 11, 14, 20, 27, 28, 30]. When we consider distance metrics for which the edges in the spanner no longer have constant complexity, another



© Sarita de Berg, Marc van Kreveld, and Frank Staals;
licensed under Creative Commons License CC-BY 4.0

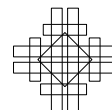
39th International Symposium on Computational Geometry (SoCG 2023).

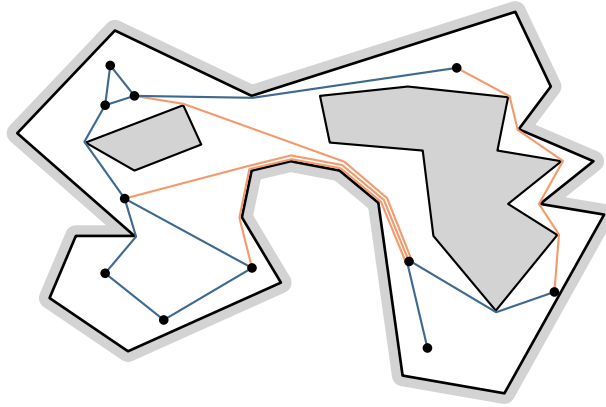
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 16; pp. 16:1–16:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** A spanner on a set of point sites in a polygonal domain. Because of the orange edges, the spanner has a relatively high complexity.

interesting property of spanners arises: the spanner *complexity*, i.e. the total complexity of all edges in the spanner. In our railway example, this corresponds to the total number of bends in the tracks. A spanner with a low number of bends may be desired, as trains can drive faster on straight tracks, and it makes construction cheaper. In this paper, we study this novel property for point sites in a polygonal domain, where the complexity of an edge is simply the number of line segments in the path. In this setting, a single edge may have complexity $\Theta(m)$. Naively, a spanner of size E could thus have complexity $\Theta(mE)$. Our goal is to compute an $O(1)$ -spanner of size $O(n \text{ polylog } n)$ with small complexity, preferably near linear in both n and m .

When studying spanning trees of points, two variants exist: with or without Steiner points. The same is true for spanners, where Steiner points can be used to obtain lighter and sparser spanners [6, 27]. In this paper we focus on the variant where Steiner points are *not* allowed, leaving the other variant to future research.

Related work. For the Euclidean distance in \mathbb{R}^d , and any fixed $\varepsilon > 0$, there is a $(1 + \varepsilon)$ -spanner of size $O(n/\varepsilon^{d-1})$ [30]. For the more general case of metric spaces of bounded doubling dimension we can also construct a $(1 + \varepsilon)$ -spanner of size $O(n/\varepsilon^{O(d)})$ [13, 21, 24]. These results do not apply when the sites lie in a polygon, and we measure their distances using the geodesic distance. Abam et al. [1] show there is a set of n sites in a simple polygon P for which any geodesic $(2 - \varepsilon)$ -spanner has $\Omega(n^2)$ edges. They also construct a geodesic $(\sqrt{10} + \varepsilon)$ -spanner of size $O(n \log^2 n)$ for sites in a simple polygon, and a geodesic $(5 + \varepsilon)$ -spanner of size $O(n\sqrt{h} \log^2 n)$ for sites in a polygonal domain. Recently, Abam et al. [3] showed that a geodesic $(2 + \varepsilon)$ -spanner with $O(n \log n)$ edges exists for points on a polyhedral terrain, thereby almost closing the gap between the upper and lower bound on the spanning ratio. However, they show only the existence of such a spanner, and leave constructing one open. Moreover, all of these spanners can have high, $\Omega(nm)$, complexity.

Abam et al. [3] make use of spanners on an *additively weighted* point set in \mathbb{R}^d . In this setting, the distance between two sites p, q is $w(p) + |pq| + w(q)$ for $p \neq q$, where $w(p)$ is the non-negative weight of a site $p \in S$ and $|pq|$ denotes the Euclidean distance, and 0 for $p = q$. Such additively weighted spanners were studied before by Abam et al. [2], who obtain an $O(5 + \varepsilon)$ -spanner of linear size, and an $O(2 + \varepsilon)$ -spanner of size $O(n \log n)$. They also provide a lower bound of $\Omega(n^2)$ on the size of any $(2 - \varepsilon)$ -spanner. Abam et al. [3] improve these results and obtain a nearly optimal additively weighted $(2 + \varepsilon)$ -spanner of size $O(n)$.

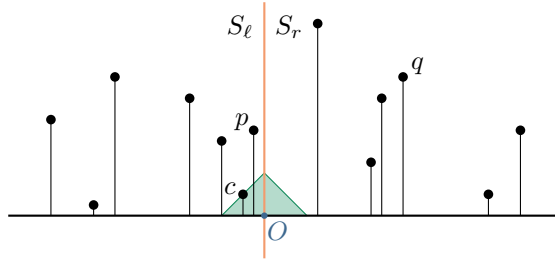
The other key ingredient for the geodesic $(2 + \varepsilon)$ -spanner of Abam et al. [3] is a *balanced shortest-path separator*. Such a separator consists of either a single shortest path between two points on the boundary of the terrain, or three shortest paths that form a *shortest-path triangle*. This separator partitions the terrain into two subterrains, and we call it balanced when each of these terrains contains roughly half of the sites in S . In their constructive proof for the existence of such a balanced separator, they assume that the three shortest paths in a shortest-path triangle are disjoint, except for their mutual endpoints. However, during their construction it can actually happen that these paths are *not* disjoint. When this happens, it is unclear exactly how to proceed. Just like for the $(2 + \varepsilon)$ -spanner, the computation of a balanced separator is left for future research. We show how to get rid of the assumption that the shortest paths are disjoint, and thereby confirm the result claimed by Abam et al. [3].

Next to spanners on the complete Euclidean geometric graph, spanners under line segment constraints were studied [8, 10, 12, 16, 17]. Here, the goal is to construct a spanner on the *visibility graph* of S with respect to a set of line segments between sites in S . If the segments form a polygonal domain P , this setting is similar to ours, except that *all* vertices of P are included as sites in S , and thus the complexity of each edge is constant.

Our results. We first consider the simple setting where the sites lie in a simple polygon, i.e. a polygonal domain without holes. We show that in this setting any $(3 - \varepsilon)$ -spanner may have complexity $\Omega(nm)$, thus implying that the $(2 + \varepsilon)$ -spanner of Abam et al. [3] may also have complexity $\Omega(nm)$, despite having $O(n \log n)$ edges.

To improve this complexity, we first introduce a simple 2-spanner with $O(n \log n)$ edges for an additively weighted point set in a 1-dimensional Euclidean space; see Section 2. In Section 3, we use this result to obtain a geodesic $2\sqrt{2}$ -spanner with $O(n \log^2 n)$ edges for a point set in a simple polygon. We recursively split the polygon by a chord λ such that each subpolygon contains roughly half of the sites, and build a 1-dimensional spanner on the sites projected to λ . We then extend this spanner into one that also has bounded complexity. For any constant $\varepsilon > 0$ and fixed integer $k \geq 1$, we obtain a $(2k + \varepsilon)$ -spanner with complexity $O(mn^{1/k} + n \log^2 n)$. Furthermore, we provide an algorithm to compute such a spanner that runs in $O(n \log^2 n + m \log n + K)$ time, where K denotes the output complexity. When we output each edge explicitly, K is equal to the spanner complexity. However, as each edge is a shortest path, we can also output an edge implicitly by only stating the two sites it connects. In this case K is equal to the size of the spanner. Additionally, for any constant $\varepsilon \in (0, 1)$ and integer constant $t \geq 2$, we show a lower bound for the complexity of any $(t - \varepsilon)$ -spanner of $\Omega(mn^{1/(t-1)} + n)$. Therefore, the $2k + \varepsilon$ spanning ratio of our $O(mn^{1/k} + n \log^2 n)$ complexity spanners is about a factor two off optimal.

In Section 4, we extend our results for a simple polygon to a polygonal domain. There are two significant difficulties in this transition: (i) we can no longer partition the polygon by a line segment such that each subpolygon contains roughly half of the sites, and (ii) the shortest path between two sites p, q may not be homotopic to the path from p to q via another site c . We solve problem (i) by using a shortest-path separator similar to Abam et al. [3]. To apply the shortest-path separator in a polygonal domain, we need new additional ideas. In particular, we allow one additional type of separator in our version of a shortest-path separator: two shortest paths from a point in P to the boundary of a single hole. We show that this way there indeed always exists such a separator in a polygonal domain, and provide an $O(n^2 \log m + nm \log m)$ time algorithm to compute one. To overcome problem (ii), we allow an edge (p, q) to be any path from p to q . In networks, the connections between two nodes are often not necessarily optimal paths, the only requirement being that the distance between



■ **Figure 2** Construction of the additively weighted 1-dimensional spanner. The green triangle represents all points that are at distance at most $d_w(c, O)$ from O .

two hubs does not become too large. Thus allowing other paths between two sites seems a reasonable relaxation. This way, we obtain in a geodesic $(2k + \epsilon)$ -spanner of size $O(n \log^2 n)$ and complexity $O(mn^{1/k} + n \log^2 n)$ that can be computed in $O(n^2 \log m + nm \log m + K)$ time. Because our edges always consist of at most three shortest paths, we can again output the edges implicitly in $O(n \log^2 n)$ time. We also provide an alternative $(2k + \epsilon)$ -spanner of size $O(\sqrt{hn} \log^2 n)$ and complexity $O(\sqrt{h}(mn^{1/k} + n \log^2 n))$ that can be constructed more efficiently, i.e., in $O(\sqrt{hn} \log^2 n + m \log m + K)$ time.

Throughout the paper, we make the general position assumption that all vertices of P and sites in S have distinct x - and y -coordinates. Symbolic perturbation, in particular a shear transformation, can be used to remove this assumption [18].

2 A 1-dimensional additively weighted 2-spanner

We consider how to compute an additively weighted spanner \mathcal{G} in 1-dimensional Euclidean space, where each site $p \in S$ has a non-negative weight $w(p)$. The distance $d_w(p, q)$ between two sites $p, q \in S$ is given by $d_w(p, q) = w(p) + |pq| + w(q)$, where $|pq|$ denotes the Euclidean distance. Without loss of generality, we can map \mathbb{R}^1 to the x -axis, and the weights to the y -axis, see Figure 2. This allows us to speak of the sites left (or right) of some site p .

To construct a spanner \mathcal{G} , we first partition the sites into two sets S_ℓ and S_r of roughly equal size by a point O with $w(O) = 0$. The set S_ℓ contains all sites left of O , and S_r all sites right of O . Sites that lie on the vertical line through O are not included in either of the sets. We then find a site $c \in S$ for which $d_w(c, O)$ is minimal. For all $p \in S$, $p \neq c$, we add the edge (p, c) to \mathcal{G} . Finally, we handle the sets S_ℓ and S_r , excluding the site c , recursively.

► **Lemma 1.** *The graph \mathcal{G} is a 2-spanner of size $O(n \log n)$ and can be constructed in $O(n \log n)$ time.*

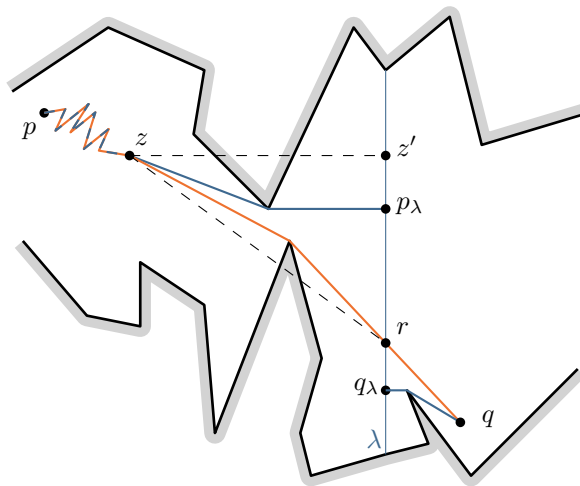
Proof sketch. For two sites p, q , consider the chosen center c at the level where p and q lie on different sides of O . As the shortest path from p to q passes via O , we have

$$d_{\mathcal{G}}(p, q) \leq d_w(p, O) + 2d_w(c, O) + d_w(q, O) \leq 2d_w(p, O) + 2d_w(q, O) = 2d_w(p, q). \quad \blacktriangleleft$$

3 Spanners in a simple polygon

3.1 A simple geodesic spanner

Just like Abam et al. [3], we use our 1-dimensional spanner to construct a geodesic spanner. We are more interested in the simplicity of the spanner than its spanning ratio, as we base our low complexity spanners, to be discussed in Section 3.2, on this simple geodesic spanner. Let



■ **Figure 3** The shortest path $\pi(p, q)$ crosses λ at r . The difference in length between the direct path from z to r and the path through p_λ can be bounded by considering the triangle $\mathcal{T} = (z, z', r)$.

P be a simple polygon, and let ∂P denote the polygon boundary. We denote by $d(p, q)$ the geodesic distance between $p, q \in P$, and by $\pi(p, q)$ the shortest (geodesic) path from p to q . We analyze the simple construction using any 1-dimensional additively weighted t -spanner of size $O(n \log n)$. We show that restricting the domain to a simple polygon improves the spanning ratio from $3t$ to $\sqrt{2}t$. The construction can be refined to achieve a spanning ratio of $t + \varepsilon$, see Section 3.2.2 and Lemma 3 of the full version [19].

As in [1] and [3], we first partition P into two subpolygons P_ℓ and P_r by a line segment λ , such that each subpolygon contains at most two thirds of the sites in S [7]. We assume, without loss of generality, that λ is a vertical line segment and P_ℓ is left of λ . Let S_ℓ be the sites in the closed region P_ℓ , and $S_r := S \setminus S_\ell$. For each site $p \in S$, we then find the point p_λ on λ closest to p . Note that this point is unique, because the shortest path to a line segment is unique in a simple polygon. We denote by S_λ the set of all projected sites. As λ is a line segment, we can define a weighted 1-dimensional Euclidean space on λ , where $w(p_\lambda) := d(p, p_\lambda)$ for each $p_\lambda \in S_\lambda$. We compute a t -spanner $\mathcal{G}_\lambda = (S_\lambda, E_\lambda)$ for this set. For each pair $(p_\lambda, q_\lambda) \in E_\lambda$, we add the edge (p, q) , which is $\pi(p, q)$, to our spanner \mathcal{G} . Finally, we recursively compute spanners for S_ℓ and S_r , and add their edges to \mathcal{G} as well.

► **Lemma 2.** *The graph \mathcal{G} is a geodesic $\sqrt{2}t$ -spanner of size $O(n \log^2 n)$.*

Proof. As \mathcal{G}_λ has $O(n \log n)$ edges (Lemma 1) that directly correspond to edges in \mathcal{G} , and the recursion has $O(\log n)$ levels, we have $O(n \log^2 n)$ edges in total. Let p, q be two sites in S . If both are in S_ℓ (or S_r), then there is a path of length $\sqrt{2}td(p, q)$ by induction. So, we assume w.l.o.g. that $p \in S_\ell$ and $q \in S_r$. Let r be the intersection point of $\pi(p, q)$ and λ . Observe that p_λ and q_λ must be on opposite sides of r , otherwise r cannot be on the shortest path. We assume, without loss of generality, that p_λ is above r and q_λ below r . Because \mathcal{G}_λ is a t -spanner, we know that there is a weighted path from p_λ to q_λ of length at most $td_w(p_\lambda, q_\lambda)$. As $w(p_\lambda) = d(p, p_\lambda)$, this directly corresponds to a path in the polygon. So,

$$d_{\mathcal{G}}(p, q) \leq d_{\mathcal{G}_\lambda}(p_\lambda, q_\lambda) \leq td_w(p_\lambda, q_\lambda) = t(d(p, p_\lambda) + |p_\lambda r| + |r q_\lambda| + d(q_\lambda, q)). \quad (1)$$

Let z be the point where the shortest paths from p to p_λ and r separate. See Figure 3 for an illustration. Consider the right triangle $\mathcal{T} = (z, z', r)$, where z' is the intersection point

of the line perpendicular to λ through z and the line containing λ . Note that z' does not necessarily lie within P . For this triangle we have that

$$|zr| \geq \frac{\sqrt{2}}{2}(|zz'| + |z'r|). \quad (2)$$

Next, we show that the path from z to p_λ is a y -monotone convex polygonal chain ending at or below z' . Consider the vertical ray through z upwards to the polygon boundary. We call the part of ∂P between where the ray hits ∂P and λ the *top* part of ∂P . Similarly, for a downwards ray, we define the *bottom* part of ∂P . There are no vertices on $\pi(z, p_\lambda)$ from the bottom part of ∂P , because such a vertex would then also occur on the shortest path to r . This is in contradiction with the definition of z . If z sees z' , then $p_\lambda = z'$, otherwise the chain must bend at one or more vertices of the top part of ∂P , and thus lie below z' . It follows that $\pi(z, p_\lambda)$ is contained within \mathcal{T} . Similarly, we conclude that $\pi(z, r)$ is contained within \mathcal{T} . Additionally, this gives us that $d(z, p_\lambda) \leq |zz'| + |z'p_\lambda|$, and $d(z, r) \geq |zr|$. Together with Equation (2) this yields $d(z, p_\lambda) + |p_\lambda r| \leq |zz'| + |z'r| \leq \sqrt{2}|zr| \leq \sqrt{2}d(z, r)$. And thus

$$d(p, p_\lambda) + |p_\lambda r| = d(p, z) + d(z, p_\lambda) + |p_\lambda r| \leq d(p, z) + \sqrt{2}d(z, r) \leq \sqrt{2}d(p, r).$$

Symmetrically, we find for q that $d(q, q_\lambda) + |q_\lambda r| \leq \sqrt{2}d(q, r)$. From this, together with Equation (1), we conclude that $d_{\mathcal{G}}(p, q) \leq t(\sqrt{2}d(p, r) + \sqrt{2}d(r, q)) = \sqrt{2}td(p, q)$. \blacktriangleleft

Applying Lemma 2 to the spanner of Section 2 yields a $2\sqrt{2}$ -spanner of size $O(n \log^2 n)$.

3.2 Low complexity geodesic spanners

In general, a geodesic spanner $\mathcal{G} = (S, E)$ in a simple polygon with m vertices may have complexity $O(m|E|)$. It is easy to see that the $2\sqrt{2}$ -spanner of Section 3.1 can have complexity $\Omega(nm)$, just like the spanners in [3]. As one of the sites, c , is connected to all other sites, the polygon in Figure 4 provides this lower bound. The construction in Figure 4 even shows that the same lower bound holds for the complexity of any $(3 - \epsilon)$ -spanner. Additionally, the following theorem implies a trade-off between the spanning ratio and the spanner complexity.

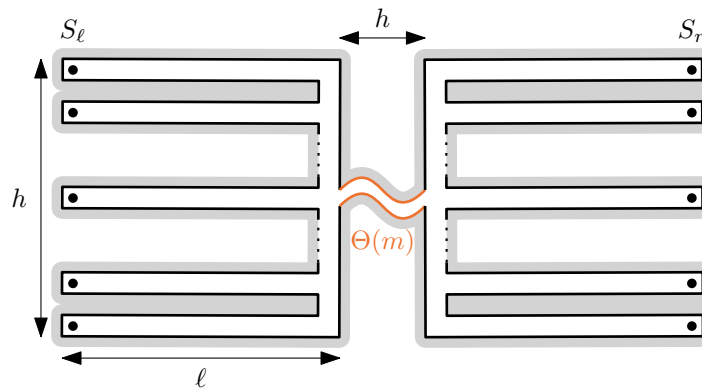
► **Theorem 3.** *For any constant $\epsilon \in (0, 1)$ and integer constant $t \geq 2$, there exists a set of n point sites in a simple polygon P with $m = \Omega(n)$ vertices for which any geodesic $(t - \epsilon)$ -spanner has complexity $\Omega(mn^{1/(t-1)})$.*

The proofs of these lower bounds are in the full version [19]. Next, we present a spanner that almost matches this bound. We first present a $4\sqrt{2}$ -spanner of bounded complexity, and then generalize the approach to obtain a $(2k + \epsilon)$ -spanner of complexity $O(mn^{1/k} + n \log^2 n)$, for any integer $k \geq 2$.

3.2.1 A $4\sqrt{2}$ -spanner of complexity $O(m\sqrt{n} + n \log^2 n)$

To improve the complexity of the geodesic spanner, we adapt our construction for the additively weighted spanner \mathcal{G}_λ as follows. After finding the site $c_\lambda \in S_\lambda$ for which $d_w(c_\lambda, O)$ is minimal, we do not add all edges (p_λ, c_λ) , $p_\lambda \in S_\lambda$, to \mathcal{G}_λ . Instead, we form groups of sites whose original sites (before projection) are “close” to each other in P . For each group S_i , we add all edges $(p_\lambda, c_{i,\lambda})$, $p_\lambda \in S_i$, to \mathcal{G}_λ , where $c_{i,\lambda}$ is the site in S_i for which $d_w(c_{i,\lambda}, O)$ is minimal. Finally, we add all edges $(c_{i,\lambda}, c_\lambda)$ to \mathcal{G}_λ .

To make sure the complexity of our spanner does not become too large, we must choose the groups in such a way that the spanner edges do not cross “bad” parts of P too often. The following lemma states the properties that we require of our groups to achieve this.



■ **Figure 4** Any $(3 - \varepsilon)$ -spanner in a simple polygon with m vertices may have complexity $\Omega(nm)$.

► **Lemma 4.** *If the groups adhere to the following properties, then \mathcal{G} has $O(m\sqrt{n} + n \log^2 n)$ complexity:*

1. *each group contains $\Theta(\sqrt{n})$ sites, and*
2. *each vertex of P is only used by shortest paths within $O(1)$ groups.*

Proof. We will first prove the complexity of the edges in one level of the 1-dimensional spanner is $O(m\sqrt{n} + n)$. Two types of edges are added to the spanner: (a) edges from some c_i to c , and (b) edges from some $p \in S_i$ to c_i . According to property 1, there are $\Theta(\sqrt{n})$ groups, and thus $\Theta(\sqrt{n})$ type (a) edges, that each have a complexity of $O(m)$. Thus the total complexity of these edges is $O(m\sqrt{n})$. Let r_i be the maximum complexity of a shortest path between any two sites in S_i and let V_i be the set of vertices this path visits. Property 2 states that for any $v \in V_i$ it holds that $|\{j \mid v \in V_j\}| = O(1)$, which implies that $\sum_i r_i = O(m)$. The complexity of all type (b) edges is thus $O(n) + \sum_i r_i O(\sqrt{n}) = O(m\sqrt{n} + n)$.

Next, we show that in both recursions, the 1-dimensional recursion and the recursion on P_ℓ and P_r , not only the number of sites, but also the complexity of the polygon is divided over the two subproblems. Splitting the sites into left and right of O corresponds to splitting the polygon horizontally at O : all sites left (right) of O in the 1-dimensional space lie in the part of the polygon below (above) this horizontal line segment. Thus, shortest paths between sites left of O use part of the polygon that is disjoint from the shortest paths between the sites right of O . This means that for two subproblems we have that $m_1 + m_2 = m$, where m_i denotes the maximum complexity of a path in subproblem i . The recursion for the complexity is now given by

$$T(n, m) = T(n/2, m_1) + T(n/2, m_2) + O(m\sqrt{n} + n), \text{ with } m_1 + m_2 = m.$$

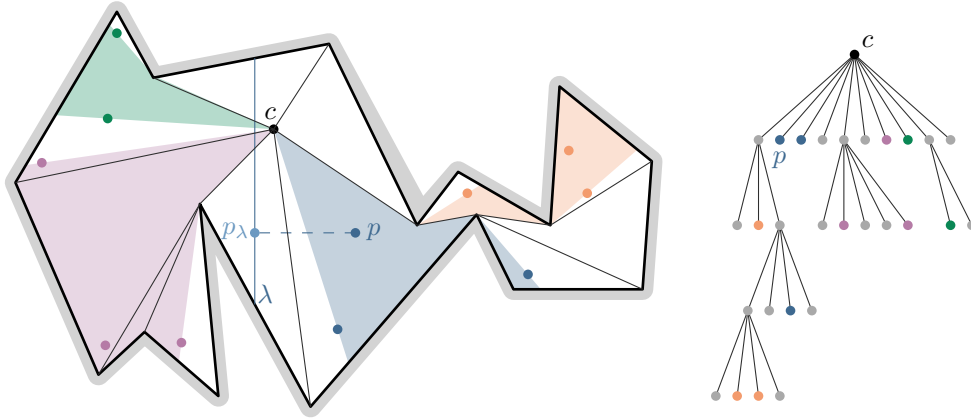
In the full version [19] we show this solves to $T(n) = O(m\sqrt{n} + n \log n)$.

Similarly, the split by λ divides the polygon into two subpolygons, while adding at most two new vertices. As all vertices, except for the endpoints of λ , are in P_ℓ or P_r (not both), the total complexity of both subpolygons is at most $m + 4$. We obtain the following recursion

$$T(n, m) = T(n/2, m_1) + T(n/2, m_2) + O(m\sqrt{n} + n \log n), \text{ with } m_1 + m_2 = m + 4,$$

which solves to $T(n) = O(m\sqrt{n} + n \log^2 n)$. ◀

To form groups that adhere to these two properties, we consider the shortest path tree SPT_c of c : the union of all shortest paths from c to the vertices of P . We include the sites $p \in S \setminus \{c\}$ as leaves in SPT_c as children of their apex, i.e., the last vertex on $\pi(c, p)$. This



■ **Figure 5** The shortest path tree of c . Each group S_i has an associated polygonal region R_i in P .

gives rise to an ordering of the sites in S , and thus of the weighted sites in S_λ , based on the in-order traversal of the tree. We assign the first $\lceil \sqrt{n} \rceil$ sites to S_1 , the second $\lceil \sqrt{n} \rceil$ to S_2 , etc. See Figure 5.

Clearly these groups adhere to property 1. Proving that they also adhere to property 2 is more involved. For each group S_i , consider the minimal subtree \mathcal{T}_i of SPT_c containing all $p \in S_i$. \mathcal{T}_i defines a polygonal region R_i in P as follows. Refer to Figure 5 for an illustration. Let v_i be the root of \mathcal{T}_i . Consider the shortest path $\pi(v_i, a)$, where a is the first site of S_i in \mathcal{T}_i by the ordering used before. Let π_a be the path obtained from $\pi(v_i, a)$ by extending the last segment of $\pi(v_i, a)$ to the boundary of P . Similarly, let π_b be such a path for the last site of S_i in \mathcal{T}_i . We take R_i to be the region in P rooted at v_i and bounded by π_a , π_b , and some part of the boundary of P , that contains the sites in S_i . In case v_i is c , we split R_i into two regions R_j and R_k , such that the angle of each of these regions at c is at most π . The set S_i is then also split into two sets S_j and S_k accordingly. The following three lemmas on R_i and \mathcal{T}_i together imply that the groups adhere to property 2.

► **Lemma 5.** *Only vertices of P that are in \mathcal{T}_i can occur in R_i .*

► **Lemma 6.** *All shortest paths between sites in S_i are contained within R_i .*

► **Lemma 7.** *Any vertex $v \in SPT_c$ occurs in at most two trees \mathcal{T}_i and \mathcal{T}_j as a non-root node.*

Note that the root r of \mathcal{T}_i is never used in a shortest path between sites in S_i , because r cannot be a reflex vertex of R_i . Consequently, Lemma 4 states that the spanner has complexity $O(m\sqrt{n} + n \log^2 n)$.

► **Lemma 8.** *The graph \mathcal{G} is a geodesic $4\sqrt{2}$ -spanner of size $O(n \log^2 n)$.*

Proof. We prove the 1-dimensional spanner \mathcal{G}_λ is a 4-spanner with $O(n \log n)$ edges. Together with Lemma 2, this directly implies \mathcal{G} is a $4\sqrt{2}$ -spanner with $O(n \log^2 n)$ edges.

In each level of the recursion, we still add only a single edge for each site. Thus, the total number of edges is $O(n \log n)$. Again, consider two sites $p_\lambda, q_\lambda \in S_\lambda$, and let c_λ be the chosen center point at the level where p_λ and q_λ are separated by O . Let S_i be the group of p_λ and S_j the group of q_λ . Both the edges $(p_\lambda, c_{i,\lambda})$ and $(c_{i,\lambda}, c_\lambda)$ are in \mathcal{G}_λ , similarly for q_λ . We thus have a path $p_\lambda \rightarrow c_{i,\lambda} \rightarrow c_\lambda \rightarrow c_{j,\lambda} \rightarrow q_\lambda$ in \mathcal{G}_λ . Using that $d_w(p_\lambda, c_{i,\lambda}) \leq d_w(p_\lambda, O) + d_w(c_{i,\lambda}, O)$, because of the triangle inequality, and $d_w(c_{i,\lambda}, O) \leq d_w(p_\lambda, O)$, we find:

$$\begin{aligned}
d_{\mathcal{G}_\lambda}(p_\lambda, q_\lambda) &= d_w(p_\lambda, c_{i,\lambda}) + d_w(c_{i,\lambda}, c_\lambda) + d_w(c_\lambda, c_{j,\lambda}) + d_w(c_{j,\lambda}, q_\lambda) \\
&\leq d_w(p_\lambda, O) + 2d_w(c_{i,\lambda}, O) + 2d_w(c_\lambda, O) + 2d_w(c_{j,\lambda}, O) + d_w(q_\lambda, O) \\
&\leq 4d_w(p_\lambda, O) + 4d_w(q_\lambda, O) \\
&= 4d_w(p_\lambda, q_\lambda)
\end{aligned}$$

◀

3.2.2 A $(2k + \varepsilon)$ -spanner of complexity $O(mn^{1/k} + n \log^2 n)$

In this section we sketch how to generalize the approach of Section 3.2.1 to obtain a spanner with a trade-off between the (constant) spanning ratio and complexity. Fix $N = n^{1/k}$, for some integer constant $k \geq 1$. Instead of $\Theta(\sqrt{n})$ groups, we create $\Theta(N)$ groups. For each of these groups we select a center, and then partition the groups further recursively. By connecting each center to its parent center, we obtain a tree of height k . This results in a spanning ratio of $k2\sqrt{2}$.

Abam et al. [3] refine their spanner construction to obtain a $(2+\varepsilon)$ -spanner. We generalize this refinement in Lemma 3 of the full version [19]. The main idea is as follows. For each point p_λ , we additionally include a collection of $O(t^2/\varepsilon^2)$ evenly spread points on λ close to p_λ in S_λ . For an edge in \mathcal{G}_λ between a point in the collection of p_λ and a point in the collection of q_λ , we add the edge (p, q) to \mathcal{G} . This way, we even obtain a $(2k + \varepsilon)$ -spanner.

► **Lemma 9.** *For any constant $\varepsilon > 0$ and integer constant $k \geq 1$, there exists a geodesic $(2k + \varepsilon)$ -spanner of size $O(c_{\varepsilon,k} n \log^2 n)$ and complexity $O(c_{\varepsilon,k}(mn^{1/k} + n \log^2 n))$, where $c_{\varepsilon,k}$ is a constant depending only on ε and k .*

3.3 Construction algorithm

In this section we propose an algorithm to construct the spanners of Section 3.2. The following gives a general overview of the algorithm, which computes a $(2k + \varepsilon)$ -spanner in $O(n \log^2 n + m \log n)$ time. In the rest of this section we will discuss the steps in more detail.

1. Preprocess P for efficient shortest path queries and build both the vertical decomposition \mathcal{VD} and horizontal decomposition \mathcal{HD} of P .
2. For each $p \in S$, find the trapezoid in \mathcal{VD} and \mathcal{HD} that contains p . For each trapezoid $\nabla \in \mathcal{VD}$, store the number of sites of S that lies in ∇ and sort these sites on their x -coordinate.
3. Recursively compute a spanner on the sites S in P :
 - a. Find a vertical chord λ of P such that λ partitions P into two polygons P_ℓ and P_r , and each subpolygon contains at most $2n/3$ sites using the algorithm of Lemma 10.
 - b. For each $p \in S$, find the point p_λ on λ and its weight using the algorithm of Lemma 11, and add this point to S_λ .
 - c. Compute an additively weighted 1-dimensional spanner \mathcal{G}_λ on the set S_λ using the algorithm of Lemma 12 or Lemma 13.
 - d. For every edge $(p_\lambda, q_\lambda) \in E_\lambda$ add the edge (p, q) to \mathcal{G} .
 - e. Recursively compute spanners for S_ℓ in P_ℓ and S_r in P_r .

In step 1, we preprocess the polygon in $O(m)$ time such that the distance between any two points $p, q \in P$ can be computed in $O(\log m)$ time [15, 22]. We also build the horizontal and vertical decompositions of P , and a corresponding point location data structure, as a preprocessing step in $O(m)$ time [15, 26]. We then perform a point location query for each site $p \in S$ in $O(n \log m)$ time in step 2 and sort the sites within each trapezoid in $O(n \log n)$

16:10 The Complexity of Geodesic Spanners

time in total. The following lemma describes the algorithm to compute a vertical chord that partitions P into two subpolygons such that each of them contains roughly half of the sites in S . It is based on the algorithm of Bose et al. [7] that finds such a chord without the constraint that it should be vertical. Because of this constraint, we use the vertical decomposition of P instead of a triangulation in our algorithm.

► **Lemma 10.** *In $O(n + m)$ time, we can find a vertical chord of P that partitions P into two subpolygons P_ℓ and P_r , such that each subpolygon contains at most $2n/3$ sites of S .*

The following lemma states that we can find the projections p_λ efficiently. The algorithm produces not only these projected sites, but also the shortest path tree SPT_λ of λ .

► **Lemma 11.** *We can compute the closest point p_λ on λ and $d(p, p_\lambda)$ for all sites $p \in S$, and the shortest path tree SPT_λ , in $O(m + n \log m)$ time.*

► **Lemma 12.** *Given SPT_λ , we can construct a 4-spanner \mathcal{G}_λ on the additively weighted points S_λ , where the groups adhere to the properties of Lemma 4, in $O(n \log n + m)$ time.*

Proof. The 4-spanner of Section 3.2.1 requires an additional step at each level of the recursion, namely the formation of $\Theta(\sqrt{n})$ groups. We first discuss the running time to construct a 4-spanner when forming the groups as in Section 3.2.1, and then improve the running time by introducing a more efficient way to form the groups.

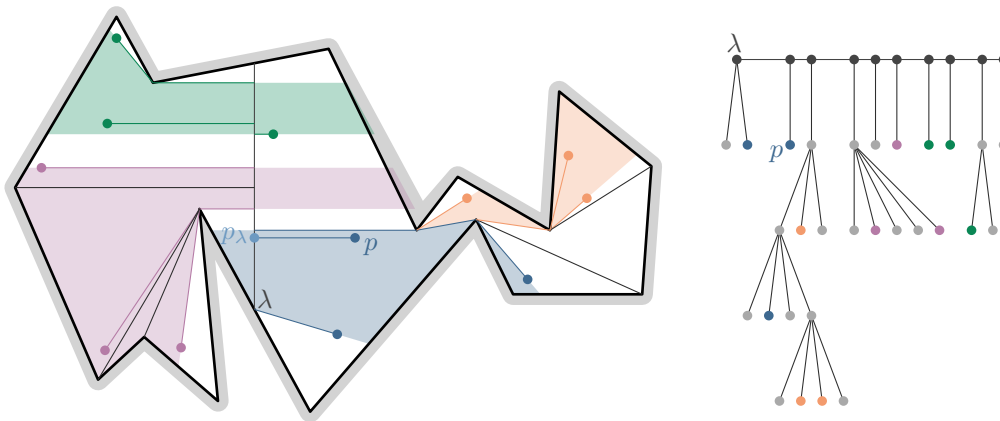
In Section 3.2.1, the groups are formed based on the shortest path tree of the site c . Building the shortest path tree, and a corresponding point location data structure, takes $O(m)$ time [23]. Then, we perform a point location query for each site to find its apex in the shortest path tree, and add the sites to the tree. These queries take $O(n \log m)$ time in total. We form groups based on the traversal of the tree. Note that we do not distinguish between sites with the same parent in the tree, as the tree \mathcal{T}_i (and thus the region R_i) obtained contains the same vertices of P regardless of the order of these sites. After obtaining the groups, we again add only $O(n)$ edges to the spanner. The overall running time of the algorithm is thus $O((m + n \log m) \log n)$.

This running time can be improved by using another approach to form the groups. To form groups that adhere to the properties of Lemma 4, and thus result in a spanner of the same complexity, we can use any partition of P into regions R_i , as long R_i as contains $\Theta(\sqrt{n})$ sites and $\sum_i r_i = O(m)$. Next, we describe how to form such groups efficiently using SPT_λ .

We first define an ordering on the sites. This is again based on the traversal of some shortest path tree. Instead of considering the shortest path tree of a point site, we consider the shortest path tree SPT_λ of λ . Again, all sites in S are included in this shortest path tree. Additionally, we split the node corresponding to λ into a node for each distinct projection point on λ (of the vertices and the sites) and add an edge between each pair of adjacent points, see Figure 6. We root the tree at the node corresponding to the bottom endpoint of λ . Whenever a node t on λ has multiple children, in other words, when multiple sites are projected to the same point t , our assumption that all y -coordinates are distinct ensures that all these sites lie either in P_ℓ or P_r .

The groups are formed based on the in-order traversal of this tree, which can be performed in $O(m + n)$ time. As before, the first $\lceil \sqrt{n} \rceil$ are in S_1 , the second in S_2 , etc. The groups thus adhere to the first property. Next, we show they also adhere to the second property.

For each group S_i , we again consider the minimal subtree \mathcal{T}_i of SPT_λ containing all $p \in S_i$. \mathcal{T}_i defines a region R_i in P as follows. Let a be the first site of S_i in \mathcal{T}_i by the ordering used before. Assume that a lies in P_ℓ . We distinguish two cases: $a_\lambda \in \mathcal{T}_i$, or $a_\lambda \notin \mathcal{T}_i$. When $a_\lambda \in \mathcal{T}_i$, then let π_a be the path obtained from $\pi(a_\lambda, a)$ by extending the last segment to



■ **Figure 6** The shortest path tree SPT_λ and associated polygonal region R_i for each group S_i .

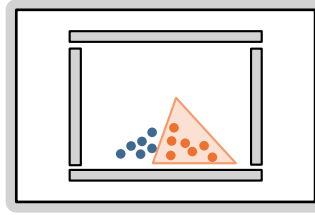
the boundary of P . Additionally, we extend π_a into P_r horizontally until we hit the polygon boundary. When $a_\lambda \notin \mathcal{T}_i$, consider the root v_i of \mathcal{T}_i . Let π_a be the path obtained from $\pi(v_i, a)$ by extending the last segment of the path to the boundary of P . Similarly, let π_b be such a path for the rightmost site of S_i in \mathcal{T}_i . We take R_i to be the region in P bounded by π_a , π_b , and some part of the boundary of P that contains the sites in S_i . See Figure 6. Note that, as before, only vertices of P that are in \mathcal{T}_i can occur in R_i . All shortest paths between sites in S_i are contained within R_i . Just as for the shortest path tree of c , Lemma 7 implies that any vertex $v \in SPT_\lambda$ occurs in at most two trees \mathcal{T}_i and \mathcal{T}_j as a non-root vertex. We conclude that any vertex is used by shortest paths within at most two groups.

After splitting λ at a point O , the tree SPT_λ is also split into two trees \mathcal{T}_ℓ and \mathcal{T}_r that contain exactly the sites in S_ℓ and S_r . We can thus reuse the ordering to form groups at each level of the recursion. This way, the total running time at a single level of the recursion is reduced to $O(n)$. The overall running time thus becomes $O(n \log n + m)$. ◀

► **Lemma 13.** *Given SPT_λ , we can construct a $2k$ -spanner \mathcal{G}_λ on the additively weighted points S_λ , where groups are formed as in Section 3.2.2, in $O(n \log n + m)$ time.*

Proof. To construct the $(2k + \varepsilon)$ -spanner of Section 3.2.2, we can use the shortest path tree of λ to form the groups as before. Note that we can select a center for each group after computing the groups, as including the center in the subgroups does not influence spanning ratio or complexity. After ordering the sites based on the in-order traversal of SPT_λ , we can build the tree of groups in linear time using a bottom up approach. As before, fix $N = n^{1/k}$. We first form the $\Theta(N^k)$ lowest level groups, containing only a single site, and select a center for each group. Each group at level i is created by merging $\Theta(N)$ groups at level $i - 1$, based on the same ordering. We do not perform this merging explicitly, but for each group we select the site closest to O of the merged level- $(i - 1)$ centers as the center. Because our center property, being the closest to O , is decomposable, this indeed gives us the center of the entire group. This way, we can compute the edges added in one level of the recursion in linear time, so the running time remains $O(n \log n + m)$. ◀

The total running time thus becomes $O((n(\log n + \log m) + m) \log n) = O(n \log^2 n + m \log n)$. By splitting the polygon alternately based on the sites and the polygon vertices, we can replace the final $O(\log n)$ factor by $O(\min(\log n, \log m))$. Using the refinement of Section 3.2.2, step 3 is performed on $c_{\varepsilon,k} n$ sites, where $c_{\varepsilon,k}$ is a constant depending only on ε and k , increasing the running time for this step by a factor $O(c_{\varepsilon,k})$. Together with Lemma 9, we obtain the following theorem.



■ **Figure 7** No shortest path between two points on P can separate the sites into two groups. We can separate the sites using three shortest paths, for example using the orange triangle.

► **Theorem 14.** *Let S be a set of n point sites in a simple polygon P with m vertices, and let $k \geq 1$ be any integer constant. For any constant $\varepsilon > 0$, we can build a geodesic $(2k+\varepsilon)$ -spanner of size $O(c_{\varepsilon,k}n \log^2 n)$ and complexity $O(c_{\varepsilon,k}(mn^{1/k} + n \log^2 n))$ in $O(c_{\varepsilon,k}n \log^2 n + m \log n + K)$ time, where $c_{\varepsilon,k}$ is a constant depending only on ε and k , and K is the output complexity.*

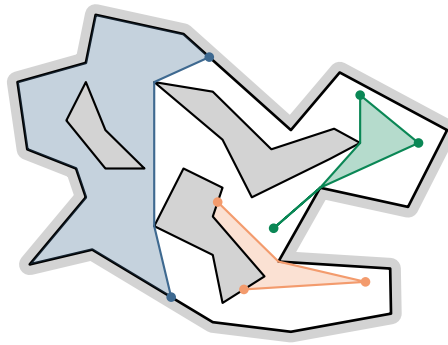
4 Spanners in a polygonal domain

We consider a set of point sites S that lie in a polygonal domain P with m vertices and h holes. Let ∂P denote the boundary of the outer polygon. Suppose we would try the same construction as we used for a simple polygon in Section 3. Then, in a polygonal domain, we run into two problems. First, we cannot split the polygonal domain into two subpolygons by a line segment λ that roughly splits the sites in S , because any line segment that appropriately partitions S might intersect one or more holes. Second, the shortest path $\pi(p, q)$ between two sites $p, q \in S$ is no longer homotopic to a path $\pi(p, c) \cup \pi(c, q)$, which means our bound on the complexity is no longer valid. In the next section, we describe these problems and how we overcome them in more detail.

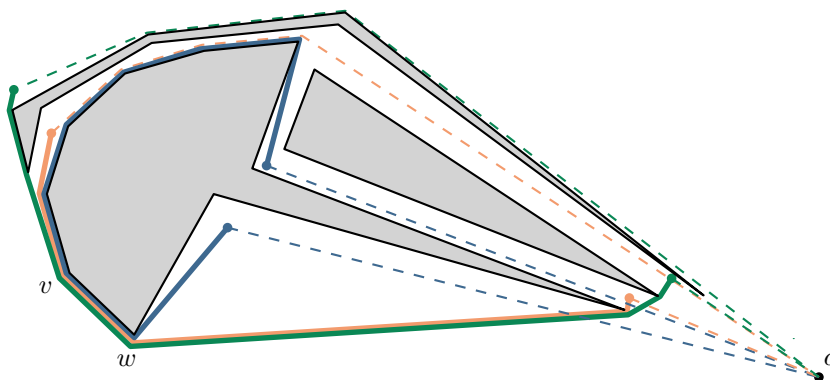
4.1 Low complexity geodesic spanners

Subdividing the domain. To apply our 1-dimensional spanner, we require only that the splitting curve λ is a shortest path in P . Instead of partitioning the domain by a line segment, we can thus use one or more shortest paths to partition the domain. Allowing only a shortest path between two points on ∂P as our separator is not sufficient, as it is not always possible to split the sites roughly equally this way. See Figure 7 for an example. Therefore, we use a *balanced shortest-path separator* (sp-separator), based on the balanced sp-separator of Abam et al. [3].

We use three types of separator: a shortest path between two points on ∂P (1-separator), two shortest paths starting at the same point and ending at the boundary of a single hole (2-separator), and three shortest paths $\pi(u, v)$, $\pi(v, w)$, and $\pi(u, w)$ with $u, v, w \in P$ (3-separator). Let P_ℓ be the polygonal domain to the left of λ , when λ is a 1-separator, and interior to λ , when λ is a 2- or 3-separator. Symmetrically, P_r is the domain to the right of, or exterior to, λ . See Figure 8 for an illustration. In the full version [19], we prove the existence of such a separator. In the proof, we correct for a missing case in the proof by Abam et al. [3] for the existence of a separator on a polyhedral terrain, thereby confirming their result. We also require an additional step in the constructive proof, because unlike a terrain, a polygonal domain is not continuous. This is where a 2-separator is required, which does not occur on a terrain. Finally, we provide an algorithm to construct a balanced sp-separator in a polygonal domain. This results in the following theorem.



■ **Figure 8** The three types of separators: A 1-separator (blue), a 2-separator (orange), and a 3-separator (green). For each separator P_ℓ is the colored region.



■ **Figure 9** Assigning the sites to groups based on the shortest path tree of c , as described in Section 3.2.1, forms these colored groups. The shortest path from each site to c is shown dashed. Each shortest path between the two sites of a group visits both vertices v and w .

► **Theorem 15.** *Let S be a set of n point sites in a polygonal domain P with m vertices. A balanced sp -separator exists and it can be computed in $O(n^2 \log m + nm \log m)$ time.*

The spanner edges. In Section 3.2.1 and 3.3, we form groups of sites such that the shortest paths within a group are (almost) disjoint from the shortest paths within other groups. We use the shortest path tree of c (or λ) to obtain these groups. In a polygonal domain, the paths $\pi(p, q)$ and $\pi(p, c) \cup \pi(c, q)$, with $p, q, c \in P$, are not necessarily homotopic. Figure 9 gives an example of how this can result in many groups that use the same vertex of P in their shortest paths.

So far, we assumed that every edge $(p, q) \in E$ is a shortest path between p and q . To obtain a spanner of low complexity, we can also allow an edge between p and q to be any path between the two sites, as long as we keep the desired spanning ratio. Note that our complexity lower bounds still hold in this case. In particular, for an edge (p_λ, q_λ) in the 1-dimensional spanner, the edge (p, q) that we add to \mathcal{G} is no longer $\pi(p, q)$. Instead, let (p, q) be the shortest path from p to q via p_λ and q_λ , excluding any overlap of the path. We denote this path by $\pi_\lambda(p, q)$. Formally, $\pi_\lambda(p, q)$ is defined as follows.

► **Definition 16.** *The path $\pi_\lambda(p, q)$ is given by:*

- $\pi(p, p_\lambda) \cup \pi(p_\lambda, q_\lambda) \cup \pi(q_\lambda, q)$, where $\pi(p_\lambda, q_\lambda) \subseteq \lambda$, if $\pi(p, p_\lambda)$ and $\pi(q, q_\lambda)$ are disjoint,
- $\pi(p, r) \cup \pi(r, q)$, where r denotes the closest point to p of $\pi(p, p_\lambda) \cap \pi(q, q_\lambda)$, otherwise.

16:14 The Complexity of Geodesic Spanners

In the full version [19], we show that using these paths as edges does not increase the spanning ratio, and prove a similar result to Lemma 4 for this new edge type. Additionally, we prove that using the shortest path tree of λ for the formation of the groups ensures that any vertex of P is used by paths $\pi_\lambda(p, q)$ within only $O(1)$ groups.

4.2 Construction algorithm

As before, let S_ℓ be the sites in the closed region P_ℓ , and $S_r := S \setminus S_\ell$. The following gives an overview of the algorithm that computes a $(2k + \varepsilon)$ -spanner of complexity $O(mn^{1/k} + n \log^2 n)$ in $O(n^2 \log m + nm \log m)$ time for a set of point sites S in a polygonal domain P .

1. Find an sp-separator such that P is partitioned into two polygons P_ℓ and P_r , and S_ℓ contains at least $2n/9$ and at most $2n/3$ sites using the algorithm of Theorem 15.
2. For each shortest path λ of the separator:
 - a. For each $p \in S$ find the weighted point p_λ on λ and add this point to S_λ .
 - b. Compute an additively weighted 1-dimensional spanner \mathcal{G}_λ on the set S_λ .
 - c. For every edge $(p_\lambda, q_\lambda) \in E_\lambda$ add the edge $(p, q) = \pi_\lambda(p, q)$ to \mathcal{G} .
3. Recursively compute spanners for S_ℓ in P_ℓ and S_r in P_r .

According to Theorem 15, step 1 takes $O(n^2 \log m + nm \log m)$ time. In step 2, we find the projected sites using the shortest path map of λ in $O((m + n) \log m)$ time [25], and then compute the 1-dimensional spanner in $O(n \log n + m)$ time as in Lemma 13. This means that the construction of the sp-separator dominates the construction time of the spanner.

► **Theorem 17.** *Let S be a set of n point sites in a polygonal domain P with m vertices, and let $k \geq 1$ be any integer constant. For any constant $\varepsilon > 0$, we can build a geodesic $(2k + \varepsilon)$ -spanner of size $O(c_{\varepsilon,k} n \log^2 n)$ and complexity $O(c_{\varepsilon,k}(mn^{1/k} + n \log^2 n))$ in $O(c_{\varepsilon,k}(n^2 \log m + nm \log m) + K)$ time, where $c_{\varepsilon,k}$ is a constant depending only on ε and k , and K is the output complexity.*

We can no longer output the edges implicitly by stating only the endpoints of the edges. Instead, we output an edge $\pi_\lambda(p, q)$ implicitly by stating the points p_λ and q_λ , when $\pi(p, p_\lambda)$ and $\pi(q, q_\lambda)$ are disjoint, or the point r from Definition 16 when they are not disjoint. The point r is the lowest common ancestor of the nodes p and q in SPT_λ . This can be computed in $O(1)$ time after preprocessing SPT_λ in $O((n + m) \log(n + m))$ [5].

4.3 A $(2 + \varepsilon)$ -spanner with a dependence on \sqrt{h}

Because the computation of a balanced shortest-path separator is quite expensive, we consider another method to partition the domain by Abam et al. [1] using \sqrt{h} simple polygons, where h is the number of holes in P . In the full version [19], we show this indeed improves the construction time, albeit at an increase in complexity.

► **Theorem 18.** *Let S be a set of n point sites in a polygonal domain P with m vertices and h holes, and let $k \geq 1$ be any integer constant. For any constant $\varepsilon > 0$, we can build a geodesic $(2k + \varepsilon)$ -spanner of size $O(c_{\varepsilon,k} \sqrt{h} n \log^2 n)$ and complexity $O(c_{\varepsilon,k} \sqrt{h}(mn^{1/k} + n \log^2 n))$ in $O(c_{\varepsilon,k} \sqrt{h} n \log^2 n + m \log m + K)$ time, where $c_{\varepsilon,k}$ is a constant depending only on ε and k , and K is the output complexity.*

References

- 1 Mohammad Ali Abam, Marjan Adeli, Hamid Homapour, and Pooya Zafar Asadollahpoor. Geometric spanners for points inside a polygonal domain. In *31st International Symposium on Computational Geometry, SoCG*, volume 34 of *LIPICs*, pages 186–197, 2015. doi:10.4230/LIPICs.SOCG.2015.186.
- 2 Mohammad Ali Abam, Mark de Berg, Mohammad Farshi, Joachim Gudmundsson, and Michiel H. M. Smid. Geometric spanners for weighted point sets. *Algorithmica*, 61(1):207–225, 2011.
- 3 Mohammad Ali Abam, Mark de Berg, and Mohammad Javad Rezaei Seraji. Geodesic spanners for points on a polyhedral terrain. *SIAM J. Comput.*, 48(6):1796–1810, 2019. doi:10.1137/18M119358X.
- 4 Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel H. M. Smid. Euclidean spanners: short, thin, and lanky. In *In Twenty-Seventh Annual ACM Symposium on Theory of Computing, STOC, Proceedings*, pages 489–498. ACM, 1995.
- 5 Michael A. Bender and Martin Farach-Colton. The LCA problem revisited. In *In Theoretical Informatics, 4th Latin American Symposium, LATIN, Proceedings*, volume 1776 of *Lecture Notes in Computer Science*, pages 88–94. Springer, 2000.
- 6 Sujoy Bhore and Csaba D. Tóth. Euclidean Steiner spanners: Light and sparse. *SIAM Journal on Discrete Mathematics*, 36(3):2411–2444, 2022.
- 7 Prosenjit Bose, Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, and Anil Maheshwari. Polygon cutting: Revisited. In *Discrete and Computational Geometry, Japanese Conference, JCDCG, Revised Papers*, volume 1763 of *LNCS*, pages 81–92, 1998. doi:10.1007/978-3-540-46515-7_7.
- 8 Prosenjit Bose, Rolf Fagerberg, André van Renssen, and Sander Verdonschot. On plane constrained bounded-degree spanners. *Algorithmica*, 81(4):1392–1415, 2019.
- 9 Prosenjit Bose, Joachim Gudmundsson, and Michiel H. M. Smid. Constructing plane spanners of bounded degree and low weight. *Algorithmica*, 42(3-4):249–264, 2005.
- 10 Prosenjit Bose and J. Mark Keil. On the stretch factor of the constrained Delaunay triangulation. In *In the 3rd International Symposium on Voronoi Diagrams in Science and Engineering, ISVD*, pages 25–31. IEEE Computer Society, 2006.
- 11 Prosenjit Bose and Michiel H. M. Smid. On plane geometric spanners: A survey and open problems. *Comput. Geom.*, 46(7):818–830, 2013. doi:10.1016/j.comgeo.2013.04.002.
- 12 Prosenjit Bose and André van Renssen. Spanning properties of Yao and Θ -graphs in the presence of constraints. *Int. J. Comput. Geom. Appl.*, 29(2):95–120, 2019.
- 13 T.-H. Hubert Chan, Anupam Gupta, Bruce M. Maggs, and Shuheng Zhou. On hierarchical routing in doubling metrics. *ACM Trans. Algorithms*, 12(4):55:1–55:22, 2016. doi:10.1145/2915183.
- 14 T.-H. Hubert Chan, Mingfei Li, Li Ning, and Shay Solomon. New doubling spanners: Better and simpler. *SIAM J. Comput.*, 44(1):37–53, 2015.
- 15 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discret. Comput. Geom.*, 6:485–524, 1991.
- 16 Kenneth L. Clarkson. Approximation algorithms for shortest path motion planning. In Alfred V. Aho, editor, *In the 19th Annual ACM Symposium on Theory of Computing, STOC*, pages 56–65. ACM, 1987.
- 17 Gautam Das. The visibility graph contains a bounded-degree spanner. In *In the 9th Canadian Conference on Computational Geometry, CCCG*, 1997.
- 18 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: Algorithms and applications, 3rd Edition*. Springer, 2008. URL: <https://www.worldcat.org/oclc/227584184>.
- 19 Sarita de Berg, Marc van Kreveld, and Frank Staals. The complexity of geodesic spanners. *CoRR*, abs/2303.02997, 2023. URL: <https://arxiv.org/abs/2303.02997>.
- 20 Michael Elkin and Shay Solomon. Optimal Euclidean spanners: Really short, thin, and lanky. *J. ACM*, 62(5):35:1–35:45, 2015.

16:16 The Complexity of Geodesic Spanners

- 21 Lee-Ad Gottlieb and Liam Roditty. An optimal dynamic spanner for doubling metric spaces. In *16th Annual European Symposium on Algorithms, ESA*, volume 5193 of *Lecture Notes in Computer Science*, pages 478–489, 2008. doi:10.1007/978-3-540-87744-8_40.
- 22 Leonidas J. Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39(2):126–152, 1989.
- 23 Leonidas J. Guibas, John Hershberger, Daniel Leven, Micha Sharir, and Robert Endre Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.
- 24 Sarel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006. doi:10.1137/S0097539704446281.
- 25 John Hershberger and Subhash Suri. An optimal algorithm for Euclidean shortest paths in the plane. *SIAM J. Comput.*, 28(6):2215–2256, 1999.
- 26 David G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM J. Comput.*, 12(1):28–35, 1983.
- 27 Hung Le and Shay Solomon. Truly optimal Euclidean spanners. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 1078–1100. IEEE Computer Society, 2019.
- 28 Christos Levcopoulos, Giri Narasimhan, and Michiel H. M. Smid. Efficient algorithms for constructing fault-tolerant geometric spanners. In *In the 13th Annual ACM Symposium on the Theory of Computing, STOC, Proceedings*, pages 186–195. ACM, 1998.
- 29 Joseph S. B. Mitchell and Wolfgang Mulzer. Proximity algorithms. In *Handbook of Discrete and Computational Geometry (3rd Edition)*, chapter 32, pages 849–874. Chapman & Hall/CRC, 2017.
- 30 Giri Narasimhan and Michiel H. M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.

An Extension Theorem for Signotopes

Helena Bergold  

Department of Computer Science, Freie Universität Berlin, Germany

Stefan Felsner  

Institut für Mathematik, Technische Universität Berlin, Germany

Manfred Scheucher  

Institut für Mathematik, Technische Universität Berlin, Germany

Abstract

In 1926, Levi showed that, for every pseudoline arrangement \mathcal{A} and two points in the plane, \mathcal{A} can be extended by a pseudoline which contains the two prescribed points. Later extendability was studied for arrangements of pseudohyperplanes in higher dimensions. While the extendability of an arrangement of proper hyperplanes in \mathbb{R}^d with a hyperplane containing d prescribed points is trivial, Richter-Gebert found an arrangement of pseudoplanes in \mathbb{R}^3 which cannot be extended with a pseudoplane containing two particular prescribed points.

In this article, we investigate the extendability of signotopes, which are a combinatorial structure encoding a rich subclass of pseudohyperplane arrangements. Our main result is that signotopes of odd rank are extendable in the sense that for two prescribed crossing points we can add an element containing them. Moreover, we conjecture that in all even ranks $r \geq 4$ there exist signotopes which are not extendable for two prescribed points. Our conjecture is supported by examples in ranks 4, 6, 8, 10, and 12 that were found with a SAT based approach.

2012 ACM Subject Classification Mathematics of computing \rightarrow Discrete mathematics; Mathematics of computing \rightarrow Solvers; Mathematics of computing \rightarrow Enumeration; Hardware \rightarrow Theorem proving and SAT solving; Theory of computation \rightarrow Automated reasoning; Theory of computation \rightarrow Computational geometry

Keywords and phrases arrangement of pseudolines, extendability, Levi’s extension lemma, arrangement of pseudohyperplanes, signotope, oriented matroid, partial order, Boolean satisfiability (SAT)

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.17

Related Version *Full Version:* <http://arxiv.org/abs/2303.04079>

Supplementary Material *Software (Data and Source Code):*

<https://github.com/manfredscheucher/supplemental-signotope-extension>

Funding *Helena Bergold:* DFG Research Training Group ‘Facets of Complexity’ (DFG-GRK 2434).
Stefan Felsner: DFG Grant FE 340/13-1.

Manfred Scheucher: DFG Grant SCHE 2214/1-1.

Acknowledgements We thank the anonymous reviewers for valuable comments.

1 Introduction

Given a family of hyperplanes \mathcal{H} in \mathbb{R}^d , any d points in \mathbb{R}^d , not all on a common hyperplane of \mathcal{H} , define a hyperplane which is distinct from the hyperplanes in \mathcal{H} . For dimension $d = 2$, Levi [14] proved in his pioneering article on pseudoline arrangements that the fundamental extendability of line arrangements also applies to the more general setting of pseudoline arrangements. A *pseudoline* is a Jordan curve in the Euclidean plane such that its removal from the plane results in two unbounded components, and a *pseudoline arrangement* is a family of pseudolines such that each pair of pseudolines intersects in exactly one point, where the two curves cross properly.



© Helena Bergold, Stefan Felsner, and Manfred Scheucher;
licensed under Creative Commons License CC-BY 4.0

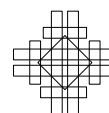
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 17; pp. 17:1–17:14

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



► **Theorem 1.1** (Levi’s extension lemma for pseudoline arrangements [14]). *Given an arrangement \mathcal{A} of pseudolines and two points in \mathbb{R}^2 , not on a common pseudoline of \mathcal{A} . Then \mathcal{A} can be extended by an additional pseudoline which passes through the two prescribed points.*

Several proofs for Levi’s extension lemma are known today (besides [14], see also [1, 9, 19]) and generalizations to higher dimensions have been studied in the context of oriented matroids, which by the representation theorem of Folkman and Lawrence [10] have representations as projective pseudohyperplane arrangements. For more about oriented matroids, see [8].

Goodman and Pollack [12] presented an arrangement of 8 pseudoplanes in \mathbb{R}^3 and a selection of three points such that there is no extension of the arrangement with a pseudoplane containing the points. Richter-Gebert [18] then investigated a weaker version with only two prescribed points in dimension 3 such that the extending pseudohyperplane contains these two points. He found an example of a rank 4 oriented matroid on 8 elements such that there is no one element extension with an element containing the two prescribed cocircuits. With the representation theorem this implies that even the weaker extendability with two prescribed points does not hold. The existence of an extension theorem or of counterexamples in higher dimensions/ranks remains open.

We present a proof of Levi’s extension lemma in a purely combinatorial setting and show that the proof can be adapted to work for higher dimensions. We represent the geometry by r -signotopes and prove extendability in even dimensions d , that is, when the rank $r = d + 1$ is odd; see Theorem 1.2. Surprisingly, there are non-extendable examples in rank 4, 6, 8, 10, and 12. We conjecture that there is no extension theorem for any even rank $r \geq 4$; see Conjecture 1.4.

Signotopes are in close relation to higher Bruhat orders which were introduced by Manin and Schechtman [15] and further studied in [21]. In rank 3, signotopes correspond to pseudoline arrangements in the plane [9]. In higher ranks they are a subclass of pseudohyperplane arrangements.

Before we formulate our extension theorem for r -signotopes, we introduce some notation and discuss the relation between pseudoline arrangements and 3-signotopes (in Section 1.1). This leads to a reformulation of Levi’s extension lemma which will be investigated in the context of signotopes of odd rank in Section 1.2.

1.1 Signotopes

Signotopes are a combinatorial structure generalizing permutations and *simple* pseudoline arrangements (i.e., no three pseudolines cross in a common point). For $r \geq 1$ a *signotope* of rank r (short: r -signotope) on n elements is a mapping σ from r -element subsets (r -subsets) of $[n]$ to $+$ or $-$, i.e., $\sigma : \binom{[n]}{r} \rightarrow \{+, -\}$ such that for every $(r + 1)$ -subset $X = \{x_1, \dots, x_{r+1}\}$ (r -packet) of $[n]$ with $x_1 < x_2 < \dots < x_{r+1}$ there is at most one sign change in the sequence

$$\sigma(X \setminus \{x_1\}), \sigma(X \setminus \{x_2\}), \dots, \sigma(X \setminus \{x_{r+1}\}).$$

Note that this sequence lists the signs of all induced r -subsets of X in reverse lexicographic order. For 3-signotopes, the following 8 sign patterns on 4-subsets are allowed:

$$++++, +++-, ++--, +- --, ----, ---+, -- ++, - + + +.$$

For sake of readability, we write $X = (x_1, \dots, x_t)$ to denote a t -subset of $[n]$ with sorted elements $x_1 < x_2 < \dots < x_t$. For such an X we denote by $X_j = (x_1, \dots, x_{j-1}, x_{j+1}, \dots, x_t)$

the set without x_j . With the convention $- < +$, the condition about sign changes in r -signotopes can be written as a monotonicity condition for r -packets $X = (x_1, \dots, x_{r+1})$:

$$\sigma(X_1) \leq \sigma(X_2) \leq \dots \leq \sigma(X_{r+1}) \quad \text{or} \quad \sigma(X_1) \geq \sigma(X_2) \geq \dots \geq \sigma(X_{r+1}).$$

It is well-known that every arrangement of pseudolines is isomorphic to an arrangement of x -monotone pseudolines [11]. In such a representation, we label the pseudolines from top to bottom on the left by $1, \dots, n$. Since two pseudolines cross exactly once, the pseudolines appear in reversed order on the right. Now the corresponding 3-signotope σ is obtained as follows: The sign of $\sigma(a, b, c)$ for $a < b < c$ indicates the orientation of the triangle formed by the pseudolines a, b, c (see Figure 1). If the crossing of a and c is below b , it is $\sigma(a, b, c) = +$ and if the crossing of a and c is above b , it is $\sigma(a, b, c) = -$. In the following we identify the crossings with the elements which cross, i.e. for 3-signotopes crossings are subsets of size 2. The 3-signotope σ gives information about the partial order of the crossings from left to right. If $\sigma(a, b, c) = +$ it holds $ab \prec ac \prec bc$ and if $\sigma(a, b, c) = -$ it is $bc \prec ac \prec ab$.



■ **Figure 1** Connection between pseudoline arrangements and 3-signotopes.

Felsner and Weil [9] showed that rank 3 signotopes are in bijection with simple pseudoline arrangements in \mathbb{R}^2 with a fixed top cell. For $r \geq 4$, r -signotopes correspond to special pseudohyperplane arrangements in \mathbb{R}^{r-1} , i.e., they are a subclass of oriented matroids of rank r . A geometric representation of r -signotopes in the plane is presented in [16] (see also [3] for the rank 3 case).

1.2 An extension theorem for signotopes

In Levi’s extension lemma for pseudoline arrangements, each of the two prescribed points can either lie in a cell of the arrangement, on one pseudoline, or be the crossing point of two pseudolines. To formulate an extension lemma in terms of 3-signotopes we restrict our considerations to simple pseudoline arrangements and to crossing points as prescribed points. Since the extending pseudoline passes through the two prescribed crossing points, the extension yields a non-simple arrangement. However, by perturbing the extending pseudoline at the non-simple crossing points, we arrive at a simple arrangement, see Figure 2.



■ **Figure 2** Perturbing an extending pseudoline at the two non-simple crossing points.

A perturbation at a prescribed crossing together with the new inserted pseudoline yields a *triangular cell* incident to the crossing. This cell is bounded by the two pseudolines

17:4 An Extension Theorem for Signotopes

defining the crossing and the extending pseudoline. Triangular cells play an important role in the study of pseudoline arrangements, since it is possible to change the orientation of a triangle by moving one of the pseudolines over the crossing of the two others. Such a local perturbation is called a *triangle flip*, it does not change the orientation of any other triangle in the arrangement. The triangular cells of the arrangement represented by a 3-signotope σ are in one to one correspondence with 3-subsets such that if we change the sign of this 3-subset in σ we obtain a new signotope σ' . We call such a 3-subset a *fliple*. The notion of fliples generalizes to higher ranks. In an r -signotope σ on $[n]$, an r -subset $X \subseteq [n]$ is a *fliple* if both assignments $+$ and $-$ to $\sigma(X)$ result in a signotope. It is worth noting that fliples in signotopes are the analogon of mutations in oriented matroids. While every signotope contains at least two fliples [9], it remains a central open problem in combinatorial geometry to decide whether every uniform oriented matroid contains a mutation [8, Chapter 7.3].

Let \mathcal{A} be an arrangement of pseudolines, which are labeled $1, \dots, n$ from top to bottom on the left. When applying Levi's extension lemma to extend \mathcal{A} the left endpoint of the extending line ℓ will be between two consecutive endpoints of pseudolines of \mathcal{A} . To re-establish the properties of the labeling, we have to set the label of ℓ accordingly and increase the label of every pseudoline that starts below ℓ by one. To cope with this relabeling-issue in terms of signotopes, we introduce the following notion. For $k \in [n]$ and a subset X of $[n]$, we define

$$X \downarrow_k = \{x \mid x \in X, x < k\} \cup \{x - 1 \mid x \in X, x > k\}.$$

Note that the cardinality of X and $X \downarrow_k$ is the same if and only if $k \notin X$. For an r -signotope σ on $[n]$, we define the *deletion* of an element $k \in [n]$ as $\sigma \downarrow_k$ by

$$\sigma \downarrow_k(X \downarrow_k) := \sigma(X)$$

for all r -sets $X \subseteq [n]$ with $k \notin X$. This is an r -signotope on $[n - 1]$ because each r -packet has been an r -packet for σ .

► **Definition.** An r -signotope σ on $[n]$ is t -extendable if for all pairwise disjoint $(r - 1)$ -subsets $I_1, \dots, I_t \in \binom{[n]}{r-1}$, there exists $k \in [n + 1]$ and an r -signotope σ^* on $[n + 1]$ with fliples I_1^*, \dots, I_t^* such that $\sigma^* \downarrow_k = \sigma$, and $I_j^* \downarrow_k = I_j$ for all $j = 1, \dots, t$. Hence the element k extends σ to σ^* .

Note that a t -extendable r -signotope on $n \geq (r - 1)t$ elements is clearly $(t - 1)$ -extendable. While the 1-extendability is a simple exercise¹, the first interesting part is the 2-extendability, which we investigate in this paper.

► **Theorem 1.2** (An extension theorem for signotopes of odd rank). *For every odd rank $r \geq 3$, every r -signotope is 2-extendable.*

Surprisingly, our proof of Theorem 1.2 (see Section 3) generalizes to the more general setting, where the $(r - 1)$ -subsets I and J , which are fliples in the extension, intersect.

► **Corollary 1.3.** *Let σ be an r -signotope on $[n]$, I and J be two $(r - 1)$ -subsets of $[n]$ such that $|I \cap J| + r$ is odd. Then σ is extendable to an r -signotope σ^* on $[n + 1]$ with fliples I^* , J^* and an extending element $k \in [n + 1]$ such that $\sigma^* \downarrow_k = \sigma$, and $I^* \downarrow_k = I$, and $J^* \downarrow_k = J$.*

¹ For the sake of completeness, we give a proof of 1-extendability in Corollary 3.2 which uses more evolved techniques.

Despite the restrictions to simple arrangements and crossing points as prescribed points we can derive Levi's extension lemma (Theorem 1.1) in its full generality with little extra work from Theorem 1.2. Details are deferred to Section 5.

The statement of Theorem 1.2 applies only to signotopes of odd rank. This is not just a defect of our proof because signotopes in even rank indeed behave differently. For ranks $r = 4, 6, 8, 10, 12$ we found signotopes on $n = 2r$ elements, which are not 2-extendable. The examples and the source code to verify their correctness are available as supplemental data [4]; see Section 4 for details. Based on these examples, we dare to conjecture:

► **Conjecture 1.4** (No extension theorem for signotopes of even rank). *For every even rank $r \geq 4$, there is an r -signotope which is not 2-extendable.*

1.3 Signotopes as a rich subclass of oriented matroids

It is well known that the number of oriented matroids of rank r on n elements is $2^{\Theta(n^{r-1})}$ [8, Corollary 7.4.3]. As shown by Balko [2, Theorem 3], r -signotopes are a rich subclass of oriented matroids of rank r ; see the full version [5] for a shorter proof of the following proposition.

► **Proposition 1.5** (Balko [2]). *For $r \geq 3$, the number of r -signotopes on $[n]$ is $2^{\Theta(n^{r-1})}$.*

In ranks 1 and 2 there are 2^n and $n!$ signotopes on $[n]$, respectively. Rank 1 signotopes are mappings from $[n]$ to $\{+, -\}$ without any additional property and 2-signotopes are permutations. For rank $r \geq 3$, the precise number of r -signotopes on $[n]$ has been computed for small values of r and n ; see A6245 (rank 3) and A60595 to A60601 (rank 4 to rank 10) on the OEIS [17].

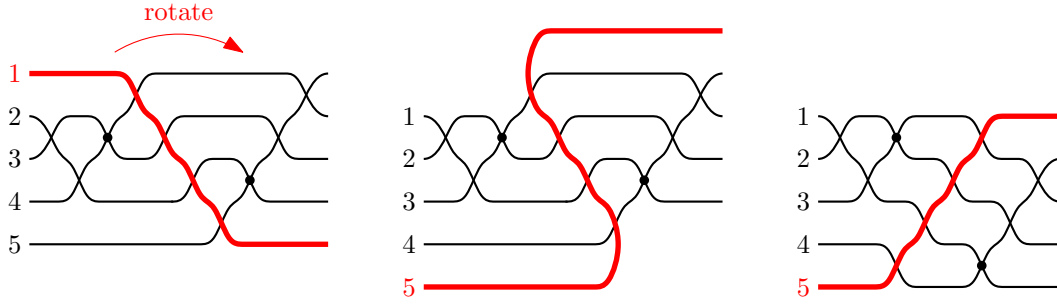
2 Preliminaries

We now prepare for the proof of Theorem 1.2. As discussed in Section 1.1, signotopes of rank 3 can be represented by an arrangement with x -monotone pseudolines. The order of the crossings from left to right gives a partial order on the 2-subsets. In general, r -signotopes can be represented by a sweepable arrangement of pseudohyperplanes in \mathbb{R}^{r-1} , which similarly allows to define a partial order on $(r-1)$ -subsets which correspond to the crossings of $r-1$ elements. This partial order is combinatorially defined as follows. For an r -signotope σ and every r -subset $X = (x_1, \dots, x_r)$ define:

$$\begin{aligned} X_1 \succ X_2 \succ \dots \succ X_r & \text{ if } \sigma(x_1, \dots, x_r) = +, & \text{ and} \\ X_1 \prec X_2 \prec \dots \prec X_r & \text{ if } \sigma(x_1, \dots, x_r) = -. \end{aligned}$$

Recall that we use the convention $x_1 \leq \dots \leq x_r$ and $X_i = X \setminus \{x_i\}$. By taking the transitive closure of all relations obtained from r -subsets, we obtain a partial order on the $(r-1)$ -subsets corresponding to σ [9, Lemma 10].

If we rotate an arrangement of pseudolines, i.e., we choose another unbounded cell as the top cell, we get a pseudoline arrangement with the same cell structure. However, the signotope does not stay the same. If we rotate only a single pseudoline, then the orientation of the triangle spanned by 3 pseudolines stays the same if and only if the rotated pseudoline is not involved (see for example the triangle spanned by $\{2, 3, 4\}$ in the left-hand side arrangement, resp. $\{1, 2, 3\}$ in the right-hand side arrangement in Figure 3). When rotating *clockwise*, the first element of σ becomes the last one in the rotated signotope σ_{rot} . In terms of the



■ **Figure 3** An illustration of a clockwise rotation of pseudolines. The rotated pseudoline is highlighted red.

3-signotope σ the signs of the rotated signotope σ_{rot} are: $\sigma_{\text{rot}}(a, b, c) = \sigma(a + 1, b + 1, c + 1)$ if $c \neq n$ and $\sigma_{\text{rot}}(a, b, n) = -\sigma(1, a + 1, b + 1)$.

In general, we define the *clockwise rotated* signotope σ_{rot} of a given r -signotope σ as:

$$\sigma_{\text{rot}}(x_1, \dots, x_r) = \begin{cases} -\sigma(1, x_1 + 1, \dots, x_{r-1} + 1) & \text{if } x_1 < x_2 < \dots < x_r = n, \\ \sigma(x_1 + 1, \dots, x_r + 1) & \text{if } x_1 < x_2 < \dots < x_r < n. \end{cases}$$

Here we use the usual convention $-\cdot + = -$ and $-\cdot - = +$. To keep track of the index shift caused by a clockwise rotation, we define $x_{\text{rot}} = x - 1$ if $x \neq 1$ and $1_{\text{rot}} = n$, and

$$X_{\text{rot}} = \{x_{\text{rot}} : x \in X\} = \begin{cases} (x_1 - 1, x_2 - 1, \dots, x_k - 1) & \text{if } x_1 > 1; \\ (x_2 - 1, \dots, x_k - 1, n) & \text{if } x_1 = 1 \end{cases}$$

for any subset $X = (x_1, \dots, x_k)$ of $[n]$ with $x_1 < \dots < x_k$. Note that this allows us to write $\sigma_{\text{rot}}(X_{\text{rot}}) = \sigma(X)$ if $1 \notin X$ and $\sigma_{\text{rot}}(X_{\text{rot}}) = -\sigma(X)$ if $1 \in X$.

As the following lemmas show, this is indeed an r -signotope, which moreover has essentially the same fliples. The proofs and further properties are deferred to the full version [5].

► **Lemma 2.1.** *Let σ be an r -signotope on $[n]$. Then σ_{rot} is an r -signotope on $[n]$.*

► **Lemma 2.2.** *Let σ be an r -signotope and let F be a fliple of σ . Then F_{rot} is a fliple in the clockwise rotated signotope σ_{rot} .*

3 Extension theorem for signotopes

In this section, we give a proof for the extension theorem for signotopes of odd rank. The central ingredient of our proof is as follows. If σ is an r -signotope on $[n]$ and the prescribed two $(r - 1)$ -sets I and J are incomparable in the partial order associated with σ (see Section 2), then σ is extendable by a “last” element $n + 1$ such that $I \cup \{n + 1\}$ and $J \cup \{n + 1\}$ are fliples in the extension. Figure 2 gives an illustration for the rank 3 case. More abstractly we can extend the signotope when there is a down-set in the partial order on $(r - 1)$ -sets which has I and J as maximal elements. A *down-set* of a partial order $(\mathcal{P}, <)$ is a subset $\mathcal{D} \subseteq \mathcal{P}$ such that for all $p \in \mathcal{P}$ and $d \in \mathcal{D}$ with $p \preceq d$ it holds $p \in \mathcal{D}$. Similarly, an *up-set* is a subset $\mathcal{U} \subseteq \mathcal{P}$ such that for all $p \in \mathcal{P}$ and $u \in \mathcal{U}$ with $p \succeq u$ it holds $p \in \mathcal{U}$.

► **Proposition 3.1** (Extension for incomparable elements). *Let $(\mathcal{P}, <)$ be the partial order on $(r - 1)$ -sets corresponding to an r -signotope σ on $[n]$. For every down-set $\mathcal{D} \subseteq \mathcal{P}$ there exists an r -signotope σ^* on $[n + 1]$ such that all r -subsets of the form $M \cup \{n + 1\}$, where M is a maximal element of \mathcal{D} , are fliples of σ^* and $\sigma^*_{\downarrow n+1} = \sigma$.*

Proof. Define the extended r -signotope σ^* on $[n + 1]$ as follows:

$$\sigma^*(x_1, \dots, x_r) = \begin{cases} \sigma(x_1, \dots, x_r) & \text{if } x_1, \dots, x_r \in [n]; \\ + & \text{if } x_r = n + 1 \text{ and } \{x_1, \dots, x_{r-1}\} \in \mathcal{D}; \\ - & \text{if } x_r = n + 1 \text{ and } \{x_1, \dots, x_{r-1}\} \notin \mathcal{D}. \end{cases}$$

First we show that σ^* is an r -signotope on $[n + 1]$. Consider an r -packet $X = (x_1, \dots, x_{r+1})$. We need to show that the sequence

$$\sigma^*(X_1), \sigma^*(X_2), \dots, \sigma^*(X_{r+1})$$

has at most one sign change.

If $x_{r+1} \leq n$, then all signs on the considered r -subsets are the same as for σ . Since σ is an r -signotope, there is at most one sign change in the sequence.

In the other case, we have $x_{r+1} = n + 1$. For all $j \leq r$ we have $n + 1 \in X_j$. Furthermore, $\sigma^*(X_{r+1}) = \sigma(X_{r+1})$ because $n + 1 \notin X_{r+1}$. We consider two cases. First, if $\sigma(X_{r+1}) = +$ we have by definition of the partial order

$$X \setminus \{x_{r+1}, x_i\} \succ X \setminus \{x_{r+1}, x_j\} \quad \text{for } i < j.$$

By the property of a down-set this means that, whenever $X \setminus \{x_{r+1}, x_i\} \in \mathcal{D}$, we also have $X \setminus \{x_{r+1}, x_j\} \in \mathcal{D}$ for $i < j$. Let i^* be the smallest integer such that $X \setminus \{x_{r+1}, x_{i^*}\} \in \mathcal{D}$. Then by definition of σ^* we have $\sigma^*(X_j) = -$ for all $j < i^*$ and $\sigma^*(X_j) = +$ for all $j \geq i^*$.

Similar arguments apply if $\sigma(X_{r+1}) = -$. Then we have

$$X \setminus \{x_{r+1}, x_i\} \prec X \setminus \{x_{r+1}, x_j\} \quad \text{for } i < j.$$

This time let i^* be the smallest integer such that $X \setminus \{x_{r+1}, x_{i^*}\} \notin \mathcal{D}$. Then by definition of σ^* we have $\sigma^*(X_j) = +$ for all $j \leq i^*$ and $\sigma^*(X_j) = -$ for all $j > i^*$.

Let M be a maximal element of the down-set \mathcal{D} . By the analysis above it follows that $M \cup \{n + 1\}$ is adjacent to a sign change in each packet in which it is contained. Hence it is a fliple. ◀

From this proposition it follows that for all $r \geq 2$ all r -signotopes are 1-extendable. Moreover the 1-extension contains the extending element at the last position.

▶ **Corollary 3.2** (1-extendability). *For $r \geq 2$ let σ be an r -signotope on $[n]$ and I an $(r - 1)$ -subset. Then there is an extending r -signotope σ^* on $[n + 1]$ elements such that $I \cup \{n + 1\}$ is a fliple and $\sigma^* \downarrow_{n+1} = \sigma$.*

The following two propositions show that, for odd rank, we can always find a rotation of the corresponding signotope such that the two prescribed $(r - 1)$ -subsets are incomparable. We can then use Proposition 3.1 to define an extension.

▶ **Proposition 3.3.** *Let σ be an r -signotope on $[n]$. For two $(r - 1)$ -subsets I, J with $I \prec J$ and $1 \notin I \cap J$, it holds I_{rot} and J_{rot} are incomparable in \prec_{rot} or $I_{\text{rot}} \prec_{\text{rot}} J_{\text{rot}}$.*

The proof of Proposition 3.3 needs some more structural properties of the partial order and its interaction with the rotation. The details are deferred to the full version [5].

▶ **Proposition 3.4.** *Let $r \geq 3$ be an odd integer, let σ be an r -signotope on $[n]$ and let I, J be two disjoint $(r - 1)$ -subsets. After at most $n - 1$ clockwise rotations, σ, I , and J are transformed into σ', I' , and J' , resp., such that I' and J' are incomparable in the partial order \prec' corresponding to σ' .*

17:8 An Extension Theorem for Signotopes

Proof. Assume I and J are comparable in the partial order \prec corresponding to the r -signotope σ with $I \prec J$, otherwise we are done. We show that after n clockwise rotations, all signs of σ are reversed. Hence the partial order \prec' corresponding to the (possible multiple times) rotated signotope σ' is the reversed relation to \prec .

The sign of an r -subset (z_1, \dots, z_r) changes from $+$ to $-$ or vice versa if and only if the rotated element is contained in (z_1, \dots, z_r) , i.e., if we rotate z_1 . Hence after rotating n times in total every z_i was rotated and thus the sign of an r -subset changes exactly r times. Since r is odd, the sign after rotating n times is opposite. The obtained signotope σ' is the reverse of the original signotope σ and the corresponding partial order is also reversed.

Furthermore we cannot reverse the order of two disjoint $(r-1)$ -sets in one rotation as shown in Proposition 3.3. Hence there will be a moment where the two disjoint sets are incomparable. \blacktriangleleft

Although the following lemma is trivial in the setting of pseudoline arrangements, we need to prove it in the context of general r -signotopes. We show that the extension of a rotated signotope when rotated back does contain the original signotope. To show this we need to investigate the interaction between the rotation and deletion of elements.

► Lemma 3.5. *Let σ be an r -signotope on $[n]$ and $x \in [n]$. Then it is $\sigma_{\text{rot}\downarrow n} = \sigma_{\downarrow 1}$ and $\sigma_{\text{rot}\downarrow x_{\text{rot}}} = (\sigma_{\downarrow x})_{\text{rot}}$ for $x \neq 1$.*

Proof. Because of the index shift it does not matter whether we delete the first element or we rotate σ such that in the first element becomes the last and delete the last element in this rotated signotope. Hence the first part $\sigma_{\text{rot}\downarrow n} = \sigma_{\downarrow 1}$ holds.

Now assume $x \neq 1$ which implies $x_{\text{rot}} \neq n$. Both mappings are r -signotopes on $[n-1]$. We need to check whether they map to the same signs. Let X be an r -subset of $[n-1]$ and let X^* be an r -subset of $[n]$ with $x_{\text{rot}} \notin X^*$ and $X^*_{\downarrow x_{\text{rot}}} = X$. We obtain

$$\sigma_{\text{rot}\downarrow x_{\text{rot}}}(X) = \sigma_{\text{rot}\downarrow x_{\text{rot}}}(X^*_{\downarrow x_{\text{rot}}}) = \sigma_{\text{rot}}(X^*).$$

We will now rewrite the term to get the statement. Recall that rotating an r -signotope on n elements exactly $2n$ times results in the original signotope. Hence rotating $2n-1$ times corresponds to a counterclockwise rotation, i.e., the inverse operation of a clockwise rotation. We denote this counterclockwise rotation by $\text{rot}(-1)$. Since $x_{\text{rot}} \notin X^*$, we have $x \notin (X^*)_{\text{rot}(-1)}$. By definition it is

$$\sigma_{\text{rot}}(X^*) = \varepsilon \cdot \sigma((X^*)_{\text{rot}(-1)}) = \varepsilon \cdot \sigma_{\downarrow x}(((X^*)_{\text{rot}(-1)})_{\downarrow x}) = \varepsilon \cdot \sigma_{\downarrow x}(X_{\text{rot}(-1)}) = (\sigma_{\downarrow x})_{\text{rot}}(X),$$

where the sign $\varepsilon = +$ (resp. $\varepsilon = -$) if $n \notin X^*$ (resp. $n \in X^*$). Note that $n \in X^*$ is equivalent to $1 \in X_{\text{rot}(-1)}$ for $x \neq 1$. This completes the proof of the lemma. \blacktriangleleft

With Proposition 3.1, Proposition 3.4 and Lemma 3.5 we are now ready to prove Theorem 1.2.

3.1 Proof of Theorem 1.2

Let σ be an r -signotope on $[n]$ and let I, J be a pair of disjoint $(r-1)$ -subsets. By Proposition 3.4 there exists $k \in \{0, \dots, n-1\}$ such that the k -fold rotated $(r-1)$ -subsets $I_{\text{rot}(k)}, J_{\text{rot}(k)}$ are incomparable in the k -fold rotated signotope $\sigma_{\text{rot}(k)}$.

To extend the signotope $\sigma_{\text{rot}(k)}$, we use the down-set \mathcal{D} consisting of $I_{\text{rot}(k)}, J_{\text{rot}(k)}$, and all $(r-1)$ -subsets below. In this down-set $I_{\text{rot}(k)}$ and $J_{\text{rot}(k)}$ are maximal elements since they are incomparable. Hence we can apply Proposition 3.1 in order to add a new element at

position $n + 1$ in the rotated signotope $\sigma_{\text{rot}(k)}$ such that $I_{\text{rot}(k)} \cup \{n + 1\}$ and $J_{\text{rot}(k)} \cup \{n + 1\}$ are flipes. The extended signotope is denoted by $\sigma_{\text{rot}(k)}^*$ and fulfills $\sigma_{\text{rot}(k)}^* \downarrow_{n+1} = \sigma_{\text{rot}(k)}$.

Finally, we need to find a rotation of $\sigma_{\text{rot}(k)}^*$ which contains the original signotope σ . For this we perform $k + 1$ counterclockwise rotations (or equivalently, $2n + 1 - k$ clockwise rotations) and denote the so-obtained signotope by σ^* . Note that we perform $k + 1$ counterclockwise rotations since the newly added element needs to be rotated and the k -fold clockwise rotation needs to be undone. After $k + 1$ counterclockwise rotations, the added element $n + 1$ in $\sigma_{\text{rot}(k)}^*$ becomes the element $k + 1$ in σ^* . It remains to show that $\sigma^* \downarrow_{k+1} = \sigma$.

After the first counterclockwise rotation, the added element $n + 1$ in $\sigma_{\text{rot}(k)}^*$ becomes the first element 1 in $(\sigma_{\text{rot}(k)}^*)_{\text{rot}(-1)}$. By Lemma 3.5 it holds $((\sigma_{\text{rot}(k)}^*)_{\text{rot}(-1)}) \downarrow_1 = (\sigma_{\text{rot}(k)}^*) \downarrow_{n+1} = \sigma_{\text{rot}(k)}$. After additional k counterclockwise rotations, the added element $n + 1$ in $\sigma_{\text{rot}(k)}^*$ becomes the element $k + 1$ in σ^* . Furthermore $I \cup \{k + 1\}$ and $J \cup \{k + 1\}$ are flipes of σ^* by Lemma 2.2. Since we do not rotate the extending element, applying the second part of Lemma 3.5 multiple times shows $((\sigma_{\text{rot}(k)}^*)_{\text{rot}(-1)}) \downarrow_1 = (\sigma^* \downarrow_{k+1})_{\text{rot}(k)}$. Together with the previous equation this shows $\sigma_{\text{rot}(k)} = (\sigma^* \downarrow_{k+1})_{\text{rot}(k)}$, which further implies $\sigma = \sigma^* \downarrow_{k+1}$. Hence we obtain the signotope σ when deleting $k + 1$ from σ^* . This completes the proof of Theorem 1.2.

3.2 Proof of Corollary 1.3

To prove Corollary 1.3, we proceed similar as in the proof of Theorem 1.2. By Proposition 3.1, it suffices to show that after some rotations the $(r - 1)$ -subsets corresponding to I and J are incomparable.

Let $s = |I \cap J|$. Since Theorem 1.2 covers the case $s = 0$, we may assume $s \geq 1$. We consider the following two cases.

First, assume that r is odd and s is even. For odd rank r , we have already seen that after n rotations, the signotope is reversed and hence the corresponding partial order is reversed. For even s , the relation between I and J is reversed s times (whenever we rotate one element $x \in I \cap J$). These are the only s times where we reverse the order in one single rotation. Since s is even and the order is reversed after n rotations, the corresponding $(r - 1)$ -subsets must be incomparable in between.

Second, assume that r is even and s is odd. For even rank r , the n -fold rotation leaves the signotope unchanged and hence also the partial orders are the same. Since s is odd, we reverse the orientation of I and J exactly s times in a single rotation step. Hence they must be incomparable in between.

The statement now follows from Proposition 3.1 and Lemma 3.5 similar as in the proof of Theorem 3.1. This completes the proof of Corollary 1.3.

4 Examples in even rank: SAT attack and properties

Since the proof for the extension theorem (Theorem 1.2) applies only for odd ranks, we had to investigate even ranks in a different manner. For rank 4, we used computer assistance to enumerate all signotopes and then tested 2-extendability for each signotope. On 6 and 7 elements all 4-signotopes are 2-extendable. On 8 elements we found non-2-extendable 4-signotopes. For both, the enumeration and the 2-extendability test, we modeled SAT instances which were then solved using the python interfaces `pycosat` [20] and `pysat` [13] to run the SAT solver `picosat`, version 965, [6] and `cadical`, version 1.0.3 [7], respectively.

Using this two-level-SAT approach we managed to find the first examples of 4-signotopes which are not 2-extendable. In order to keep symmetries and similarities of our nicely structured example of rank 4, we restricted our search space to examples in rank r on

17:10 An Extension Theorem for Signotopes

$2r$ elements. While for rank 4 all signotopes on 8 elements can be enumerated within a few seconds, the complete enumeration in higher ranks is unpractical as the number of r -signotopes on $2r$ elements grows faster than doubly exponential in r (cf. Proposition 1.5). Hence, to be able to approach higher ranks, we further analyzed the structure of our non-2-extendable rank 4 examples together with an analyze of the already found rank 6 examples. These made it possible to find a recursive construction. See Section 4.3 for more details.

With the observed properties as additional constraints, we further restricted the search space so that only “reasonable” candidates were enumerated. Under these restrictions, we managed to find examples for rank 6, 8, 10, and 12 which are not 2-extendable.

4.1 SAT model for enumeration

To encode r -signotope on n elements, we proceed as following. We use Boolean variables S_X for every $X \in \binom{[n]}{r}$ to indicate whether $\sigma(X) = +$. To ensure that these variables model a valid signotope, we add constraints which ensure that for every r -packet $Y = \{y_1, \dots, y_{r+1}\} \in \binom{[n]}{r+1}$ there is at most one sign-change in the sequence

$$\sigma(Y_1), \dots, \sigma(Y_{r+1}).$$

More precisely, since there are exactly $2r + 2$ possible assignment of this sequence, we introduce auxiliary variables $T_{Y,t}$ for $t \in \{1, \dots, 2r + 2\}$ to indicate which of the assignments applies.²

Next we introduce auxiliary variables $F_{X,Y}$ for every r -packet $Y \in \binom{[n]}{r+1}$ and every r -tuple $X \in \binom{[n]}{r}$ to indicate whether X is a fliple when σ is restricted to Y . This is done in a similar fashion as for the S_X variables. Using the $F_{X,Y}$ variables, we can assert the variables $F_X = \bigvee_{Y \in \binom{[n]}{r+1}: X \subset Y} F_{X,Y}$ for every $X \in \binom{[n]}{r}$ to indicate whether X forms a fliple. Last but not least, we introduce variables $L_{X,k}$ to indicate whether X is the k -th fliple. This will allow us to enumerate only configurations with a prescribed number of fliples.

4.2 SAT model for testing 2-extendability

We are now ready to formulate a SAT instance to decide whether a given signotope σ on $[n]$ and given disjoint $(r - 1)$ -tuples I, J can be extended by an additional element $n + 1$ such that $I \cup \{n + 1\}$ and $J \cup \{n + 1\}$ are fliples in the extension σ^* . This is sufficient to test extendability since whenever there is an extension, there is a rotation such that the signotope is extendable by an element at the last position. As in Section 4.1, we create a SAT instance to find an $(n + 1)$ -element signotope but we add constraints to fix σ and to assert that $I \cup \{n + 1\}$ and $J \cup \{n + 1\}$ are fliples in σ^* .

For a given signotope σ on elements $[n]$ we can now iterate over all disjoint $(r - 1)$ -tuples I, J and test whether there is a rotation of σ and I, J such that in the extension σ^* by the element $n + 1$ the r -tuples $I \cup \{n + 1\}$ and $J \cup \{n + 1\}$ are fliples. If for some I, J no such rotations exists, we have certified that σ is not 2-extendable.

² Alternatively one can assert $\neg S_{Y_i} \vee S_{Y_j} \vee \neg S_{Y_k}$ and $S_{Y_i} \vee \neg S_{Y_j} \vee S_{Y_k}$ for every Y and $1 \leq i < j < k \leq r + 1$. Even though this approach does not require auxiliary variables to indicate the types of $(r + 1)$ -tuples, we need these auxiliary variables to assign the variables for fliples later anyhow.

4.3 Structure of the examples supporting Conjecture 1.4

In order to find the first witnessing examples for Conjecture 1.4 in rank 4, we used the two-step SAT approach as described in Sections 4.1 and 4.2. To make investigations in higher ranks, we had to get a better understanding of the examples found in rank 4. Hence we filtered those with regularities and symmetries to come up with a generalization of the observed properties and analyzed their structure. Our aim was to find a relation between examples in different ranks, for example using projection and deletion arguments. For this we investigated the structure of our rank 4 examples together with some already found rank 6 examples.

One of the first and crucial observations was that there exist signotopes such that for every choice of even indices $I \subset E_r := \{2, 4, \dots, 2r\}$ and every choice of odd indices $J \subset O_r := \{1, 3, \dots, 2r - 1\}$ there is no such extension. In fact, for such examples it is sufficient to check $I = \{2, 4, \dots, 2r - 2\}$ and $J = \{1, 3, \dots, 2r - 3\}$ to verify the non-2-extendability. This observation not only allowed us to restrict the search space, but also to speed up the extendability-test by a factor of $\Theta(r^2)$ since not all pairs of $(r - 1)$ -tuples I, J need to be tested.

While we came up with further observations one by one over the time, we here give a summary of all the properties, which we desire from the examples in rank r with $n = 2r$ elements. In the following we denote by $X = (x_1, x_2, \dots, x_r)$ an r -tuple and use the notation $(-)^i = +$ if i is even and $(-)^i = -$ if i is odd.

- (a) $\sigma = \sigma_{\text{rot}(4)}$, where $\sigma_{\text{rot}(4)}$ is obtained by the 4-fold rotation of σ .
- (b) $\sigma(2, 4, \dots, 2r) = -$ and $\sigma(1, 3, \dots, 2r - 1) = +$.
- (c) If there is only one even or only one odd element in X , then the sign $\sigma(X)$ depends only on the position of that element in X . More specifically: If $e = x_i$ is the only even element in X , then $\sigma(X) = (-)^i$. If $o = x_i$ is the only odd element in X , then it is $\sigma(X) = (-)^{i+1}$.
- (d) If $x_1, \dots, x_i \in E_r$ and $x_{i+1}, \dots, x_r \in O_r$ with $2 \leq i \leq r - 2$, then the sign is $\sigma(X) = (-)^{i+1}$.
- (e) Let $x_1, \dots, x_i \in O_r$ and $x_{i+1}, \dots, x_r \in E_r$ for $2 \leq i < r - 2$. If $x_r < 2r$, then $\sigma(X) = -$. If $x_j = 2j$ for all $j = i + 1, \dots, r$, then $\sigma(X) = +$.

Furthermore, we fix the following set of 8 fliples for rank 4.

$$F_4 = \{(1, 3, 5, 7), (2, 4, 6, 8), (2, 3, 7, 8), (1, 3, 4, 8), \\ (1, 2, 4, 7), (3, 5, 6, 8), (4, 5, 7, 8), (3, 4, 6, 7)\}$$

Together with the 4-fold symmetry it is sufficient to mention only some of them:

$$\widehat{F}_4 = \{(1, 3, 5, 7), (2, 4, 6, 8), (4, 5, 7, 8), (3, 4, 6, 7), (1, 2, 4, 7)\}$$

In rank 4, there are only four signs which are not determined by the above properties:

$$(1, 3, 4, 8), \quad (4, 5, 7, 8), \quad (2, 3, 7, 8), \quad (3, 4, 6, 7)$$

By the 4-fold symmetry, the assignment of $(1, 3, 4, 8)$ also determines the sign of $(4, 5, 7, 8)$, and vice versa. The third and fourth tuple have a similar interaction. Hence, there are precisely 4 signotopes in rank 4 which fulfill the above properties. We fix one of the four configurations (the choice does not play a role) and refer to it as σ_4 in the following.

In order to find examples in higher ranks, we use the following property.

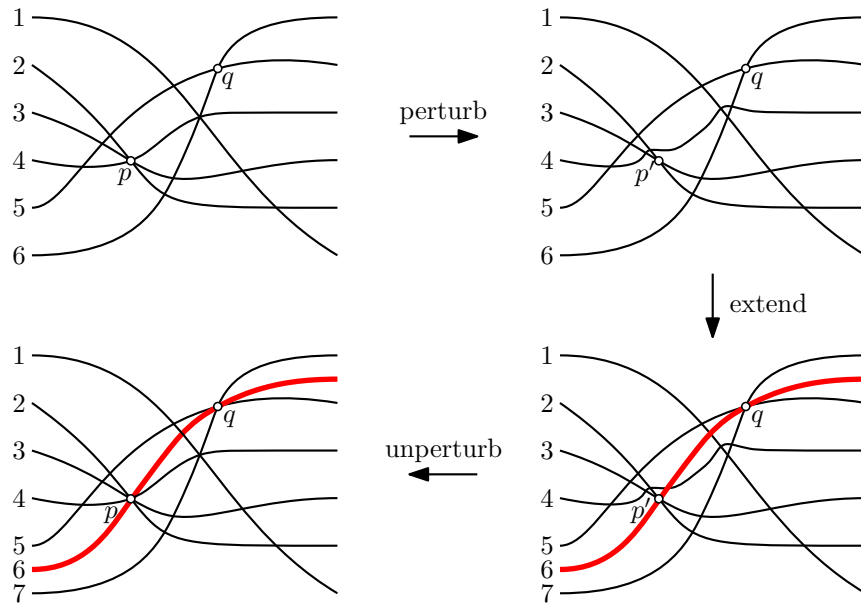


Figure 4 An illustration how Theorem 1.2 implies Levi’s extension lemma (Theorem 1.1). When perturbing the top-left arrangement, the multi-crossing point p (the intersection of 2, 3, and 4) is split into three simple crossing points, including the point p' (the intersection of 2 and 3). After the extension, we again contract these three crossing points to one multi-crossing point.

- (f) Let σ_{r-2} be an example of rank $r - 2$ on $2r - 4$ elements. For an r -tuple $X \subseteq [2r]$ with $1, 3 \notin X$ and $2, 4 \in X$, we define the sign

$$\sigma_r(X) = \sigma_{r-2}(X \downarrow_{1,2,3,4}),$$

where $X \downarrow_{1,2,3,4} = (((X \downarrow_4) \downarrow_3) \downarrow_2) \downarrow_1$ denotes the $(r - 2)$ tuple on $[2r - 4]$. Note that $X \downarrow_{1,2,3,4}$ is obtained by deleting the elements 2 and 4 from X and a further index shift by -2 caused by deleting³ 1 and 3, which are not contained in X .

Altogether, if we start with one example from rank 4 and recursively construct examples in higher ranks with the desired properties and further prescribe $(r/2)^2 + (r/2) + 2$ flips for rank r , it finally turned out that there is a unique example in each of the ranks $r = 6, 8, 10, 12$. All examples and the source code to verify their correctness are available as supplemental data [4].

In the future we hope to find an argument for the non-2-extendability based on the described properties and construct an infinite family of examples. Even though we conjecture that there is an infinite family, we want to clarify that we found examples in rank 4 and 6 which do not have the above properties and hence the assumptions might also be too strong.

5 Theorem 1.2 implies Levi’s extension lemma (Theorem 1.1)

As outlined in Section 1.2, it is sufficient to prove Levi’s extension lemma for simple arrangements of pseudolines and for crossing points as prescribed points. Given a non-simple arrangement, we can slightly perturb the multiple crossing points (as depicted in Figure 2)

³ Inspired by the language of oriented matroids, such an operation might be called *contraction*.

to obtain a simple arrangement. We obtain simplicial cells instead of the multiple crossings. This simple arrangement can then be extended, and each of the multiple crossing points of the original arrangement can again be obtained by contracting the simplicial cells to a point. Whenever a prescribed point lies on a pseudosegment or inside a cell, we can extend the arrangement through an adjacent crossing. By perturbing the extending pseudoline, we can ensure that the pseudoline passes through the originally prescribed point.

References

- 1 Alan Arroyo, Dan McQuillan, R. Bruce Richter, and Gelasio Salazar. Levi's lemma, pseudo-linear drawings of K_n , and empty triangles. *Journal of Graph Theory*, 87(4):443–459, 2018. doi:10.1002/jgt.22167.
- 2 Martin Balko. Ramsey numbers and monotone colorings. *Journal of Combinatorial Theory, Series A*, 163:34–58, 2019. doi:10.1016/j.jcta.2018.11.013.
- 3 Martin Balko, Radoslav Fulek, and Jan Kynčl. Crossing numbers and combinatorial characterization of monotone drawings of K_n . *Discrete & Computational Geometry*, 53(1):107–143, 2015. doi:10.1007/s00454-014-9644-z.
- 4 Helena Bergold, Stefan Felsner, and Manfred Scheucher. Supplemental source code and data. URL: https://page.math.tu-berlin.de/~scheuch/supplemental/signotopes/extend/sigext_suppl_socg.zip.
- 5 Helena Bergold, Stefan Felsner, and Manfred Scheucher. An extension theorem for signotopes. arXiv:2303.04079, 2023.
- 6 Armin Biere. PicoSAT essentials. *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, 4:75–97, 2008. URL: <http://satassociation.org/jsat/index.php/jsat/article/view/45>.
- 7 Armin Biere. CaDiCaL at the SAT Race 2019. In *Proc. of SAT Race 2019 – Solver and Benchmark Descriptions*, volume B-2019-1 of *Department of Computer Science Series*, pages 8–9. University of Helsinki, 2019. URL: <http://researchportal.helsinki.fi/en/publications/proceedings-of-sat-race-2019-solver-and-benchmark-descriptions>.
- 8 Anders Björner, Michel Las Vergnas, Bernd Sturmfels, Neil White, and Günter M. Ziegler. *Oriented Matroids*, volume 46 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 2 edition, 1999. doi:10.1017/CB09780511586507.
- 9 Stefan Felsner and Helmut Weil. Sweeps, Arrangements and Signotopes. *Discrete Applied Mathematics*, 109(1):67–94, 2001. doi:10.1016/S0166-218X(00)00232-8.
- 10 Jon Folkman and Jim Lawrence. Oriented matroids. *Journal of Combinatorial Theory, Series B*, 25(2):199–236, 1978. doi:10.1016/0095-8956(78)90039-4.
- 11 Jacob E. Goodman. Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Mathematics*, 32(1):27–35, 1980. doi:10.1016/0012-365X(80)90096-5.
- 12 Jacob E. Goodman and Richard Pollack. Three points do not determine a (pseudo-) plane. *Journal of Combinatorial Theory, Series A*, 31(2):215–218, 1981. doi:10.1016/0097-3165(81)90017-0.
- 13 Alexey Ignatiev, Antonio Morgado, and Joao Marques-Silva. PySAT: A Python toolkit for prototyping with SAT oracles. In *SAT*, pages 428–437, 2018. doi:10.1007/978-3-319-94144-8_26.
- 14 Friedrich Levi. Die Teilung der projektiven Ebene durch Gerade oder Pseudogerade. *Berichte über die Verhandlungen der Sächsischen Akademie der Wissenschaften zu Leipzig, Mathematisch-Physische Klasse*, 78:256–267, 1926.
- 15 Yuri I. Manin and Vadim V. Schechtman. Arrangements of hyperplanes, higher braid groups and higher bruhat orders. *Advanced Studies in Pure Mathematics*, pages 289–308, 1989.

17:14 An Extension Theorem for Signotopes

- 16 Hiroyuki Miyata. On combinatorial properties of points and polynomial curves. arXiv:1703.04963, 2021.
- 17 OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences. Published electronically at <http://oeis.org>.
- 18 Jürgen Richter-Gebert. Oriented matroids with few mutations. In *Discrete & Computational Geometry*, volume 10, pages 251–269. Springer, 1993. doi:10.1007/BF02573980.
- 19 Marcus Schaefer. A proof of Levi’s extension lemma. arXiv:1910.05388, 2019.
- 20 Ilan Schnell et al. pycosat: bindings to PicoSAT (a SAT solver). <http://pypi.python.org/pypi/pycosat>.
- 21 Günter M. Ziegler. Higher Bruhat orders and cyclic hyperplane arrangements. *Topology*, 32(2):259–279, 1993. doi:10.1016/0040-9383(93)90019-R.


Extending Orthogonal Planar Graph Drawings Is Fixed-Parameter Tractable

Sujoy Bhore ✉ 

Indian Institute of Technology Bombay, India

Robert Ganian ✉ 

Technische Universität Wien, Austria

Liana Khazaliya ✉ 

Technische Universität Wien, Austria

Fabrizio Montecchiani ✉ 

University of Perugia, Italy

Martin Nöllenburg ✉ 

Technische Universität Wien, Austria

Abstract

The task of finding an extension to a given partial drawing of a graph while adhering to constraints on the representation has been extensively studied in the literature, with well-known results providing efficient algorithms for fundamental representations such as planar and beyond-planar topological drawings. In this paper, we consider the extension problem for bend-minimal orthogonal drawings of planar graphs, which is among the most fundamental geometric graph drawing representations. While the problem was known to be NP-hard, it is natural to consider the case where only a small part of the graph is still to be drawn. Here, we establish the fixed-parameter tractability of the problem when parameterized by the size of the missing subgraph. Our algorithm is based on multiple novel ingredients which intertwine geometric and combinatorial arguments. These include the identification of a new graph representation of bend-equivalent regions for vertex placement in the plane, establishing a bound on the treewidth of this auxiliary graph, and a global point-grid that allows us to discretize the possible placement of bends and vertices into locally bounded subgrids for each of the above regions.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Graph algorithms

Keywords and phrases orthogonal drawings, bend minimization, extension problems, parameterized complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.18

Related Version *Full Version:* <http://arxiv.org/abs/2302.10046>

Funding *Robert Ganian:* Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029]; Austrian Science Fund (FWF) [Y1329].

Liana Khazaliya: Austrian Science Fund (FWF) [Y1329]; European Union's Horizon 2020 COFUND programme [LogiCS@TUWien, grant agreement No. 101034440].

Fabrizio Montecchiani: Department of Engineering, University of Perugia, grants RICBA21LG: Algoritmi, modelli e sistemi per la rappresentazione visuale di reti and RICBA22CB: Modelli, algoritmi e sistemi per la visualizzazione e l'analisi di grafi e reti.

Martin Nöllenburg: Vienna Science and Technology Fund (WWTF) [10.47379/ICT22029].



© Sujoy Bhore, Robert Ganian, Liana Khazaliya, Fabrizio Montecchiani, and Martin Nöllenburg;

licensed under Creative Commons License CC-BY 4.0

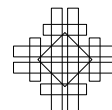
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 18; pp. 18:1–18:16



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Extending partial drawings of graphs while preserving certain desirable properties such as planarity is an algorithmic problem that received considerable attention in the last decade in graph theory, graph drawing, and computational geometry. Drawing extension problems are motivated, for instance, by visualizing networks, in which certain subgraphs represent important motifs that require a specific drawing, or by visualizing dynamic networks, in which new edges and vertices must be integrated in an existing, stable drawing. Generally speaking, we are given a graph G and a (typically connected) subgraph H of G with a drawing $\Gamma(H)$, which is called a *partial* drawing of G . The drawing $\Gamma(H)$ typically satisfies certain topological or geometric properties, e.g., planarity, upward planarity, or 1-planarity, and the goal of the corresponding extension problem is to extend $\Gamma(H)$ to a drawing $\Gamma(G)$ of the whole graph G (if possible) by inserting the missing vertices and edges into $\Gamma(H)$ while maintaining the required drawing properties.

A fundamental result in this line of research is the work of Angelini et al. [1], who showed that for planar graphs with a given partial planar drawing, the extension problem can be solved in linear time, thus matching the time complexity of unconstrained planarity testing. In fact, there is also a corresponding combinatorial characterization of planar graphs with extensible partial planar drawings via forbidden substructures [25]. In contrast to the above results, which consider topological graph embeddings, the planar drawing extension problem is NP-hard in its geometric variant, where one has to decide if a partial planar straight-line drawing $\Gamma(H)$ can be extended to a planar straight-line drawing of G [30].

In this paper, we study the geometric drawing extension problem arising in the context of one of the most fundamental graph drawing styles: orthogonal drawings [12, 16, 19, 29]. In a planar orthogonal drawing, edges are represented as polylines comprised of (one or more) horizontal and vertical segments with as few overall bends as possible, where edges are not allowed to intersect except at common endpoints. Orthogonal drawings find applications in various domains from VLSI and printed circuit board (PCB) design, to schematic network visualizations, e.g., UML diagrams in software engineering, argument maps, or flow charts.

Given the above, a key optimization goal in orthogonal drawings is bend minimization. This task is known to be NP-hard [22] when optimizing over all possible combinatorial embeddings of a given graph, but can be solved in polynomial time for a fixed combinatorial embedding using the network flow model of Tamassia [31]. Interestingly, the complexity of the bend minimization problem without a fixed embedding depends on the vertex degrees, which in the classical case of vertices being represented as points is naturally bounded by 4. If, however, the maximum vertex degree is 3, then there is a polynomial-time algorithm for bend minimization [4], and this result has recently been improved to linear time [15]; more generally, the problem is fixed-parameter tractable (FPT) in the number of degree-4 vertices [14]. In addition, it has been recently shown that the bend minimization problem is in XP (slice-wise polynomial) parameterized by the treewidth of the input graph [13].

Despite the general popularity of planar orthogonal graph drawings, the corresponding extension problem has only been considered recently [2]. While the authors of that paper showed that the existence of a planar orthogonal extension can be decided in linear time, the orthogonal bend-minimal drawing extension problem in general is easily seen to be NP-complete as it generalizes the case in which the pre-drawn part of the graph is empty [22]. Our paper addresses the parameterized complexity of the bend-minimal extension problem for planar orthogonal graph drawings under the most natural parameterization of the problem, which is the size of the subgraph that is still missing from the drawing. This parameter can

be assumed to be small in many applications, e.g., when extending drawings of dynamic graphs with few added edges and vertices, and has been used broadly in the study of previous topological drawing extension problems (see, e.g., [17, 18]).

Contributions. In this paper, we establish the fixed-parameter tractability of the BEND-MINIMAL ORTHOGONAL EXTENSION (BMOE) problem when parameterized by the size κ of the missing subgraph (see the formal problem statement in Section 2). A general difficulty we had to overcome on our way to obtain our fixed-parameter algorithm is the fact that while there have been numerous recent advances in the parameterized study of drawing extension problems [18, 21, 23], the specific drawing styles considered in those papers were primarily topological in nature, while for bend minimization the geometry of the instance is crucial. In order to overcome this difficulty, we develop a new set of tools summarized below.

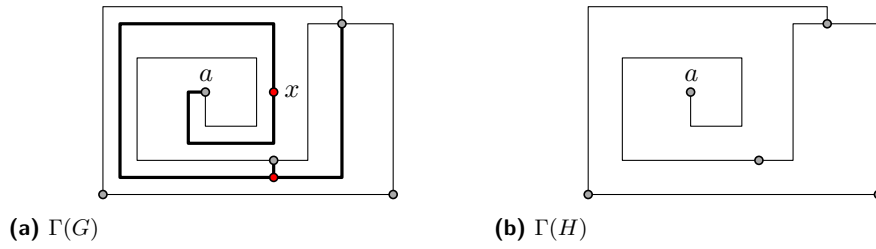
In Section 3, we make the first and simplest step towards fixed-parameter tractability of BMOE by applying an initial branching step to simplify the problem. This step allows us to reduce our target problem to BEND-MINIMAL ORTHOGONAL EXTENSION ON A FACE (F-BMOE), where the missing edges and vertices are drawn only in a marked face f and we have some additional information about how the edges are geometrically connected.

Next, in Section 4, we focus on solving an instance of F-BMOE. We show that certain parts of the marked face f are irrelevant and can be pruned away, and also use an involved argument to reduce the case of f being the outer face to the case of f being an inner face.

Once that is done, we enter the centerpiece of our approach in Section 5, where the aim is to obtain a suitable discretization of our instance. To this end, we split the face f into so-called *sectors*, which group together points that have the same “bend distances” to all of the connecting points on the boundary of f . Furthermore, we construct a *sector-grid* – a point-set such that each sector contains a bounded number of points from this set, and every bend-minimal extension can be modified to only use points from this set for all vertices and bends. While this latter result would make it easy to handle each individual sector by brute force, the issue is that the number of sectors can be very large, hindering tractability.

To deal with this obstacle, we capture the connections between sectors via a *sector graph* whose vertices are precisely the sectors and edges represent geometric adjacencies between sectors. Crucially, in Section 6 we show that the sector graph has treewidth bounded by a function of κ . This is non-trivial and relies on the previous application of the pruning step in Section 4. Having obtained this bound on the treewidth, the last step simply combines the already constructed sector grid with dynamic programming to solve F-BMOE (and hence also BMOE). It is perhaps worth pointing out the interesting contrast between the use of treewidth here as an implicit structural property of the sector graph – a crucial tool in our fixed-parameter algorithm – with the previously considered use of treewidth directly on the input graph – which is not known to lead to fixed-parameter tractability [13].

Related work. Several variants of drawing extension problems have been studied over the years. For instance, Chambers et al. [10] studied the problem of drawing a planar graph using straight-line edges with a prescribed convex polygon as the outer face, and proposed a method that produces drawings with polynomial area. Mchedlidze et al. [28] provide a characterization (which can be tested in linear time) to determine whether given a planar straight-line convex drawing of a biconnected subgraph G' of a planar graph G with a fixed planar embedding, this drawing can be extended to a planar straight-line drawing of G . Recently, Eiben et al. studied the problem of extending 1-planar drawings. While the problem was known to be NP-complete, they showed [18] that the problem is FPT when parameterized by the edge deletion distance. Later, in [17], they showed that the 1-planar



■ **Figure 1** An orthogonal drawing of (a) a graph G and (b) a subgraph H of G .

extension is polynomial-time solvable when the number of vertices and edges to be added to the partial drawing is bounded. Hamm and Hliněný also studied the parameterized complexity of the extension problem in the setting of crossing minimization [23].

Other types of extension problems have also been investigated, e.g., Da Lozzo et al. [27] studied the upward planarity extension problem, and showed that this is NP-complete even for very restricted settings. Brückner and Rutter [9] showed that the partial level planarity problem is NP-complete again in severely restricted settings. For non-planar graph drawings, it is even NP-hard to determine whether a single edge can be inserted into a simple partial drawing of the remaining graph, i.e., a drawing in which any two edges intersect in at most one point [3]. Extension problems have been investigated also for other types of graph representations, in particular for intersection representations such as circular arc graphs [20] or circle graphs [8]. In the context of bend-minimal planar orthogonal drawing extension, Angelini et al. showed that the problem remains NP-hard even when a planar embedding of the whole graph is provided in the input [2].

2 Preliminaries and Basic Tools

We assume familiarity with basic concepts in parameterized complexity theory, notably fixed-parameter tractability and treewidth [11], and with standard graph drawing terminology. Recall that a planar drawing $\Gamma(G)$ is *orthogonal* if each edge is a polyline consisting of horizontal and vertical segments. A *bend* in a polygonal chain representing an edge in $\Gamma(G)$ is a point shared by two consecutive segments of the chain. For instance, Figure 1a shows an orthogonal drawing of a graph G in which edge ax has three bends.

Problem Statement. Let G be a planar graph and H be a connected subgraph of G . We call the complement $X = V(G) \setminus V(H)$ the *missing vertex set* of G , and $E_X = E(G) \setminus E(H)$ the *missing edge set*. Let $\Gamma(H)$ be a planar orthogonal drawing of H . A planar orthogonal drawing $\Gamma(G)$ *extends* $\Gamma(H)$ if its restriction to the vertices and edges of H coincides with $\Gamma(H)$. Moreover, $\Gamma(G)$ is a β -extension of $\Gamma(H)$ if it extends $\Gamma(H)$ and the total number of bends along the edges of E_X is at most β , for some $\beta \in \mathbb{N}$. For example, Figure 1a shows a 7-extension $\Gamma(G)$ of the drawing $\Gamma(H)$ in Figure 1b, with the missing vertices drawn in red.

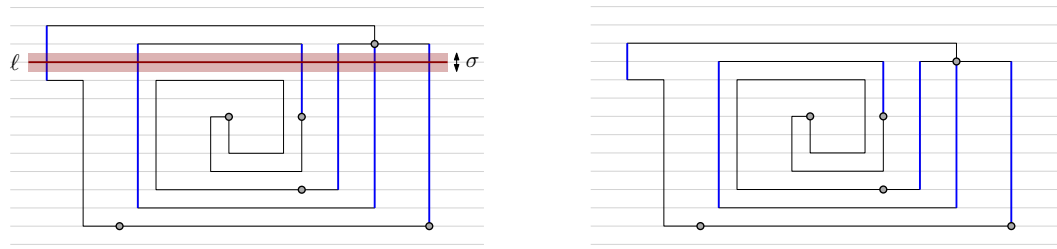
Our problem of interest is defined as follows.

BEND-MINIMAL ORTHOGONAL EXTENSION (BMOE)

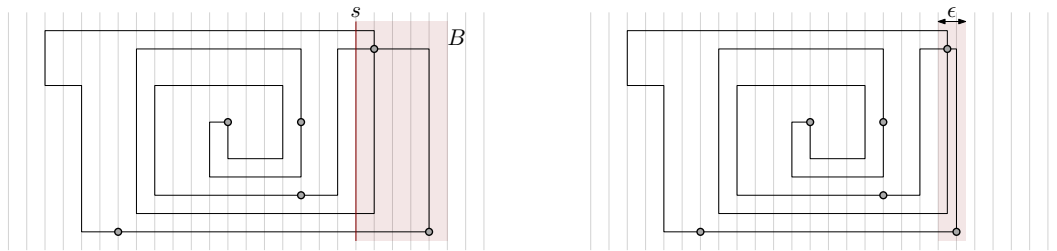
Input: $(G, H, \Gamma(H))$, integer β

Problem: Is there a β -extension $\Gamma(G)$ of $\Gamma(H)$?

We remark that BMOE is known to be NP-hard even when restricted to the case where $\beta = 0$ and $V(H) = \emptyset$ [22]. Also, unless specified otherwise, in the rest of the paper we only consider orthogonal drawings which are planar. Our parameter of interest is the number of vertices and edges missing from H , i.e., $\kappa = |V(G) \setminus V(H)| + |E(G) \setminus E(H)|$.



■ **Figure 2** Two shape-equivalent orthogonal drawings such that the one on the right is obtained from the one on the left by applying a (σ, ℓ) -strip removal operation.



■ **Figure 3** Two shape-equivalent orthogonal drawings such that the one on the right is obtained from the one on the left by applying Lemma 2 with the v-selection B .

Basic Tools. We introduce a set of redrawing operations that will be used as basic tools in several proofs. It is worth noting that similar operations as the ones introduced here, which are based on shortening or prolonging sets of parallel edges in orthogonal drawings, are well known (see, e.g., [6]). However, in our specific setting we have parts of the drawing that are given and cannot be modified, and handling this requires additional care in our arguments.

A *feature point* of an orthogonal drawing is a point representing either a vertex or a bend of an edge. An *edge-segment* of an orthogonal drawing is a segment that belongs to a polyline representing an edge. Two orthogonal drawings $\Gamma(G)$ and $\Gamma'(G)$ of a planar graph G are *shape-equivalent* if one can be obtained from the other by only shortening or lengthening some edge-segments. Figure 2 shows an example of two shape-equivalent drawings; in particular, the one on the right can be obtained from the one on the left by suitably shortening the blue (thicker) edge-segments. (We note that in the literature on orthogonal drawings, this is equivalent to saying that $\Gamma(G)$ and $\Gamma'(G)$ have the same *shape* but two different *metrics*.)

Let $\Gamma(G)$ be an orthogonal drawing of a graph G . Let ℓ be a horizontal (vertical) line that contains no feature points of $\Gamma(G)$ but intersects a set S of vertical (horizontal) edge-segments. Let l be the shortest distance between the endpoints of the segments in S and ℓ . For any $0 < \sigma < l$, a (σ, ℓ) -strip removal operation consists of decreasing the y -coordinates (x -coordinates) of all feature points above (to the right of) ℓ by σ . Analogously, for any $\sigma > 0$, a (σ, ℓ) -strip addition operation consists of increasing the y -coordinates (x -coordinates) of all feature points above (to the right of) ℓ by σ . See Figure 2 for an illustration of a (σ, ℓ) -strip removal operation. The following property readily follows.

► **Property 1.** *Let $\Gamma(G)$ and $\Gamma'(G)$ be two orthogonal drawings such that $\Gamma'(G)$ is obtained from $\Gamma(G)$ by applying a (σ, ℓ) -strip removal or addition operation. Then $\Gamma(G)$ and $\Gamma'(G)$ are shape-equivalent.*

Let B be a rectangle that intersects $\Gamma(G)$ such that only one side s of B is crossed by edges of G . We call B a *v-selection* if s is vertical or a *h-selection* otherwise. Also, the subdrawing of $\Gamma(G)$ inside B is called the *B-selected drawing*; see Figure 3 for an illustration of a v-selection and of the next lemma (whose proof easily follows from Property 1).

► **Lemma 2.** *Let $\Gamma(G)$ be an orthogonal drawing and let B be a v -selection (h -selection) of $\Gamma(G)$. For any $\epsilon > 0$, there is a drawing $\Gamma'(G)$ that is shape-equivalent to $\Gamma(G)$ and such that the B -selected drawing has width (height) at most ϵ and height (width) equal as in $\Gamma(G)$.*

3 Initial Branching

In this section, we make the first step towards the fixed-parameter tractability of BMOE by applying an initial branching step to simplify the problem – notably, this will allow us to focus on only extending the drawing inside a single face of H , and to assume that H is an induced subgraph of G .

We begin by introducing some additional notation that will be useful throughout the paper. Let $\langle(G, H, \Gamma(H)), \beta\rangle$ be an instance of BMOE. A vertex $w \in V(H)$ is called an *anchor* if it is incident to an edge in the missing edge set E_X . For a missing edge $vw \in E_X$ incident to a vertex $v \in V(H)$, we will use “ports” to specify a direction that vw could potentially use to reach v in an extension of $\Gamma(H)$; we denote these directions as d which is an element from $\{\downarrow$ (north), \uparrow (south), \leftarrow (east), \rightarrow (west) $\}$. Formally, a *port candidate* for $vw \in E_X$ and $v \in V(H)$ is a pair (v, d) . A *port-function* is an ordered set of port candidates which contains precisely one port candidate for each $vw \in E_X, v \in V(H)$, ordered lexicographically by v and then by w .

We can now formalize the target problem that we will obtain from BMOE via our exhaustive branching, which will be the focus of our considerations in Sections 4–6.

BEND-MINIMAL ORTHOGONAL EXTENSION ON A FACE (F-BMOE)

Input: A planar graph G_f ; an induced subgraph H_f of G_f with $k = |X_f|$, where $X_f = V(G_f) \setminus V(H_f)$; a drawing $\Gamma(H_f)$ of H_f consisting of a single inner face f ; a port-function \mathcal{P} .

Task: Compute the minimum β for which a β -extension of $\Gamma(H_f)$ exists and such that (1) missing edges and vertices are only drawn in the face f and (2) each edge $vx \in E_X$ where $v \in V(H)$ connects to x via its port candidate defined by \mathcal{P} , or determine that no such extension exists.

For the Turing reduction formalized in the next lemma, it will be useful to recall the definition of BMOE and κ from Section 2.

► **Lemma 3.** *There is an algorithm that solves an instance \mathcal{I} of BMOE in time $3^{\mathcal{O}(\kappa)} \cdot T(|\mathcal{I}|, k)$, where $T(a, b)$ is the time required to solve an instance of F-BMOE with instance size a and parameter value b .*

Proof Sketch. We exhaustively branch over all possible faces in which a missing vertex can be drawn, as well as over possible ports that will be used by each edge incident to an anchor in H . Also, additional care is needed with each missing edge with both endpoints in H in order to make H an induced subgraph of G . For this, we branch to determine whether the edge will be drawn as a straight-line segment (in which case we simply add it to H), or whether it will have at least one bend (in which case we subdivide it, mark the newly created vertex as missing, and remember that the total number of bends will decrease by 1). ◀

We note that the marked face f can be either the single inner face of $\Gamma(H_f)$ or the outer face. On a different note, while BMOE was stated as a decision problem for complexity-theoretic purposes, the output for F-BMOE is either an integer or “No”. Two instances of F-BMOE are said to be *equivalent* if their outputs are the same. Note that checking whether an instance of F-BMOE admits some β -extension can be done in polynomial time by using the

algorithm in [2]. The pre-drawn graph given as input to the algorithm in [2] will be $\Gamma(H_f)$ with a slight modification: if a vertex v makes an angle larger than $\frac{\pi}{2}$ in the non-marked face g of $\Gamma(H_f)$, then we add dummy vertices and connect them to v until all angles around v in g are $\frac{\pi}{2}$. This guarantees that a solution only draws missing vertices inside the marked face f (and not in g). Hence, we will assume to be dealing with instances where such an extension exists, and the task is to identify the minimum value of β . We will call a β -extension minimizing the value of β a *solution*.

4 Preprocessing

We can now focus on solving an instance of F-BMOE with only a single marked face f being of interest. The aim of this section is to make the first two steps that will allow us to solve F-BMOE. This includes pruning out certain parts of the face which are provably irrelevant, and reducing the case of f being the outer face to the case of f being an inner face.

4.1 Pruning

Let $\Gamma(G)$ be an orthogonal drawing of a graph G and let f be a face of $\Gamma(G)$. A *reflex corner* p of f is a feature point that makes an angle larger than π inside f . Also, if p is an anchor, then it is called an *essential* reflex corner. A projection ℓ of a reflex corner p is a horizontal or vertical line-segment in the interior of f that starts at p and ends at its first intersection with the boundary of f . Figure 4 (left) shows two projections ℓ_1 and ℓ_2 of a reflex corner p .

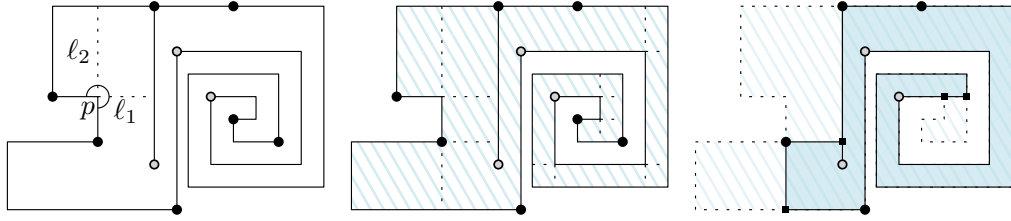
Observe that each projection ℓ of a reflex corner p divides the face f into two connected regions, which are themselves orthogonal polygons. If p is not essential and one of the two regions contains no reflex corners of its own (notice that inside this region, p needs no longer be a reflex corner) and no anchors, we call the region *redundant*. Our aim will be to show that such regions can be safely removed from the instance. More formally, recall that ℓ intersects the boundary of f in p on one side and in an element e that is either a vertex u or a point q on an edge of H on the other side of f . The *pruning operation* at ℓ for a redundant region ι works as follows. (1) If both p and e are vertices (which are therefore vertically or horizontally aligned) we add the edge pu into H , whose representation in $\Gamma(H)$ is ℓ . (2) If p is a vertex and e is an edge, we modify H by replacing q with a dummy vertex v_q that subdivides e and by adding the edge pv_q (whose representation in $\Gamma(H)$ is ℓ). (3) If p is part of an edge e' and e is also an edge, we modify H by replacing p and q with two dummy vertices v_p and v_q that subdivide e' and e and by adding the edge $v_p v_q$ (whose representation in $\Gamma(H)$ is ℓ). We finally remove the boundary of ι from H and $\Gamma(H)$, except for the edge-segment ℓ and its end-vertices. The proof of the next lemma easily follows by suitably using v-/h-selections, see also Figure 5 for an illustration.

► **Lemma 4.** *Let $\mathcal{I} = \langle G_f, H_f, \Gamma(H_f), f, \mathcal{P} \rangle$ be an instance of F-TBOE. Let ℓ be a projection of some non-essential reflex corner in f , which gives rise to a redundant region ι . Then pruning ι at ℓ results in an instance \mathcal{I}^ι that is equivalent to \mathcal{I} .*

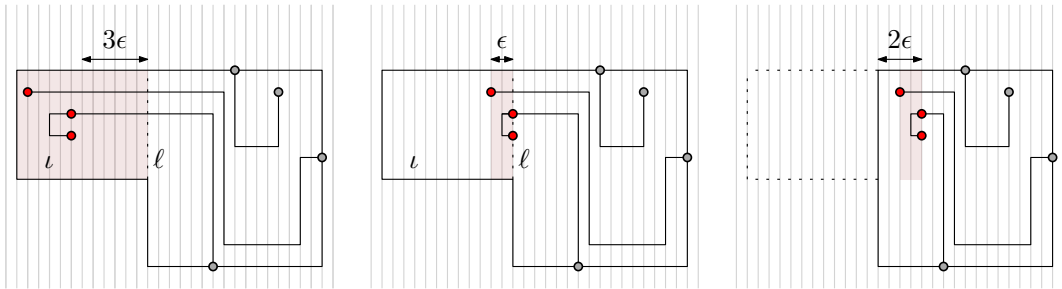
We can show that exhaustively applying Lemma 4 results in an instance with the following property: each projection of each non-essential reflex corner in f splits f into two faces, each of which has at least one port on its boundary. We call such instances *clean*; see Figure 4.

► **Lemma 5.** *There is a polynomial-time algorithm that takes as input an arbitrary instance of F-TBOE and outputs an equivalent instance which is clean.*

Given Lemma 5, we will hereinafter assume that our instances of F-TBOE are clean.



■ **Figure 4** Left: A reflex corner p and its projections ℓ_1 and ℓ_2 . Middle: A face (striped) with all its non-essential reflex corners and projections (anchor vertices have a gray filling while non-anchors are solid). Right: The corresponding clean instance (dummy vertices are drawn as small squares).



■ **Figure 5** Illustration for the proof of Lemma 4.

4.2 Outer Face

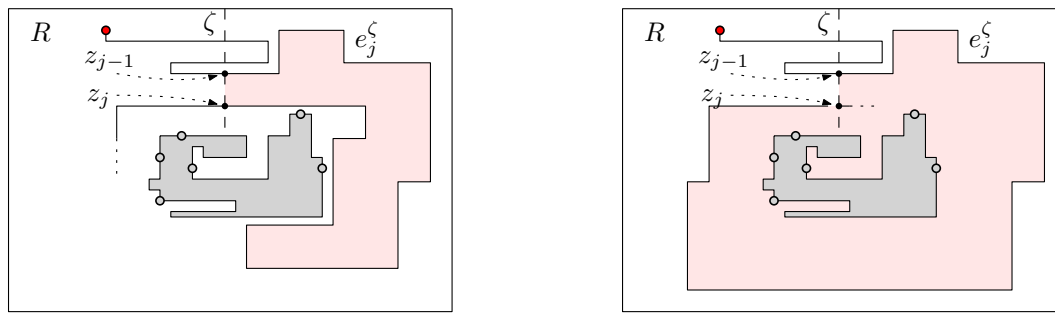
Given an instance of F-BMOE where the marked face f is the outer face of $\Gamma(H_f)$, let us begin by constructing a rectangle that bounds $\Gamma(H_f)$ and will serve as a “frame” for any solution.

► **Observation 6.** *Let \mathcal{I} be an instance of F-BMOE and let R be a rectangle that contains $\Gamma(H_f)$ in its interior. Then \mathcal{I} admits a solution that lies in the interior of R .*

Based on Observation 6, we shall assume that any instance \mathcal{I} is modified such that the outer face of $\Gamma(H_f)$ is a rectangle R containing no anchors (e.g., with four dummy vertices at its corners connected in a cycle). Notice that, while this ensures that f is no longer the outer face, f now contains a hole (that is, H_f is not connected anymore). The aim for the rest of this section is to remove this hole by connecting it to the boundary of R .

To do so, let us consider an arbitrary horizontal or vertical line-segment ζ that connects the boundary of R with an edge-segment in the drawing $\Gamma(H_f)$ and intersects no other edge-segment of $\Gamma(H_f)$. Observe that, w.l.o.g., we can assume that each edge-segment in a solution $\Gamma(G_f)$ only intersects ζ in single points (and not in a line-segment); otherwise, one may shift ζ by a sufficiently small ϵ to avoid such intersections. Roughly speaking, our aim will be to show that the instance \mathcal{I} can be “cut open” along ζ to construct an equivalent instance where the boundary of the polygon includes R , and to branch in order to determine how the edges in a hypothetical solution cross through ζ . However, to do so we need to ensure that there is a solution, in which the number of such crossings through ζ is bounded.

Let us consider the drawing of a missing edge $e \in E_X$ in $\Gamma(G_f)$. The intersection points of e with ζ partition the drawing of e into polylines $e_1^\zeta, e_2^\zeta, \dots, e_q^\zeta$, where each pair of consecutive polylines e_i^ζ and e_{i+1}^ζ touch ζ at a point, which we denote by z_i ($i = 1, \dots, q - 1$). We distinguish two cases depending on the structure of these polylines. A polyline e_j^ζ , $1 < j < q$, is called a ζ -handle if the unique region of the plane enclosed by e_j^ζ and ζ does not contain $\Gamma(H_f)$; otherwise the polyline is called a ζ -spiral. See Figure 6 for an illustration.



■ **Figure 6** Illustration of ζ -handles (left) and ζ -spirals (right).

► **Lemma 7.** *Assume \mathcal{I} and ζ are fixed as above. Then \mathcal{I} admits a solution such that no missing edge contains a ζ -handle.*

Proof Sketch. By planarity, the polyline e^* representing any ζ -handle is not crossed by any edge (except possibly at common endpoints). Consider the subdrawing Γ_ζ of $\Gamma(G_f)$ formed by all vertices and edge-segments in the interior of the unique region of the plane enclosed by e^* and ζ . At high-level, we scale-down Γ_ζ and then define suitably h-/v-selections such that the transformed version of Γ_ζ can be moved to the other side of ζ without introducing crossings. At this point we can redraw e^* such that it does not cross ζ anymore and its number of bends is not increased. ◀

Next we deal with ζ -spirals: while they cannot be completely avoided, we show that one can bound the number of ζ -spirals for each edge by a function of the parameter k .

► **Lemma 8.** *Assume \mathcal{I} and ζ are fixed as above. Then \mathcal{I} admits a solution with no ζ -handles and at most $4k(k+1)$ ζ -spirals.*

Proof Sketch. The first part of the statement follows by Lemma 7. The second part can be proved by observing that pairs of consecutive ζ -spirals of the same edge can be shortcut and merged together into a single ζ -spiral if they do not enclose any vertex. On the other hand a vertex blocking this operation must be a missing vertex, and hence we have at most k consecutive blocked pairs for each of the at most $4k$ missing edges. ◀

With Lemma 8, we obtain that there exists a solution where the total number of edge-segments crossing through ζ is at most $4k(k+1)$. We can use this to branch on which edges cross through ζ and use this to make a “bridge” connecting R to the hole in f , thus resulting in an equivalent instance where f is modified to become an inner face with no holes.

► **Lemma 9.** *There is an algorithm that takes as input an instance \mathcal{I} of F-BMOE where f is the outer face and solves it in time $2^{\mathcal{O}(k^2 \log k)} \cdot Q(|\mathcal{I}|, k)$, where $Q(a, b)$ is the time to solve an instance of F-BMOE with instance size a and parameter value b such that f is the inner face.*

Proof Sketch. We can assume that f is an inner face bounded by a rectangle R and containing a segment ζ defined as above. By Lemma 8, it is not restrictive to consider solutions such that each missing edge drawn in f contains no ζ -handles and at most $4k(k+1)$ ζ -spirals. That is, we shall consider solutions in which ζ is crossed at most $4k(k+1)$ times. The first task here is to branch over which missing edges will cross ζ (possibly multiple times) and in which order. The second task is to show that the precise position of these crossings along ζ is not important, because a hypothetical solution can always be redrawn so to use the

given crossing points without increasing the number of bends. Once this is done, each such crossing point can be replaced with a dummy vertex that subdivides ζ and belongs to the new boundary of f , which now has no holes anymore. To this aim, we can use suitably defined h-/v-selections and (σ, ℓ) -strip additions. ◀

5 Discretizing the Instances

Our next aim is to define the sector graph and show that it suffices to consider only a bounded number of possible points in each sector for extending $\Gamma(H_f)$. Essentially, this allows us to combinatorially extract those properties of $\Gamma(H_f)$ that are relevant for solving F-BMOE.

5.1 Sectors and the Sector Graph

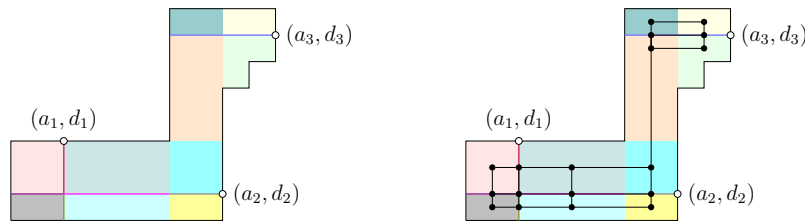
For a point $p \in f$, the *bend distance* $\text{bd}(p, (a, d))$ to a port candidate (a, d) is the minimum integer q such that there exists an orthogonal polyline with q bends connecting p and a in the interior of f which arrives to a from direction d .

► **Definition 10.** Let $\mathcal{P} = ((a_1, d_1), \dots, (a_q, d_q))$ be an ordered set of port candidates. For each point $p \in f$, we define a *bend-vector* as the tuple $\text{vect}(p) = (\text{bd}(p, (a_1, d_1)), \dots, \text{bd}(p, (a_q, d_q)))$.

► **Definition 11.** Given an ordered set of port candidates \mathcal{P} , a *sector* F is a maximal connected set of points with the same bend-vector w.r.t. \mathcal{P} .

When \mathcal{P} is not specified explicitly, we will assume it to be the set of port candidates provided by the considered instance of F-BMOE. The face f is now partitioned into a set \mathcal{F} of sectors. It is worth noting that sectors are connected regions in the face f , they do not overlap, and they cover the whole interior of f . We further notice that a sector can be degenerate, it may be a single point or a line-segment, and that pairs of (non-adjacent) sectors may have the same bend-vectors. At this point, we can define a graph representation capturing the adjacencies between the sectors in our instance; see Figure 7 for an illustration.

► **Definition 12.** Sectors A and B are adjacent if there exists a point p in A and a direction $d \in \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ such that the first point outside of A hit by the ray starting from p in direction d is in B .



► **Figure 7** Left: partitioning a face f into a set \mathcal{F} of sectors, with three anchors marked using white circles. Right: the graph representation of \mathcal{F} .

Observe that the relationship of being adjacent is symmetric; furthermore, for a specific direction d we say that sector A is *d-adjacent* to B if A is adjacent to B for this choice of d . The *sector graph* \mathcal{G} is the graph whose vertex set is the set of sectors \mathcal{F} , and adjacencies of vertices are defined via the adjacency of sectors.

It will be useful to establish some basic properties of the sector graph. For instance, it is not difficult to observe that the sector graph is a connected planar graph. Furthermore, we can show that the boundary between two sectors is, in a sense, simple. Concerning its size, we observe that each sector contains at least one intersection point between two projections and that any such intersection point can be shared by at most nine sectors (four non-degenerate sectors plus five degenerate sectors). Hence:

► **Observation 13.** *The number of vertices in \mathcal{G} is upper-bounded by $9x^2$, where x is the number of feature points in $\Gamma(H_F)$.*

5.2 The Sector-Grid

A property of sectors that will become important later is that, inside each sector, we only need a bounded number of positions for the placement of feature points in a hypothetical solution. In particular, our aim will be to construct a “universal” point-set with the property that there exists a solution which places feature points only on these points, and where the intersection of the point-set with each sector is upper-bounded by a function of the parameter. Before we construct such a universal point-set, we will first need to subdivide sectors into “subsectors” which have grid-like connections to each other. Crucially, we will show that the number of subsectors in each sector is upper-bounded by a function of k .

Let us fix a sector S and a direction $d \in \{\uparrow, \downarrow, \leftarrow, \rightarrow\}$, say w.l.o.g. $d = \rightarrow$. Let a reflex corner be *critical* if it is incident to at least two distinct sectors, and (S, d) -critical if it is critical and also can be reached by a ray from some point in S traveling in direction d . To construct the subsectors of S , let us project all (S, d) -critical reflex corners (for all four choices of d) into S to obtain a grid, and make each induced grid cell in S a *subsector* of S . Observe that for each subsector in each sector S , it holds that its entire boundary in each direction is either the boundary of f , or touches the boundary of a single other “adjacent” subsector (which may or may not belong to S).

Crucially, we show that the number of such subsectors obtained from each sector is not too large. This will be important when using sectors for dynamic programming in Section 6, since it will allow us to bound the size of the universal point-set in each sector.

► **Lemma 14.** *For each S, d , there are at most $4k$ (S, d) -critical reflex corners.*

By applying Lemma 14 on all sides of each sector S , we obtain that S is partitioned into at most $(8k)^2$ subsectors. Observe that we may refine the sector graph constructed earlier by partitioning sectors into subsectors, with adjacencies between subsectors defined in the same way as between sectors. Note that by definition, each pair of adjacent subsectors share the complete side of the boundary that connects them. Hence, we can define a *subsector-column* as a set of subsectors which form a path in the subsector graph and span the same vertical strip in $\Gamma(H_f)$, and similarly a *subsector-row* is a set of subsectors which forms a path in the subsector graph and span the same horizontal strip in $\Gamma(H_f)$.

With the above in mind, we proceed to build the universal point-set. As our first step, we construct an auxiliary set of points we call a *skeleton*. Let us now choose an arbitrary horizontal line-segment for each subsector-row that intersects it, and similarly an arbitrary vertical line-segment for each subsector-column that intersects it. To construct the skeleton, for each subsector v , we define the point p_v to be the point at the intersection of the two line-segments intersecting the subsector.

Let $\text{subgridsize}(k) = 112k^3 + 202k^2 + 85k$. We place a set of $\text{subgridsize}(k) \times \text{subgridsize}(k)$ points in a grid-like arrangement into each subsector v , where the points are centered at p_v and the grid underlying these points occupies a square area of $\epsilon \times \epsilon$ for a

sufficiently small ϵ . In particular, we choose ϵ to be sufficiently small so that a horizontal or vertical projection of any pair of grid points intersects with the same line-segment of $\Gamma(H_f)$. We call this point-set \mathcal{S}_v the *subsector-grid* of a subsector v ; in the degenerate cases where v is a line-segment or single point, the subsector-grid is a set of points on that segment or just a single point, respectively.

► **Lemma 15.** *There exists a solution such that each feature point not in $\Gamma(H_f)$ lies on a subsector-grid point of some subsector.*

Proof Sketch. The proof undergoes several steps. We first argue that if a polyline representing (part of) an edge drawn inside a sector-column (or analogously inside a sector-row) \mathcal{A} contains a large number of bends, then we can redraw it and obtain an equivalent solution. This requires similar arguments as in Lemmas 7 and 8 (although in a different setting), together with new arguments dealing with edges that have a “staircase” shape. The second step is then to prove that similar redrawing arguments can be adopted to show that there are not too many disjoint polylines that represent the same edge inside \mathcal{A} . The last step is to show that the feature points of a hypothetical solution that lies in a subsector can always be mapped to the specific point-set defined by the subsector-grid. ◀

From Lemmas 14, 15 and by setting $\text{gridsize}(k) = \text{subgridsize}(k)^2 \cdot (8k)^2$, we obtain:

► **Corollary 16.** *Given an instance \mathcal{I} of F-BMOE we can construct a point-set (called a sector grid) in time $\mathcal{O}(|\mathcal{I}|)$ with the following properties: (1) \mathcal{I} admits a solution whose feature points all lie on the sector grid, and (2) each sector contains at most $\text{gridsize}(k)$ points of the sector grid.*

6 Exploiting the Treewidth of Sector Graphs

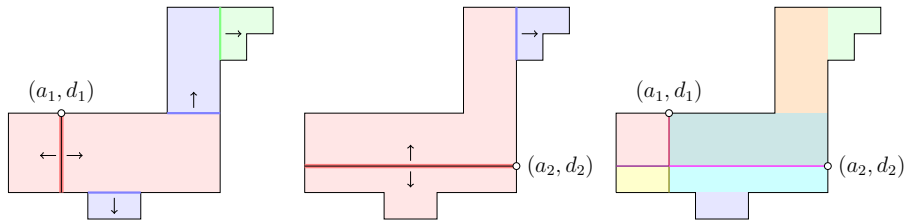
In this section, we complete the proof of our fixed-parameter tractability result by first showing that the sector graphs in fact have treewidth bounded by a function of the parameter k , and then by using this fact to design a dynamic programming algorithm solving F-BMOE.

6.1 Sector Graphs Are Tree-Like

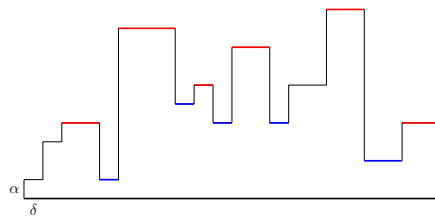
We begin by introducing some notation that will be useful in this subsection. Let $\mathcal{P} = ((a_1, d_1), \dots, (a_q, d_q))$ be the ordered set of port candidates for the considered face f . Also, $q \leq 4k$, because the degree of the vertices being added is at most 4. For each $1 \leq i \leq q$, let $\mathcal{P}_i = ((a_1, d_1), \dots, (a_i, d_i))$ be a prefix of length i of \mathcal{P} . For each $1 \leq i \leq q$, we denote by \mathcal{F}_i and \mathcal{G}_i the set of sectors and the sector graph, respectively, obtained by considering the bend distances to \mathcal{P}_i . Using this terminology, we obtain that the graph \mathcal{G}_q is precisely the sector graph of our initial instance, which we will also simply denote as \mathcal{G} . Furthermore, for a sector $F \in V(\mathcal{G}_i)$ we denote by \mathcal{U}_F^{t+1} the set of sectors in \mathcal{G}_{t+1} that F is partitioned into when one additionally considers bend distances to (a_{t+1}, d_{t+1}) .

► **Lemma 17.** *The sector graph \mathcal{G}_1 is a tree.*

Lemma 17 will be used as a base of an inductive argument establishing a bound on the treewidth of \mathcal{G} . See Figure 8 for an example of the sectors for two port candidates. We start by considering how each sector $F \in \mathcal{F}_t$ maps to a subset \mathcal{U}_F^{t+1} of sectors in \mathcal{F}_{t+1} . Towards this aim, let us now consider an arbitrary sector $F \in \mathcal{F}_t$ for some $1 \leq t \leq q$. We say that a line-segment δ on the boundary of F is an *F-baseline* if (1) each point in F can be reached by a ray starting at and orthogonal to δ , and (2) δ touches F on one side and points in $f \setminus F$ on the other side. When F is clear from context, we simply use baseline for brevity.



■ **Figure 8** Sectors with respect to (a) the first port; (b) the second port; (c) \mathcal{P}_2 . For a sector of each color, the segment on the border highlighted with the same color is its baseline; for (c) different sectors have different colors, and notice that at the intersection of the rays from (a_1, d_1) and (a_2, d_2) there is also a single point sector.



■ **Figure 9** The segments colored red (blue) are local maxima (minima).

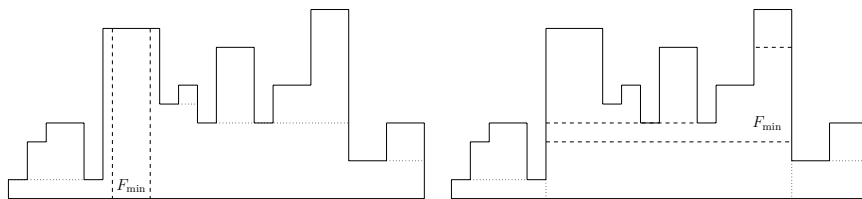
► **Lemma 18.** *Each sector in \mathcal{F}_t , $1 \leq t \leq q$, admits at least one baseline.*

The existence of a baseline is already quite helpful to obtain the desired bound on the treewidth, but not yet sufficient on its own. In particular, this implies that each sector has the shape of a histogram. Next, we show that the bend distances to any “additional port” cannot differ too much within a sector.

► **Lemma 19.** *For every sector $F \in \mathcal{F}_t$, $t \in [1, q - 1]$, and every pair $F_1, F_2 \in \mathcal{U}_F^{t+1}$, $|\text{bd}(p, (a_{t+1}, d_{t+1})) - \text{bd}(q, (a_{t+1}, d_{t+1}))| \leq 3$ for every pair of points $p \in F_1$, $q \in F_2$.*

With Lemmas 18 and 19, we are ready to proceed to the most difficult part of establishing our bound on the treewidth of the sector graph. Let us fix some F -baseline δ for a sector F in the sector graph \mathcal{G}_t , $1 \leq t \leq q$. Consider the polyline α obtained when traversing F in clockwise fashion from one endpoint of δ to the other, where α does not intersect δ . We call a line-segment in α a *local maximum (minimum)* if α makes a right (left) turn both before and after the line-segment (see Figure 9). Let $\xi_{\max}(F)$ ($\xi_{\min}(F)$) denote the number of local maxima (local minima) in F ; note that since each sector is a histogram, $\xi_{\max}(F) = \xi_{\min}(F) + 1$.

► **Lemma 20.** *For every sector $F \in \mathcal{F}_t$, $1 \leq t \leq q - 1$, we have $|\mathcal{U}_F^{t+1}| \leq 4 + \xi_{\max}(F)$ and $\max_{F' \in \mathcal{U}_F^{t+1}} \xi_{\max}(F') \leq \xi_{\max}(F)$.*



■ **Figure 10** Cases of relative location of the F_{\min} sector in F relative to the F -baseline, Lemma 20.

To obtain the main result of this section (Theorem 22), we will combine Lemma 20 with the following lemma that bounds the number of local maxima in each sector.

► **Lemma 21.** *For each sector F in $V(\mathcal{G})$, $\xi_{\max}(F) \leq 4k$.*

► **Theorem 22.** *Let \mathcal{G} be a sector graph of a face f of the drawing $\Gamma(G)$. Then $\text{tw}(\mathcal{G}) \leq (4 + 4k)^{4k}$.*

Proof Sketch. We prove the claim by induction on the number of port candidates for f , where the base of an induction exactly follows from the result of Lemma 17. For the inductive step, we assume that $\text{tw}(\mathcal{G}_t) = \mathcal{O}(k^t)$ and our aim will be to show that $\text{tw}(\mathcal{G}_{t+1})$ is $\mathcal{O}(k^{t+1})$. To do so, we replace each occurrence of a sector v in a bag with all of the sectors in \mathcal{U}_F^{t+1} . ◀

6.2 The Final Step

At this point, we have shown that an instance $\mathcal{I} = \langle G_f, H_f, \Gamma(H_f), f, \mathcal{P} \rangle$ with $k = |V(G_f) \setminus V(H_f)|$ of F-BMOE admits a sector graph \mathcal{G} of treewidth at most $(4 + 4k)^{4k}$ (Theorem 22), and that a bend-minimal extension of $\Gamma(H_f)$ to an orthogonal planar drawing of G_f can be assumed to only contain feature points on the sector-grid points as per Corollary 16, of which there are at most $\text{gridsize}(k)$ many per sector. This allows us to proceed to the final ingredient for our algorithm:

► **Lemma 23.** *F-BMOE can be solved in time $2^{k^{\mathcal{O}(1)}} \cdot |V(G_f)|$.*

Proof Sketch. Thanks to Theorem 22, we can use known results to compute a nice tree decomposition (T, χ) of \mathcal{G} of small width. Next we design a dynamic program that runs along T and at each point stores all possible options of how a hypothetical bend-minimal extension can intersect the sector-grid points of the sectors in the current bag. ◀

By combining Lemma 23 with Lemma 3 and Observation 13, we conclude:

► **Corollary 24.** *BMOE can be solved in time $2^{\kappa^{\mathcal{O}(1)}} \cdot n$, where n is the number of feature points of $\Gamma(H)$.*

7 Concluding Remarks

We have established the fixed-parameter tractability of the extension problem for bend-minimal orthogonal drawings, marking a notable addition to our understanding of drawing extension problems. What distinguishes this result from some of its predecessors on, e.g., extending 1-planar [17], simple k -planar [21] or crossing-minimal [23] drawings, is that these examples were topological while orthogonal planar drawings are geometric in nature. We believe this is one of the reasons why it seems impossible to use previously developed techniques in our setting, a fact which inspired the development of a novel machinery that we believe will find applications beyond the specific context of the problem studied here.

As an example of this, a minor adjustment of our technique is already sufficient to also obtain a fixed-parameter algorithm for the problem of extending an orthogonal planar drawing while preserving a bound δ on the number of bends per edge [5, 7] parameterized by $\kappa + \delta$. But the technique could also possibly be applied to more general drawing styles, such as extending drawings restricted to boundedly many allowed edge slopes [24, 26].

References

- 1 Patrizio Angelini, Giuseppe Di Battista, Fabrizio Frati, Vít Jelínek, Jan Kratochvíl, Maurizio Patrignani, and Ignaz Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32:1–32:42, 2015. doi:10.1145/2629341.
- 2 Patrizio Angelini, Ignaz Rutter, and T. P. Sandhya. Extending partial orthogonal drawings. *J. Graph Algorithms Appl.*, 25(1):581–602, 2021. doi:10.7155/jgaa.00573.
- 3 Alan Arroyo, Fabian Klute, Irene Parada, Birgit Vogtenhuber, Raimund Seidel, and Tilo Wiedera. Inserting one edge into a simple drawing is hard. *Discrete & Computational Geometry*, pages 1–26, 2022. doi:10.1007/978-3-030-60440-0_26.
- 4 Giuseppe Di Battista, Giuseppe Liotta, and Francesco Vargiu. Spirality and optimal orthogonal drawings. *SIAM J. Comput.*, 27(6):1764–1811, 1998. doi:10.1137/S0097539794262847.
- 5 Therese Biedl and Goos Kant. A better heuristic for orthogonal graph drawings. *Comput. Geom. Theory Appl.*, 9(3):159–180, 1998. doi:10.1016/S0925-7721(97)00026-6.
- 6 Therese Biedl, Anna Lubiw, Mark Petrick, and Michael J. Spriggs. Morphing orthogonal planar graph drawings. *ACM Trans. Algorithms*, 9(4):29, 2013. doi:10.1145/2500118.
- 7 Thomas Bläsius, Ignaz Rutter, and Dorothea Wagner. Optimal orthogonal graph drawing with convex bend costs. *ACM Trans. Algorithms*, 12(3):33:1–33:32, 2016. doi:10.1145/2838736.
- 8 Guido Brückner, Ignaz Rutter, and Peter Stumpf. Extending partial representations of circle graphs in near-linear time. In *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria*, volume 241 of *LIPICs*, pages 25:1–25:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.MFCS.2022.25.
- 9 Guido Brückner and Ignaz Rutter. Partial and constrained level planarity. In *Discrete Algorithms (SODA'17)*, pages 2000–2011. SIAM, 2017. doi:10.1137/1.9781611974782.130.
- 10 Erin W. Chambers, David Eppstein, Michael T. Goodrich, and Maarten Löffler. Drawing graphs in the plane with a prescribed outer face and polynomial area. *J. Graph Algorithms Appl.*, 16(2):243–259, 2012. doi:10.7155/jgaa.00257.
- 11 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.
- 12 Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice-Hall, 1999.
- 13 Emilio Di Giacomo, Giuseppe Liotta, and Fabrizio Montecchiani. Orthogonal planarity testing of bounded treewidth graphs. *J. Comput. Syst. Sci.*, 125:129–148, 2022. doi:10.1016/j.jcss.2021.11.004.
- 14 Walter Didimo and Giuseppe Liotta. Computing orthogonal drawings in a variable embedding setting. In Kyung-Yong Chwa and Oscar H. Ibarra, editors, *Algorithms and Computation (ISAAC'98)*, volume 1533 of *LNCS*, pages 79–88. Springer, 1998. doi:10.1007/3-540-49381-6_10.
- 15 Walter Didimo, Giuseppe Liotta, Giacomo Ortali, and Maurizio Patrignani. Optimal orthogonal drawings of planar 3-graphs in linear time. In *Discrete Algorithms (SODA'20)*, pages 806–825. SIAM, 2020. doi:10.1137/1.9781611975994.49.
- 16 Christian A. Duncan and Michael T. Goodrich. Planar orthogonal and polyline drawing algorithms. In Roberto Tamassia, editor, *Handbook of Graph Drawing and Visualization*, chapter 7, pages 223–246. CRC Press, 2013.
- 17 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending nearly complete 1-planar drawings in polynomial time. In Javier Esparza and Daniel Král', editors, *Mathematical Foundations of Computer Science (MFCS'20)*, volume 170 of *LIPICs*, pages 31:1–31:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.MFCS.2020.31.
- 18 Eduard Eiben, Robert Ganian, Thekla Hamm, Fabian Klute, and Martin Nöllenburg. Extending partial 1-planar drawings. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors,

- Automata, Languages, and Programming (ICALP'20)*, volume 168 of *LIPICs*, pages 43:1–43:19. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.43.
- 19 Markus Eiglsperger, Sándor P. Fekete, and Gunnar W. Klau. Orthogonal graph drawing. In Michael Kaufmann and Dorothea Wagner, editors, *Drawing Graphs: Methods and Models*, volume 2025 of *LNCS*, chapter 6, pages 121–171. Springer-Verlag, 2001. doi:10.1007/3-540-44969-8_6.
 - 20 Jirí Fiala, Ignaz Rutter, Peter Stumpf, and Peter Zeman. Extending partial representations of circular-arc graphs. In *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers*, volume 13453 of *Lecture Notes in Computer Science*, pages 230–243. Springer, 2022. doi:10.1007/978-3-031-15914-5_17.
 - 21 Robert Ganian, Thekla Hamm, Fabian Klute, Irene Parada, and Birgit Vogtenhuber. Crossing-optimal extension of simple drawings. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP 2021*, volume 198 of *LIPICs*, pages 72:1–72:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.ICALP.2021.72.
 - 22 Ashim Garg and Roberto Tamassia. On the computational complexity of upward and rectilinear planarity testing. *SIAM J. Comput.*, 31(2):601–625, 2001. doi:10.1137/S0097539794277123.
 - 23 Thekla Hamm and Petr Hliněný. Parameterised partially-predrawn crossing number. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022*, volume 224 of *LIPICs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SoCG.2022.46.
 - 24 Udo Hoffmann. On the complexity of the planar slope number problem. *J. Graph Algorithms Appl.*, 21(2):183–193, 2017. doi:10.7155/jgaa.00411.
 - 25 Vít Jelínek, Jan Kratochvíl, and Ignaz Rutter. A Kuratowski-type theorem for planarity of partially embedded graphs. *Comput. Geom. Theory Appl.*, 46(4):466–492, 2013. doi:10.1016/j.comgeo.2012.07.005.
 - 26 Balázs Keszegh, János Pach, and Dömötör Pálvölgyi. Drawing planar graphs of bounded degree with few slopes. *SIAM J. Discrete Math.*, 27(2):1171–1183, 2013. doi:10.1137/100815001.
 - 27 Giordano Da Lozzo, Giuseppe Di Battista, and Fabrizio Frati. Extending upward planar graph drawings. *Comput. Geom. Theory Appl.*, 91:101668, 2020. doi:10.1016/j.comgeo.2020.101668.
 - 28 Tamara Mchedlidze, Martin Nöllenburg, and Ignaz Rutter. Extending convex partial drawings of graphs. *Algorithmica*, 76(1):47–67, 2016. doi:10.1007/s00453-015-0018-6.
 - 29 Takao Nishizeki and Md. Saidur Rahman. *Planar Graph Drawing*, volume 12 of *Lecture Notes Series on Computing*. World Scientific, 2004. doi:10.1142/5648.
 - 30 Maurizio Patrignani. On extending a partial straight-line drawing. *Int. J. Found. Comput. Sci.*, 17(5):1061–1070, 2006. doi:10.1142/S0129054106004261.
 - 31 Roberto Tamassia. On embedding a graph in the grid with the minimum number of bends. *SIAM J. Comput.*, 16(3):421–444, 1987. doi:10.1137/0216030.

Improved Bounds for Covering Paths and Trees in the Plane

Ahmad Biniiaz ✉

School of Computer Science, University of Windsor, Canada

Abstract

A *covering path* for a planar point set is a path drawn in the plane with straight-line edges such that every point lies at a vertex or on an edge of the path. A *covering tree* is defined analogously. Let $\pi(n)$ be the minimum number such that every set of n points in the plane can be covered by a noncrossing path with at most $\pi(n)$ edges. Let $\tau(n)$ be the analogous number for noncrossing covering trees. Dumitrescu, Gerbner, Keszegh, and Tóth (Discrete & Computational Geometry, 2014) established the following inequalities:

$$\frac{5n}{9} - O(1) < \pi(n) < \left(1 - \frac{1}{601080391}\right)n, \quad \text{and} \quad \frac{9n}{17} - O(1) < \tau(n) \leq \left\lfloor \frac{5n}{6} \right\rfloor.$$

We report the following improved upper bounds:

$$\pi(n) \leq \left(1 - \frac{1}{22}\right)n, \quad \text{and} \quad \tau(n) \leq \left\lfloor \frac{4n}{5} \right\rfloor.$$

In the same context we study rainbow polygons. For a set of colored points in the plane, a *perfect rainbow polygon* is a simple polygon that contains exactly one point of each color in its interior or on its boundary. Let $\rho(k)$ be the minimum number such that every k -colored point set in the plane admits a perfect rainbow polygon of size $\rho(k)$. Flores-Peñaloza, Kano, Martínez-Sandoval, Orden, Tejel, Tóth, Urrutia, and Vogtenhuber (Discrete Mathematics, 2021) proved that $20k/19 - O(1) < \rho(k) < 10k/7 + O(1)$. We report the improved upper bound of $\rho(k) < 7k/5 + O(1)$.

To obtain the improved bounds we present simple $O(n \log n)$ -time algorithms that achieve paths, trees, and polygons with our desired number of edges.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Discrete mathematics

Keywords and phrases planar point sets, covering paths, covering trees, rainbow polygons

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.19

Related Version *Full Version*: <https://arxiv.org/abs/2303.04350>

Funding Research supported by NSERC.

1 Introduction

Traversing a set of points in the plane by a polygonal path possessing some desired properties has a rich background. For example the famous traveling salesperson path problem asks for a polygonal path with minimum total edge length [6, 29]. In recent years there has been an increased interest in paths with properties such as being noncrossing [2, 9], minimizing the longest edge length [8], maximizing the shortest edge length [4], minimizing the total or the largest turning angle [1, 18], and minimizing the number of turns (which is the same as minimizing the number of edges) [15, 30] to name a few.

The main focus of this paper is polygonal paths with a small number of edges. It is related to the classical nine dots puzzle which asks for covering the vertices of a 3×3 grid by a polygonal path with no more than 4 segments (Figure 1). It appears in Sam Loyd’s Cyclopedia of Puzzles from 1914 [27].



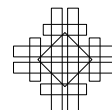
© Ahmad Biniiaz;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

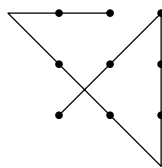
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 19; pp. 19:1–19:15



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** The nine dots puzzle.

Let P be a set of n points in the plane. A *spanning path* for P is a path drawn in the plane with straight-line edges such that every point of P lies at a vertex of the path and every vertex of the path lies at a point of P . In other words, it is a Hamiltonian path which has exactly $n - 1$ edges. The path in the figure above is not a spanning path because two of its vertices do not lie on given points. A *covering path* for P is a path drawn in the plane with straight-line edges such that every point of P lies at a vertex or on an edge of the path. A vertex of a covering path can be any point in the plane (not necessarily in P). The path in the figure above is a covering path with 4 edges. With these definitions, any spanning path is also a covering path, but a covering path may not be a spanning path. A *covering tree* for P is defined analogously as a tree drawn in the plane with straight-line edges such that every point of P lies at a vertex or on an edge of the tree. A covering path or a tree is called *noncrossing* if its edges do not cross each other. The edges of covering paths and trees are also referred to as *links* in the literature [5].

Covering paths and trees have received considerable attention in recent years, see e.g. [5, 15, 25]. In particular covering paths with a small number of edges find applications in robotics and heavy machinery for which turning is an expensive operation [30]. Covering trees with a small number of edges are useful in red-blue separation [20] and in constructing rainbow polygons [19]. In 2010 F. Morić [14] and later Dumitrescu, Gerbner, Keszegh, and Tóth [15] raised many challenging questions about covering paths and trees. Specifically they asked the following two questions which are the main topics of this paper. As noted in [14], analogous questions were asked by E. Welzl in Gremo's Workshop on Open Problems 2011.

1. What is the minimum number $\pi(n)$ such that every set of n points in the plane can be covered by a noncrossing path with at most $\pi(n)$ edges?
2. What is the minimum number $\tau(n)$ such that every set of n points in the plane can be covered by a noncrossing tree with at most $\tau(n)$ edges?

For both $\pi(n)$ and $\tau(n)$, a trivial upper bound is $n - 1$ (which comes from the existence of a noncrossing spanning path) and a trivial lower bound is $\lceil \frac{n}{2} \rceil$ (because if no three points are collinear then each edge covers at most two points). In 2014, Dumitrescu et al. [15] established, among other interesting results, the following nontrivial bounds:

$$\frac{5n}{9} - O(1) < \pi(n) < \left(1 - \frac{1}{601080391}\right)n, \quad \text{and} \quad \frac{9n}{17} - O(1) < \tau(n) \leq \left\lfloor \frac{5n}{6} \right\rfloor.$$

The following is a related question that has recently been raised by Flores-Peñaloza, Kano, Martínez-Sandoval, Orden, Tejel, Tóth, Urrutia, and Vogtenhuber [19] in the context of rainbow polygons. For a set of colored points in the plane, a *rainbow polygon* is a simple polygon that contains at most one point of each color in its interior or on its boundary. A rainbow polygon is called *perfect* if it contains exactly one point of each color. The *size* of a polygon is the number of its edges (which is the same as the number of its vertices).

3. What is the minimum number $\rho(k)$ (known as *the rainbow index*) such that every k -colored point set in the plane, with no three collinear points, admits a perfect rainbow polygon of size $\rho(k)$?

Question 3 is related to covering trees in the sense that (as we will see later in Section 4) particular covering trees could lead to better upper bounds for $\rho(k)$. Flores-Peñaloza et al. [19] established the following inequalities:

$$\frac{20k}{19} - O(1) < \rho(k) < \frac{10k}{7} + O(1).$$

The upper bounds on $\pi(n)$, $\tau(n)$, and $\rho(n)$ are universal (i.e., any point set admits these bounds) and they are obtained by algorithms that achieve paths, trees, and polygons of certain size [15, 19]. The lower bounds, however, are existential (i.e., there exist point sets that achieve these bounds) and they are obtained by the same point set that is exhibited in [15]. Perhaps there should be configurations of points that achieve better lower bounds for each specific number.

1.1 Our contributions

Narrowing the gaps between the lower and upper bounds for $\pi(n)$, $\tau(n)$, and $\rho(n)$ are open problems which are explicitly mentioned in [15, 19]. In this paper we report the following improved upper bounds for the three numbers:

$$\pi(n) \leq \left(1 - \frac{1}{22}\right)n, \quad \tau(n) \leq \left\lceil \frac{4n}{5} \right\rceil, \quad \text{and} \quad \rho(k) < \frac{7k}{5} + O(1).$$

The new bounds for $\pi(n)$ and $\tau(n)$ are the first improvements in 8 years. To obtain these bounds we present algorithms that achieve noncrossing covering paths, noncrossing covering trees, and rainbow polygons with our desired number of edges. The algorithms are simple and run in $O(n \log n)$ time where n is the number of input points. The running time is optimal for paths because computing a noncrossing covering path has an $\Omega(n \log n)$ lower bound [15]. A noncrossing covering tree, however, can be computed in $O(n)$ time by taking a spanning star. We extend our path algorithm and achieve an upper bound of $(1 - \frac{1}{22})n + 2$ for noncrossing *covering cycles*. This is a natural variant of the traveling salesperson tour problem with the objective of minimizing the number of links, which is NP-hard [5].

Our algorithms share some similarities with previous algorithms in the sense that both are iterative and use the standard plane sweep technique which scans the points from left to right. However, to achieve the new bounds we employ new geometric insights and make use of convex layers and the Erdős-Szekeres theorem [16].

Regardless of algorithmic implications, our results are important because they provide new information on universal numbers $\pi(n)$, $\tau(n)$, and $\rho(n)$ similar to the crossing numbers [3, 13, 22], the size of crossing families (pairwise crossing edges) [28], the Steiner ratio [6, 23], and other numbers and constants studied in discrete geometry (such as [8, 10, 17]).

Remark. Collinear points are beneficial for covering paths and trees as they usually lead to paths and trees with fewer edges. To avoid the interruption of our arguments we first describe our algorithms for point sets with no three collinear points. In the end we show how to handle collinearities.

1.2 Related problems and results

If we drop the noncrossing property, Dumitrescu et al. [15] showed that every set of n points in the plane admits a (possibly self-crossing) covering path with $n/2 + O(n/\log n)$ edges. Covering paths have also been studied from the optimization point of view. The problem of

computing a covering path with minimum number of edges for a set of points in the plane (also known as the *minimum-link* covering path problem and the *minimum-bend* covering path problem) is shown to be NP-hard by Arkin et al. [5]. Stein and Wagner [30] presented an $O(\log z)$ -approximation algorithm where z is the maximum number of collinear points.

Keszegh [25] determined exact values of $\pi(n)$ and $\tau(n)$ for vertices of square grids. The axis-aligned version of covering paths is also well-studied and various lower bounds, upper bounds, and approximation algorithms are presented to minimize the number of edges of such paths; see e.g. [7, 12, 24]. Covering trees are studied also in the context of separating red and blue points in the plane [20]. The problem of covering points in the plane with minimum number of lines is another related problem which is also well-studied, see e.g. [11, 21, 26].

For problems and results related to rainbow polygons we refer the reader to the paper of Flores-Peñalosa et al. [19]. In particular, they determine the exact rainbow indices for small values of k by showing that $\rho(k) = k$ for $k \in \{3, 4, 5, 6\}$ and $\rho(7) = 8$.

1.3 Preliminaries

For two points p and q in the plane we denote by $\ell(p, q)$ the line through p and q , and by pq the line segment with endpoints p and q . For two paths δ_1 and δ_2 , where δ_1 ends at the same vertex at which δ_2 starts, we denote their concatenation by $\delta_1 \oplus \delta_2$.

A point set P is said to be in *general position* if no three points of P are collinear. We denote the convex hull of P by $\text{CH}(P)$. A set K of k points in the plane in convex position, with no two points on a vertical line, is a *k-cap* (resp. a *k-cup*) if all points of K lie on or above (resp. below) the line through the leftmost and rightmost points of K . A classical result of Erdős and Szekeres [16] implies that every set of at least $\binom{2k-4}{k-2} + 1$ points in the plane in general position, with no two points on a vertical line, contains a *k-cap* or a *k-cup*. This bound is tight in the sense that there are point sets of size $\binom{2k-4}{k-2}$ that do not contain any *k-cap* or *k-cup* [16].

2 Noncrossing covering paths

In this section we prove that $\pi(n) \leq (1 - 1/22)n$. We start by the following folklore result on the existence of noncrossing polygonal paths among points in the plane; see e.g. [15, 20].

► **Lemma 1.** *Let P be a set of points in the plane in the interior of a convex region C , and let p and q be two points on the boundary of C . Then $P \cup \{p, q\}$ admits a noncrossing spanning path with $|P| + 1$ edges such that its endpoints are p and q , and its relative interior lies in the interior of C .*

In fact the spanning path that is obtained by Lemma 1 is a noncrossing covering path for $P \cup \{p, q\}$ and it lies in the convex hull of $P \cup \{p, q\}$. The following lemma shows that any set of 23 points can be covered by a noncrossing path with 21 edges.

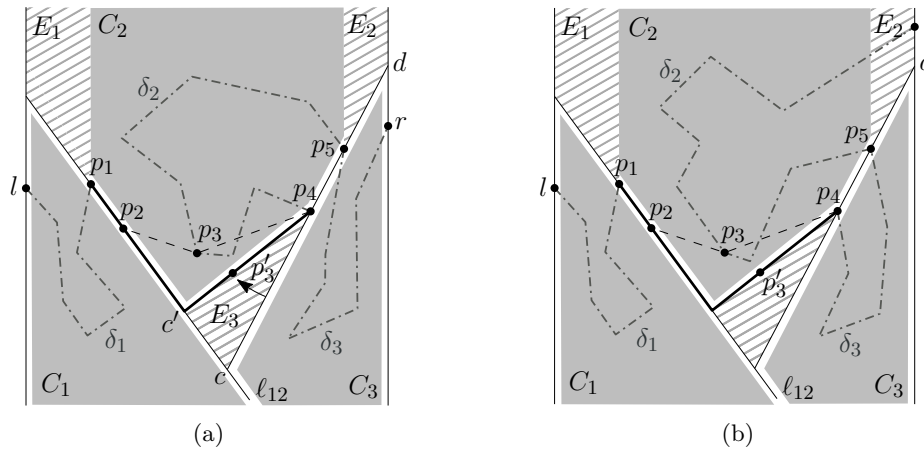
► **Lemma 2.** *Let P be a set of at least 23 points in the plane such that no two points have the same x -coordinate. Let H be the vertical strip bounded by the vertical lines through the leftmost and rightmost points of P . Then there exists a noncrossing covering path for P with $|P| - 2$ edges that is contained in H and its endpoints are the leftmost and rightmost points of P .*

Proof. Our proof is constructive. Let l and r be the leftmost and rightmost points of P , respectively. Let $P' = P \setminus \{l, r\}$, and notice that $|P'| \geq 21$. We assume that P' is in general position. In the end of the proof we briefly describe how to handle collinearities. By the

result of [16] the set P' has a 5-cap or a 5-cup. After a suitable reflection we may assume that it has a 5-cup K with points p_1, p_2, p_3, p_4, p_5 from left to right, as in Figure 2(a). Among all 5-cups in P' we may assume that K is one for which p_1 is the leftmost possible point. Also among all such 5-cups (with leftmost point p_1) we may assume that K is the one for which p_5 is the rightmost possible point. This choice of K implies that the region that is the intersection of H with the halfplane above $\ell(p_1, p_2)$ and the halfplane to the left of the vertical line through p_1 is empty of points of P' ; this region is denoted by E_1 in Figure 2(a). Similarly the region that is the intersection of H with the halfplane above $\ell(p_4, p_5)$ and the halfplane to the right of the vertical line through p_5 is empty of points of P' ; this region is denoted by E_2 in Figure 2(a).

For brevity let $\ell_{12} = \ell(p_1, p_2)$ and $\ell_{45} = \ell(p_4, p_5)$. We distinguish two cases: (i) l lies below ℓ_{12} or r lies below ℓ_{45} , and (ii) l lies above ℓ_{12} and r lies above ℓ_{45} .

(i) In this case we may assume, up to symmetry, that l lies below ℓ_{12} as in Figure 2. Let c be the intersection point of ℓ_{12} with ℓ_{45} , and d be the intersection point of ℓ_{45} with the right boundary of H . Since K is a cup, c lies below K and hence in H . Consider the ray emanating from p_4 and passing through c . Rotate this ray clockwise around p_4 and stop as soon as hitting a point in the triangle $\Delta p_2 p_4$; see Figure 2(a). Notice that such a point exists because p_3 is in $\Delta p_2 p_4$. Denote this first hit by p'_3 (it might be the case that $p'_3 = p_3$). Then p_1, p_2, p'_3, p_4, p_5 is a 5-cup which we denote by K' (again, it might be the case that $K' = K$). Let c' be the intersection point of the rotated ray with ℓ_{12} . Our choice of p'_3 implies that the triangle $\Delta c p_4 c'$ is empty, i.e. its interior has no points of P ; this triangle is denoted by E_3 in Figure 2(a).

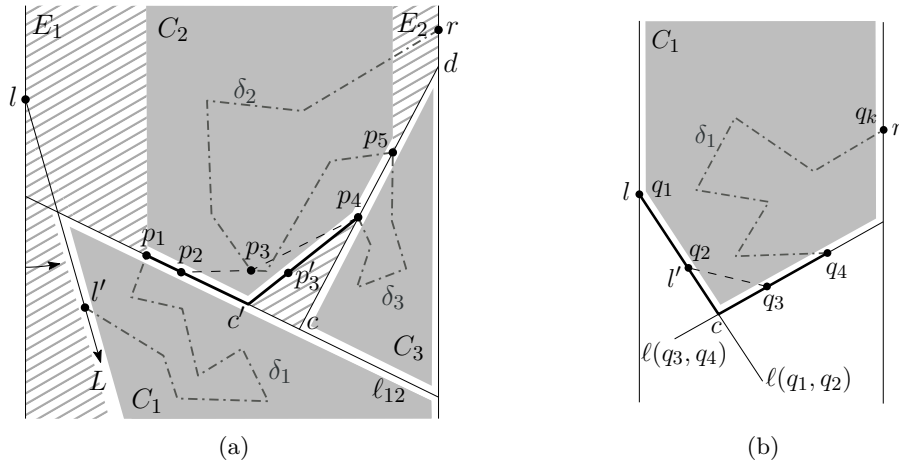


■ **Figure 2** Illustration of the proof of Lemma 2. (a) l lies below ℓ_{12} and r lies below ℓ_{45} . (b) l lies below ℓ_{12} and r lies above ℓ_{45} .

The points of P' lie in the interior or on the boundary of three convex regions C_1, C_2, C_3 as depicted in Figures 2(a) and 2(b). The region C_1 is the intersection of H and the halfplane below ℓ_{12} . The region C_3 is the intersection of H and the halfplane above ℓ_{12} and the halfplane below ℓ_{45} . The region C_2 is the intersection of H and five halfplanes (the halfplanes above the lines $\ell_{12}, \ell_{45}, \ell(c', p_4)$, the halfplane to the right of the vertical line through p_1 , and the halfplane to the left of the vertical line through p_5). Let P_i be the set of points of P in the interior (but not on the boundary) of each C_i . Then $P_1 \cup P_2 \cup P_3 = P \setminus \{l, p_1, p_2, p'_3, p_4, p_5, r\}$, and thus $|P_1| + |P_2| + |P_3| = |P| - 7$.

We construct a covering path for P as follows. The four points p_1, p_2, p'_3, p_4 can be covered by the path (p_1, c', p_4) which has two edges p_1c' and $c'p_4$. Let δ_1 be the noncrossing path with $|P_1| + 1$ edges that is obtained by applying Lemma 1 on (P_1, C_1, l, p_1) where l and p_1 play the roles of p and q in the lemma. We now consider two subcases.

- r lies below ℓ_{45} . In this case r is on the boundary of C_3 , as in Figure 2(a). Let δ_2 and δ_3 be the noncrossing paths with $|P_2| + 1$ and $|P_3| + 1$ edges that are obtained by applying Lemma 1 on (P_2, C_2, p_4, p_5) and (P_3, C_3, p_5, r) , respectively. By interconnecting these paths we obtain a noncrossing covering path $\delta_1 \oplus (p_1, c', p_4) \oplus \delta_2 \oplus \delta_3$ for P . This path has $(|P_1| + 1) + 2 + (|P_2| + 1) + (|P_3| + 1) = |P| - 2$ edges, and it lies in H .
- r lies above ℓ_{45} . In this case r is on the boundary of the convex region $C_2 \cup E_2$, as in Figure 2(b). Let δ_2 and δ_3 be the noncrossing paths obtained by applying Lemma 1 on $(P_2, C_2 \cup E_2, p_5, r)$ and (P_3, C_3, p_4, p_5) , respectively. Then $\delta_1 \oplus (p_1, c', p_4) \oplus \delta_3 \oplus \delta_2$ is a noncrossing covering path for P . This path has $|P| - 2$ edges, and it lies in H .



■ **Figure 3** Illustration of the proof of Lemma 2 where l lies above ℓ_{12} and r lies above ℓ_{45} . (a) $l' \neq p_1$, and (b) $l' = p_1$ and $r' = p_5$ (here $p_1 = q_2$ and $p_5 = q_{k-1}$).

(ii) In this case l lies above ℓ_{12} and r lies above ℓ_{45} . Let L and R be the downward rays emanating from l and r , respectively. Rotate L counterclockwise around l and stop as soon as hitting a point l' of P . Since E_1 is empty, l' is either p_1 or a point below ℓ_{12} ; see Figure 3(a). Rotate R clockwise around r and stop as soon as hitting a point r' of P . Since E_2 is empty, r' is either p_5 or a point below ℓ_{45} . We distinguish two subcases.

- $l' \neq p_1$ or $r' \neq p_5$. Up to symmetry we assume that $l' \neq p_1$ as depicted in Figure 3(a). Define c, c', d, p'_3 and the 5-cup K' as in case (i), and recall that the triangle $\Delta cp_4c'$ is empty. The points of P' lie in the interior or on the boundary of three convex regions C_1, C_2, C_3 as depicted in Figures 3(a). The region C_1 is the intersection of H and the halfplane below ℓ_{12} and the halfplane above $\ell(l, l')$. The regions C_2 and C_3 are defined as in case (i). Let P_i be the set of points of P in the interior (but not on the boundary) of each C_i . Then $P_1 \cup P_2 \cup P_3 = P \setminus \{l, l', p_1, p_2, p'_3, p_4, p_5, r\}$, and thus $|P_1| + |P_2| + |P_3| = |P| - 8$. We cover l and l' by the edge (l, l') and cover the four points p_1, p_2, p'_3, p_4 by the path (p_1, c', p_4) which has two edges. Let δ_1, δ_2 , and δ_3 be the noncrossing paths with $|P_1| + 1, |P_2| + 1$, and $|P_3| + 1$ edges obtained by applying Lemma 1 on (P_1, C_1, l, p_1) , $(P_2, C_2 \cup E_2, p_5, r)$, and (P_3, C_3, p_4, p_5) , respectively; see Figures 3(a). By interconnecting these paths we obtain a noncrossing covering path $(l, l') \oplus \delta_1 \oplus (p_1, c', p_4) \oplus \delta_3 \oplus \delta_2$ for P . This path has $1 + (|P_1| + 1) + 2 + (|P_3| + 1) + (|P_2| + 1) = |P| - 2$ edges, and it lies in H .

- $l' = p_1$ and $r' = p_5$. In this case the lower chain on the boundary of $\text{CH}(P)$ has at least 5 vertices, including l, l', r, r' , and a point in the triangle formed by L, R , and $\ell(p_1, p_5)$. Let $k \geq 5$ be the number of vertices of this chain. Let q_1, q_2, \dots, q_k denote the vertices of this chain that appear in this order from left to right, as in Figure 3(b). Then $q_1 = l, q_2 = l' = p_1, q_k = r$, and $q_{k-1} = r' = p_5$.

Let c be the intersection point of $\ell(q_1, q_2)$ and $\ell(q_3, q_4)$, which lies in H . Then, the four points q_1, q_2, q_3, q_4 can be covered by the path (q_1, c, q_4) . All points of P lie in the interior or on the boundary of a convex region C_1 that is the intersection of H with the halfplanes above $\ell(q_1, q_2)$ and $\ell(q_3, q_4)$; this region is shaded in Figure 3(b). Let P_1 be the points of P that lie in the interior (but not on the boundary) of C_1 . Then $P_1 = P \setminus \{q_1, q_2, q_3, q_4, q_k\}$ and $|P_1| = |P| - 5$. Let δ_1 be the covering path with $|P_1| + 1$ edges that is obtained by applying Lemma 1 on (P_1, C_1, q_4, q_k) where q_4 and q_k play the roles of p and q in the lemma. Then $(q_1, c, q_4) \oplus \delta_1$ is a noncrossing covering path for P . This path has $2 + (|P_1| + 1) = |P| - 2$ edges, and it lies in H .

This is the end of our proof (for P' being in general position).

One can simply adjust the above construction to work even if P' is not in general position. For the sake of completeness here we give a brief description of an alternative (and perhaps simpler) construction when P' has three or more collinear points. Let p_1, p_2, p_3 be three collinear points in P' from left to right and let ℓ_{13} be the line through these points. We choose p_1, p_2, p_3 in such a way that there is no point of P' on ℓ_{13} to the left of p_1 or to the right of p_3 . Up to symmetry we have two cases: (i) l lies on or above ℓ_{13} and r lies on or below ℓ_{13} , and (ii) both l and r lie below ℓ_{13} .

In case (i) we first obtain a path by applying Lemma 1 on l, p_1 and all points above ℓ_{13} . Then we connect p_1 and p_3 by one edge which also covers p_2 . Then we obtain another path by applying Lemma 1 on r, p_3 and all points below ℓ_{13} . This gives a covering path with $|P| - 2$ edges.

In case (ii) we start from l and walk on the vertices of $\text{CH}(P)$ in clockwise direction (and at the same time cover the visited vertices) and stop at the first vertex, say p_0 , for which the next vertex, say x , is on or above ℓ_{13} (it could be the case that $p_0 = l$). Denote the traversed path between l and p_0 by δ_l . First assume that x is above ℓ_{13} . We connect p_0 to x . Then we obtain a path by applying Lemma 1 on x, p_1 and all points above ℓ_{13} . Then we connect p_1 to p_3 by one edge which also covers p_2 . Then we extend the current path to a covering path for P by applying Lemma 1 on p_3, r and the remaining points below ℓ_{13} . Now assume that x is on ℓ_{13} , in which case $x = p_1$. If there is no point of P' above ℓ_{13} then we connect p_1 to p_3 by one edge and then extend it to a covering path for P by applying Lemma 1 on p_3, r and the remaining points below ℓ_{13} . Assume that there are points above ℓ_{13} . We repeat the above process from r by a counterclockwise walk on the vertices of $\text{CH}(P)$, and due to symmetry, assume that p_3 is the first visited vertex that lies on or above ℓ_{13} . Let p_4 denote the vertex of $\text{CH}(P)$ after p_3 . Notice that p_4 lies above ℓ_{13} . Let c be the intersection point of the lines $\ell(p_0, p_1)$ and $\ell(p_3, p_4)$. To obtain a covering path, we start with δ_l , connect its endpoint p_0 to c , and connect c to p_3 ; these two edges cover p_1 and p_4 . Then we continue by a path obtained from Lemma 1 applied on p_3, r and the remaining points. ◀

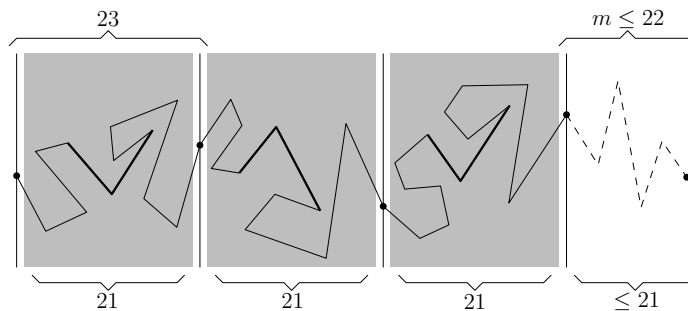
The following corollary, although very simple, will be helpful in the analysis of our algorithm.

► **Corollary 3.** *Let Q be a set of at least 22 points in the plane and let l be its leftmost point. Then there exists a noncrossing covering path for Q with $|Q| - 2$ edges that lies to the right of the vertical line through l and has l as an endpoint.*

Proof. We add a dummy point r to the right of all points in Q . Let $P = Q \cup \{r\}$. We obtain a noncrossing covering path δ for P with $|P| - 2$ edges by Lemma 2. Recall that r is an endpoint of δ . Also recall from the proof of Lemma 2 that in all cases r gets connected to δ by a path that is obtained from Lemma 1. No edge of this path has a point of P in its interior (even if two consecutive edges happen to be collinear, we treat them as two different edges). Thus, the edge of δ that covers r has no point in its interior. Therefore, by removing r and its incident edge from δ we obtain a covering path with $|Q| - 2$ edges for Q that satisfies the conditions of the corollary. ◀

► **Theorem 4.** *Every set of n points in the plane admits a noncrossing covering path with at most $\lceil 21n/22 \rceil - 1$ edges. Thus, $\pi(n) \leq (1 - 1/22)n$. Such a path can be computed in $O(n \log n)$ time.*

Proof. Let P be a set of n points in the plane. After a suitable rotation we may assume that no two points of P have the same x -coordinate. Draw vertical lines in the plane such that each line goes through a point of P , there are exactly 21 points of P between any pair of consecutive lines, no point of P lies to the left of the leftmost line, and at most 21 points of P lie to the right of the rightmost line; see Figure 4. Each pair of consecutive lines defines a vertical strip containing 23 points; 21 points in its interior and 2 points on its boundary (the point on the boundary of two consecutive strips is counted for both strips). For the 23 points in each strip we obtain a noncrossing covering path with 21 edges using Lemma 2. Each path lies in its corresponding strip and its endpoints are the two points on the boundary of the strip. By assigning to each strip the point on its left boundary, it turns out that for every 22 points we get a path with 21 edges.



■ **Figure 4** Illustration of the proof of Theorem 4.

Let m be the number of points on or to the right of the rightmost line, and notice that $m \leq 22$. We distinguish between two cases $m = 22$ and $m < 22$.

If $m = 22$ (in this case n is divisible by 22) then we cover these 22 points by a noncrossing path with 20 edges using Corollary 3. The union of this path and the paths constructed within the strips is a noncrossing covering path for P . The total number of edges in this path is $21n/22 - 1$.

If $m < 22$ then $m = n - 22\lfloor n/22 \rfloor$. In this case we cover the m points by an x -monotone path with $m - 1$ edges (dashed segments in Figure 4). Again, the union of this path and the paths constructed within the strips is a noncrossing covering path for P . The total number of edges in this path is $21\lfloor n/22 \rfloor + m - 1 = n - \lfloor n/22 \rfloor - 1 = \lceil 21n/22 \rceil - 1$.

Each call to Lemma 2 and Corollary 3 takes constant time. Therefore, after rotating and sorting the points in $O(n \log n)$ time, the rest of the algorithm takes linear time. ◀

Our path construction in Theorem 4 achieves a similar bound for covering cycles.

► **Corollary 5.** *Every set of n points in the plane admits a noncrossing covering cycle with at most $\lceil 21n/22 \rceil + 1$ edges. Such a cycle can be computed in $O(n \log n)$ time.*

Proof. Let δ be the path constructed by Theorem 4 on a point set P of size n . Recall m from the proof of this theorem. If $m < 22$ then the two endpoints of δ are the leftmost and rightmost points of P . Thus, by introducing a new point p with a sufficiently large y -coordinate and connecting it to the two endpoints of δ , we obtain a noncrossing covering cycle for P . If $m = 22$ then the dummy point that was introduced in Corollary 3 could be chosen suitably to play the role of p . ◀

3 Noncrossing covering trees

In this section we prove the following theorem which gives an algorithm for computing a noncrossing covering tree with roughly $4n/5$ edges. We should clarify that the number of *edges* of a tree is different from the number of its *segments* (where each segment is either a single edge or a chain of several collinear edges of the tree). For example the tree in Figure 6(b) has 10 edges and 7 segments, where the segments p_1p_7 and p_5p_8 consist of 3 and 2 collinear edges, respectively.

► **Theorem 6.** *Every set of n points in the plane admits a noncrossing covering tree with at most $\lceil 4n/5 \rceil$ edges. Thus, $\tau(n) \leq \lceil 4n/5 \rceil$. Such a tree can be computed in $O(n \log n)$ time.*

Proof. Let P be a set of n points in the plane. After a suitable rotation we may assume that no two points of P have the same x -coordinate. We present an iterative algorithm to compute a noncrossing covering tree for P that consists of at most $\lceil 4n/5 \rceil$ edges. In a nutshell, the algorithm scans the points from left to right and in every iteration (except possibly the last iteration) it considers 4 or 5 new points and covers them with 3 or 4 new edges, respectively. Thus the ratio of the number of new edges to the number of covered points would be at most $4/5$. We begin by describing an intermediate iteration of the algorithm; the first and last iterations will be described later. We assume that the scanned points in each iteration are in general position. In the end of the proof we describe how to handle collinearities. Let m be the number of points that have been scanned so far and let l be the rightmost scanned point (our choice of the letter l will become clear shortly). We maintain the following invariant at the beginning of every intermediate iteration.

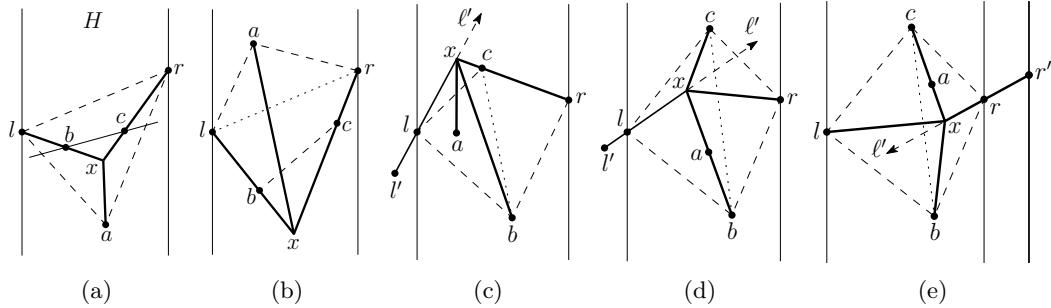
Invariant. All the m points that have been scanned so far, are covered by a noncrossing tree T with at most $4\lceil m/5 \rceil$ edges. The tree T lies to the left of the vertical line through l and the degree of l in T is one.

In the current (intermediate) iteration we scan four new points, namely a, b, c , and r where r is the rightmost point. Let H be the vertical strip bounded by the vertical lines through l and r (l is the leftmost point and r is the rightmost point in H); see Figure 5(a). Let $Q = \{l, a, b, c, r\}$. We consider three cases depending on the number of vertices of $\text{CH}(Q)$. Notice that r and l are two vertices of $\text{CH}(Q)$.

- $\text{CH}(Q)$ has three vertices. Let a be the third vertex of $\text{CH}(Q)$. Then b and c lie in the interior of $\text{CH}(Q)$, as in Figure 5(a). In this case two vertices of $\text{CH}(Q)$, say l and r , lie on the same side of $\ell(b, c)$. Thus l, b, c , and r form a convex quadrilateral. After a suitable relabeling assume that l, b, c, r appear in this order along the boundary of the quadrilateral. Let x be the intersection point of $\ell(l, b)$ and $\ell(r, c)$, which lies in the

19:10 Improved Bounds for Covering Paths and Trees in the Plane

triangle $\triangle lra$. We cover the four scanned points a , b , c , and r by three edges xl , xr , and xa which lie in H . We add these edges to T . The degree of r is one in the new tree (no matter which two vertices of $\text{CH}(Q)$ lay on the same side of $\ell(b, c)$). The invariant holds and we proceed to the next iteration.



■ **Figure 5** Illustration of the proof of Theorem 6. (a) $\text{CH}(Q)$ has three vertices. (b) $\text{CH}(Q)$ has five vertices. (c)-(e) $\text{CH}(Q)$ has four vertices.

- $\text{CH}(Q)$ has five vertices. We explain this case first as our argument is shorter, and also it will be used for the next case. In this case Q contains a 4-cap or a 4-cup with endpoints l and r . After a suitable reflection and relabeling assume it has the 4-cup l, b, c, r as in Figure 5(b). Let x be the intersection point of $\ell(l, b)$ and $\ell(r, c)$, and observe that it lies in H . We cover a, b, c , and r by three edges xl , xr , and xa which lie in H . We add these edges to T . The degree of r is one in the new tree. The invariant holds for the next iteration.
- $\text{CH}(Q)$ has four vertices. After a suitable relabeling assume that b and c are two vertices of $\text{CH}(Q)$ (other than l and r). Thus a lies in the interior of $\text{CH}(Q)$. If both b and c lie above or below $\ell(l, r)$ then l, b, c, r form a 4-cap or a 4-cup, in which case we cover the points as in the previous case. Therefore we may assume that one point, say b , lies below $\ell(l, r)$ and c lies above $\ell(l, r)$ as in Figures 5(c)-(e). We consider two subcases.
 - a lies in the triangle $\triangle lbc$. By the invariant, l has degree one in T . Let l' be the neighboring vertex of l in T . Let ℓ' be the ray emanating from l' and passing through l . We consider two subcases: (i) the segment bc does not intersect ℓ' and (ii) the segment bc intersects ℓ' .
 - In case (i) the segment bc lies below or above ℓ' . By symmetry assume that it lies below ℓ' . Then a and r also lie below ℓ' , as in Figure 5(c). In this case $\ell(r, c)$ intersects ℓ' . Let x be their intersection point, and observe that it lies in H . We replace the edge $l'l$ of T by $l'x$ (this does not increase the number of edges because l has degree one). Notice that $l'x$ contains l . Then we cover a, b, c , and r by adding three edges xr , xb , and xa to T . Therefore the number of edges of T is increased by 3. Moreover, r has degree one in the new tree, and all the newly introduced edges lie to the left of the vertical line through r . Thus the invariant holds for the next iteration.
 - In case (ii) the ray ℓ' goes through $\triangle lbc$. The point a lies below or above ℓ' . By symmetry assume that it lies below ℓ' , as in Figure 5(d). Let x be the intersection point of $\ell(a, b)$ and ℓ' , which lies in $\triangle lbc$. We replace the edge $l'l$ of T by $l'x$. Then we cover a, b, c , and r by adding three edges xr , xb , and xc to T . Thus, the number of edges of T is increased by 3, the vertex r has degree one in the new tree, and all new edges lie to the left of the vertical line through r . The invariant holds for the next iteration.

- *a* lies in the triangle $\triangle rbc$. Here is the place where we use four new edges to cover five vertices. In fact the ratio $4/5$ comes from this case (In previous cases we were able to cover four points by three new edges). In this case we scan the next point after r which we denote by r' , as in Figure 5(e). Now let ℓ' be the ray emanating from r' and passing through r . The current setting is essentially the vertical reflection of the previous case where r and r' play the roles of l and l' , respectively. We handle this case analogous to the previous case. Our analysis is also analogous except that now we consider the edge $r'x$ as a new edge. Thus we use four new edges to cover five points a, b, c, r , and r' . All new edges lie to the left of the vertical line through r' , and the degree of r' is one in the new tree. Thus the invariant holds for the next iteration.

This is the end of an intermediate iteration. The noncrossing property of the resulting tree follows from our construction. This iteration suggests a covering tree with roughly $4n/5$ edges. To get the exact claimed bound we need to have a closer look at the first and last iterations of the algorithm.

For the first iteration of the algorithm we scan only the leftmost input point. This point will play the role of l for the second iteration (which is the first intermediate iteration). The invariant holds for the second iteration because the tree has no edges at this point. If we happen to use the edge $l'l$ in the second iteration, then we take $l' = l$ and give the ray ℓ' an arbitrary direction to the right. Based on the above construction this could happen only when we scan four points (a, b, c, r) in the second iteration. In this case the first five points (l, a, b, c, r) are covered by four edges, and thus the invariant holds for the following iteration. In the last iteration of the algorithm we are left with $w \leq 4$ points that are not being scanned. We connect these w points by w edges to the rightmost scanned point. Therefore, the algorithm covers all points by a noncrossing tree with at most $\lceil 4n/5 \rceil$ edges.

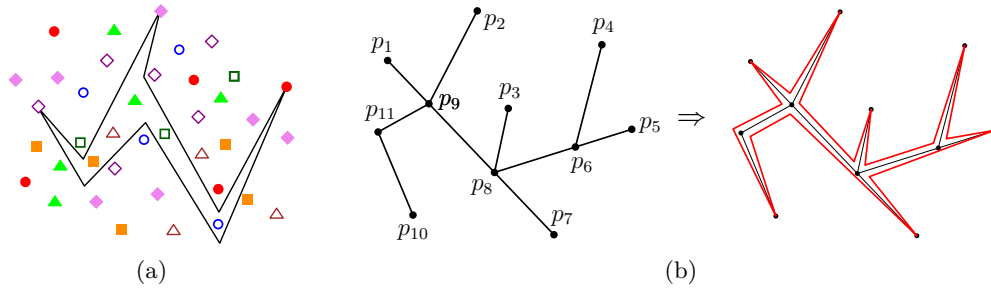
If three or more of the scanned points are collinear then cover all collinear points by one edge and connect the left endpoint of this edge to l . Then we connect every remaining scanned point to l . The number of new edges is at most 3 (for 4 scanned points) and 4 (for 5 scanned points).

Each iteration takes constant time. Therefore, after rotating and sorting the points in $O(n \log n)$ time, the rest of the algorithm takes linear time. ◀

4 Perfect rainbow polygons

Recall that a perfect rainbow polygon for a set of colored points, is a simple polygon that contains exactly one point of each color in its interior or on its boundary. Figure 6(a) shows a perfect rainbow polygon of size 9 (nine edges) for an 8-colored point set (i.e. colored by 8 different colors). There is a relation (as described below) between rainbow polygons and noncrossing covering trees. We employ this relation (similar to [19]) and present an algorithm that achieves a perfect rainbow polygon of size at most $7k/5 + O(1)$ for any k -colored point set.

We adopt the following notation and definitions from [19]. Let T be a noncrossing geometric tree. Recall that a segment of T is a chain of collinear edges in T . Let M be a partition of the edges of T into a minimal number of pairwise noncrossing segments. Let s denote the number of segments in M . A *fork* of T (with respect to M) is a vertex f that lies in the interior of a segment $ab \in M$ and it is an endpoint of another segment of M . The *multiplicity* of f is a number in $\{1, 2\}$ that is determined as follows. If the segments that have f as an endpoint lie on both sides of $\ell(a, b)$ then f has multiplicity 2, otherwise (the



■ **Figure 6** (a) A perfect rainbow polygon of size 9 for an 8-colored point set. (b) Left: A noncrossing tree with ten edges that can be partitioned into seven segments $\{p_1p_7, p_2p_9, p_3p_8, p_4p_6, p_5p_8, p_{10}p_{11}, p_{11}p_9\}$; p_6 and p_8 are two forks with multiplicity 1 and p_9 is a fork with multiplicity 2. Right: Obtaining a simple enclosing polygon from the tree.

segments lie on one side of the line) f has multiplicity 1. See the tree in Figure 6(b) for an example. Let t denote the sum of multiplicities of all forks in T . The following lemma expresses the size of a polygon enclosing T in terms of s and t .

► **Lemma 7** (Flores-Peñaloza et al. [19]). *Let T be a noncrossing geometric tree and M be a partition of its edges into a minimal number of pairwise noncrossing segments. Let s be the number of segments in M and t be the total multiplicity of forks in T . If $s \geq 2$ and $t \geq 0$, then for every $\varepsilon > 0$ there exists a simple polygon of size $2s + t$ and of area at most ε that encloses T .*

There are simple intuitions behind Lemma 7. For example if we cut out the tree T from the plane, then the resulting hole could be expressed as a desired polygon. Alternatively, if we start from a vertex of T and walk around T (arbitrary close to its edges) until we come back to the starting vertex, then the traversed tour could be represented as a desired polygon. See Figure 6(b).

In view of Lemma 7, a better covering tree (i.e. for which $2s + t$ is smaller) leads to a better polygon (i.e. with fewer edges). The following theorem (proven in the full version of the paper at <https://arxiv.org/abs/2303.04350>) gives a covering tree for which $2s + t$ is smaller (compared to that of [19]). Our construction in the proof of this theorem shares some similarities with the construction in our proof of Theorem 6. However, the details of the two constructions are different because they have different objectives.

► **Theorem 8.** *Let K be a set of k points in the plane in general position. Then, in $O(k \log k)$ time, one can construct a noncrossing covering tree for K consisting of at most $\lceil \frac{3k}{5} \rceil + 2$ pairwise noncrossing segments with at most $\lceil \frac{k}{5} \rceil$ forks of multiplicity 1.*

With this lemma and theorem in hand, we present our algorithm for computing a perfect rainbow polygon.

Algorithm. (in a nutshell). Let P be a set of n points in the plane in general position that are colored by k distinct colors. The algorithm picks one point from each color (arbitrarily), covers the chosen points by a noncrossing tree (using Theorem 8), and then obtains a perfect rainbow polygon from the tree (using Lemma 7).

Analysis. Let K be the set of k chosen points, and let T be the covering tree for K obtained by Theorem 8. Then $s \leq \lceil \frac{3k}{5} \rceil + 2$ and $t \leq \lceil \frac{k}{5} \rceil$. Thus, the perfect rainbow polygon obtained by Lemma 7 has size

$$2s + t \leq 2 \left(\left\lceil \frac{3k}{5} \right\rceil + 2 \right) + \left\lceil \frac{k}{5} \right\rceil \leq \left\lceil \frac{7k}{5} \right\rceil + 6.$$

The tree T can be obtained in $O(k \log k)$ time, by Theorem 8. To obtain a polygon (avoiding points of $P \setminus K$) from T we need to choose a suitable ε in Lemma 7. As noted in [19], half of the minimum distance between the edges of T and the points of $P \setminus K$ is a suitable ε , which can be found in $O(n \log n)$ time by computing the Voronoi diagram of the edges of T together with the points of $P \setminus K$. Thus the total running time of the algorithm is $O(n \log n)$. The following theorem summarizes our result in this section.

► **Theorem 9.** *Every k -colored point set of size n in the plane in general position admits a perfect rainbow polygon of size at most $\lceil 7k/5 \rceil + 6$. Thus, $\rho(k) \leq \lceil 7k/5 \rceil + 6$. Such a polygon can be computed in $O(n \log n)$ time.*

Remark. The general position assumption is necessary for our algorithm because if a non-selected point (i.e. a point of $P \setminus K$) lies on a segment of T then the resulting polygon is not a valid rainbow polygon as it contains two or more points of the same color.

5 Concluding remarks

A natural open problem is to improve the presented upper bounds or the known lower bounds for $\pi(n)$, $\tau(n)$, and $\rho(k)$. Here are some directions for further improvements:

- For the proof of Lemma 2 we used a 5-cap or a 5-cup which forced us to scan 21 points in each iteration (due to the result of Erdős and Szekeres). If one could manage to use a 4-cap or a 4-cup instead, then it could improve the upper bound for $\pi(n)$ further.
- Our iterative algorithm in the proof of Theorem 6, covers 4 points by 3 edges in all cases except in the last case (where $\text{CH}(Q)$ has four vertices and a lies in $\triangle rbc$) for which it covers 5 points by 4 edges. The upper bound $4n/5$ for $\tau(n)$ comes from this case. If one could argue that this case won't happen often (for example by showing that it won't happen in three consecutive iterations or by choosing a different ordering for points), then it would lead to a slightly improved upper bound for $\tau(n)$.

References

- 1 Alok Aggarwal, Don Coppersmith, Sanjeev Khanna, Rajeev Motwani, and Baruch Schieber. The angular-metric traveling salesman problem. *SIAM Journal on Computing*, 29(3):697–711, 1999. Also in *SODA '97*.
- 2 Oswin Aichholzer, Sergio Cabello, Ruy Fabila Monroy, David Flores-Peñaloza, Thomas Hackl, Clemens Huemer, Ferran Hurtado, and David R. Wood. Edge-removal and non-crossing configurations in geometric graphs. *Discrete Mathematics & Theoretical Computer Science*, 12(1):75–86, 2010.
- 3 Oswin Aichholzer, Frank Duque, Ruy Fabila Monroy, Oscar E. García-Quintero, and Carlos Hidalgo-Toscano. An ongoing project to improve the rectilinear and the pseudolinear crossing constants. *Journal of Graph Algorithms and Applications*, 24(3):421–432, 2020.
- 4 Esther M. Arkin, Yi-Jen Chiang, Joseph S. B. Mitchell, Steven Skiena, and Tae-Cheon Yang. On the maximum scatter traveling salesperson problem. *SIAM Journal on Computing*, 29(2):515–544, 1999. Also in *SODA '97*.

- 5 Esther M. Arkin, Joseph S. B. Mitchell, and Christine D. Piatko. Minimum-link watchman tours. *Information Processing Letters*, 86(4):203–207, 2003.
- 6 Sanjeev Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *Journal of the ACM*, 45(5):753–782, 1998.
- 7 Sergey Bereg, Prosenjit Bose, Adrian Dumitrescu, Ferran Hurtado, and Pavel Valtr. Traversing a set of points with a minimum number of turns. *Discrete & Computational Geometry*, 41(4):513–532, 2009. Also in *SoCG'07*.
- 8 Ahmad Biniiaz. Euclidean bottleneck bounded-degree spanning tree ratios. *Discrete & Computational Geometry*, 67(1):311–327, 2022. Also in *SODA'20*.
- 9 Jakub Cerný, Zdenek Dvorák, Vít Jelínek, and Jan Kára. Noncrossing Hamiltonian paths in geometric graphs. *Discrete Applied Mathematics*, 155(9):1096–1105, 2007.
- 10 Timothy M. Chan. Euclidean bounded-degree spanning tree ratios. *Discrete & Computational Geometry*, 32(2):177–194, 2004. Also in *SoCG 2003*.
- 11 Jianer Chen, Qin Huang, Iyad Kanj, and Ge Xia. Near-optimal algorithms for point-line covering problems. *CoRR*, abs/2012.02363, 2020.
- 12 Michael J. Collins. Covering a set of points with a minimum number of turns. *International Journal of Computational Geometry & Applications*, 14(1-2):105–114, 2004.
- 13 Éva Czabarka, Ondrej Sýkora, László A. Székely, and Imrich Vrto. Biplanar crossing numbers. II. Comparing crossing numbers and biplanar crossing numbers using the probabilistic method. *Random Structures & Algorithms*, 33(4):480–496, 2008.
- 14 Erik D. Demaine and Joseph O'Rourke. Open problems from CCCG 2010. In *Proceedings of the 22nd Canadian Conference on Computational Geometry*, 2011.
- 15 Adrian Dumitrescu, Dániel Gerbner, Balázs Keszegh, and Csaba D. Tóth. Covering paths for planar point sets. *Discrete & Computational Geometry*, 51(2):462–484, 2014.
- 16 P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compositio Mathematica*, 2:463–470, 1935.
- 17 Sándor P. Fekete and Henk Meijer. On minimum stars and maximum matchings. *Discrete & Computational Geometry*, 23(3):389–407, 2000. Also in *SoCG 1999*.
- 18 Sándor P. Fekete and Gerhard J. Woeginger. Angle-restricted tours in the plane. *Computational Geometry: Theory and Applications*, 8:195–218, 1997.
- 19 David Flores-Peñaloza, Mikio Kano, Leonardo Martínez-Sandoval, David Orden, Javier Tejel, Csaba D. Tóth, Jorge Urrutia, and Birgit Vogtenhuber. Rainbow polygons for colored point sets in the plane. *Discrete Mathematics*, 344(7):112406, 2021.
- 20 Radoslav Fulek, Balázs Keszegh, Filip Morić, and Igor Uljarević. On polygons excluding point sets. *Graphs and Combinatorics*, 29(6):1741–1753, 2013.
- 21 Magdalene Grantson and Christos Levcopoulos. Covering a set of points with a minimum number of lines. In *Proceedings of the 6th International Conference on Algorithms and Complexity (CIAC)*, pages 6–17, 2006.
- 22 Frank Harary and Anthony Hill. On the number of crossings in a complete graph. *Proceedings of the Edinburgh Mathematical Society*, 13:333–338, 1963.
- 23 Alexandr O. Ivanov and Alexey A. Tuzhilin. The Steiner ratio Gilbert–Pollak conjecture is still open: Clarification statement. *Algorithmica*, 62(1-2):630–632, 2012.
- 24 Minghui Jiang. On covering points with minimum turns. *International Journal of Computational Geometry & Applications*, 25(1):1–10, 2015.
- 25 Balázs Keszegh. Covering paths and trees for planar grids. *Geombinatorics Quarterly*, 24, 2014.
- 26 Stefan Langerman and Pat Morin. Covering things with things. *Discrete & Computational Geometry*, 33(4):717–729, 2005. Also in *ESA'02*.
- 27 Samuel Loyd. *Cyclopedia of 5000 Puzzles, Tricks & Conundrums*. The Lamb Publishing Company, 1914.
- 28 János Pach, Natan Rubin, and Gábor Tardos. Planar point sets determine many pairwise crossing segments. *Advances in Mathematics*, 386:107779, 2021. Also in *STOC'19*.

- 29 Christos H. Papadimitriou. The Euclidean traveling salesman problem is NP-complete. *Theoretical Computer Science*, 4(3):237–244, 1977.
- 30 Clifford Stein and David P. Wagner. Approximation algorithms for the minimum bends traveling salesman problem. In *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 406–422, 2001.

Sparse Higher Order Čech Filtrations

Mickaël Buchet

Institute of Geometry, TU Graz, Austria

Bianca B. Dornelas ✉

Institute of Geometry, TU Graz, Austria

Michael Kerber ✉

Institute of Geometry, TU Graz, Austria

Abstract

For a finite set of balls of radius r , the k -fold cover is the space covered by at least k balls. Fixing the ball centers and varying the radius, we obtain a nested sequence of spaces that is called the k -fold filtration of the centers. For $k = 1$, the construction is the union-of-balls filtration that is popular in topological data analysis. For larger k , it yields a cleaner shape reconstruction in the presence of outliers. We contribute a sparsification algorithm to approximate the topology of the k -fold filtration. Our method is a combination and adaptation of several techniques from the well-studied case $k = 1$, resulting in a sparsification of linear size that can be computed in expected near-linear time with respect to the number of input points.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Computational complexity and cryptography

Keywords and phrases Sparsification, k -fold cover, Higher order Čech complexes

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.20

Related Version *Full Version*: <http://arxiv.org/abs/2303.06666> [9]

Funding This research has been supported by the Austrian Science Fund (FWF), grant numbers P 33765-N and W1230.

Acknowledgements We thank Alexander Rolle for his valuable input and encouragement of this work.

1 Introduction

Persistent homology [10,22,23] is a major branch of topological data analysis with applications, for instance, in shape recognition [5], material science [34] and biology [29,35]. It studies the homological properties of sequences of topological spaces. A standard construction is to take the homogeneous union of balls, with increasing radius, centered at finitely many points of \mathbb{R}^d . We call these points *sites* and refer to that filtration as the *union-of-balls filtration*. For computational purposes, one considers the homologically equivalent *Čech filtration*, which is a sequence of simplicial complexes that captures the intersection patterns of the balls in the union-of-balls filtration [22, Chap.3; 30].

The drawback of the Čech filtration (as well as of the closely-related Vietoris-Rips filtration) is that for n sites, it consists of up to $\binom{n}{m+1}$ m -simplices because every $(m+1)$ -subset of balls intersects at a sufficiently large radius. A technique to overcome this large size is to *approximate* the Čech (or Vietoris-Rips) filtration with another, much smaller simplicial filtration with similar topological properties. Technically, that means that the persistence modules induced by the homology of the Čech filtration and its approximation are ϵ -interleaved for an arbitrary $\epsilon > 0$ [12]. Several strategies have been devised to construct such approximations with total size linear in n for any fixed ϵ (see related work). Many of these approaches work by selecting only a subset of the simplices of the Čech filtration, in which case we refer to the approximation as a *sparsification*.



© Mickaël Buchet, Bianca B. Dornelas, and Michael Kerber;
licensed under Creative Commons License CC-BY 4.0

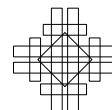
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 20; pp. 20:1–20:17

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



20:2 Sparse Higher Order Čech Filtrations

The union-of-balls filtration is a special case of the k -fold filtration built upon the k -fold cover. For n sites in \mathbb{R}^d and $k \geq 1$ fixed, the k -fold cover is the subset of \mathbb{R}^d consisting of points contained in at least k balls of radius r centered at the sites. Besides being a natural extension, k -fold filtrations are tightly related with the k th neighbor distance that arises in the context of outlier removal and processing of non-homogeneous data densities [13, 25, 42, 44]. For that reason, they have received increased attention recently, both with regards to computational [17, 24] and structural aspects [4].

For fixed k , the k -fold filtration can be equivalently expressed by its nerve, which is a simplicial filtration called the k th order Čech filtration. It captures the intersection patterns of all k -wise intersections of balls, which we call *lenses*. The aforementioned size issue for Čech filtrations is even more important in the k th order case: the filtration is defined over $\binom{n}{k}$ vertices (one for each k -subset of sites) and consequently consists of $\binom{\binom{n}{k}}{m+1}$ m -simplices, making it unrealistic to compute even for small values of n . Therefore we need to reduce its size considerably while maintaining a good approximation quality.

Contributions. We propose the first sparsification of the k -fold filtration for a fixed k . It is a simplicial filtration that, for n sites in \mathbb{R}^d (with constant d) and a given parameter $\epsilon > 0$, is (multiplicatively) $(1 + \epsilon)$ -interleaved with the k -fold filtration. Moreover, the number of m -simplices in our sparsification is

$$\mathcal{O} \left(nk^{k(m+1)} \left(\frac{96}{\epsilon} \right)^{\delta k(m+1)} \right), \quad (1)$$

where δ is the doubling dimension of \mathbb{R}^d . We point out that for constant k and ϵ , the size of the filtration is linear in the number of sites. This is remarkable because the k th order Čech filtration, which captures the k -fold filtration exactly, already contains $\binom{n}{k}$ vertices. Hence our construction avoids including the vast majority of lenses into the sparsification.

We give an output-sensitive algorithm to compute our sparsification up to dimension m_{max} in

$$\mathcal{O} \left(nk \log n \log \Phi + X k^{k+1} \left(\frac{96}{\epsilon} \right)^{k\delta} \cdot m_{max} \right)$$

expected time. Here Φ is the spread of the point set (i.e., the ratio of diameter and smallest distance of two distinct points) and X is the size of the output complex, upper bounded by (1) with m replaced by m_{max} . Again considering everything but n as constant, we get a running time of $\mathcal{O}(n \log n)$.

Techniques and related work. The seminal work by Sheehy [45] was the first one to introduce a sparsification technique for Vietoris-Rips filtrations yielding linear size and $\mathcal{O}(n \log n)$ running time (assuming all other parameters as constant). His technique extends to Čech complexes as well with minor adaptations. Subsequent work [6, 8, 11, 19, 46] introduces several extensions, variations, and simplifications of Sheehy's original sparsification; all these works share essentially the same size and complexity bounds.

Our results are achieved by combining several of these techniques used for approximating in the case $k = 1$, which required non-trivial adaptation for larger values of k . The main idea is that for every site p , we define a *removal radius* such that, for radii larger than this removal radius, all lenses involving p are ignored. That means, for larger and larger radii, we construct simplicial complexes with fewer and fewer sites to keep the size small. To determine

the removal radii of sites, we introduce the *k-distance permutation* which is an ordering of the sites based on the distance to the k th closest neighbor. The k -distance permutation is a generalization of the farthest point sampling [31] used in some sparsification schemes [19, 46] and induces covering and packing properties analogous to those of nets.

Although we opted to extend sparsification techniques, there is an alternative line of research by Choudhary et al. [14–16] that defines approximations of Čech complexes which are not sparsifications. They arrive at slightly improved bounds than the sparsification for $k = 1$. Approximate filtrations are also actively researched in practice [3, 7, 20, 37, 41].

The k -fold cover and the higher order Čech complexes are also studied with relation to *multiparameter persistence*: considering the order k as a second varying parameter, we obtain the *multicover bifiltration*. Blumberg and Lesnick [4] survey different multiparameter persistence approaches and show a particularly strong stability result for multicovers. Sheehy [44] introduces the barycentric bifiltrations, which is equivalent to the multicover but whose size is prohibitively large. The question of computing the multicover bifiltration exactly has been studied by Edelsbrunner and Osang [25], whose results have been refined by Corbet et al. [17]. The latter authors obtain an equivalent bifiltration to the multicover one but has total size (over all choices of k) $\mathcal{O}(n^{d+1})$ for n points in \mathbb{R}^d [17, Prop. 5]. Their construction rely on using *higher order Voronoi diagrams* and *Delaunay complexes* [26]. That reduces the size of Čech complexes, but cannot lead to linear size without further improvements: the Delaunay filtration's d -skeleton is of size $\mathcal{O}(n^{\lceil d/2 \rceil})$ [43], which is a substantial improvement over the $\mathcal{O}(n^d)$ size of the Čech d -skeleton, but still super-linear for $d \geq 3$. Our approximation foregoes those constructions to reduce the size dependency on n further, with the trade-off that we get an exponential dependency on k .

The size reduction in our construction is a consequence of ignoring lenses after their removal radius. The idea of removing a lens beyond a certain radius is justified geometrically by the fact that the remaining lenses cover its entire area after a certain radius. This is only true, however, if we *freeze* a lens before removing it, that is, keep it unchanged for a short time while the surrounding lenses keep growing. This concept was already introduced in [11], from where we also adapt the elegant technique of lifting the lenses to convex *cones* in $\mathbb{R}^d \times \mathbb{R}$. The additional dimension, which is the radius r , is needed because removing simplices is not possible in filtrations.

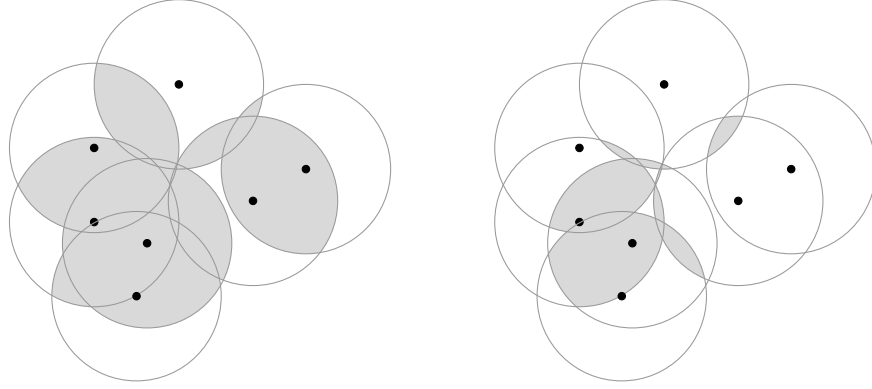
The major geometric predicate for our computation is whether a set of balls is intersecting, which can be dualized to computing the radius of the minimal enclosing ball of the ball's centers [21, 27, 28, 38]. However, the aforementioned freezing of lenses makes this problem technically more challenging. This question seems to be unaddressed in previous work, and we give an efficient solution in the Euclidean setting.

Outline. Section 2 provides background definitions and results. Section 3 defines a k -distance and uses it to construct the k -distance permutation of a point set P . In Section 4 the permutation is used to define a sparse lens filtration that approximates the k -fold cover. That results in a nerve filtration that approximates the k th order Čech complex, as shown in Section 5. The size bound of that filtration is given in Section 6. Section 7 provides an algorithm for computing the discrete sparse Čech filtration. We conclude with Section 8.

2 Background

Lenses and k -fold covers. Given a point set $P \subseteq \mathbb{R}^d$ and a fixed $k \in \mathbb{N}$, an element $p \in P$ is called a *site* and a *k-subset* of P is a subset with k sites. Let $\binom{P}{k}$ be the collection of all k -subsets of P and $A \in \binom{P}{k}$. Let also $B_r(a)$ denote the closed ball centered at a of radius r .

20:4 Sparse Higher Order Čech Filtrations



■ **Figure 1** Example of 2- (left) and 3-fold (right) covers for a fixed radius.

The *lens* corresponding to the k -subset A at scale r is

$$L_r(A) := \bigcap_{a \in A} B_r(a).$$

The k -fold cover of P at scale r is the union of lenses at scale r over all k -subsets:

$$L(r, P) := \bigcup_{A \in \binom{P}{k}} L_r(A).$$

See Figure 1 for an example. Note the omission of k , which we consider fixed, in the notation. When P is clear from context, we drop it from the notation as well and write L_r instead.

Nerves. We assume that the reader is familiar with (abstract) simplicial complexes [22, Chap. 3]. For a finite collection C of subsets of \mathbb{R}^d , we can define a simplicial complex with vertex set C , called the *nerve* of C , as the set of all subsets of C that have a non-empty mutual intersection. Note that the nerve can contain simplices of larger dimension than d . The nerve of the set of all lenses of k -subsets of P at scale r is called the k th order Čech complex with radius r over P , denoted by $\check{\text{Cech}}_r(P, k)$.

Filtrations and equivalence. A collection of topological spaces (e.g., subsets of \mathbb{R}^d) $\mathcal{C} = \{C_r\}_{r \geq 0}$ is called a *filtration* if for all $r \leq r'$, it holds that $C_r \subseteq C_{r'}$. The letter r denotes the *scale parameter* of the filtration. For P and k fixed, the previous concepts yield two different ways of obtaining filtrations. On the one hand, since $L_r \subseteq L_{r'}$ for $r \leq r'$, we get the k -fold filtration $\mathcal{L} := \{L_r\}_{r \geq 0}$. On the other hand, we observe that $\check{\text{Cech}}_r(P, k)$ is a subcomplex of $\check{\text{Cech}}_{r'}(P, k)$ for $r \leq r'$ and hence we get the k th order Čech filtration $\{\check{\text{Cech}}_r(P, k)\}_{r \geq 0}$.

Let \mathcal{C} and \mathcal{D} be two filtrations. We say that \mathcal{C} is (*homotopy*) *equivalent* [40, Chap. 9] to \mathcal{D} if there exists a family of maps $\{f_r : C_r \rightarrow D_r\}_{r \geq 0}$ that are homotopy equivalences of spaces and additionally commute with the inclusion maps of \mathcal{C} and \mathcal{D} .

Interleaving and approximations. Let $\epsilon \geq 0$. Two filtrations \mathcal{C} and \mathcal{D} are (*multiplicatively*) $(1 + \epsilon)$ -*interleaved* if there exist families of linear maps f, g such that the diagram

$$\begin{array}{ccccc} C_r & \xrightarrow{\quad} & C_{r(1+\epsilon)^2} & \xrightarrow{\quad} & C_{r(1+\epsilon)^4} \\ & \searrow f_r & \nearrow g_{r(1+\epsilon)} & \searrow f_{r(1+\epsilon)^2} & \nearrow g_{r(1+\epsilon)^3} \\ & & D_{r(1+\epsilon)} & \xrightarrow{\quad} & D_{r(1+\epsilon)^3} \end{array}$$

commutes for all r . Informally, interleaved filtrations with small ϵ are good approximations of each other because every D_r sits in between two instances of \mathcal{C} with close-by scale parameters.

If $\{C_r\}_{r \geq 0}$, $\{D_r\}_{r \geq 0}$ are $(1 + \epsilon_1)$ -interleaved and $\{D_r\}_{r \geq 0}$, $\{E_r\}_{r \geq 0}$ are $(1 + \epsilon_2)$ -interleaved, then $\{C_r\}_{r \geq 0}$ and $\{E_r\}_{r \geq 0}$ are $(1 + \epsilon_1)(1 + \epsilon_2)$ -interleaved. Moreover, if $C_r \subseteq D_r \subseteq C_{r(1 + \epsilon)}$ for all $r \geq 0$, then \mathcal{C} and \mathcal{D} are $(1 + \epsilon)$ -interleaved.

A filtration \mathcal{C} is a $(1 + \epsilon)$ -approximation of another filtration \mathcal{D} if there exist filtrations \mathcal{C}' and \mathcal{D}' such that \mathcal{C}' is equivalent to \mathcal{C} , \mathcal{D}' is equivalent to \mathcal{D} and \mathcal{C}' and \mathcal{D}' are $(1 + \epsilon)$ -interleaved. This is a symmetric relationship, so we can say that \mathcal{C} and \mathcal{D} are $(1 + \epsilon)$ -approximate. If additionally $C_r \subseteq D_r$ for all $r \geq 0$ we call \mathcal{C} a $(1 + \epsilon)$ -sparsification of \mathcal{D} . We point out that an approximation between two filtrations implies interleaved persistence modules (see [22, Chap. 7]) in the sense of [12].

The Persistent Nerve Theorem. Consider a finite index set I and a family of filtrations $\{U_r^{(i)}\}_{r \geq 0}$ over \mathbb{R}^d , one for each $i \in I$. The union filtration is $\{U_r\}_{r \geq 0}$, where $U_r := \bigcup_{i \in I} U_r^{(i)}$, and the nerve filtration is $\{N_r\}_{r \geq 0}$, where N_r is the nerve of U_r . The Persistent Nerve Theorem [2, Thm. 3.9] states that if every $U_r^{(i)}$ is closed and convex, then $\{U_r\}_{r \geq 0}$ and $\{N_r\}_{r \geq 0}$ are equivalent. As a consequence, the Persistent Nerve Theorem implies that the k -fold and the k th order Čech filtrations are equivalent: choose I as the set of all k -subsets of P and $U_r^{(i)}$ as the lens indexed by i at radius r , which is a closed and convex set.

Doubling dimension. The doubling constant Δ of \mathbb{R}^d is such that any ball of radius r can be covered with at most Δ balls of radius $r/2$, for all $r \geq 0$. The doubling dimension of \mathbb{R}^d is $\delta := \log_2 \Delta$, which is of order $\Theta(d)$ and hence constant for this paper. Note that for finite point sets in \mathbb{R}^d the doubling dimension can be significantly smaller than d , for instance if the points all lie close to a low-dimensional subspace.

To cover a ball B of radius r with balls of radius $r/4$, one needs at most Δ^2 balls; with balls of radius $r/8$ one needs Δ^3 balls and so on. Thus, to cover B with balls of radius r' , we have to find the smallest t such that $r/2^t \leq r'$. That is $t = \lceil \log_2 r/r' \rceil$. Then, $\Delta^t \leq \Delta^{\log_2 r/r'+1} = 2^\delta (r/r')^\delta$ and $(2r/r')^\delta$ balls of radius r' are sufficient to cover B .

Quadtreaps. A quadtreap [39] is a dynamic data structure for spherical range search. We summarize its properties in a simplified form suitable for us: for a set X of n points in \mathbb{R}^d (with d constant), it can be built in $\mathcal{O}(n \log n)$ expected time. It supports deletions of points in X in expected $\mathcal{O}(\log n)$ time. Moreover, given a query point q and a radius r , it returns a list $S \subseteq X$ which is guaranteed to contain all points in X of distance $\leq r$ from q , and is guaranteed not to contain any point in X of distance $\geq 2r$ from q . The running time for such a query is $\mathcal{O}(\log n + |S|)$.

3 k -distance permutation

Given some integer $k \geq 1$ and a finite data set $P \subseteq \mathbb{R}^d$ of $n \geq k$ sites, we define an order on the points in P in which the sites are denoted by p_1, \dots, p_n . Writing $P_i := \{p_1, \dots, p_i\}$, our order ensures that the k -fold cover over P_i approximates the k -fold cover over P , with increasing approximation quality when i increases.

The k -distance of $x \in \mathbb{R}^d$ to P , denoted by $d^k(x, P)$, is the distance from x to its k th closest neighbor in P . We define the k -distance permutation incrementally as follows: we choose p_1, \dots, p_k as arbitrary, pairwise distinct sites from P . If p_1, \dots, p_{i-1} are chosen for

20:6 Sparse Higher Order Čech Filtrations

$k < i \leq n$, we set

$$p_i := \operatorname{argmax}_{q \in P \setminus P_{i-1}} d^k(q, P_{i-1}).$$

Note that for $k = 1$, we obtain the well-known farthest point sampling. We also define

$$\lambda_i := d^k(p_i, P_{i-1})$$

for $k + 1 \leq i \leq n$ and set $\lambda_1, \dots, \lambda_k$ to ∞ , so that the sequence $(\lambda_1, \lambda_2, \dots, \lambda_n)$ is non-increasing. The next two properties of the k -distance permutation are reminiscent of the packing and covering properties of ϵ -nets [47, Chap. 14].

► **Lemma 1 (Covering).** *For all $k \leq i \leq n - 1$, we have $L(r, P_i) \subseteq L(r, P) \subseteq L(r + \lambda_{i+1}, P_i)$.*

Proof. Recall the notation $L_r = L(r, P)$. $P_i \subseteq P$ immediately implies $L(r, P_i) \subseteq L_r$. Consider $x \in L_r$. Then, $x \in L_r(A)$ for some $A = \{a_1, a_2, \dots, a_k\} \subseteq P$. If $A \subseteq P_i$, the result follows. Otherwise, without loss of generality let $a_1 \notin P_i$. By definition of λ_{i+1} , $d^k(a_1, P_i) \leq \lambda_{i+1}$ and hence there are sites $b_1, b_2, \dots, b_k \in P_i$ with $d(a_1, b_j) \leq \lambda_{i+1}$ for all $1 \leq j \leq k$. Consequently, $d(x, b_j) \leq d(x, a_1) + d(a_1, b_j) \leq r + \lambda_{i+1}$ and the k closest sites to x in P_i are within distance $r + \lambda_{i+1}$ of x , implying $x \in L(r + \lambda_{i+1}, P_i)$. ◀

► **Lemma 2 (Packing).** *For all $k + 1 \leq i \leq n$, each $p \in P_i$ has $d^k(p, P_i \setminus \{p\}) \geq \lambda_i/2$.*

Proof. We do induction on i . For $i = k + 1$, let q be the k th closest neighbor of p_{k+1} in P_k . We have $d^k(p_{k+1}, P_{k+1} \setminus \{p_{k+1}\}) = \lambda_{k+1} \geq \lambda_{k+1}/2$ and, for any $p \in P_{k+1} \setminus \{p_{k+1}\}$,

$$d^k(p, P_{k+1} \setminus \{p\}) = \max_{p' \in P_{k+1} \setminus \{p\}} d(p, p') \geq \frac{d(p, q) + d(p, p_{k+1})}{2} \geq \frac{d(q, p_{k+1})}{2} = \frac{\lambda_{k+1}}{2}.$$

Hence the statement is true for $i = k + 1$. Next we assume, for some $i \geq k + 1$, that for every $p \in P_i$, $d^k(p, P_i \setminus \{p\}) \geq \lambda_i/2$, and show the statement for $i + 1$.

For p_{i+1} , we have $d^k(p_{i+1}, P_{i+1} \setminus \{p_{i+1}\}) = \lambda_{i+1} \geq \lambda_{i+1}/2$ and the statement follows. Consider $p \in P_{i+1} \setminus \{p_{i+1}\}$. If p_{i+1} is not among the k nearest neighbors of p in P_{i+1} , then

$$d^k(p, P_{i+1} \setminus \{p\}) = d^k(p, P_i \setminus \{p\}) \geq \frac{\lambda_i}{2} \geq \frac{\lambda_{i+1}}{2}$$

by the induction hypothesis and because the λ -values are non-increasing. Otherwise, p_{i+1} is among the k nearest neighbors of p in $P_{i+1} \setminus \{p\}$ and $d^k(p, P_{i+1} \setminus \{p\}) \geq d(p, p_{i+1})$.

If $d(p, p_{i+1}) \geq \lambda_{i+1}/2$, the claim follows. Otherwise, every site at distance smaller than $\lambda_{i+1}/2$ of p is at distance smaller than λ_{i+1} of p_{i+1} . Since $\lambda_{i+1} = d^k(p_{i+1}, P_i)$, there can be at most $k - 2$ sites of $P_i \setminus \{p\}$ at distance smaller than λ_{i+1} of p_{i+1} . Thus, counting p_{i+1} as well, there can be at most $k - 1$ sites of $P_{i+1} \setminus \{p\}$ at distance smaller than $\lambda_{i+1}/2$ of p and it follows that $d^k(p, P_{i+1} \setminus \{p\}) \geq \lambda_{i+1}/2$. ◀

Computation. We give a simple algorithm for computing the k -distance permutation that has quadratic running time in the number of input points and discuss an approach for improving it. We call a site *ordered* if it has already been assigned its index in the k -distance permutation and *unordered* otherwise.

The simple approach is the following. Pick k sites p_1, \dots, p_k and compute, for each $y \in P \setminus P_k$, the distances from y to p_i , $1 \leq i \leq k$. Store them in a max-heap T_y that also has a fixed entry identifying y . Up until this point we need $\mathcal{O}(nk)$ time. The next steps are repeated iteratively. For all unordered y , group the T_y in a list L . When p_1, \dots, p_{i-1}

are chosen, the algorithm picks p_i by scanning over L and choosing the point with largest distance to its k th nearest ordered neighbor, which takes $\mathcal{O}(n)$ time. When p_i is picked and becomes ordered, remove its entry from L . Then, by traversing all remaining elements in L , identify each unordered y with p_i among y 's k nearest neighbors in P_i and insert $d(p_i, y)$ to T_y . The $(k + 1)$ -distance from y to the ordered sites, which was a previous entry in T_y , is removed. This takes $\mathcal{O}(\log k)$ time per element of L and hence $\mathcal{O}(n \log k)$ per iteration. Since there are n iterations, the total running time is of $\mathcal{O}(n^2 \log k)$.

This simple algorithm can be improved with the main insight that when p_i is determined, the k th nearest ordered neighbor of all remaining unordered sites is at most λ_i away. Hence, unordered sites further than λ_i away from p_i do not have to be updated. Whenever a site p_i is ordered, we can employ a quadtree (Section 2) to only update the unordered sites within distance λ_i . This last step can also be done in general metric spaces with elementary but rather tedious techniques; see [33, Sec. 3.1]. Using the packing property, the total number of updates reduces to $\mathcal{O}(nk \log \Phi)$ (with a constant that depends exponentially on the doubling dimension of the point set). Finally, we replace the list L by a max-heap to avoid the linear scan to search for the next ordered points. Appendix A of [9] provides further details on how to achieve this improvement, which results in the next theorem.

► **Theorem 3.** *The k -distance permutation can be computed in expected time $\mathcal{O}(nk \log n \log \Phi)$ with Φ the spread of the point set.*

4 A sparse union of lenses

We define several spaces in this section and the following. Figure 2 has an overview.

Recall that the k -fold cover is defined as the union of all lenses at radius r , where every lens is given by k sites. For large values of r , most of these lenses intersect, yielding a size explosion in its nerve, the k th order Čech complex. At the same time, many lenses are eventually covered by the union of other lenses and so may be removed from consideration.

To define the precise threshold for removal of a lens, recall that in Section 3 we ordered the sites as p_1, \dots, p_n and obtained values $\lambda_1, \lambda_2, \dots, \lambda_k = \infty, \lambda_{k+1} \geq \dots \geq \lambda_n$. Fix $\epsilon \in (0, 1]$. Since it is fixed, we drop ϵ from the upcoming notation. The *freezing radius* of a site p_i is

$$\text{frz}(p_i) := \frac{(1 + \epsilon)\lambda_i}{\epsilon}.$$

We extend the definition to lenses by setting $\text{frz}(A) = \min_{p \in A} \text{frz}(p)$. Then at radius r we only consider lenses whose freezing radius is at least r and set

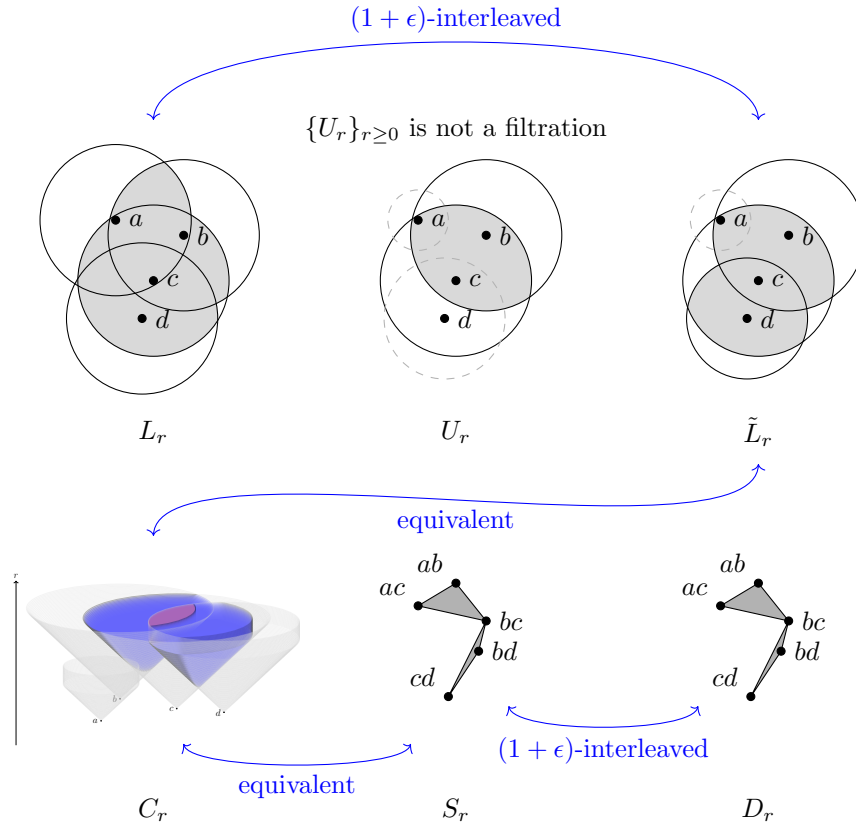
$$U_r := \bigcup_{\text{frz}(A) \geq r} L_r(A).$$

Notice that $\mathcal{U} := \{U_r\}_{r \geq 0}$ is *not* a filtration: Figure 3 illustrates that U_r might not be a subset of $U_{r'}$ for $r < r'$. Even so, some useful inclusions hold as we see on the next lemma.

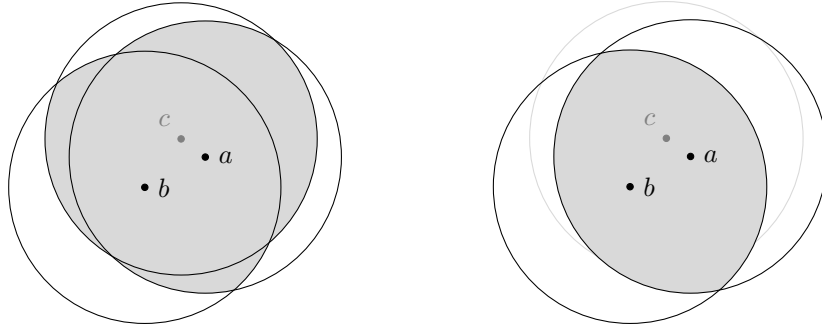
► **Lemma 4.** $U_r \subseteq L_r \subseteq U_{(1+\epsilon)r}$.

Proof. The first inclusion is clear. For the second inclusion, consider $x \in L_r$ and let i be the maximal index such that

$$r \leq \frac{\lambda_i}{\epsilon}. \tag{*}$$



■ **Figure 2** Schematic view of the different filtrations introduced in Sections 4 and 5. We consider $k = 2$, $r > (1 + \epsilon) \text{frz}(a)$ and $r \in (\text{frz}(d), (1 + \epsilon) \text{frz}(d)]$.



■ **Figure 3** Example in \mathbb{R}^2 with $k = 2$. *Left:* U_r at radius $r = \text{frz}(c)$. *Right:* $U_{r'}$ at radius immediately after $\text{frz}(c)$. Even though $r < r'$, $U_r \not\subseteq U_{r'}$.

If $i = n$, then there is $A \subseteq P_n$ with $x \in L_r(A)$ because $x \in L_r$ and $P = P_n$. By definition of the freezing radius and inequality (*), $\text{frz}(A) \geq \text{frz}(p_i) = (1 + \epsilon)\lambda_i/\epsilon \geq r(1 + \epsilon)$ and thus $L_{r(1+\epsilon)}(A) \subseteq U_{(1+\epsilon)r}$. Since $L_r(A) \subseteq L_{(1+\epsilon)r}(A)$, the result follows.

For $i < n$, notice that the Covering Property (Lemma 1) guarantees that x is contained in a lens $L_{r+\lambda_{i+1}}(A)$ for some $A \subseteq P_i$. Since i is maximal, $\lambda_{i+1}/\epsilon < r$ and so $L_{r+\lambda_{i+1}}(A) \subseteq L_{(1+\epsilon)r}(A)$. Moreover, $A \subseteq P_i$ and inequality (*) imply $\text{frz}(A) \geq \text{frz}(p_i) \geq (1 + \epsilon)r$. Hence the lens of A contributes to $U_{(1+\epsilon)r}$ and as it contains x , the statement follows. ◀

This lemma suggests that a $(1 + \epsilon)$ -interleaving with the filtration $\mathcal{L} = \{L_r\}_{r \geq 0}$ consisting of all lenses should be possible if we adjust \mathcal{U} to obtain an actual filtration. To do that we slightly delay the removal of lenses. More precisely, we define

$$\tilde{L}_r(A) := \begin{cases} L_r(A) & r < \text{frz}(A) \\ L_{\text{frz}(A)}(A) & \text{frz}(A) \leq r \leq (1 + \epsilon) \text{frz}(A) \\ \emptyset & (1 + \epsilon) \text{frz}(A) < r. \end{cases}$$

One can visualize the evolution of a lens as a continuous process for increasing r : the lens \tilde{L}_r grows until it reaches its freezing radius and remains unchanged (it is “frozen”) for the interval $[\text{frz}(A), (1 + \epsilon) \text{frz}(A)]$. Afterwards it completely disappears. We call $(1 + \epsilon) \text{frz}(A)$ the *removal radius* of A . The construction is an adaptation of a similar one by Sheehy [45].

We write \tilde{L}_r for the union of $\tilde{L}_r(A)$ over all $A \in \binom{P}{k}$ and $\tilde{\mathcal{L}} := \{\tilde{L}_r\}_{r \geq 0}$. We show next that $\tilde{\mathcal{L}}$ is a filtration.

► **Lemma 5.** *$\tilde{\mathcal{L}}$ is a filtration, i.e., for any $r \leq r'$, $\tilde{L}_r \subseteq \tilde{L}_{r'}$.*

Proof. If the interval $(r, r']$ does not contain any removal radius, $\tilde{L}_r \subseteq \tilde{L}_{r'}$ because the inclusions hold lens-wise. Since the number of different removal radii is bounded by the number of sites and hence finite, it suffices to show that at a removal radius s , any lens that is removed is already covered by lenses that are not being removed at s . In fact, we show that such a lens is covered by lenses that are not yet frozen at s .

Let A be the k -subset associated with a lens being removed at s and $x \in \tilde{L}_s(A)$. By definition, $s = (1 + \epsilon)t$ with $t = \text{frz}(A)$. This implies that $x \in \tilde{L}_t(A) = L_t(A) \subseteq L_s$ because the lens is frozen from radius t on. By Lemma 4, it follows that $x \in U_{(1+\epsilon)t} = U_s$, and therefore x is contained in a lens $L_s(B)$ with $\text{frz}(B) \geq s$. Thus $\tilde{L}_s(A)$ is covered by lenses $L_s(B)$ with $\text{frz}(B) \geq s$ and $\tilde{\mathcal{L}}$ is a filtration. ◀

► **Lemma 6.** *$\tilde{\mathcal{L}}$ and \mathcal{L} are $(1 + \epsilon)$ -interleaved.*

Proof. We show that $\tilde{L}_r \subseteq L_r \subseteq \tilde{L}_{(1+\epsilon)r}$. Note that by definition, $\tilde{L}_r \subseteq L_r$. For the second inclusion, observe that $U_r \subseteq \tilde{L}_r$ follows directly from their definition. Then, Lemma 4 yields $L_r \subseteq U_{(1+\epsilon)r} \subseteq \tilde{L}_{(1+\epsilon)r}$. ◀

5 A sparse simplicial filtration

Since $\tilde{L}_r(A)$ is closed and convex for every r , the nerve of all (non-empty) $\tilde{L}_r(A)$ yields a simplicial complex with the same homotopy type as the k -fold cover at radius r . However, the collection of simplicial complexes obtained when varying r does not form a filtration because simplices disappear from the nerve when passing a removal radius. To overcome this problem, we adapt a construction of Cavanna et al. [11] that is similar to a function’s graph.

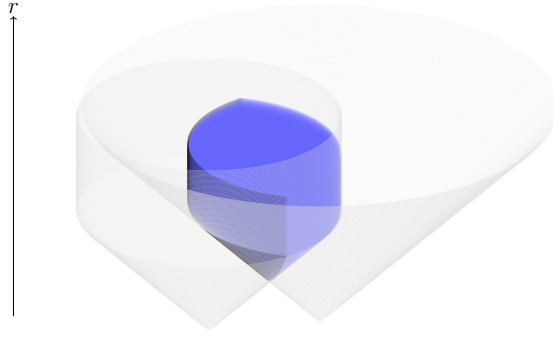
Cones. The idea is to “stack-up” the lenses $\tilde{L}_r(A)$ for all radii: the *cone* of A at radius r is

$$C_r(A) := \bigcup_{\alpha \in [0, r]} (\tilde{L}_\alpha(A) \times \{\alpha\}) \subseteq \mathbb{R}^d \times \mathbb{R}.$$

We write C_r for the union of $C_r(A)$ over all $A \in \binom{P}{k}$. Figure 4 shows one cone.

► **Lemma 7.** *The filtrations $\mathcal{C} = \{C_r\}_{r \geq 0}$ and $\tilde{\mathcal{L}}$ are equivalent.*

20:10 Sparse Higher Order Čech Filtrations



■ **Figure 4** A cone representing the evolution of a lens for $k = 2$. At first the lens grows, until it is frozen. Then, the lens has static size and afterwards it disappears. At the radius where the lens disappears, it is completely covered by other lenses (which are not displayed in the figure).

Proof. In the same sense as above C_r is a stacked-up version of \tilde{L}_α for all $\alpha \leq r$, and we can consider \tilde{L}_r as a subspace of C_r via the map $x \mapsto (x, r)$ for $x \in \tilde{L}_r$. Since $\tilde{\mathcal{L}}$ is a filtration, there is a strong deformation retraction R from C_r to \tilde{L}_r , given by $R((x, \alpha), t) = (x, (1-t)\alpha + tr)$, which naturally commutes with the canonical inclusions. The result follows. ◀

We define the nerve of the cones as the *sparse k th order Čech complex*,

$$S_r := \text{Nrv} \left\{ C_r(A) \mid A \in \binom{P}{k} \right\}.$$

S_r is a subcomplex of the k th order Čech complex for every r because $\tilde{L}_r \subseteq L_r$. Moreover, if $r \leq r'$, $C_r(A) \subseteq C_{r'}(A)$ for all A . Hence $\mathcal{S} = \{S_r\}_{r \geq 0}$ is a filtration. By the Persistent Nerve Theorem [2, Thm. 3.9] and Lemmas 6 and 7 we obtain:

► **Lemma 8.** *The filtrations \mathcal{S} and \mathcal{C} are equivalent. As a consequence, \mathcal{S} is a $(1 + \epsilon)$ -approximation of the k -fold filtration.*

Discretization of the radius. The filtration \mathcal{S} is challenging to compute, due to the freezing of lenses. We elaborate on these issues in Section 7. We now define a variant of \mathcal{S} which is easier to compute and also is $(1 + \epsilon)$ -interleaved with the k -fold filtration.

Recall that the filtrations $\tilde{\mathcal{L}}$, \mathcal{C} and \mathcal{S} are defined based on the freezing radii of sites, which depend on a parameter $\epsilon > 0$. To obtain a $(1 + \epsilon)$ -approximation for $\epsilon \in (0, 1]$ in the end, we consider the above construction of \mathcal{S} with parameter $\epsilon' = \frac{\epsilon}{3}$, obtaining a $(1 + \frac{\epsilon}{3})$ -approximation of the k -fold filtration.

Next, for every $r \geq 0$, let $z \in \mathbb{Z}$ be such that $(1 + \frac{\epsilon}{3})^z \leq r < (1 + \frac{\epsilon}{3})^{z+1}$ and define

$$D_r := S_{(1+\epsilon/3)^z}.$$

We call $\mathcal{D} := \{D_r\}_{r \geq 0}$ the *discrete sparse k th order Čech filtration*. It is formed by a discrete set of snapshots of \mathcal{S} and kept unchanged except when passing over a snapshot radius (this is also referred to as the *Left Kan extension* of a discrete filtration [36, Chap. 10]).

► **Theorem 9.** *\mathcal{D} is a $(1 + \epsilon)$ -approximation of the k -fold filtration.*

Proof. From the definition, $D_r \subseteq S_r \subseteq D_{(1+\epsilon/3)r}$. This interleaving implies that \mathcal{D} is a $(1 + \frac{\epsilon}{3})$ -approximation of \mathcal{S} . Since \mathcal{S} is a $(1 + \frac{\epsilon}{3})$ -approximation of the k -fold filtration, by transitivity, we get that D_r is a $(1 + \frac{\epsilon}{3})^2$ -approximation of the k -fold filtration. The result follows by noting that $(1 + \frac{\epsilon}{3})^2 \leq 1 + \epsilon$ for all $\epsilon \in (0, 1]$. ◀

6 Size analysis

We bound the size of \mathcal{D} , i.e., the number of simplices it contains. Since $D_r \subseteq S_r$ for all $r \geq 0$, it is enough to bound the size of \mathcal{S} with parameter $\epsilon' = \epsilon/3$. Let

$$C_\infty(A) := \bigcup_{r \geq 0} C_r(A)$$

be the *cone of A* (without dependence on a radius). Then, the size of \mathcal{S} equals the size of the nerve of the cones $C_\infty(A)$, where A ranges over all k -subsets of sites. However, the number of vertices is not necessarily $\binom{n}{k}$ because many cones are empty: this happens in particular when the smallest radius for which the balls around the sites of A intersect is larger than the removal radius of the lens. In fact, our argument shows that this is the case for the vast majority of cones. The proof for vertices extends readily to the case of m -wise intersections of cones, i.e. for $(m - 1)$ -simplices, without change and thus we treat the general case directly.

For fixed $m \geq 1$, we derive an upper bound for the number of sets $\{A_1, \dots, A_m\}$ such that the cones $C_\infty(A_1), \dots, C_\infty(A_m)$ intersect. Such sets are in one-to-one correspondence to the $(m - 1)$ -simplices of the sparse k th order Čech filtration, hence we refer to these sets as $(m - 1)$ -simplices. Let $\sigma = \{C_\infty(A_1), \dots, C_\infty(A_m)\}$ be a $(m - 1)$ -simplex and the set of sites P be ordered according to the k -distance permutation. We call a site p_i *involved* in σ if p_i belongs to one of the sets A_1, \dots, A_m . Note that there are at most km sites involved in σ . We say that σ is *associated* to a site p_i if p_i is involved in σ and all other involved sites have index smaller than i . Our strategy is to upper bound the number of $(m - 1)$ -simplices associated to an arbitrary p_i . We only need to consider simplices associated to p_i that appear in the filtration, i.e., simplices whose defining cones intersect.

Fix p_i and $\omega_i := (1 + \epsilon') \text{frz}(p_i)$. Let B denote the ball of radius $2\omega_i$ centered at p_i .

► **Lemma 10.** *If $\sigma := \{C_\infty(A_1), \dots, C_\infty(A_m)\}$ is an $(m - 1)$ -simplex associated to p_i whose cones intersect, then all sites involved in σ are contained in B .*

Proof. Let α denote the minimal radius such that all the balls around sites involved in σ intersect. This is the radius of the minimum enclosing ball of the involved sites. Any common intersection of the cones must happen at scale $r \geq \alpha$.

On the other hand, assume wlog that $p_i \in A_1$. Since p_i has maximal index in A_1 , we have that $\text{frz}(A_1) = \text{frz}(p_i)$. Hence the removal radius of A_1 is equal to ω_i and it follows that the cone of A_1 is empty for all radii greater than ω_i . Therefore any common intersection of the cones of σ must happen at scale $r \leq \omega_i$. Hence, as we assume that the cones do intersect, we must have that $\alpha \leq \omega_i$.

Then, since $d(q, p_i) \leq 2\alpha$, any involved site q lies within distance $2\omega_i$ from p_i . ◀

Hence the involved sites of σ are close to p_i in the sense of the lemma. We can furthermore guarantee that the points of P_i are not too densely packed in B using the packing property of the k -distance permutation.

► **Lemma 11.** *The ball B contains at most $\Gamma := k \left(\frac{96}{\epsilon}\right)^\delta$ sites of $P_i = \{p_1, \dots, p_i\}$, where δ is the doubling dimension of \mathbb{R}^d .*

Proof. We cover B by balls of radius $\lambda_i/4$. That can be done with at most $\zeta = \left(\frac{16(1 + \epsilon')^2}{\epsilon'}\right)^\delta$ balls (see *Doubling Dimension* in Section 2). By the Packing Lemma 2, each open ball of radius $\lambda_i/4$ contains at most k sites of P_i , thus the total number of sites in B is at most $k\zeta$. The bound follows because $\epsilon' = \frac{\epsilon}{3}$ and $\epsilon \leq 1$, hence $(1 + \epsilon')^2 \leq \frac{16}{9} < 2$. ◀

20:12 Sparse Higher Order Čech Filtrations

Bounding the number of non-empty m -intersections of cones is now a matter of simple combinatorics. Recall the Γ notation from Lemma 11.

► **Theorem 12.** *The number of $(m - 1)$ -simplices of the sparse k th order Čech complex with non-empty cone intersection is at most*

$$n \cdot \Gamma^{km} = nk^{km} \left(\frac{96}{\epsilon} \right)^{\delta km}.$$

Proof. Fix p_i . Every $(m - 1)$ -simplex associated to p_i with non-empty cone intersection has up to km involved sites, which all lie in B by Lemma 10. Moreover, all involved sites are in P_i and, by Lemma 11, there are at most Γ of those sites in B to choose from. It follows that there are at most Γ^{km} different choices possible. This upper bound holds for every p_i , so multiplying by the number of sites n yields the result. ◀

We remark that the bounds on this section are not tight and slightly better ones could be easily achieved, by keeping binomials in place or avoiding some approximations. However the improvements would be minor.

7 Computation

We now present an algorithm to construct the discrete sparse Čech filtration. As in the previous sections, let us fix a finite set $P \subseteq \mathbb{R}^d$, an integer $k > 0$ and $\epsilon \in (0, 1]$. Assume that $P = \{p_1, \dots, p_n\}$ has the indices ordered with respect to the k -distance permutation, and that we have computed the corresponding values $\lambda_1, \dots, \lambda_n$ as discussed in Section 3. The algorithm outputs the discrete sparse k th order Čech filtration $\{D_r\}_{r \geq 0}$ as a list of simplices with their corresponding *critical value*, i.e., the smallest parameter value r for which the simplex is part of the filtration. Note that by definition of the discrete sparse Čech filtration, every critical value is of the form $(1 + \epsilon/3)^z$ for some integer z .

Friends. Our algorithm follows the approach and notation of Section 6. For every p_i , we compute all simplices associated to p_i in the filtration together with their critical value. To do so, we first find, among p_1, \dots, p_{i-1} , all sites of distance at most $2\omega_i$ from p_i , where $\omega_i = (1 + \epsilon') \text{frz}(p_i)$ (compare Lemma 10). We call these points *friends* of p_i . We compute friends using a quadtrees data structure, as introduced in Section 3, which we query for every p_i at $2\omega_i$. p_i is added to the quadtrees after the i th iteration (adding an element costs $\mathcal{O}(\log n)$ in expectation as well). Hence the expected running time for this loop is $\mathcal{O}(n \log n + \Sigma)$, where Σ is the number of reported points. These reported points have distance at most $4\omega_i$ from the respective p_i (since the queries are approximate), and by the same argument as in Lemma 11, the number of sites reported for p_i is at most $k \left(\frac{192}{\epsilon} \right)^\delta = \mathcal{O}(k(1/\epsilon)^\delta)$. We traverse the list and remove all “false friends” of distance more than $2\omega_i$. Thus we get the friends of p_i for all sites p_i in expected time

$$\mathcal{O}(n \log n + nk(1/\epsilon)^\delta). \tag{2}$$

Note that the number of friends is bounded by Γ as defined in Lemma 11.

Vertices. Next, we compute the vertices of the filtration associated to p_i , for each $i \geq k$. We proceed by brute-force, just enumerating all k -tuples formed by p_i and $k - 1$ of its friends and checking for every k -tuple whether their cone is non-empty. The last condition is simple

to check, as the cone is non-empty if and only if the radius α of the minimum enclosing ball of the k sites is at most $\text{frz}(p_i)$. In this case, the critical radius of the vertex is set to $(1 + \epsilon/3)^z$, where z is the smallest integer such that $(1 + \epsilon/3)^z \geq \alpha$. Computing α and z per vertex requires $\mathcal{O}(k)$ in expectation [18]. Hence we calculate all vertices in $\mathcal{O}((n - k)k\Gamma^{k-1})$ time.

Simplices. For higher-dimensional simplices associated to p_i , we proceed inductively by dimension, up to a maximal dimension m_{max} . Fix a $(m - 1)$ -simplex $\sigma = \{A_1, \dots, A_m\}$ associated to p_i . We compute all cofacets of σ in the filtration, that is, all m -simplices that contain σ and one further vertex A_{m+1} . Notice that one could order the k -subsets and avoid computing all cofacets, computing instead only cofacets with larger index in the ordering. This would remove a k factor from the computation expected time locally, but does not change the final bound in Theorem 13, which has a k^k factor. We compute all cofacets for simplicity. Since each element of A_{m+1} is either a friend of p_i or p_i itself, we enumerate all k -tuples consisting of these sites and check whether they form a vertex of the filtration. This takes $\mathcal{O}(k)$ time per vertex, just by re-doing the check from the previous step, except that k -tuples associated to p_j must be checked at radius $\min\{\text{frz}(p_j), \omega_i\}$. That is because if the k -tuple cone becomes non empty only after ω_i , then it cannot contribute to a coface of σ .

For a vertex A_{m+1} of the filtration, check whether the cones of A_1, \dots, A_{m+1} intersect is technically challenging because the cones might intersect for a radius where one or several cones are frozen. One cannot resolve this question by a simple minimal enclosing ball computation. In fact, we are not aware of an efficient way to compute the smallest intersection radius of such cones in general. However, as demonstrated in [9, App. B], given a collection of cones A_1, \dots, A_{m+1} and a fixed radius r , we can decide whether the cones (or rather, the corresponding lenses \tilde{L}_r) intersect at radius r by a reduction to a minimum enclosing ball of balls instance [28] in expected $\mathcal{O}(k(m + 1))$ time. We use this predicate and query whether the cones intersect at the smallest removal radius of A_1, \dots, A_{m+1} . That decides whether the m -simplex is in the filtration, and its critical radius can be computed by two more minimal enclosing ball computations. This is only possible because we have discretized the filtration, as discussed in [9, App. B]. It follows that the expected running time spent per $(m - 1)$ -simplex of the filtration is $\mathcal{O}(mk\Gamma^k)$, and doing this over all simplices of the filtration up to dimension m_{max} yields a total expected complexity of $\mathcal{O}(X \cdot m_{max} \cdot k \cdot \Gamma^k)$ for this step, where X is the total number of simplices in the filtration.

This concludes the description of the algorithm. Recalling (2) we obtain a complexity of

$$\mathcal{O}\left(n \log n + \frac{nk}{\epsilon^\delta} + (n - k)k\Gamma^{k-1} + X \cdot m_{max} \cdot k \cdot \Gamma^k\right),$$

where the second and third terms are dominated by the last one because $X \geq n - k$ (see [9, App. C]). Together with the algorithm from Section 3, we arrive at the result

► **Theorem 13.** *Given a set P of n points, $k \geq 0$ and $\epsilon \in (0, 1]$, the discrete sparse k th order Čech filtration up to dimension m_{max} can be computed in time*

$$\mathcal{O}\left(nk \log n \log \Phi + Xk^{k+1}m_{max} \left(\frac{96}{\epsilon}\right)^{k\delta}\right),$$

where Φ is the spread of P and X is the total number of simplices in the filtration.

8 Conclusions

We introduced the first $(1 + \epsilon)$ -approximate filtration of the higher order k -fold filtration and provided an algorithm for computing it. If k and ϵ are considered as constants and the input point set has constant spread, the algorithm runs in time $\mathcal{O}(n \log n)$ and yields a filtration of size $\mathcal{O}(n)$, which are the same favorable properties of the well-studied case $k = 1$.

There are various avenues to strengthen and generalize our results. First of all, our method has concentrated on the Euclidean case, but our approach mostly generalizes to point sets in arbitrary metric spaces – the algorithm cannot use the quadtree data structure anymore, but there is no need for it, since the algorithm by Har-Peled and Mendel [33, Sec. 3.1] can be adapted to the k -distance case with little effort. Also, the friends of p_i (Section 7) can be computed with a slight adaptation of their techniques; we used quadtrees mostly for the ease of presentation. However, the computation of critical values of simplices described in [9, App. B] is for the Euclidean case only, and the complexity of this step remains unspecified for a general metric space. This is common in related work; see, for instance [11, Sec. 5].

Another natural goal is to remove the dependence on the spread. This dependence is caused by the computation of the k -distance permutation which is inspired by the algorithm of [33, Sec. 3.1]. In the same paper [33, Sec. 3.2–3.3], they describe an approach to remove the spread from the bound (for $k = 1$) using an approximate version of the greedy permutation. While our construction of the sparsified filtration can be easily adapted to work with an approximate version of the k -distance permutation, it seems less straight-forward to generalize the computation to the k -distance, even in the Euclidean case. We leave this for future work.

The k -distance permutation relates to the *Distance to Measure* (DTM) [13], which is the square average of the distances to the k nearest neighbors. The DTM has the advantage of being robust, in terms of the Wasserstein distance, to perturbations on the sample [13, Sec. 3]. However, most of the existing methods for sparsifying filtrations obtained via the DTM [1, 8, 32] require a preliminary approximation by weighted distances. Our approach might be adaptable to directly sparsify DTM filtrations.

While we concentrate on the case of a single value of k , we pose the question whether our methods can be used to approximate the multi-cover bifiltration, as studied in [17]. The extension is not straight-forward because there is no direct relation between the approximate filtrations on level k and $k + 1$. We speculate that the technique of double-nerve constructions of [17] could be useful in this context. The presence of an exponential factor on k in our size bounds suggests a restriction of our approach’s usability to small portions of the bifiltration, for small k . The exponential factor on k also carries over to the expected computation time. Reducing that dependency on k is another possible line of future work. Note that [17, Prop. 5] gives a size bound of $\mathcal{O}(n^{d+1})$ for the exact version, but we ask whether a polynomial bound on k could be achieved without such a blow-up in the dependency on n .

Finally, a natural question is the practicality of our algorithm. We remark that even for $k = 1$, while some work has been devoted to practical aspects of computing sparsifications [3, 7, 20, 37, 41], the actual practical computation is still an unresolved problem. We think that the natural order for a practically efficient solution would be to first identify best practices in the simpler $k = 1$ case and subsequently try to adapt them to larger values of k . So, while we would be curious about the performance of our algorithm, such an evaluation seems to be premature at the moment.



References

- 1 Hirokazu Anai, Frédéric Chazal, Marc Glisse, Yuichi Ike, Hiroya Inakoshi, Raphaël Tinarrage, and Yuhei Umeda. Dtm-based filtrations. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, volume 129 of *LIPICs*, pages 58:1–58:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPICs.SoCG.2019.58.
- 2 Ulrich Bauer, Michael Kerber, Fabian Roll, and Alexander Rolle. A unified view on the functorial nerve theorem and its variations, 2022. arXiv:2203.03571.
- 3 Nello Blaser and Morten Brun. Sparse nerves in practice. In Andreas Holzinger, Peter Kieseberg, A M. Tjoa, and Edgar R. Weippl, editors, *Machine Learning and Knowledge Extraction - CD-MAKE 2019, Canterbury, UK, August 26-29, 2019, Proceedings*, volume 11713 of *Lecture Notes in Computer Science*, pages 272–284. Springer, 2019. doi:10.1007/978-3-030-29726-8_17.
- 4 Andrew J. Blumberg and Michael Lesnick. Stability of 2-parameter persistent homology. *CoRR*, 2020. arXiv:2010.09628.
- 5 Thomas Bonis, Maks Ovsjanikov, Steve Oudot, and Frédéric Chazal. Persistence-based pooling for shape pose recognition. In Alexandra Bac and Jean-Luc Mari, editors, *Computational Topology in Image Context - 6th International Workshop, CTIC 2016, Marseille, France, June 15-17, 2016, Proceedings*, volume 9667 of *Lecture Notes in Computer Science*, pages 19–29. Springer, 2016. doi:10.1007/978-3-319-39441-1_3.
- 6 Magnus B. Botnan and Gard Spreemann. Approximating persistent homology in euclidean space through collapses. *Applicable Algebra in Engineering, Communication and Computing*, 26(1-2):73–101, January 2015. doi:10.1007/s00200-014-0247-y.
- 7 Bernhard Brehm and Hanne Hardering. Sparrips. *CoRR*, 2018. arXiv:1807.09982.
- 8 Mickaël Buchet, Frédéric Chazal, Steve Y. Oudot, and Donald R. Sheehy. Efficient and robust persistent homology for measures. *Comput. Geom.*, 58:70–96, 2016. doi:10.1016/j.comgeo.2016.07.001.
- 9 Mickaël Buchet, Bianca B. Dornelas, and Michael Kerber. Sparse higher order Čech filtrations, 2023. arXiv:2303.06666.
- 10 Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009. doi:10.1090/S0273-0979-09-01249-X.
- 11 Nicholas J. Cavanna, Mahmoodreza Jahanseir, and Donald R. Sheehy. A geometric perspective on sparse filtrations. In *Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG 2015, Kingston, Ontario, Canada, August 10-12, 2015*. Queen’s University, Ontario, Canada, 2015. URL: <http://research.cs.queensu.ca/cccg2015/CCCG15-papers/01.pdf>.
- 12 Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Oudot. Proximity of persistence modules and their diagrams. In John Hershberger and Efi Fogel, editors, *Proceedings of the 25th ACM Symposium on Computational Geometry, Aarhus, Denmark, June 8-10, 2009*, pages 237–246. ACM, 2009. doi:10.1145/1542362.1542407.
- 13 Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Found. Comput. Math.*, 11(6):733–751, 2011. doi:10.1007/s10208-011-9098-0.
- 14 Aruni Choudhary, Michael Kerber, and Sharath Raghvendra. Improved topological approximations by digitization. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2675–2688. SIAM, 2019. doi:10.1137/1.9781611975482.166.
- 15 Aruni Choudhary, Michael Kerber, and Sharath Raghvendra. Polynomial-sized topological approximations using the permutahedron. *Discret. Comput. Geom.*, 61(1):42–80, 2019. doi:10.1007/s00454-017-9951-2.
- 16 Aruni Choudhary, Michael Kerber, and Sharath Raghvendra. Improved approximaterips filtrations with shifted integer lattices and cubical complexes. *J. Appl. Comput. Topol.*, 5(3):425–458, 2021. doi:10.1007/s41468-021-00072-4.

- 17 René Corbet, Michael Kerber, Michael Lesnick, and Georg Osang. Computing the multicover bifiltration. *Discret. Comput. Geom.*, 2023. doi:10.1007/s00454-022-00476-8.
- 18 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition*. Springer, 2008. URL: <https://www.worldcat.org/oclc/227584184>.
- 19 Tamal K. Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. In Siu-Wing Cheng and Olivier Devillers, editors, *30th International Symposium on Computational Geometry, SoCG 2014, Kyoto, Japan, June 08 - 11, 2014*, page 345. ACM, 2014. doi:10.1145/2582112.2582165.
- 20 Tamal K. Dey, Dayu Shi, and Yusu Wang. Simba: An efficient tool for approximating rips-filtration persistence via simplicial batch collapse. *ACM J. Exp. Algorithmics*, 24(1):1.5:1–1.5:16, 2019. doi:10.1145/3284360.
- 21 Martin E. Dyer. A class of convex programs with applications to computational geometry. In David Avis, editor, *Proceedings of the Eighth Annual Symposium on Computational Geometry, Berlin, Germany, June 10-12, 1992*, pages 9–15. ACM, 1992. doi:10.1145/142675.142681.
- 22 Herbert Edelsbrunner and John L. Harer. *Computational topology*. American Mathematical Society, Providence, RI, 2010. An introduction. doi:10.1090/mbk/069.
- 23 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. *Discret. Comput. Geom.*, 28(4):511–533, 2002. doi:10.1007/s00454-002-2885-2.
- 24 Herbert Edelsbrunner and Georg Osang. A simple algorithm for higher-order delaunay mosaics and alpha shapes. *CoRR*, 2020. arXiv:2011.03617.
- 25 Herbert Edelsbrunner and Georg Osang. The multi-cover persistence of euclidean balls. *Discret. Comput. Geom.*, 65(4):1296–1313, 2021. doi:10.1007/s00454-021-00281-9.
- 26 Herbert Edelsbrunner and Raimund Seidel. Voronoi diagrams and arrangements. *Discret. Comput. Geom.*, 1:25–44, 1986. doi:10.1007/BF02187681.
- 27 Kaspar Fischer. *Smallest enclosing balls of balls: Combinatorial structure & algorithms*. PhD thesis, Swiss Federal Institute of Technology, ETH Zürich, 2005. URL: <https://people.inf.ethz.ch/emo/DoctThesisFiles/fischer05.pdf>.
- 28 Kaspar Fischer and Bernd Gärtner. The smallest enclosing ball of balls: combinatorial structure and algorithms. *Int. J. Comput. Geom. Appl.*, 14(4-5):341–378, 2004. doi:10.1142/S0218195904001500.
- 29 Marcio Gameiro, Yasuaki Hiraoka, Shunsuke Izumi, Miroslav Kramar, Konstantin Mischaikow, and Vidit Nanda. A topological measurement of protein compressibility. *Japan Journal of Industrial and Applied Mathematics*, 32(1):1–17, 2015. doi:10.1007/s13160-014-0153-5.
- 30 Robert W. Ghrist. *Elementary applied topology*, volume 1. Createspace Seattle, WA, 2014.
- 31 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.
- 32 Leonidas J Guibas, Quentin Mérigot, and Dmitriy Morozov. Witnessed k-distance. *Discret. Comput. Geom.*, 49(1):22–45, 2013. doi:10.1007/s00454-012-9465-x.
- 33 Sarel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006. doi:10.1137/S0097539704446281.
- 34 Yasuaki Hiraoka, Takenobu Nakamura, Akihiko Hirata, Emerson G. Escobar, Kaname Matsue, and Yasumasa Nishiura. Hierarchical structures of amorphous solids characterized by persistent homology. *Proceedings of the National Academy of Sciences*, 113(26):7035–7040, 2016. doi:10.1073/pnas.1520877113.
- 35 Lida Kanari, Pawel Dlotko, Martina Scolamiero, Ran Levi, Julian C. Shillcock, Kathryn Hess, and Henry Markram. A topological representation of branching neuronal morphologies. *Neuroinformatics*, 16(1):3–13, 2018. doi:10.1007/s12021-017-9341-1.
- 36 Saunders Mac Lane. *Categories for the working mathematician.*, volume 5. New York, NY: Springer, 1998. doi:10.1007/978-1-4757-4721-8.

- 37 Clément Maria, Pawel Dlotko, Vincent Rouvreau, and Marc Glisse. Rips complex. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 3.5.0 edition, 2022. URL: https://gudhi.inria.fr/doc/3.5.0/group__rips__complex.html.
- 38 Nimrod Megiddo. On the ball spanned by balls. *Discret. Comput. Geom.*, 4:605–610, 1989. doi:10.1007/BF02187750.
- 39 David M. Mount and Eunhui Park. A dynamic data structure for approximate range searching. In David G. Kirkpatrick and Joseph S. B. Mitchell, editors, *Proceedings of the 26th ACM Symposium on Computational Geometry, Snowbird, Utah, USA, June 13-16, 2010*, pages 247–256. ACM, 2010. doi:10.1145/1810959.1811002.
- 40 James Munkres. *Topology*. Prentice Hall, 2ed edition, 2000.
- 41 Julian B. Pérez, Sydney Hauke, Umberto Lupo, Matteo Caorsi, and Alberto Dassatti. Giotto-ph: A python library for high-performance computation of persistent homology of vietoris-rips filtrations. *CoRR*, 2021. arXiv:2107.05412.
- 42 Jeff M. Phillips, Bei Wang, and Yan Zheng. Geometric inference on kernel density estimates. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry, SoCG 2015, June 22-25, 2015, Eindhoven, The Netherlands*, volume 34 of *LIPICs*, pages 857–871. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICs.SOCG.2015.857.
- 43 Raimund Seidel. On the number of faces in higher-dimensional voronoi diagrams. In D. Soule, editor, *Proceedings of the Third Annual Symposium on Computational Geometry, Waterloo, Ontario, Canada, June 8-10, 1987*, pages 181–185. ACM, 1987. doi:10.1145/41958.41977.
- 44 Donald R. Sheehy. A multicover nerve for geometric inference. In *Proceedings of the 24th Canadian Conference on Computational Geometry, CCCG 2012, Charlottetown, Prince Edward Island, Canada, August 8-10, 2012*, pages 309–314, 2012. URL: <http://2012.cccg.ca/papers/paper52.pdf>.
- 45 Donald R. Sheehy. Linear-size approximations to the vietoris-rips filtration. *Discret. Comput. Geom.*, 49(4):778–796, 2013. doi:10.1007/s00454-013-9513-1.
- 46 Donald R. Sheehy. A sparse delaunay filtration. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry, SoCG 2021, June 7-11, 2021, Buffalo, NY, USA (Virtual Conference)*, volume 189 of *LIPICs*, pages 58:1–58:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.SOCG.2021.58.
- 47 Wilson A. Sutherland. *Introduction to metric and topological spaces*. Oxford Mathematics Series. Oxford University Press, 2 edition, 2009.

Finding Large Counterexamples by Selectively Exploring the Pachner Graph

Benjamin A. Burton  

The University of Queensland, Brisbane, Australia

Alexander He   

The University of Queensland, Brisbane, Australia

Abstract

We often rely on censuses of triangulations to guide our intuition in 3-manifold topology. However, this can lead to misplaced faith in conjectures if the smallest counterexamples are too large to appear in our census. Since the number of triangulations increases super-exponentially with size, there is no way to expand a census beyond relatively small triangulations – the current census only goes up to 10 tetrahedra. Here, we show that it is feasible to search for large and hard-to-find counterexamples by using heuristics to *selectively* (rather than exhaustively) enumerate triangulations. We use this idea to find counterexamples to three conjectures which ask, for certain 3-manifolds, whether one-vertex triangulations always have a “distinctive” edge that would allow us to recognise the 3-manifold.

2012 ACM Subject Classification Mathematics of computing → Topology

Keywords and phrases Computational topology, 3-manifolds, Triangulations, Counterexamples, Heuristics, Implementation, Pachner moves, Bistellar flips

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.21

Related Version *Full Version*: <https://arxiv.org/abs/2303.06321> [9]

Supplementary Material *Software (Source Code)*:
<https://github.com/AlexHe98/triang-counterex>

Funding *Alexander He*: Supported by an Australian Government Research Training Program Scholarship.

Acknowledgements We thank the referees for their helpful comments.

1 Introduction

Many conjectures in computational geometry and topology are true in small cases, but turn out to have a relatively large counterexample. Perhaps the most notable example of this is the *Hirsch conjecture*, which posited that in a d -dimensional polytope with n facets, any two vertices can be connected by a path of at most $n - d$ edges. This is true when the dimension is small (specifically, when $d \leq 3$ [20]), as well as when the number of facets is small (specifically, when $n \leq d + 6$ [2]). However, this is not indicative of the general behaviour: Santos showed that there is a 43-dimensional counterexample with 86 facets [31].

For an example from 3-manifold topology, consider the Seifert fibre spaces (see Hatcher’s notes [15] for a definition). There are 302 Seifert fibre spaces that can be triangulated with no more than 7 tetrahedra; for all of these, at least one **minimal triangulation** (triangulation with the smallest possible number of tetrahedra) uses a standard prism-and-layering construction. This pattern does not persist: there is a Seifert fibre space whose unique (8-tetrahedron) minimal triangulation is given by the isomorphism signature `iLLLPQcbcgffghhhtsmhgsof` (the software *Regina* [8] can convert this back into a triangulation), and this is instead



© Benjamin A. Burton and Alexander He;

licensed under Creative Commons License CC-BY 4.0

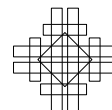
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 21; pp. 21:1–21:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



constructed from a gadget (called the *brick* B_5) that was first identified in Martelli and Petronio’s census of minimal triangulations up to 9 tetrahedra [22]; although the first author [3] and Matveev [24] have since extended the census, no other gadgets like this have been found.

We present a technique for finding large counterexamples in a similar setting; we study arbitrary one-vertex triangulations, not just minimal ones. The obvious source of counterexamples is a census of all triangulations up to a given number of tetrahedra. However, this only captures small counterexamples because the number of triangulations grows super-exponentially as we increase the number of tetrahedra; to date, the census of all closed 3-manifold triangulations only goes up to 10 tetrahedra, and this already includes over 2 billion triangulations, constituting over 63 GB of data [3].

How can we find a counterexample that is too large to appear in our census? We showcase a method of *selectively* (rather than exhaustively) enumerating triangulations, which yields large counterexamples to three conjectures posed by Saul Schleimer at the 2022 Dagstuhl workshop on *Computation and Reconfiguration in Low-Dimensional Topological Spaces*.

1.1 The conjectures

Each conjecture concerns edges of one-vertex triangulations. Since such edges realise embedded closed curves, we can ask whether they are embedded in an interesting way.

For instance, in a lens space, consider an edge e that forms a **core curve**, meaning that the complement of a regular neighbourhood of the curve is a solid torus; in this case, we call e a **core edge**. Finding such an edge certifies that the 3-manifold is a lens space.

► **Conjecture 1.** *Every one-vertex triangulation of a lens space has a core edge.*

Since solid torus recognition can be solved efficiently in practice [11, 5], proving Conjecture 1 would have provided a relatively efficient method for recognising lens spaces. Lens space recognition can also be used to determine whether a 3-manifold is **elliptic** (i.e., admits spherical geometry). Indeed, Lackenby and Schleimer [21] recently used the following variant of Conjecture 1 to show that recognising elliptic manifolds is in NP:

► **Theorem** ([21], Theorem 9.4). *Let \mathcal{M} be a lens space that is neither $\mathbb{R}P^3$ nor a prism manifold, and let \mathcal{T} be any triangulation of \mathcal{M} . Then the 86th iterated barycentric subdivision of \mathcal{T} contains a sequence of edges that forms a core curve.*

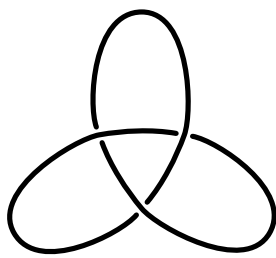
The other two conjectures have similar motivations: special edges can help us recognise certain 3-manifolds, so we would like to know whether such edges always exist.

The second conjecture concerns **tunnel number**: the smallest number of arcs that we need to add to a knot so that the complement becomes a handlebody. Let \mathcal{T} be an ideal triangulation of the complement of a knot K , so that each edge e in \mathcal{T} forms an arc α that meets K at its endpoints. We call e a **tunnel edge** if the complement of $K \cup \alpha$ is a genus-2 handlebody. If K is not the unknot, then the existence of a tunnel edge implies that K has tunnel number equal to one; Figure 1 shows three knots with tunnel number one.

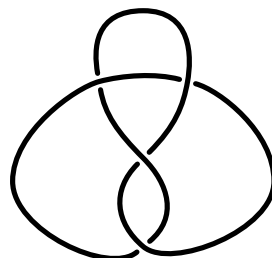
► **Conjecture 2.** *Let K be a knot with tunnel number equal to one. Then every one-vertex ideal triangulation of the complement of K has a tunnel edge.*

The third conjecture concerns **Seifert fibre spaces**, which are fibred by circles in a particular way (again, see Hatcher’s notes [15]); we call each circle fibre a **Seifert fibre**. Seifert fibre spaces play a central role in torus decompositions for irreducible 3-manifolds:

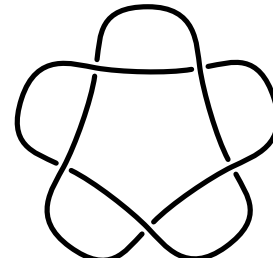
- The JSJ decomposition (due to Jaco and Shalen [17], and Johannson [19]) cuts such 3-manifolds into pieces that are either atoroidal or Seifert fibred.



(a) The trefoil knot.



(b) The figure-eight knot.



(c) The (5, 2) torus knot.

■ **Figure 1** Three knots with tunnel number equal to one.

- The geometric decomposition from Thurston’s geometrisation conjecture (famously proved by Perelman [25, 27, 26]) cuts such 3-manifolds into pieces that are either hyperbolic or graph manifolds; the graph manifolds can be further decomposed into Seifert fibre spaces. The **small** Seifert fibre spaces do not contain any embedded two-sided incompressible surfaces, which makes them relatively difficult to work with.

► **Conjecture 3.** *Every one-vertex triangulation of a small Seifert fibre space has an edge isotopic to a Seifert fibre.*

1.2 Using a targeted search to find counterexamples

As mentioned earlier, we have found counterexamples to all three conjectures listed in section 1.1. The key ingredient for finding these examples is a heuristic for measuring how “far away” a triangulation is from having no “distinguished” edges (such as core edges).

To see why such a heuristic is necessary, we note that the census of triangulations up to 10 tetrahedra contains 422 533 279 one-vertex triangulations of the 3-sphere (the simplest possible lens space). All of these have at least one core edge; verifying this required 22 hours of wall time on 12 threads, but this does not include the time required to: (1) generate the census, and (2) identify the 3-spheres in this census (which had been done previously [3, 4]).

The upshot is that this exhaustive search was both expensive and unsuccessful. In contrast, our heuristic enabled a targeted search, which produced all of our counterexamples in just *minutes* of wall time. See section 5 for detailed computational results, as well as an explanation of how to turn our counterexamples into infinite families.

We introduce our heuristic and our targeted search algorithm in section 4. In section 3, we discuss some auxiliary algorithms, which are of independent interest; in particular, we present (and implement) an improved algorithm for handlebody recognition. We finish by mentioning some unanswered questions in section 6.

2 Preliminaries

2.1 Triangulations

A (**generalised**) **triangulation** \mathcal{T} is a finite collection of tetrahedra whose triangular faces may be affinely identified in pairs; each equivalence class of such faces is a **face** of \mathcal{T} . The **size** of \mathcal{T} , denoted $|\mathcal{T}|$, is the number of tetrahedra in \mathcal{T} . The **boundary** of \mathcal{T} consists of all the faces that are not identified with any other faces. We allow faces of the same tetrahedron to be identified, so our triangulations need not be simplicial complexes.

The face identifications also yield equivalence classes of vertices and edges of the tetrahedra; we call these classes the **vertices** and **edges** of \mathcal{T} , respectively. In this paper, we are particularly interested in **one-vertex triangulations**, which have exactly one vertex. The **degree** of an edge e of \mathcal{T} is the number of tetrahedra that meet e , counted with multiplicity. The **1-skeleton** of \mathcal{T} is the subcomplex consisting only of the vertices and edges of \mathcal{T} .

If a point lies on the boundary of a triangulation, then we say that it is **boundary**; otherwise, we say that it is **internal**. If a face, edge or vertex consists *entirely* of boundary points, then we say that it is **boundary**; otherwise, we say that it is **internal**.

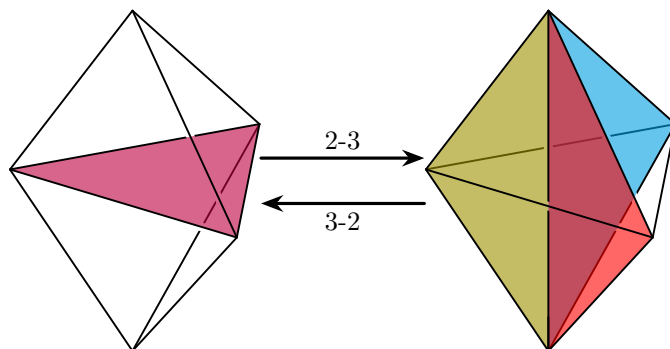
For a triangulation \mathcal{T} to be a 3-manifold, every point p in \mathcal{T} must be **non-singular** – i.e., have a small neighbourhood bounded by either a disc (if p is boundary) or a 2-sphere (if p is internal). This could fail for vertices and midpoints of edges. For this paper, we insist that no edge is identified with itself in reverse; this ensures that midpoints of edges are non-singular.

We call \mathcal{T} a **3-manifold triangulation** if all its vertices are non-singular, because in this case \mathcal{T} will genuinely be a 3-manifold. However, we also allow vertices to be **ideal**, meaning that a small neighbourhood of the vertex is bounded by a closed surface other than the 2-sphere. **Ideal triangulations** (which contain one or more ideal vertices) give a useful representation of the 3-manifold obtained by truncating the ideal vertices; this idea originated with Thurston’s ideal triangulation of the figure-eight knot complement [33, Example 1.4.8].

To concretely encode a triangulation, we give each tetrahedron both a label and an ordering of its four vertices. Two triangulations are **isomorphic** if they are identical up to relabelling tetrahedra and/or reordering tetrahedron vertices. We can uniquely identify any isomorphism class of triangulations using an efficiently-computable string called an **isomorphism signature**. There are many ways to formulate isomorphism signatures; we use the implementation in **Regina** [5, 8], which is described in [4].

2.2 The 2-3 and 3-2 moves

Given any one-vertex triangulation \mathcal{T} with at least two tetrahedra, we can produce a new one-vertex triangulation of the same 3-manifold by performing a **2-3 move** about a (triangular) face that meets two distinct tetrahedra. This replaces these two tetrahedra with three tetrahedra attached around a new edge e ; see Figure 2. We call the inverse move a **3-2 move** about e ; it is possible to perform a 3-2 move about an edge if and only if this edge is an internal degree-3 edge that actually meets three distinct tetrahedra.



■ **Figure 2** The 2-3 and 3-2 moves.

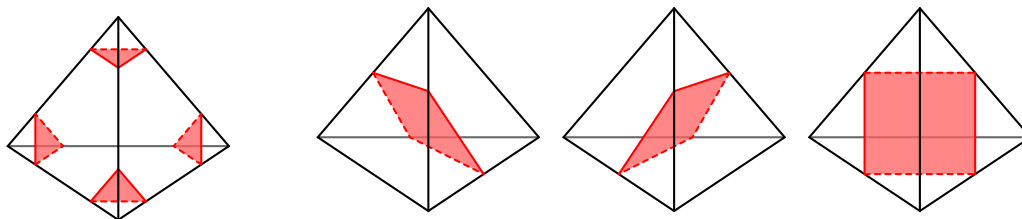
Consider a 3-manifold \mathcal{M} . We can think of the one-vertex triangulations of \mathcal{M} with at least two tetrahedra as nodes of an infinite graph, with two nodes connected by an undirected arc if and only if the corresponding triangulations are related by a 2-3 move. This graph, called the **Pachner graph** of \mathcal{M} , is known to be connected [1, 23, 28, 30].

2.3 Normal surfaces

We now outline the basics of normal surface theory; see [14] and [16] for more comprehensive discussion. A properly embedded surface S in a triangulation \mathcal{T} is a **normal surface** if:

- S meets each simplex (i.e., vertex, edge, triangle, or tetrahedron) of \mathcal{T} transversely; and
- S meets each tetrahedron Δ of \mathcal{T} in a finite (and possibly empty) collection of discs – known as **elementary discs** – where each such disc is a curvilinear triangle or quadrilateral whose vertices lie on different edges of Δ .

Up to **normal isotopy** – an ambient isotopy that preserves every simplex of \mathcal{T} – every elementary disc has one of the seven types shown in Figure 3.



(a) The four triangle types. (b) The three quadrilateral types.

■ **Figure 3** The seven elementary disc types in a tetrahedron.

The simplest example of a normal surface is a **vertex-linking surface**, which consists entirely of triangles. Such surfaces always exist, so finding these surfaces never gives us any new information about the underlying 3-manifold. For this reason, we consider a normal surface to be **non-trivial** when it includes at least one quadrilateral.

If \mathcal{T} has size n , then we can represent any normal surface in \mathcal{T} as a **normal coordinate vector** in \mathbb{R}^{7n} that counts the number of elementary discs of each type in each tetrahedron. It is often sufficient to focus on a class of surfaces called the **vertex normal surfaces**; roughly, these form a finite and algorithmically enumerable “basis” for the set of all normal surfaces.

3 Key tools

This section discusses some new algorithms and implementations that were crucial for our work. These algorithms may be of independent interest.

3.1 Handlebody recognition

To find counterexamples to Conjectures 1 and 2, we need algorithms to recognise solid tori and genus-2 handlebodies. **Regina** includes an implementation of solid torus recognition [13, 18] that is remarkably efficient in practice [8, 5, 11]. We generalise this to recognise handlebodies of arbitrary genus, improving the earlier algorithm of Jaco and Tollefson [18, Algorithm 9.3].

Handlebodies are **irreducible** and have **compressible** boundary (see Definitions 2.2.1 and 3.3.2 of [23]), so we can exploit the following consequence of Corollary 6.4 of [18]:

► **Theorem 4.** *Let \mathcal{T} be a triangulation of a compact irreducible 3-manifold with compressible boundary. Then \mathcal{T} contains a vertex normal essential compressing disc.*

The strategy of our algorithm is to repeatedly cut along vertex normal compressing discs; a triangulation \mathcal{T} is a handlebody if and only if \mathcal{T} eventually gets decomposed into a collection of 3-balls. For computational reasons, we modify this approach in two ways.

First, we expand our focus beyond discs, to include any vertex normal surface S whose Euler characteristic $\chi(S)$ is positive. The rationale is that $\chi(S)$ can be expressed as a linear function in normal coordinates, which allows us to exploit linear programming techniques. This leads to an algorithm for detecting non-trivial normal spheres and discs that is fast in practice (even though, in theory, this is actually the exponential-time bottleneck for both solid torus recognition and handlebody recognition); see [11] for details.

Second, because cutting along a normal surface can increase the number of tetrahedra exponentially, we use a technique called **crushing**. Crushing was introduced by Jaco and Rubinstein [16] following earlier unpublished work of Casson, and was later refined by the first author [6]. Crushing a non-trivial normal surface has the following crucial benefit: it produces a new triangulation with strictly fewer tetrahedra than before. The trade-off is that crushing could have topological side-effects, as detailed in the following theorem:

► **Theorem 5** ([6], Theorem 2). *Let \mathcal{T} be a triangulation of a compact orientable 3-manifold \mathcal{M} , and let S be a normal sphere or disc in \mathcal{T} . Crushing S yields a triangulation \mathcal{T}' whose underlying 3-manifold \mathcal{M}' is obtained from \mathcal{M} by zero or more of the following operations:*

- *undoing connected sums;*
- *cutting along properly embedded discs;*
- *filling boundary 2-spheres with 3-balls; or*
- *deleting 3-ball, 3-sphere, $\mathbb{R}P^3$, $L_{3,1}$ or $S^2 \times S^1$ components.*

For the rest of this section, call a connected orientable 3-manifold **interesting** if it has exactly one boundary component and its first homology is \mathbb{Z}^g , where g is the genus of the boundary surface. Our handlebody recognition algorithm relies on the following result:

► **Proposition 6.** *Let \mathcal{T} be a triangulation of an interesting 3-manifold, and suppose that \mathcal{T} contains a non-trivial normal sphere or disc S . Let \mathcal{T}' denote the triangulation obtained by crushing S . Then each component of \mathcal{T}' is either closed or interesting. Moreover, \mathcal{T} is a handlebody if and only if every component of \mathcal{T}' is either a 3-sphere or a handlebody.*

Proof Outline. To understand the effect of crushing the surface S , it suffices to consider the operations listed in Theorem 5 one at a time. The most intricate part of the proof is showing that if we perform one of these operations on an interesting 3-manifold, then the components of the resulting 3-manifold must all be either closed or interesting. This mostly involves some fairly routine homology arguments. See the full version [9] for details. ◀

The last thing we rely on is an algorithm to recognise the 3-ball; this is a well-known variant of 3-sphere recognition [16, 29, 32], and an implementation is available in **Regina** [5, 8].

The following handlebody recognition algorithm is now also available in **Regina** [8]:

► **Algorithm 7** (Handlebody recognition). *To test whether a triangulation \mathcal{T} is a handlebody:*

- (1) *Check that \mathcal{T} is connected, orientable, and has exactly one boundary component. If \mathcal{T} fails to satisfy any of these conditions, then terminate and return **false**.*
- (2) *Compute the genus g of the boundary of \mathcal{T} .*
 - *In the case that $g = 0$, check whether \mathcal{T} is a 3-ball. If it is, terminate and return **true**; otherwise, terminate and return **false**.*

- In the case that $g > 0$, check whether the first homology of \mathcal{T} is \mathbb{Z}^g . If it is not, terminate and return **false**.
- (3) Create a list \mathcal{L} of triangulations to process, which initially contains \mathcal{T} (and nothing else). While \mathcal{L} is non-empty:
 - (i) Let \mathcal{F} be the first triangulation that appears in \mathcal{L} . Remove \mathcal{F} from \mathcal{L} .
 - (ii) Find a non-trivial normal sphere or disc S in \mathcal{F} . If no such surface exists, then terminate and return **false**.
 - (iii) Crush S . For each component \mathcal{C} of the triangulation \mathcal{R} that results from crushing:
 - If \mathcal{C} is closed, check whether \mathcal{C} is a 3-sphere. If it is, discard \mathcal{C} and move on to the next component of \mathcal{R} ; otherwise, terminate and return **false**.
 - If \mathcal{C} has sphere boundary, check whether \mathcal{C} is a 3-ball. If it is, discard \mathcal{C} and move on to the next component of \mathcal{R} ; otherwise, terminate and return **false**.
 - In any other case, add \mathcal{C} to the list \mathcal{L} , and move on to the next component of \mathcal{R} .
- (4) Once there are no more triangulations in \mathcal{L} , terminate and return **true**.

► **Theorem 8.** *Algorithm 7 correctly determines whether a given triangulation is a handlebody.*

Proof Outline. By Proposition 6, every time we make a full pass through the loop in step 3, we preserve the following invariant: every triangulation in \mathcal{L} is interesting, and \mathcal{T} is a handlebody if and only if every triangulation in \mathcal{L} is actually a handlebody with positive genus. See the full version [9] for details. ◀

3.2 Detecting edges isotopic to Seifert fibres

To find counterexamples to Conjecture 3, we need to be able to test whether an edge is “bad” in the sense that it is isotopic to a Seifert fibre. This is difficult to test conclusively. We resort to searching for normal surfaces that must occur if an edge is bad; if any such surface fails to exist, then we can certify that the edge is not bad. See the full version [9] for details.

3.3 Tracking edges as we perform 2-3 and 3-2 moves

Suppose that in some triangulation \mathcal{T} , we already know which edges are (for instance) core edges. Observe that if we create a new triangulation \mathcal{T}' using a 3-2 move about an edge e of \mathcal{T} , then the only change to the 1-skeleton is that we remove the edge e . Thus, in principle, we do not need to recompute which edges of \mathcal{T}' are core edges.

In the other direction, if we create a new triangulation using a 2-3 move, then the only change to the 1-skeleton is that we introduce a new edge e . In this case, we need to check whether e is a core edge, but there should be no need to recompute this for the other edges.

In practice, the situation is complicated by the fact that performing 2-3 and 3-2 moves in **Regina** could arbitrarily renumber the edges. Our solution is to use a bespoke implementation of 2-3 and 3-2 moves that provides, as part of the output, a description of how the edges are renumbered. The source code is available at <https://github.com/AlexHe98/triang-counterex>. There may be other applications for this idea of tracking how edges (or more generally, simplices) get renumbered as we perform moves.

4 Removing “unwanted” edges using a targeted search

This section discusses the algorithm for finding counterexamples to Conjectures 1, 2 and 3. We give the main ideas in 4.1 and 4.2, and then some implementation details in 4.3 and 4.4.

4.1 The degree defect and the multiplicity defect

► **Definitions 9.** Let e be an edge in a one-vertex triangulation. Let $d(e)$ denote the degree of e , and let $n(e)$ denote the number of *distinct* tetrahedra that meet e .

- The **degree defect** of e , denoted $\delta(e)$, is given by $|d(e) - 3|$.
- The **multiplicity defect** of e , denoted $\mu(e)$, is given by $d(e) - n(e)$.

Observe that we can perform a 3-2 move about an edge e , and hence remove this edge e , if and only if both the degree and multiplicity defects of e are zero. Thus, a natural way to remove all “unwanted” edges (such as core edges) from a triangulation is to minimise these defects across all “unwanted” edges. More precisely, our goal will be to minimise the following complexity with respect to the *lexicographical ordering*:

► **Definition 10.** Let \mathcal{T} be a one-vertex triangulation with some “unwanted” edges e_1, \dots, e_k , where $k \geq 1$. The **complexity** of \mathcal{T} (with respect to the “unwanted” edges) is given by

$$\left(k, \max_{1 \leq i \leq k} \mu(e_i), \max_{1 \leq i \leq k} \delta(e_i), |\mathcal{T}| \right).$$

Depending on context, the “unwanted” edges may be the core edges, tunnel edges, or edges isotopic to Seifert fibres. Our primary objective is to reduce the number of these edges, which is why this quantity appears as the first entry of our complexity. The last entry is the size of the triangulation because, all else being equal, it would be nice for our counterexamples to be as small as possible.

We have already discussed the rationale for the other two entries: we would like to reduce the degree and multiplicity defects across all “unwanted” edges. The reason for placing a higher priority on reducing multiplicity defect is best illustrated by recounting our initial approach to this problem. At first, our complexity did not involve the multiplicity defect at all. With this naïve approach, the search tended to get stuck enumerating lots of triangulations with small degree defect, but without ever actually reducing this defect to 0. We eventually realised that the search was getting trapped in a region of the Pachner graph where the “unwanted” edges had multiplicity defect equal to 2. Placing a high priority on reducing the multiplicity defect gives the search some impetus to avoid such regions.

4.2 The targeted search algorithm

► **Algorithm 11.** In a one-vertex triangulation, let P be a property of edges that is invariant under ambient isotopy. Call an edge **bad** if it satisfies property P , and **good** otherwise. This algorithm takes the following inputs:

- A one-vertex triangulation \mathcal{T} with n bad edges, where $n \geq 1$.
- A non-negative integer x (the number of extra bad edges that we are allowed to use).

To search for a new triangulation with $n - 1$ bad edges:

- (1) For each bad edge in \mathcal{T} , check whether it is possible to perform a 3-2 move about this edge. If such a move is possible, terminate and return the triangulation that results from performing this move.
- (2) Create a set \mathcal{S} . Also create a priority queue \mathcal{Q} that stores triangulations in order of increasing complexity (with respect to the bad edges). Add \mathcal{T} to both \mathcal{S} and \mathcal{Q} .
- (3) While \mathcal{Q} is non-empty:
 - (i) Remove the first triangulation \mathcal{F} (i.e., a triangulation with smallest complexity) from \mathcal{Q} , and let m denote the number of bad edges in \mathcal{F} .

- (ii) Call a 3-2 move about an edge e **eligible** if e is a good edge; for a 2-3 move, let e denote the new edge that is created by this move, and call this move **eligible** if either e is a good edge, or e is a bad edge but $m < n + x$. For each eligible move on \mathcal{F} , check whether (up to isomorphism) the set \mathcal{S} already contains the triangulation \mathcal{G} that we obtain after performing this move. If not:
- Add \mathcal{G} to both \mathcal{S} and \mathcal{Q} .
 - Perform all possible sequences of 3-2 moves about bad edges in \mathcal{G} . For each triangulation \mathcal{T}' that we obtain from such moves, if \mathcal{T}' does not already appear in the set \mathcal{S} (up to isomorphism), then add \mathcal{T}' to both \mathcal{S} and \mathcal{Q} .
 - If we can find such a sequence consisting of $m - n + 1$ 3-2 moves, then the final triangulation \mathcal{T}^* in this sequence has $n - 1$ bad edges. Terminate and return \mathcal{T}^* .

Visit <https://github.com/AlexHe98/triang-counterex> to see an implementation of this algorithm. We discuss the major details of this implementation in sections 4.3 and 4.4.

4.3 Troublesome regions, concurrent computation, and instability

In section 4.1, we mentioned regions of the Pachner graph where the “unwanted” edges have low degree defect, but multiplicity defect equal to 2 (such edges naturally occur, for instance, at the hearts of layered solid tori). Although placing a high priority on reducing multiplicity defect does help the search avoid such regions, if the search nevertheless gets trapped in such a region then it can be difficult to escape because the only way out is to perform moves that *increase* the degree defect of the “unwanted” edges.

This phenomenon occurs more generally: a locally optimal move can send the search into a “troublesome” region (which could, a priori, be *infinite*) of the Pachner graph where none of the subsequently available moves decrease the complexity. In the case above, we get stuck enumerating lots of triangulations with similar complexity; in section 5.1, we describe a case where we start enumerating lots of triangulations with *rapidly increasing* complexity.

Algorithm 11 is especially vulnerable to falling into such troublesome regions if we deal with triangulations one at a time in step 3. However, when we instead use multiple processes to deal with several triangulations concurrently, the search is sometimes able to either avoid or escape these troublesome regions. There are probably two drivers for this: (1) using multiple processes causes the search to explore with more “breadth” than a purely greedy approach, and (2) the search gains some randomness because the order in which triangulations are inserted into the priority queue \mathcal{Q} could vary each time we run the search. There may be more direct methods to achieve similar behaviour; our method was good enough, and had a low cost (both in human effort and in computational complexity).

One drawback is that it is impossible to know in advance how many concurrent processes we should use. We will see in section 5 that there is often a “sweet spot” where the search successfully terminates, but if we significantly increase or decrease the number of processes, then the search tends to get trapped in a troublesome region.

Our solution is to implement Algorithm 11 so that it periodically prints an update on the complexity of the triangulations that it is currently dealing with. If we manually observe that the search is not making satisfactory progress, then we simply restart the search, possibly with a change to the number of processes. Although it is not ideal that some human intervention is required, this is not unheard of for difficult problems in computational topology that only need to be solved once (such as finding a counterexample). Another example of this is the work done in [7] to extend the census of prime knots up to 19 crossings.

The last thing that we mention in this section is that our implementation is quite unstable. Indeed, we have already noted that the search can produce different results when we change the number of processes, or when we simply rerun the search. We have encountered one other manifestation of this instability: refactoring the source code can drastically change the behaviour of the search. We have not found a way to make the algorithm more stable.

4.4 Other implementation details

We now mention some other important details of our implementation of Algorithm 11.

First, since Algorithm 11 always considers triangulations up to isomorphism, we actually store isomorphism signatures in the set \mathcal{S} and the priority queue \mathcal{Q} . As described in [4], isomorphism signatures are an indispensable tool when exploring the Pachner graph.

Second, when the priority queue \mathcal{Q} contains multiple triangulations with the same complexity, we choose to prioritise such triangulations by insertion order.

Finally, we usually set the input variable x to be 0, in which case Algorithm 11 discards all triangulations with more bad edges than the input triangulation \mathcal{T} . However, there are occasions where taking $x > 0$ may be beneficial; we discuss one such case in section 5.3.

5 The counterexamples

We present our experimental results in sections 5.1, 5.2 and 5.3. We then show how to turn our counterexamples into infinite families in section 5.4.

Our computations were run on a laptop with an Intel Core i5-7200U processor, which has just two physical cores divided into four logical processors. It is therefore remarkable that we obtained all of our counterexamples in just a few *minutes* of wall time. This is mostly because our targeted search was able to home in on an extremely small portion of the search space: in total, we only needed to enumerate on the order of thousands of triangulations.

5.1 Triangulations with no core edges

The 20-tetrahedron triangulation \mathcal{T} with isomorphism signature

```
uLLvQQvLAPvPAQccdfeghhgmklnorsqssttthsaaggggaaaaaanaagb
```

is a one-vertex 3-sphere with no core edges (for the 3-sphere, this means that every edge is non-trivially knotted). To find this, we arbitrarily selected a one-vertex 3-sphere \mathcal{T}_2 , which has isomorphism signature `cMcabbgqs` and complexity $(2, 5, 4, 2)$. After running Algorithm 11 twice, we obtained a 22-tetrahedron one-vertex triangulation \mathcal{T}_0 of the 3-sphere with no core edges; the results are summarised in Figure 4.

To turn \mathcal{T}_0 into the smaller triangulation \mathcal{T} , we ran a breadth-first search through the Pachner graph, but with the restriction that we ignored any 2-3 move that introduces a core edge. This took approximately 367 seconds of wall time.

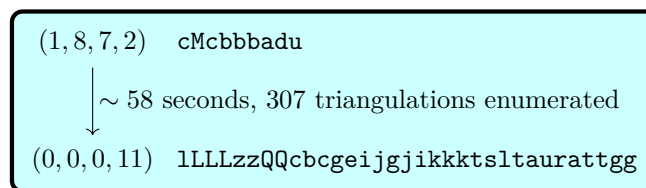
It is worth noting that to remove one core edge from \mathcal{T}_2 , we tried running Algorithm 11 with different numbers of processes, but using 24 processes produces the best results. In particular, we initially tried 12 processes, but this produced the isomorphism signature

```
sLvAAvLAzMMQQcdceflkmjmqonprqprhrvrqnkkksqeeekocksf,
```

which has complexity $(1, 2, 1, 18)$. The core edge in this triangulation has multiplicity defect 2 but degree defect 1; this is exactly one of the troublesome cases that we mentioned in section 4.3. We needed to increase the number of processes to 24 to avoid this.



■ **Figure 4** Removing core edges from cMcabbgqs (3-sphere); $x = 0$.



■ **Figure 5** Removing tunnel edges from cPcbbbadu (trefoil knot); $x = 0$, 8 processes.

Surprisingly, increasing the number of processes significantly beyond 24 also has an adverse affect on the effectiveness of Algorithm 11. For example, with 36 processes, although the search is sometimes able to remove a core edge, it seems to do so less reliably. Instead, we find that the search has a tendency to get trapped in a different type of troublesome region:

- After about 10 seconds, the search reaches a triangulation with complexity (2, 4, 16, 15).
- After about 20 seconds, the search reaches a triangulation with complexity (2, 4, 23, 22).
- After about 30 seconds, the search reaches a triangulation with complexity (2, 4, 26, 25).

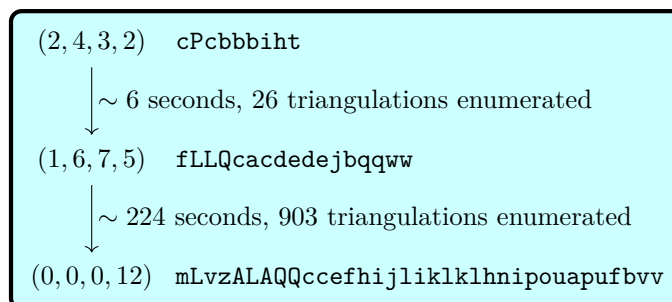
When this happens, the complexity only increases further if we allow the search to continue.

5.2 Triangulations with no tunnel edges

We found ideal triangulations with no tunnel edges for three knots with tunnel number one:

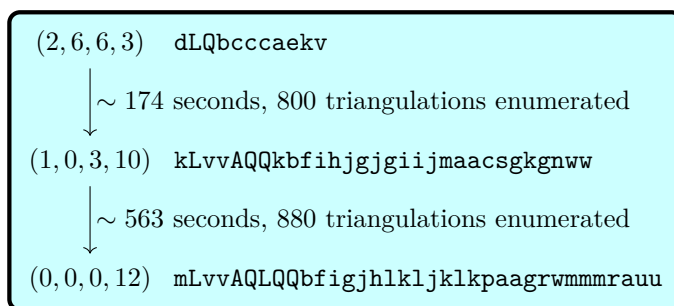
- the trefoil knot (results summarised in Figure 5);
- the figure-eight knot (results summarised in Figure 6); and
- the (5, 2) torus knot (results summarised in Figure 7).

Although the triangulations here are relatively small, and although we only enumerated relatively few triangulations, our running times are much longer than those in section 5.1.



■ **Figure 6** Removing tunnel edges from cPcbbbiht (figure-eight knot); $x = 0$, 24 processes.

21:12 Finding Large Counterexamples by Selectively Exploring the Pachner Graph



■ **Figure 7** Removing tunnel edges from dLQbcccaekv ((5, 2) torus knot); $x = 0$, 24 processes.

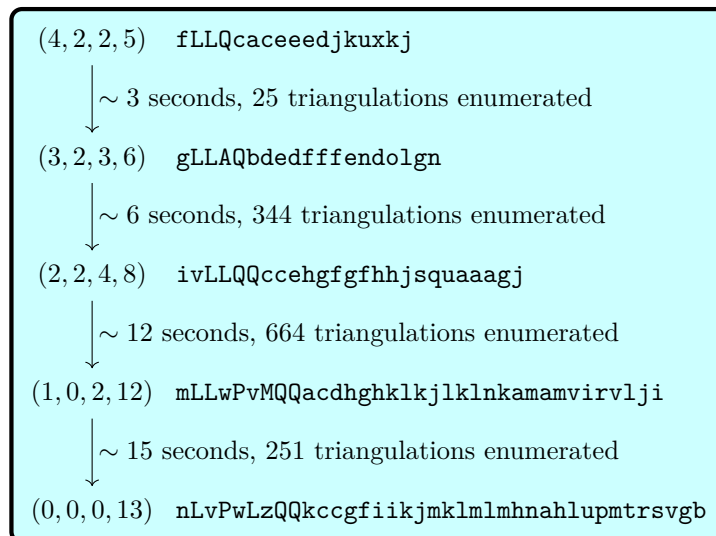
This is probably because genus-2 handlebodies are harder to recognise than solid tori.

5.3 Triangulations with no edges isotopic to Seifert fibres

We found the following 11-tetrahedron one-vertex triangulation \mathcal{T} of a small Seifert fibre space¹ with no edges isotopic to Seifert fibres:

lLLLLPMQccddfjiihikkkpkrwaaacttvc.

To do this, we began with the isomorphism signature fLLQcaceeedjkuxkj. Running Algorithm 11 four times with $x = 1$ and 12 processes produced a 13-tetrahedron triangulation \mathcal{T}^* with no edges isotopic to Seifert fibres; the results are summarised in Figure 8. We then used a breadth-first search (similar to the one from section 5.1) to turn \mathcal{T}^* into \mathcal{T} .



■ **Figure 8** Removing edges isotopic to Seifert fibres from fLLQcaceeedjkuxkj; $x = 1$, 12 processes.

The reason for taking $x = 1$ instead of $x = 0$ comes from early versions of our algorithm for detecting good edges (in this context, edges not isotopic to Seifert fibres): we initially included fewer normal surfaces in the analysis, so the results were not particularly conclusive.

¹ Specifically, this is fibred over the 2-sphere with three exceptional fibres with slopes $\frac{1}{2}$, $\frac{2}{3}$ and $-\frac{1}{3}$.

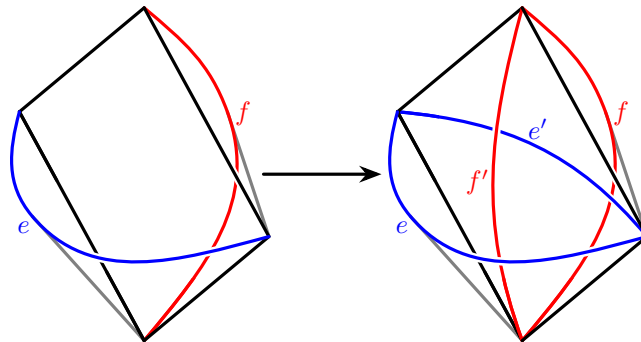
At the time, we encountered triangulations for which every 2-3 move would create an edge that we could not certify as good, so our short-term solution was to allow extra bad edges if required. In the current version of the code, taking $x = 1$ no longer seems strictly necessary.

5.4 From one counterexample to infinitely many

The following result allows us to turn our counterexamples into infinite families:

► **Proposition 12.** *Let \mathcal{M} be either a closed 3-manifold, or a 3-manifold with a single boundary component of positive genus (and no other boundary components). In the closed case, let \mathcal{T} be a one-vertex triangulation of \mathcal{M} ; in the bounded case, let \mathcal{T} be a one-vertex ideal triangulation of \mathcal{M} . Let P be a property of edges of \mathcal{T} that is invariant under ambient isotopy. Call \mathcal{T} **interesting** if every edge in \mathcal{T} satisfies P . If \mathcal{M} has an interesting triangulation, then \mathcal{M} has infinitely many interesting triangulations.*

Proof. Let \mathcal{T} be an interesting triangulation of \mathcal{M} . Fix a tetrahedron Δ of \mathcal{T} , and let e and f denote a pair of opposite edges of Δ . We obtain a new triangulation \mathcal{T}' of \mathcal{M} by replacing Δ with a three-tetrahedron gadget, as shown in Figure 9. In terms of 1-skeletons, observe that all we have done is introduce two new edges e' and f' such that e' is isotopic to e and f' is isotopic to f . Thus, \mathcal{T}' is an interesting triangulation of \mathcal{M} . Repeating this procedure indefinitely gives the desired infinite family of interesting triangulations of \mathcal{M} . ◀



■ **Figure 9** Building a new interesting triangulation.

► **Corollary 13.** *The 3-sphere has infinitely many triangulations with no core edges.*

► **Corollary 14.** *Let K be the trefoil knot, figure-eight knot, or $(5, 2)$ torus knot. Then K has infinitely many ideal triangulations with no tunnel edges.*

► **Corollary 15.** *There exists a small Seifert fibre space that has infinitely many triangulations with no edges isotopic to Seifert fibres.*

6 Discussion

6.1 Unanswered questions

For knots with tunnel number one, we found triangulations with no tunnel edges for all three knots that we considered. There are many other such knots in the census of knots [7] – in particular, every torus knot has tunnel number one – so it would be interesting to see whether such counterexamples exist for all of these knots. A key obstacle to a systematic investigation is the requirement for human supervision when running Algorithm 11, as discussed in section 4.3. In any case, we tentatively propose the following conjecture:

► **Conjecture 16.** *Every knot with tunnel number one admits infinitely many ideal triangulations with no tunnel edges.*

In contrast, we have only been able to find triangulations with no core edges for the 3-sphere; we have tried two other lens spaces – $L_{3,1}$ and $L_{5,1}$ – but without success. Similarly, although we have considered several small Seifert fibre spaces, we currently only have counterexamples for one case. We nevertheless suspect that the answer to the following two questions is “yes”, but new insights may be required to verify this:²

► **Question 17.** *Is there a lens space other than the 3-sphere that admits infinitely many triangulations with no core edges?*

► **Question 18.** *Is there another small Seifert fibre space that admits infinitely many triangulations with no edges isotopic to Seifert fibres?*

6.2 Future applications

Elementary moves such as 2-3 and 3-2 moves have many computational applications beyond how we used them in this paper. The two most common are:

- (1) using moves to improve a triangulation, which is often critical for making exponential-time computations feasible; and
- (2) finding a sequence of moves that transforms a triangulation \mathcal{T} into another triangulation \mathcal{T}' , which gives a computational proof that \mathcal{T} and \mathcal{T}' are homeomorphic.

Finding the right sequence of moves can be difficult, so a targeted search could be useful. In particular, increasing the number of tetrahedra is sometimes unavoidable [4]:

- (1) There are many triangulations \mathcal{T} of the 3-sphere such that to simplify \mathcal{T} to a smallest-possible triangulation via 2-3 and 3-2 moves, we must visit at least one intermediate triangulation with two more tetrahedra than \mathcal{T} . There is also a triangulation of a graph manifold for which simplification requires *three* additional tetrahedra.
- (2) In the census of minimal triangulations up to 9 tetrahedra, there are many 3-manifolds for which two additional tetrahedra are required to connect all the minimal triangulations using 2-3 and 3-2 moves. There is also one 3-manifold – the lens space $L_{3,1}$ – for which *three* additional tetrahedra are required to connect all the minimal triangulations.

There are also situations where we might actually *want* to increase the number of tetrahedra, provided this improves the triangulation with respect to another quantity. For example, some algorithms in 3-manifold topology are known to be fixed-parameter tractable in a quantity called the **treewidth** [10, 12]; for such algorithms, minimising the treewidth is more important than minimising the number of tetrahedra.

As mentioned in section 1, increasing the number of tetrahedra is accompanied by a super-exponential increase in the number of triangulations; even if we fix a 3-manifold, this is still at least exponential. This means that exhaustive searches quickly run into problems with not only running time, but also (often more importantly) memory management. However, our work suggests that, with the right heuristics, we may be able to circumvent these problems; the challenge is to actually devise such heuristics in the settings mentioned above.

² The full version [9] presents new results that answer these questions (but also raise new questions).

References

- 1 Gennaro Amendola. A calculus for ideal triangulations of three-manifolds with embedded arcs. *Math. Nachr.*, 278(9):975–994, 2005. doi:10.1002/mana.200310285.
- 2 David Bremner and Lars Schewe. Edge-graph diameter bounds for convex polytopes with few facets. *Experiment. Math.*, 20(3):229–237, 2011. doi:10.1080/10586458.2011.564965.
- 3 Benjamin A. Burton. Detecting genus in vertex links for the fast enumeration of 3-manifold triangulations. In *Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, pages 59–66. Association for Computing Machinery, 2011. doi:10.1145/1993886.1993901.
- 4 Benjamin A. Burton. Simplification paths in the Pachner graphs of closed orientable 3-manifold triangulations. [arXiv:1110.6080](https://arxiv.org/abs/1110.6080), 2011.
- 5 Benjamin A. Burton. Computational topology with Regina: Algorithms, heuristics and implementations. In *Geometry and Topology Down Under*, volume 597 of *Contemporary Mathematics*, pages 195–224. American Mathematical Society, 2013. doi:10.1090/conm/597.
- 6 Benjamin A. Burton. A new approach to crushing 3-manifold triangulations. *Discrete Comput. Geom.*, 52:116–139, 2014. doi:10.1007/s00454-014-9572-y.
- 7 Benjamin A. Burton. The Next 350 Million Knots. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 25:1–25:17. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.SoCG.2020.25.
- 8 Benjamin A. Burton, Ryan Budney, William Pettersson, et al. Regina: Software for low-dimensional topology. <https://regina-normal.github.io>, 1999–2022.
- 9 Benjamin A. Burton and Alexander He. Finding large counterexamples by selectively exploring the Pachner graph. [arXiv:2303.06321](https://arxiv.org/abs/2303.06321), 2023.
- 10 Benjamin A. Burton, Clément Maria, and Jonathan Spreer. Algorithms and complexity for Turaev-Viro invariants. *Journal of Applied and Computational Topology*, 2:33–53, 2018. doi:10.1007/s41468-018-0016-2.
- 11 Benjamin A. Burton and Melih Özlen. A fast branching algorithm for unknot recognition with experimental polynomial-time behaviour. [arXiv:1211.1079](https://arxiv.org/abs/1211.1079), 2012. To appear in *Math. Program.*
- 12 Benjamin A. Burton and Jonathan Spreer. The complexity of detecting taut angle structures on triangulations. In *Proceedings of the 2013 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 168–183. Society for Industrial and Applied Mathematics, 2013. doi:10.1137/1.9781611973105.13.
- 13 Wolfgang Haken. Theorie der Normalflächen. *Acta Math.*, 105:245–375, 1961. doi:10.1007/BF02559591.
- 14 Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *J. ACM*, 46(2):185–211, 1999. doi:10.1145/301970.301971.
- 15 Allen Hatcher. Notes on Basic 3-Manifold Topology. <https://pi.math.cornell.edu/~hatcher/3M/3Mfds.pdf>, 2007.
- 16 William Jaco and J. Hyam Rubinstein. 0-efficient triangulations of 3-manifolds. *J. Differential Geom.*, 65:61–168, 2003. doi:10.4310/jdg/1090503053.
- 17 William Jaco and Peter B. Shalen. A new decomposition theorem for irreducible sufficiently large 3-manifolds. In R. James Milgram, editor, *Algebraic and Geometric Topology, Part 2*, volume XXXII of *Proceedings of Symposia in Pure Mathematics*, pages 71–84. American Mathematical Society, 1978. doi:10.1090/pspum/032.2.
- 18 William Jaco and Jeffrey L. Tollefson. Algorithms for the complete decomposition of a closed 3-manifold. *Illinois J. Math.*, 39(3):358–406, 1995. doi:10.1215/ijm/1255986385.
- 19 Klaus Johannson. *Homotopy Equivalences of 3-Manifolds with Boundaries*, volume 761 of *Lecture Notes in Mathematics*. Springer-Verlag, 1979. doi:10.1007/BFb0085406.
- 20 Victor Klee. Paths on polyhedra. II. *Pacific J. Math.*, 17(2):249–262, 1966. doi:10.2140/pjm.1966.17.249.

21:16 Finding Large Counterexamples by Selectively Exploring the Pachner Graph

- 21 Marc Lackenby and Saul Schleimer. Recognising elliptic manifolds. [arXiv:2205.08802](https://arxiv.org/abs/2205.08802), 2022.
- 22 Bruno Martelli and Carlo Petronio. Three-manifolds having complexity at most 9. *Experiment. Math.*, 10(2):207–236, 2001. doi:10.1080/10586458.2001.10504444.
- 23 Sergei V. Matveev. *Algorithmic Topology and Classification of 3-Manifolds*, volume 9 of *Algorithms and Computation in Mathematics*. Springer-Verlag, second edition, 2007. doi:10.1007/978-3-540-45899-9.
- 24 Sergei V. Matveev. Atlas of 3-Manifolds. <http://www.matlas.math.csu.ru/>, accessed November 2022.
- 25 Grisha Perelman. The entropy formula for the Ricci flow and its geometric applications. [arXiv:math/0211159](https://arxiv.org/abs/math/0211159), 2002.
- 26 Grisha Perelman. Finite extinction time for the solutions to the Ricci flow on certain three-manifolds. [arXiv:math/0307245](https://arxiv.org/abs/math/0307245), 2003.
- 27 Grisha Perelman. Ricci flow with surgery on three-manifolds. [arXiv:math/0303109](https://arxiv.org/abs/math/0303109), 2003.
- 28 Riccardo Piergallini. Standard moves for standard polyhedra and spines. In *III Convegno Nazionale Di Topologia: Trieste, 9-12 Giugno 1986 : Atti*, number 18 in Rend. Circ. Mat. Palermo (2) Suppl., pages 391–414, 1988. doi:11581/244540.
- 29 J. Hyam Rubinstein. An algorithm to recognize the 3-sphere. In *Proceedings of the International Congress of Mathematicians*, pages 601–611. Birkhäuser, 1995. doi:10.1007/978-3-0348-9078-6_54.
- 30 J. Hyam Rubinstein, Henry Segerman, and Stephan Tillmann. Traversing three-manifold triangulations and spines. *Enseign. Math.*, 65(1/2):155–206, 2019. doi:10.4171/LEM/65-1/2-5.
- 31 Francisco Santos. A counterexample to the Hirsch Conjecture. *Ann. of Math.*, 176(1):383–412, 2012. doi:10.4007/annals.2012.176.1.7.
- 32 Abigail Thompson. Thin position and the recognition problem for S^3 . *Math. Res. Lett.*, 1(5):613–630, 1994. doi:10.4310/MRL.1994.v1.n5.a9.
- 33 William P. Thurston. *Three-Dimensional Geometry and Topology*. Princeton University Press, 1997. doi:10.1515/9781400865321.

Improved Algebraic Degeneracy Testing

Jean Cardinal  

Université Libre de Bruxelles, Belgium

Micha Sharir  

School of Computer Science, Tel Aviv University, Israel

Abstract

In the classical linear degeneracy testing problem, we are given n real numbers and a k -variate linear polynomial F , for some constant k , and have to determine whether there exist k numbers a_1, \dots, a_k from the set such that $F(a_1, \dots, a_k) = 0$. We consider a generalization of this problem in which F is an arbitrary constant-degree polynomial, we are given k sets of n real numbers, and have to determine whether there exists a k -tuple of numbers, one in each set, on which F vanishes. We give the first improvement over the naïve $O^*(n^{k-1})$ algorithm for this problem (where the $O^*(\cdot)$ notation omits subpolynomial factors).

We show that the problem can be solved in time $O^*\left(n^{k-2+\frac{4}{k+2}}\right)$ for even k and in time $O^*\left(n^{k-2+\frac{4k-8}{k^2-5}}\right)$ for odd k in the real RAM model of computation. We also prove that for $k = 4$, the problem can be solved in time $O^*(n^{2.625})$ in the algebraic decision tree model, and for $k = 5$ it can be solved in time $O^*(n^{3.56})$ in the same model, both improving on the above uniform bounds.

All our results rely on an algebraic generalization of the standard meet-in-the-middle algorithm for k -SUM, powered by recent algorithmic advances in the polynomial method for semi-algebraic range searching. In fact, our main technical result is much more broadly applicable, as it provides a general tool for detecting incidences and other interactions between points and algebraic surfaces in any dimension. In particular, it yields an efficient algorithm for a general, algebraic version of Hopcroft's point-line incidence detection problem in any dimension.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms; Theory of computation \rightarrow Computational geometry

Keywords and phrases Degeneracy testing, k -SUM problem, incidence bounds, Hocroft's problem, polynomial method, algebraic decision trees

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.22

Related Version *Full Version:* <https://arxiv.org/abs/2212.03030>

Funding *Micha Sharir:* Work by Micha Sharir was partially supported by ISF grant 260/18.

Acknowledgements This work was initiated during a visit of the authors to the group of Pr. Emo Welzl at the Swiss Federal Institute of Technology (ETH) in Zürich, Switzerland.

1 Introduction

Linear degeneracy testing is a computational problem in which we are given n real numbers and a k -variate linear polynomial F , for some fixed constant k , as input, and we seek k numbers a_1, \dots, a_k from the input set such that $F(a_1, \dots, a_k) = 0$ [4, 22]. An important special case is the 3SUM problem, in which $k = 3$ and F is simply the sum of the three variables. This problem was first studied as a bottleneck problem in computational geometry [30], since it is reducible, in subquadratic time, to many problems in computational geometry, which are now known as *3SUM-hard problems*. It has acquired over the years the status of a basic hard problem in fine-grained complexity theory [45]. The case where k is an arbitrary fixed constant and F is the sum of the k variables is a fixed-parameter version of the NP-complete subset sum problem, usually referred to as the k -SUM problem [13, 31, 35, 39]. Linear



© Jean Cardinal and Micha Sharir;

licensed under Creative Commons License CC-BY 4.0

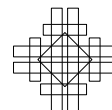
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 22; pp. 22:1–22:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



degeneracy testing has a higher-dimensional counterpart that is of crucial importance in computational geometry, called *affine degeneracy testing in \mathbb{R}^d* : Given a set of n points in \mathbb{R}^d , decide whether there exist a $(d+1)$ -tuple of points lying on a $(d-1)$ -flat [23]. Many problems reduce to degeneracy testing in that sense, including degeneracy of Voronoi diagrams or checking for incidences between geometric objects. Quoting Ailon and Chazelle [4], “the list of problems (...) that can be reduced to degeneracy testing is nearly endless”.

In this contribution, we consider an algebraic generalization of the linear degeneracy testing problem that we call *k-POL*, in which the polynomial F can be an arbitrary, bounded-degree k -variate polynomial. For simplicity, we consider the k -partite version of the problem, in which we are given k sets of n real numbers, and we consider k -tuples formed by picking one number in each set.¹

► **Problem 1** (*k-POL*). *Given k sets A_1, \dots, A_k , each of n real numbers, and a constant-degree real k -variate polynomial $F \in \mathbb{Z}[x_1, \dots, x_k]$, determine whether F has a zero in the Cartesian product $A_1 \times A_2 \times \dots \times A_k$, that is, determine whether there exist $a_1 \in A_1, \dots, a_k \in A_k$ such that $F(a_1, \dots, a_k) = 0$.*

The k -POL problem can be solved in $O(n^{k-1} \log n)$ time using standard algebraic tools, such as the ones described in Basu, Pollack, and Roy [11]. To do so, we iterate over all $(k-1)$ -tuples (a_1, \dots, a_{k-1}) in $A_1 \times \dots \times A_{k-1}$, find the $O(1)$ real roots of $F(a_1, \dots, a_{k-1}, x) = 0$ for each such tuple, and sort the overall set of $O(n^{k-1})$ roots. Then, for each $a_k \in A_k$ we search a_k in the resulting sequence, and declare a positive solution to the k -POL problem if and only if one of the searches succeeds. To this date, no better algorithm is known for this problem.

In this work, we present the first algorithm improving over this elementary method, that solves k -POL in time $O^*(n^{k-2+f(k)})$ for any $k > 2$, where $f(k) = 4/k + O(1/k^2)$ (a more precise expression for $f(k)$ is stated in the abstract and also given later), and the $O^*(\cdot)$ notation hides subpolynomial factors. We also show how to further improve on our bounds in the cases $k = 4$ and $k = 5$, in the nonuniform algebraic decision tree model. Before stating our results in full detail, we briefly review the previous works on both classical and more recent variants of the problem.

1.1 Previous related work

The best known upper bounds on the complexity of linear degeneracy testing in the uniform real RAM model are $O(n^{k/2} \log n)$ for even values of k , and $O(n^{\lceil k/2 \rceil})$ for odd values of k . The folklore algorithm yielding these upper bounds is referred to as the *meet-in-the-middle* algorithm, due to its similarity to meet-in-the-middle attacks in cryptography. In the nonuniform k -linear decision tree model, where only linear sign tests involving k distinct input numbers are allowed and accounted for, Erickson proved that $\Omega(n^{\lceil k/2 \rceil})$ queries are necessary [22]. Ailon and Chazelle [4] gave a similar lower bound for t -linear decision trees, where t is slightly larger than k . Note that linear degeneracy testing has long been known to be solvable in polynomial time in the n -linear decision tree model (hence with unrestricted linear sign tests), with the degree of the polynomial in the bound independent of n [8, 38]. The bound was subsequently improved by Cardinal, Iacono, and Ooms [13] and Ezra and

¹ The single set version of the problem can be recovered by letting all sets be identical. This, however, allows us to use the same number more than once. The reduction to the case where numbers can be picked at most once is nontrivial, see for instance [15, 19]. We will skip over these issues here.

Sharir [24]. In a remarkable breakthrough paper, Kane, Lovett, and Moran [33] finally managed to show that linear degeneracy testing could be solved in $O(n \log^2 n)$ time in the $2k$ -linear decision tree model.

The k -SUM problem is the special case of linear degeneracy testing in which the polynomial F is simply the sum of the k variables. It is known to not be solvable in the uniform model in time $n^{o(k)}$ under the exponential time hypothesis (ETH) [39]. The simplest instance of k -SUM is the well-known 3SUM problem, which is the case where $k = 3$. For a long time, the 3SUM problem has been conjectured to require $\Omega(n^2)$ time. Subquadratic algorithms in the word RAM model of computation were first presented by Baran, Demaine, and Pătraşcu for the case where the input consists of integers [9]. It is only in 2014 that the real version was shown to be solvable in (slightly) subquadratic time by Grønlund and Pettie [32]. Further improvements were given by Chan [14]. Improvements on the decision tree complexity of 3SUM [32], and later in [29, 31], have been proposed before it was shown to be solvable with $O(n \log^2 n)$ 6-linear queries, as a special case of the aforementioned algorithm for linear degeneracy testing [33]. While some slightly subquadratic-time uniform algorithms exist, the existence of a uniform algorithm solving 3SUM in time $O(n^{2-\delta})$ for some positive constant δ is still a major open problem. It has recently been shown that all nontrivial linear degeneracy testing problems for $k = 3$ are equivalent under subquadratic reductions [19]. This makes 3SUM one of the cornerstone computational problems in fine-grained complexity theory [45]. The 4-SUM problem is closely related to the so-called “Sorting $X + Y$ ” problem, in which we are asked to sort the pairwise sums of elements in two sets of n real numbers [27]. See also [35] for recent results on k -SUM.

The 3-POL problem, in which we look for three input numbers on which an arbitrary given bounded-degree polynomial F vanishes, has first been studied in Barba et al. [10] in both the real RAM and the *algebraic decision tree* models. In an algebraic decision tree, we only count sign tests of constant-degree polynomials in the input data, and again forbid any other operation to access the data explicitly; see [12, 40] and below. As shown in [10], the 3-POL problem can be solved in this model with only $O^*(n^{12/7})$ sign tests, an improvement over the $O^*(n^2)$ uniform upper bound.

When the three input sets A, B, C are sets of points in the plane, we obtain the *collinearity testing* problem as a special case, in which we want to determine whether $A \times B \times C$ contains a collinear triple. Collinearity testing is a classical 3SUM-hard problem in computational geometry [30] for which no subquadratic algorithm is known, even in the algebraic decision tree model; see [7, 10] for a discussion. In the uniform model, the problem can be solved in $O(n^2)$ time. The primitive operation needed to test for collinearity of a specific triple (a, b, c) is the so-called *orientation test*, in which we test for the sign of the determinant

$$\begin{vmatrix} 1 & x_a & y_a \\ 1 & x_b & y_b \\ 1 & x_c & y_c \end{vmatrix},$$

which is a quadratic polynomial in the six coordinates of a triple of points in $A \times B \times C$. Consequently, it is natural, and in fact necessary, to use the more general algebraic decision tree model mentioned above (instead of linear decision trees). Partial results with subquadratic algorithms in the algebraic decision tree model, both for the general 3-POL problem in the plane and for collinearity testing, have been obtained by Aronov, Ezra, and Sharir [7].

For the more general problem of affine degeneracy testing in \mathbb{R}^d , Erickson and Seidel [23] proved a lower bound of $\Omega(n^d)$ on the number of *sidedness queries*, in which one asks whether a point lies above, on, or below some hyperplane. The lower bound is matched by well-known algorithms [21]. The existence of a better real RAM algorithm, that would use more sophisticated operations than only sidedness queries, is a major open problem.

■ **Table 1** Upper bounds on the uniform complexity of k -POL for the first few values of $k \geq 4$.

k	4	5	6	7	8
Exponent	2.666...	3.6	4.5	5.4545...	6.4

1.2 Our results

We provide the first algorithm for the k -POL problem that achieves a polynomial-factor improvement over the naïve method.

► **Theorem 2.** *The k -POL problem is solvable in randomized expected time $O^*\left(n^{k-2+\frac{4}{k+2}}\right)$ for even k and $O^*\left(n^{k-2+\frac{4k-8}{k^2-5}}\right)$ for odd k in the real RAM computation model.*

Note that the speedup factor over the simple $O(n^{k-1} \log n)$ algorithm mentioned earlier gets close to linear for large k . The exponents for a few small values of $k \geq 4$ are given in Table 1. In the cases $k = 4$ and $k = 5$, we can be further improve those bounds, albeit only in the more powerful algebraic decision tree model.

► **Theorem 3.** *The 4-POL problem is solvable in time $O^*(n^{21/8})$ in the algebraic decision tree computation model.*

► **Theorem 4.** *The 5-POL problem is solvable in time $O^*(n^{210/59})$ in the algebraic decision tree computation model.*

The exponents in the nonuniform bounds in Theorems 3 and 4 are 2.625 and ~ 3.56 , respectively, which improve on the corresponding uniform bounds for $k = 4$ (2.666...) and $k = 5$ (3.6); see Table 1.

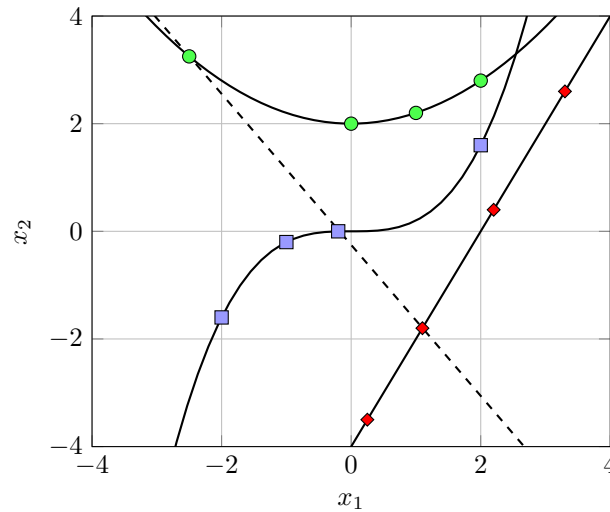
Application to affine degeneracy testing in d -space. One motivation for studying the k -POL problem is the following restricted version of affine degeneracy testing in \mathbb{R}^d . Let A_1, \dots, A_k be k sets of n points in \mathbb{R}^d , where each of the sets A_i lies on its own algebraic curve γ_i . The goal is to decide, in the real RAM computation model, whether there exist k points, one in each set, that lie on a common $(k-2)$ -flat. We suppose that each γ_i is polynomially parameterizable. That is, γ_i is given by the equations $x_j = f_{i,j}(t)$, for $j = 1, \dots, d$, $t \in \mathbb{R}$, where the $f_{i,j}$ are polynomials of some constant maximum degree. Hence, we may represent each set A_i as a set of n real numbers, which are the values of the parameter t that define its points. An example of an instance with $n = 4$ and $k = 3$ is shown in Figure 1.

Up to a simple randomized preprocessing, we can assume that $k = d + 1$, that is, we reduce the problem to one in which we look for $d + 1$ points lying on a common $(d-1)$ -flat. Indeed, take a generic projection $\pi : \mathbb{R}^d \rightarrow \mathbb{R}^{k-1}$ with respect to $S = A_1 \cup \dots \cup A_k$, namely a projection that does not introduce additional degeneracies in S . Thus a k -tuple of points of $\pi(S)$ lies on a hyperplane in \mathbb{R}^{k-1} if and only if its preimage in S lies on a $(k-2)$ -flat. Note that for natural choices of distributions, a *random projection* is generic with respect to S with probability one (see for instance [5] for a discussion).

We now consider the reduced problem, and note that the condition that $d + 1$ points $a_1 \in A_1, \dots, a_{d+1} \in A_{d+1}$ lie on a common hyperplane is that the matrix

$$\begin{bmatrix} 1 & f_{1,1}(a_1) & f_{1,2}(a_1) & \cdots & f_{1,d}(a_1) \\ 1 & f_{2,1}(a_2) & f_{2,2}(a_2) & \cdots & f_{2,d}(a_2) \\ & & \cdots & & \\ 1 & f_{d+1,1}(a_{d+1}) & f_{d+1,2}(a_{d+1}) & \cdots & f_{d+1,d}(a_{d+1}) \end{bmatrix}$$

has determinant zero. This determinant is a bounded-degree polynomial in the input. Applying Theorem 2, we directly get the following result.



■ **Figure 1** An instance of the affine degeneracy testing in \mathbb{R}^2 for $n = 4$ and $k = 3$.

► **Corollary 5** (Constrained affine degeneracy testing in \mathbb{R}^d). *Let A_1, \dots, A_k be k sets of n points in \mathbb{R}^d , such that each of the sets A_i lies on its own constant-degree algebraic curve, where all these curves are polynomially parameterizable. Then one can decide, in the real RAM computation model, whether there exists k points, one in each set, that lie on a common $(k - 2)$ -flat, in randomized expected time $O^*(n^{k-2+\frac{4}{k+2}})$ for even k and $O^*(n^{k-2+\frac{4k-8}{k^2-5}})$ for odd k .*

Hopcroft’s problem. Hopcroft’s classical problem is that of determining, given two collections of n points and of n lines in the plane, whether some point lies on some line. An elegant algorithm relying on cuttings was proposed by Matoušek [36], with running time $n^{4/3}2^{O(\log^* n)}$. It has recently been slightly improved to $O(n^{4/3})$ by Chan and Zheng [16]. Our main technical result, given in Theorem 6, is an algebraic generalization of this result, in which we wish to detect incidences between points and algebraic surfaces of codimension 1 and of constant degree in arbitrary dimension. This involves a careful use of recent algorithmic methods for hierarchical polynomial partitions, which also arise in the context of semi-algebraic range searching; see below.

Organization of the paper. The next section is dedicated to solving this generalized algebraic version of Hopcroft’s problem. In Section 3, we apply this result to the k -POL problem and prove Theorem 2. In Section 4, we consider algorithms in the algebraic decision tree model and prove Theorems 3 and 4.

2 Hopcroft’s Problem generalized

Our main technical result is of independent interest, and provides a broad generalization of the classical Hopcroft’s problem [16, 36], originally formulated for points and lines in the plane.

2.1 Statement

► **Theorem 6** (General Hopcroft's problem). *Let $s \geq 1$ and $t \geq 1$ be a pair of integers, and put $k = s + t$. Let F be a real k -variate polynomial of constant degree. For any set P of N points in \mathbb{R}^t , and any set Q of M points in \mathbb{R}^s , deciding whether there exists a pair $(p, q) \in P \times Q$ such that $F(p_1, \dots, p_t, q_1, \dots, q_s) = 0$ can be done in randomized expected time*

$$O^* \left(M^{1 - \frac{t-1}{ts-1}} N^{1 - \frac{s-1}{ts-1}} + M + N \right)$$

in the real RAM model. Furthermore, one can obtain, within the same time and output size bounds, a compact encoding of the signs $\text{sign}(F(p_1, \dots, p_t, q_1, \dots, q_s)) \in \{0, +, -\}$ for all $(p, q) \in P \times Q$.

Hopcroft's problem itself is a special case with $t = s = 2$ and $F(p_1, p_2, q_1, q_2) = p_1 q_1 + p_2 - q_2$ (with a suitable parameterization of the input lines). Our bound in this case is $O^*(M^{2/3} N^{2/3} + M + N)$, which is close to the best known upper bound $O(M^{2/3} N^{2/3} + M + N)$, due to Chan and Zheng [16]. The case $s = t = 6$ was stated by Aronov et al. [6] for another application. A special case of Theorem 6 with $t = s = 2$ and $N = M$ was also stated by Barba et al. [10, Lemma 6.8]. When $s \neq t$, we obtain an *asymmetric* incidence detection problem, with distinct primal and dual space dimensions. (For a simple instance of this asymmetric setup, think of points and arbitrary circles in the plane.)

2.2 Proof

To prove Theorem 6, we employ a modified version of the recent machinery of Matoušek and Patáková [37, Theorems 1.1 and 1.3] for range searching with semi-algebraic sets in higher dimensions (see also [1, 2, 3, 25, 26] for related results and applications). The original study [37] gives, for N input points in \mathbb{R}^t , a data structure of size $O^*(N)$ that can be constructed in $O^*(N)$ randomized expected time, so that a query with a constant-complexity semi-algebraic range can be answered in $O^*(N^{1-1/t})$ time. The main technical result on which their algorithm is based is the following lemma, which will be the main technical tool for our algorithm too. (We have changed some of the notations in the lemma statement to conform with the other notations used in this work.)

► **Lemma 7** (Matoušek and Patáková [37]). *For every integer $t > 1$ there is a constant K such that the following hold. Given an N -point set $P \subset \mathbb{R}^t$ and a parameter $r > 1$, there are numbers $r_1, r_2, \dots, r_t \in [r, r^K]$, positive integers $\ell_1, \ell_2, \dots, \ell_t$, all of which are $O(r^C)$, for a suitable constant $C = C(t)$ depending on t , a partition*

$$P = P^* \cup \bigcup_{i=1}^t \bigcup_{j=1}^{\ell_i} P_{ij}$$

of P into disjoint subsets, and for every i, j , a connected set $S_{ij} \subseteq \mathbb{R}^t$, which is semi-algebraic of constant complexity (that depends on r), containing P_{ij} , such that $|P_{ij}| \leq n/r_i$ for all i, j , $|P^| \leq r^K$, and the following holds:*

(★) *If h is a t -variate real polynomial of bounded degree and $Z(h)$ is its zero set then, for every $i = 1, 2, \dots, t$, the number of the sets S_{ij} crossed by (intersected by but not contained in) $Z(h)$ is at most $O(r_i^{1-1/t})$.*

The implied constants and parameters depend on the maximum degree of the polynomials defining the partition, and the constant in the crossing bound for $Z(h)$ also depends on the degree of h . Furthermore, the sets P^ , P_{ij} , and S_{ij} can be constructed, in the standard model of algebraic computation [11], in randomized expected time $O(nr^C)$.*

Proof of Theorem 6. We map each point $q = (q_1, \dots, q_s) \in Q$ to the surface $\sigma_q = \{(x_1, \dots, x_t) \mid F(x_1, \dots, x_t, q_1, \dots, q_s) = 0\}$ in \mathbb{R}^t . Let Σ denote the resulting collection of these M surfaces. Note that the surfaces of Σ have s degrees of freedom. If any of these surfaces is the entire t -space we terminate the algorithm right away with a positive outcome. Otherwise, the problem has been reduced to the problem of detecting an incidence between some point of P and some surface of Σ . Note that the problem is fully symmetric in P and Q , so we can also represent it in \mathbb{R}^s , where the points of Q are represented as points, and those of P as surfaces.

Fix a sufficiently large constant parameter r (see below for more concise details concerning this choice). If $N \geq M$ we apply Lemma 7 to $P \subset \mathbb{R}^t$, and if $N \leq M$ we apply Lemma 7 to the corresponding points and surfaces in the dual representation in \mathbb{R}^s . Assume without loss of generality that $N \geq M$, and follow the notations in Lemma 7.

For each $i = 1, 2, \dots, t$, each surface in Σ crosses (intersects but does not contain) at most $O\left(r_i^{1-1/t}\right)$ sets S_{ij} . Denote by Σ_{ij} the subset of surfaces that cross S_{ij} . If there is a surface σ_q that fully contains S_{ij} then we have an incidence between σ_q and each point $p \in P_{ij}$ (assuming, as we may, that none of the sets P_{ij} is empty). Otherwise, let $q_{ij} = |\Sigma_{ij}|$. Then from (\star) we have, for each i ,

$$\sum_{j=1}^{\ell_i} q_{ij} \leq cMr_i^{1-1/t},$$

for a suitable constant $c > 0$, independent of r . For each $i = 1, \dots, t$ and $j = 1, \dots, \ell_i$, we face a subproblem involving P_{ij} and Σ_{ij} , of respective sizes at most N/r_i and q_{ij} , which we solve recursively, possibly switching to the dual setup, depending on which of these two sizes is larger. In addition, we have the leftover set P^* , which is of constant size, so we can detect an incidence between some point of P^* and a surface of Σ in $O(M)$ time. We stop the recursion when the size of one of the sets becomes smaller than some constant threshold n_0 , and then solve the problem using brute force, in time linear in the size of the other set. Note that at each recursive step we can, in addition, keep track of the signs of the values of $F(p, q)$ for the points p in P_{ij} and the points q defining surfaces in Σ that do not intersect S_{ij} , in order to construct the desired encoding of the signs.

Let $T(N, M)$ denote the maximum running time of the procedure for sets P, Σ of respective sizes at most N, M .

▷ **Claim 8.** For any fixed positive integer constants s and t , and real $\varepsilon > 0$, there is a constant A such that for any set P of at most N points in \mathbb{R}^t and any set Σ of at most M algebraic surfaces in \mathbb{R}^t with s degrees of freedom and constant maximum degree, we have

$$T(N, M) \leq A \left(M^{1-\frac{t-1}{ts-1}+\varepsilon} N^{1-\frac{s-1}{ts-1}+\varepsilon} + M^{1+\varepsilon} + N^{1+\varepsilon} \right). \tag{1}$$

Proof. The proof is by induction on N and M . The base case is $N \leq n_0$ or $M \leq n_0$. In either case we have $T(N, M) = O((N + M)n_0) = O(N + M)$, which is subsumed by the right-hand side of (1) if we choose A to be sufficiently large. Consider then the case where, say, $N \geq M > n_0$, and assume that (1) holds for all smaller values $N' \leq N, M' < M$ or $N' < N, M' \leq M$. Apply Lemma 7 to the primal setup, where $P \subset \mathbb{R}^t$ is the set of points; we would apply it in the dual setup, in which $Q \subset \mathbb{R}^s$ is the set of points that represent the surfaces of Σ in the complementary case where $M \geq N > n_0$. We use the notations in that lemma. Since r is a constant, the nonrecursive cost of the procedure is at most $B(N + M)$, where B is a constant that depends on r and on the various other constant parameters (this

22:8 Improved Algebraic Degeneracy Testing

bound also accounts for the cost of processing of P^*). By induction hypothesis, for each i and j , the cost of the recursive processing of P_{ij} and Σ_{ij} is at most

$$A \left(p_{ij}^{1 - \frac{s-1}{ts-1} + \varepsilon} q_{ij}^{1 - \frac{t-1}{ts-1} + \varepsilon} + p_{ij}^{1+\varepsilon} + q_{ij}^{1+\varepsilon} \right), \quad (2)$$

where $p_{ij} = |P_{ij}|$. Observe that we have $p_{ij} \leq N/r_i$ for each j , and the quantities $N_i := \sum_{j=1}^{\ell_i} p_{ij}$ satisfy $\sum_{i=1}^t N_i \leq N$ (since the decomposition in Lemma 7 is into disjoint subsets). Recall also that $\sum_{j=1}^{\ell_i} q_{ij} \leq cMr_i^{1-1/t}$ for each i .

We now sum the bounds in (2) over j for each fixed i . We first note that

$$\sum_{j=1}^{\ell_i} p_{ij}^{1+\varepsilon} \leq (N/r_i)^\varepsilon \sum_{j=1}^{\ell_i} p_{ij} = (N/r_i)^\varepsilon N_i.$$

Using Hölder's inequality, the sum is upper bounded by

$$\begin{aligned} & A \left(\sum_{j=1}^{\ell_i} p_{ij}^{1 - \frac{s-1}{ts-1} + \varepsilon} q_{ij}^{1 - \frac{t-1}{ts-1} + \varepsilon} + \sum_{j=1}^{\ell_i} p_{ij}^{1+\varepsilon} + \sum_{j=1}^{\ell_i} q_{ij}^{1+\varepsilon} \right) \\ & \leq A \left((N/r_i)^{1 - \frac{t+s-2}{ts-1} + 2\varepsilon} \sum_{j=1}^{\ell_i} p_{ij}^{\frac{t-1}{ts-1} - \varepsilon} q_{ij}^{1 - \frac{t-1}{ts-1} + \varepsilon} + (N/r_i)^\varepsilon N_i + (cMr_i^{1-1/t})^{1+\varepsilon} \right) \\ & \leq A \left((N/r_i)^{1 - \frac{t+s-2}{ts-1} + 2\varepsilon} N_i^{\frac{t-1}{ts-1} - \varepsilon} (cMr_i^{1-1/t})^{1 - \frac{t-1}{ts-1} + \varepsilon} + (N/r_i)^\varepsilon N_i + (cMr_i^{1-1/t})^{1+\varepsilon} \right) \\ & = A \left(N^{1 - \frac{t+s-2}{ts-1} + 2\varepsilon} N_i^{\frac{t-1}{ts-1} - \varepsilon} M^{1 - \frac{t-1}{ts-1} + \varepsilon} \cdot \frac{c^{1 - \frac{t-1}{ts-1} + \varepsilon}}{r_i^{(1 + \frac{1}{t})\varepsilon}} + (N/r_i)^\varepsilon N_i + (cr_i^{1-1/t})^{1+\varepsilon} M^{1+\varepsilon} \right) \\ & \leq A \left(N^{1 - \frac{s-1}{ts-1} + \varepsilon} M^{1 - \frac{t-1}{ts-1} + \varepsilon} \cdot \frac{c^{1 - \frac{t-1}{ts-1} + \varepsilon}}{r_i^{(1 + \frac{1}{t})\varepsilon}} + (N/r_i)^\varepsilon N_i + (cr_i^{1-1/t})^{1+\varepsilon} M^{1+\varepsilon} \right), \end{aligned}$$

where in the last inequality we used the fact that $N_i \leq N$ for each i .

We then sum these bounds over $i = 1, \dots, t$, add the nonrecursive cost $B(M + N)$, and obtain the overall upper bound (recalling that $r_i \geq r$ for each i)

$$A \left(N^{1 - \frac{s-1}{ts-1} + \varepsilon} M^{1 - \frac{t-1}{ts-1} + \varepsilon} \cdot \frac{tc^{1 - \frac{t-1}{ts-1} + \varepsilon}}{r^{(1 + \frac{1}{t})\varepsilon}} + \frac{N^{1+\varepsilon}}{r^\varepsilon} + \sum_{i=1}^t (cr_i^{1-1/t})^{1+\varepsilon} M^{1+\varepsilon} \right) + B(M + N). \quad (3)$$

For r chosen sufficiently large, the factors $\frac{tc^{1 - \frac{t-1}{ts-1} + \varepsilon}}{r^{(1 + \frac{1}{t})\varepsilon}}$ and $\frac{1}{r^\varepsilon}$ are both smaller than $1/4$.

The only problematic factor is $Z := \sum_{i=1}^t (cr_i^{1-1/t})^{1+\varepsilon}$, which in general will be larger than 1. To address this issue, one can easily verify that, for $M \leq N$, we have

$$M^{1+\varepsilon} \leq \frac{1}{N^{\frac{(s-1)(t-1)}{st-1} + \varepsilon}} \cdot N^{1 - \frac{s-1}{ts-1} + \varepsilon} M^{1 - \frac{t-1}{ts-1} + \varepsilon}.$$

That is,

$$ZM^{1+\varepsilon} \leq \frac{Z}{N^{\frac{(s-1)(t-1)}{st-1} + \varepsilon}} \cdot N^{1 - \frac{s-1}{ts-1} + \varepsilon} M^{1 - \frac{t-1}{ts-1} + \varepsilon},$$

and we can make the factor $\frac{Z}{N^{\frac{(s-1)(t-1)}{st-1} + \varepsilon}}$ smaller than $1/4$, assuming that n_0 is sufficiently large, an assumption that, as already noted, affects the choice of A .

Substituting these bounds into (3), we obtain

$$T(N, M) \leq A \left(\frac{1}{2} N^{1 - \frac{s-1}{ts-1} + \epsilon} M^{1 - \frac{t-1}{ts-1} + \epsilon} + \frac{1}{4} N^{1+\epsilon} \right) + B(M + N),$$

which, by choosing A sufficiently large, is smaller than the right-hand side of (1). This establishes the induction step and thereby completes the proof of the lemma for the case $M \leq N$. The complementary case $M \geq N$ is treated in a fully symmetric manner. This completes the proof of Claim 8. \triangleleft

Switching back to the $O^*(\cdot)$ notation, we have established Theorem 6. \blacktriangleleft

3 An improved real RAM algorithm for algebraic degeneracy testing

We can now apply Theorem 6 to obtain an improved algebraic degeneracy testing algorithm. We first briefly summarize the best known algorithm for k -SUM.

3.1 The meet-in-the-middle algorithm for k -SUM

The meet-in-the-middle algorithm for k -SUM, assuming k even, proceeds by computing the $n^{k/2}$ sums of all $(k/2)$ -tuples of numbers in the first $k/2$ sets, as well as all $n^{k/2}$ sums of $(k/2)$ -tuples from the last $k/2$ sets. It then searches a pair of opposite sums that sum to 0, by sorting each of the collections of sums, in time $O(n^{k/2} \log n)$, and then by merging the sequence of the former sums with the negated sequence of the latter. When k is odd, we sort the collection of sums of the $(k-1)/2$ -tuples composed from numbers of the first $(k-1)/2$ sets, and of sums of the $(k-1)/2$ -tuples composed from the last $(k-1)/2$ sets, in time $O(n^{(k-1)/2} \log n)$. Then, for each number $x \in A_{(k+1)/2}$, we shift the negated sequence of the latter sums by $-x$, then merge it with the former sequence to detect a coincident pair of values, which implies a positive answer. This takes time $O(n^{(k+1)/2}) = O(n^{\lceil k/2 \rceil})$ overall. This algorithm, for odd values of k , is hinted at by Erickson [22], and described by Ailon and Chazelle [4].

3.2 An algebraic meet-in-the-middle algorithm

Our algorithm can be seen as an algebraic generalization of this elementary method. We note that in the very special case where k is even, say, and F has the form

$$G(F_1(x_1, \dots, x_{k/2}), F_2(x_{k/2+1}, \dots, x_k))$$

for suitable constant-degree polynomials F_1, F_2 and G , the meet-in-the-middle algorithm can be straightforwardly generalized to solve the k -POL problem within the same time bound. In general, however, F does not have this separation-of-variables property, and we have to resort to a more involved algorithm, with the running time asserted in Theorem 2.

Proof of Theorem 2. Let A_1, \dots, A_k and F be as defined in the formulation of the k -POL problem, and let t, s be integers satisfying $t + s = k$. Define $P = A_1 \times \dots \times A_t$ as a set of $N = n^t$ points in \mathbb{R}^t , and $Q = A_{t+1} \times \dots \times A_k$ as a set of $M = n^s$ points in \mathbb{R}^s . We now apply Theorem 6 with these values, and obtain a main term in the running time bound that is proportional to

$$n^{s - \frac{st-s}{st-1}} n^{t - \frac{st-t}{st-1}} = n^{k-2 + \frac{k-2}{st-1}},$$

22:10 Improved Algebraic Degeneracy Testing

up to some subpolynomial factors. It is easily checked that this term dominates the other terms n^s and n^t (for $1 \leq s, t \leq k-1$). The running time bound is therefore $O^*\left(n^{k-2+\frac{k-2}{st-1}}\right)$. To minimize this bound, s and $t = k-s$ should be as close to $k/2$ as possible. Thus for even values of k , we take $t = s = k/2$ and obtain the bound

$$O^*\left(n^{k-2+\frac{4}{k+2}}\right),$$

which is indeed an improvement over the simpler $O^*(n^{k-1})$ solution discussed earlier. For odd values of k , we take $t = (k-1)/2$ and $s = (k+1)/2$, and obtain the bound

$$O^*\left(n^{k-2+\frac{4k-8}{k^2-5}}\right),$$

again an improvement. This proves Theorem 2. ◀

4 Improved algorithms in the algebraic decision tree model

We present faster algorithms in the algebraic decision tree model for the special cases of 4-POL and 5-POL, and prove Theorems 3 and 4. In the algebraic decision tree model we only count the number of sign tests of constant-degree polynomials in the input data; all other operations are free of charge, but they are not allowed to explicitly access the real numbers in the input, and can only manipulate them via the results of the sign tests. We refer to [12, 41, 44] for seminal works on this model, and to [6, 10, 13, 33] for recent results under this model. Before giving a description of the algorithms, we briefly recall the point location data structure described in Aronov et al. [6], that will be used in both algorithms.

4.1 A simple point location data structure

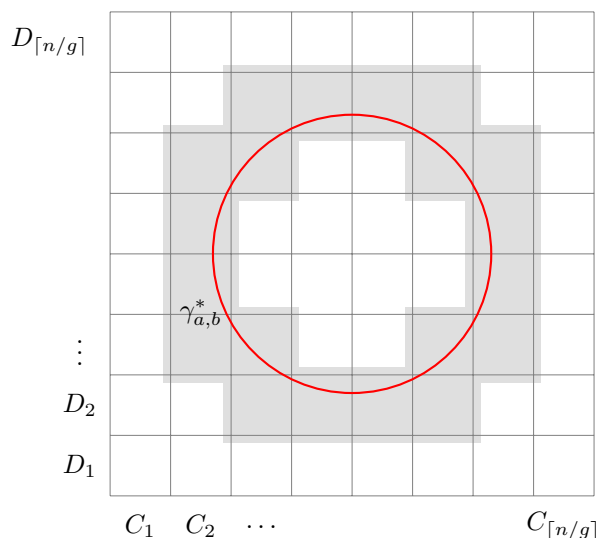
Let Γ be a finite collection of x -monotone constant-degree algebraic curves in the plane from $-\infty$ to $+\infty$, and let $\mathcal{A}(\Gamma)$ denote the arrangement of those curves. We define the *order type* of $\mathcal{A}(\Gamma)$ as the following information:

1. The vertical order of the curves at $x = -\infty$.
2. For each curve $\gamma \in \Gamma$, the left-to-right order of the intersection points of γ with the other curves of Γ , where each point is tagged by its *index*, which is the number of intersections of the same pair of curves that lie to its left.

We use the following lemma.

► **Lemma 9** (Aronov et al. [6]). *Let Γ be a finite collection of x -monotone constant-degree algebraic curves in the plane. Using only the order type of $\mathcal{A}(\Gamma)$, one can construct a data structure that allows to answer point location queries in $\mathcal{A}(\Gamma)$ in time $O(\log^2 |\Gamma|)$.*

The data structure itself is the one proposed by Lee and Preparata [34] (see also [20]). It stores the information related to the levels of the arrangement, each consisting of a sequence of vertices defined as intersections of pairs of curves. Each such intersection point q is encoded by a triple (i, j, k) , indicating that this is the k th intersection, in the left-to-right order, of the curves $\gamma_i, \gamma_j \in \Gamma$ (where $k-1$ is the index of q , as defined above). To answer a query, we perform a binary search on the levels of the arrangement. At each step of this binary search, we need to know whether the query point lies above or below (or on) some level. For this, we use a secondary binary search on the x -coordinates of the vertices of the level, which we can obtain as a left-to-right sorted sequence from the discrete information in the order type.



■ **Figure 2** Illustration of the algorithm for 4-POL. For each pair of values $(a, b) \in A \times B$, we test whether the dual curve $\gamma_{a,b}^*$ (here a circle of equation $w^2 + z^2 = 1$) is incident to any pair $(c, d) \in C \times D$. We only need to search the data structures for the pairs of blocks $C_i \times D_j$ whose boxes are intersected by $\gamma_{a,b}^*$, shown in gray. Note that the grid does not have to be uniform, but all boxes contain the same number g^2 of points of $C \times D$. The number of intersected boxes is $O(n/g)$.

The overall cost is therefore $O(\log^2 |\Gamma|)$. We refer to [6] for a more detailed description. We emphasize again that in the uniform model this structure requires quadratic storage and construction time, but they are cost-free in the algebraic decision tree model, once the order type has been produced.

The main point in using this simple method, even though its queries are less efficient than standard point location techniques [18, 42], is that in order to construct the data structure, we only need to know the order type of the arrangement, and that this order type can be encoded by a predicate that involves only *triples* of curves of Γ (or by pairs for the vertical order of the curves at $x = -\infty$). The small arity of this predicate is a crucial factor in the improvement of the running time in the algebraic decision tree model, compared to the uniform real RAM model.

The assumption that the curves are x -monotone and extend along the whole x -axis can be lifted, since every constant-degree algebraic curve in the plane can be decomposed into $O(1)$ x -monotone arcs, where the constant involved only depends on the degree of the curve. This extension of the algorithm to the case of bounded arcs requires some care but is not difficult. Using a segment tree over the x -projections of the arcs, we only need to pay an extra logarithmic factor in the point location mechanism, hence point location queries are answered in time $O(\log^3 |\Gamma|)$.

4.2 Algorithm for 4-POL

We consider the special case $k = 4$ of the k -POL problem, with four input sets A, B, C, D of n real numbers each.

22:12 Improved Algebraic Degeneracy Testing

Proof of Theorem 3. Recall that we need to locate the points of $P = A \times B$ in the arrangement of the curves in $\Gamma = \{\gamma_{c,d} \mid (c,d) \in C \times D\}$, where $\gamma_{c,d} = \{(x,y) \in \mathbb{R}^2 \mid F(x,y,c,d) = 0\}$. We can safely assume these are indeed one-dimensional curves and not the entire plane.

We want to preprocess the arrangement $\mathcal{A}(\Gamma)$ into a point location data structure. However, instead of computing a single data structure for the whole set, we partition each of C and D into n/g blocks, each consisting of g consecutive points (maybe less for the last block), for a suitable parameter $g \ll n$ that we will fix later. Denote the C -blocks as $C_1, \dots, C_{\lceil n/g \rceil}$ and the D -blocks as $D_1, \dots, D_{\lceil n/g \rceil}$. We construct the point location data structure described in Section 4.1 for each of the (n^2/g^2) cells of the form $C_i \times D_j$, namely for the curves $\gamma_{c,d}$ for $(c,d) \in C_i \times D_j$. We then search these structures with each of the $O(n^2)$ pairs in $A \times B$, to detect whether any such pair, regarded as a point in \mathbb{R}^2 , lies on any of the curves. For each pair $(a,b) \in A \times B$, define the dual curve $\gamma_{a,b}^* = \{(z,w) \in \mathbb{R}^2 \mid F(a,b,z,w) = 0\}$. Again we may assume that this is a one-dimensional curve and not the entire zw -plane. We observe that we need to search with a pair $(a,b) \in A \times B$ only in the point location data structures corresponding to pairs of blocks $C_i \times D_j$ for which $\gamma_{a,b}^*$ crosses the axis-parallel box in the wz -plane that defines $C_i \times D_j$. (See Figure 2 for an illustration.) Observe that there are only $O(n/g)$ such pairs, and the cost of one query in one of the blocks of size g^2 is $O(\log^3 g)$, so the total cost of the n^2 searches is

$$O^* \left(\frac{n^3}{g} \right). \quad (4)$$

It remains to describe the preprocessing phase, in which the point location data structures are constructed. We proceed as in the earlier recent works [6, 10], using the so-called Fredman's trick [27, 28, 32].

▷ **Claim 10.** The point location data structures for all the cells $C_i \times D_j$ can be constructed in time $O^*(n^{3/2}g^3)$ in the algebraic decision tree model.

Proof of Claim 10. In order to construct the point location data structure for the cell $C_i \times D_j$, we need to determine the order type of the arrangement of curves represented by points in $C_i \times D_j$. To determine this order type, we define a Boolean predicate $H_{k,k'}(c_1, c_2, c_3; d_1, d_2, d_3)$ with the following arguments:

1. a triple of curves $(\gamma_{c_1,d_1}, \gamma_{c_2,d_2}, \gamma_{c_3,d_3})$, where $c_1, c_2, c_3 \in C_i$ and $d_1, d_2, d_3 \in D_j$,
2. two positive integers k, k' bounded by a constant depending on the degree of the curves.

$H_{k,k'}(c_1, c_2, c_3; d_1, d_2, d_3)$ determines the relative order along γ_{c_1,d_1} of its k th leftmost intersection with γ_{c_2,d_2} and its k' th leftmost intersection with γ_{c_3,d_3} . Specifically, it is true if and only if the first intersection point lies to the left of the second point. (This suffices if we assume that no pair of intersection points coincide. Otherwise we add a predicate that is true when the two intersection points coincide.) Note that in practice $H_{k,k'}$ involves a number of quantifiers proportional to k and k' , and eliminating these quantifiers is somewhat involved. Still, since the curves are of constant degree, all of this can be done in constant time using standard algebraic geometry techniques [11, 17].

To efficiently resolve all such comparisons, we split the predicate $H_{k,k'}(c_1, c_2, c_3; d_1, d_2, d_3)$ by considering (c_1, c_2, c_3) as a point in \mathbb{R}^3 and (d_1, d_2, d_3) as defining a constant-complexity semi-algebraic range

$$\sigma_{(d_1,d_2,d_3)} := \{(x_1, x_2, x_3) \mid H_{k,k'}(x_1, x_2, x_3; d_1, d_2, d_3) \text{ is true}\}.$$

For each of the $O(1)$ pairs k, k' , the number of points (c_1, c_2, c_3) involved is equal to the number of blocks in C multiplied by the number of ordered triples in each block, namely $(n/g) \cdot g^3 = ng^2$. Similarly, there are ng^2 ranges of the form $\sigma_{(d_1, d_2, d_3)}$. We now have a semi-algebraic batch range searching problem in \mathbb{R}^3 , which has a symmetric dual version, in which the c -coordinates define ranges and the d -coordinates define points, also involving ng^2 points and ng^2 ranges. We can thus apply Theorem 6 with $N = M = ng^2$ and $t = s = 3$, since both points and curves have three degrees of freedom, and conclude that this problem can be solved in time

$$O^* \left((ng^2)^{3/2} \right) = O^* (n^{3/2} g^3).$$

This gives us a compact encoding of all the values $H_{k,k'}(c_1, c_2, c_3; d_1, d_2, d_3)$, hence the outcome of all the necessary comparisons to construct the point location structures, one for each pair C_i, D_j . The rest of the construction is free in the algebraic decision tree model. This proves Claim 10. \triangleleft

It remains to (roughly) balance the cost in Claim 10 with that of the search phase given in (4); that is, ignoring subpolynomial factors, we set

$$\frac{n^3}{g} = n^{3/2} g^3 \quad \Rightarrow \quad g = n^{3/8}.$$

With this choice of g , the overall cost is $O^*(n^{21/8}) = O^*(n^{2.625})$, a polynomial improvement over the $O^*(n^{2.667})$ uniform algorithm. This proves Theorem 3. \blacktriangleleft

4.3 Algorithm for 5-POL

We now consider the case $k = 5$, with five input sets A, B, C, D, E of n real numbers each.

Proof of Theorem 4. We will locate the points of $P = A \times B$ in the arrangement of curves in $\Gamma = \{\gamma_{c,d,e} \mid (c, d, e) \in C \times D \times E\}$, where $\gamma_{c,d,e} = \{(x, y) \in \mathbb{R}^2 \mid F(x, y, c, d, e) = 0\}$. We first partition each of the three sets C, D, E into blocks of g consecutive values. We refer to the i th block of C, D, E as C_i, D_i, E_i , respectively, where $i \in \{1, \dots, \lceil n/g \rceil\}$.

Following the previous approach, we map each $(a, b) \in A \times B$ to the 2-surface $\sigma_{a,b}^* = \{(z, w, u) \in \mathbb{R}^3 \mid F(a, b, z, w, u) = 0\}$. One can show that $\sigma_{a,b}^*$ crosses only $O((n/g)^2)$ cells of the form $C_i \times D_j \times E_\ell$. (This property, and the corresponding property in the 4-POL algorithm, can be regarded as extensions of the Schwartz–Zippel lemma; see [43, 46].) For each of the cells, we compute the point location data structure of Lemma 9, such that detecting an incidence can be performed in $O(\log^3 g)$ time. Hence, the time spent on the query phase is

$$O^* \left(n^2 \cdot (n/g)^2 \right) = O^* \left(\frac{n^4}{g^2} \right). \tag{5}$$

As for the preprocessing phase, we need to construct the point location data structure of Lemma 9 for each cell of the form $C_i \times D_j \times E_\ell$.

\triangleright **Claim 11.** The point location data structures for all the cells $C_i \times D_j \times E_\ell$ can be constructed in time $O^*(n^{42/17} g^{84/17})$ in the algebraic decision tree model.

Proof of Claim 11. We need to infer the order type of the arrangements in each of these cells. The order type can be inferred from the relative order of all pairs of intersections along a curve $\gamma_{c,d,e}$. This involves three curves $\gamma_{c_1, d_1, e_1}, \gamma_{c_2, d_2, e_2}, \gamma_{c_3, d_3, e_3}$, where $c_1, c_2, c_3 \in C_i$,

$d_1, d_2, d_3 \in D_j$, and $e_1, e_2, e_3 \in E_\ell$ and amounts to determining the signs of a constant number of 9-variate real polynomials of the form $H_{k,k'}(c_1, c_2, c_3; d_1, d_2, d_3; e_1, e_2, e_3)$, defined in complete analogy to the predicates in Claim 10. This can be solved using again batch semi-algebraic range searching. We have $(n/g) \cdot g^3 = ng^2$ triples of the form $(c_1, c_2, c_3) \in \mathbb{R}^3$ and $(n/g)^2 \cdot g^6 = n^2g^4$ 6-tuples of the form $(d_1, d_2, d_3; e_1, e_2, e_3) \in \mathbb{R}^6$. We can therefore apply Theorem 6 with $N = ng^2$, $M = n^2g^4$, $t = 3$ and $s = 6$, and obtain the claimed running time of $O^*((n^2g^4)^{15/17}(ng^2)^{12/17}) = O^*(n^{42/17}g^{84/17})$. \triangleleft

Balancing (roughly) the preprocessing cost of Claim 11 with the query cost in (5), we obtain

$$\frac{n^4}{g^2} = n^{42/17}g^{84/17} \quad \Rightarrow \quad g = n^{13/59},$$

yielding an overall complexity of $O^*(n^{4-26/59}) \simeq O^*(n^{3.56})$, an improvement over the uniform bound $O(n^{3.6})$ obtained in the previous section. This proves Theorem 4. \blacktriangleleft

Open problems

An interesting open question is to give lower bounds for the k -POL problem that are asymptotically larger than that for the k -SUM problem. Also, the techniques we use for algebraic decision trees do not seem to allow any speedup over the real RAM algorithm for $k \geq 6$. It would be interesting to design faster nonuniform algorithms for any value of k .

References

- 1 Pankaj K. Agarwal. Simplex range searching and its variants: A review. In *A Journey through Discrete Mathematics: A Tribute to Jiří Matoušek*, pages 1–30. Springer-Verlag, 2017. doi:10.1007/978-3-319-44479-6_1.
- 2 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, Matthew J. Katz, and Micha Sharir. Intersection queries for flat semi-algebraic objects in three dimensions and related problems. In *38th International Symposium on Computational Geometry, SoCG 2022*, pages 4:1–4:14, 2022. doi:10.4230/LIPIcs.SocG.2022.4.
- 3 Pankaj K. Agarwal, Boris Aronov, Esther Ezra, and Joshua Zahl. Efficient algorithm for generalized polynomial partitioning and its applications. *SIAM J. Comput.*, 50(2):760–787, 2021. Also in Proceedings of SoCG’20. doi:10.1137/19M1268550.
- 4 Nir Ailon and Bernard Chazelle. Lower bounds for linear degeneracy testing. *J. ACM*, 52(2):157–171, 2005. doi:10.1145/1059513.1059515.
- 5 Boris Aronov and Jean Cardinal. Geometric pattern matching reduces to k -SUM. *Discrete Comput. Geom.*, 68(3):850–859, 2022. Also in Proceedings of ISAAC’20. doi:10.1007/s00454-021-00324-1.
- 6 Boris Aronov, Mark de Berg, Jean Cardinal, Esther Ezra, John Iacono, and Micha Sharir. Subquadratic algorithms for some 3SUM-hard geometric problems in the algebraic decision-tree model. *Comput. Geom.*, 109:101945, 2023. Also in Proceedings of ISAAC’21. doi:10.1016/j.comgeo.2022.101945.
- 7 Boris Aronov, Esther Ezra, and Micha Sharir. Testing polynomials for vanishing on cartesian products of planar point sets: Collinearity testing and related problems. *Discrete Comput. Geom.*, 68(4):997–1048, 2022. Also in Proceedings of SoCG’20. doi:10.1007/s00454-022-00437-1.
- 8 Friedhelm Meyer auf der Heide. A polynomial linear search algorithm for the n -dimensional knapsack problem. *J. ACM*, 31(3):668–676, 1984. doi:10.1145/828.322450.


- 9 Ilya Baran, Erik D. Demaine, and Mihai Pătraşcu. Subquadratic algorithms for 3SUM. *Algorithmica*, 50(4):584–596, 2008. doi:10.1007/s00453-007-9036-3.
- 10 Luis Barba, Jean Cardinal, John Iacono, Stefan Langerman, Aurélien Ooms, and Noam Solomon. Subquadratic algorithms for algebraic 3SUM. *Discrete Comput. Geom.*, 61(4):698–734, 2019. Also in Proceedings of SoCG’17. doi:10.1007/s00454-018-0040-y.
- 11 Saugata Basu, Richard Pollack, and Marie-Françoise Roy. *Algorithms in Real Algebraic Geometry*. Algorithms and Computation in Mathematics. Springer-Verlag, Berlin, Heidelberg, 2006. doi:10.1007/3-540-33099-2.
- 12 Michael Ben-Or. Lower bounds for algebraic computation trees (preliminary report). In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing, STOC 1983*, pages 80–86. ACM, 1983. doi:10.1145/800061.808735.
- 13 Jean Cardinal, John Iacono, and Aurélien Ooms. Solving k -SUM using few linear queries. In *24th Annual European Symposium on Algorithms, ESA 2016*, pages 25:1–25:17, 2016. doi:10.4230/LIPIcs.ESA.2016.25.
- 14 Timothy M. Chan. More logarithmic-factor speedups for 3SUM, (median, +)-convolution, and some geometric 3SUM-hard problems. *ACM Trans. Algorithms*, 16(1):7:1–7:23, 2020. doi:10.1145/3363541.
- 15 Timothy M. Chan and Qizheng He. Reducing 3SUM to convolution-3SUM. In *3rd Symposium on Simplicity in Algorithms, SOSA 2020*, pages 1–7. SIAM, 2020. doi:10.1137/1.9781611976014.1.
- 16 Timothy M. Chan and Da Wei Zheng. Hopcroft’s problem, log-star shaving, 2D fractional cascading, and decision trees. In *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms, SODA 2022*, pages 190–210. SIAM, 2022. doi:10.1137/1.9781611977073.10.
- 17 David Cox, John Little, and Donal O’Shea. *Ideals, Varieties, and Algorithms. An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer Verlag, Heidelberg, 2007. doi:10.1007/978-3-319-16721-3.
- 18 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational Geometry: Algorithms and Applications, 3rd Edition*. Springer, 2008. doi:10.1007/978-3-540-77974-2.
- 19 Bartłomiej Dudek, Pawel Gawrychowski, and Tatiana Starikovskaya. All non-trivial variants of 3-LDT are equivalent. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020*, pages 974–981. ACM, 2020. doi:10.1145/3357713.3384275.
- 20 Herbert Edelsbrunner, Leonidas J. Guibas, and Jorge Stolfi. Optimal point location in a monotone subdivision. *SIAM J. Comput.*, 15(2):317–340, 1986. doi:10.1137/0215023.
- 21 Herbert Edelsbrunner, Joseph O’Rourke, and Raimund Seidel. Constructing arrangements of lines and hyperplanes with applications. *SIAM J. Comput.*, 15(2):341–363, 1986. doi:10.1137/0215024.
- 22 Jeff Erickson. Bounds for linear satisfiability problems. *Chic. J. Theor. Comput. Sci.*, 1999, 1999. URL: <http://cjtcs.cs.uchicago.edu/articles/1999/8/contents.html>.
- 23 Jeff Erickson and Raimund Seidel. Better lower bounds on detecting affine and spherical degeneracies. *Discrete Comput. Geom.*, 13:41–57, 1995. doi:10.1007/BF02574027.
- 24 Esther Ezra and Micha Sharir. A nearly quadratic bound for point-location in hyperplane arrangements, in the linear decision tree model. *Discrete Comput. Geom.*, 61(4):735–755, 2019. doi:10.1007/s00454-018-0043-8.
- 25 Esther Ezra and Micha Sharir. Intersection searching amid tetrahedra in 4-space and efficient continuous collision detection. In *30th Annual European Symposium on Algorithms, ESA 2022*, pages 51:1–51:17, 2022. doi:10.4230/LIPIcs.ESA.2022.51.
- 26 Jacob Fox, János Pach, Adam Sheffer, Andrew Suk, and Joshua Zahl. A semi-algebraic version of Zarankiewicz’s problem. *J. Eur. Math. Soc.*, 19(6):1785–1810, 2017. doi:10.4171/JEMS/705.
- 27 Michael L. Fredman. How good is the information theory bound in sorting? *Theor. Comput. Sci.*, 1(4):355–361, 1976. doi:10.1016/0304-3975(76)90078-5.

- 28 Michael L. Fredman. New bounds on the complexity of the shortest path problem. *SIAM J. Comput.*, 5(1):83–89, 1976. doi:10.1137/0205006.
- 29 Ari Freund. Improved subquadratic 3SUM. *Algorithmica*, 77(2):440–458, 2017. doi:10.1007/s00453-015-0079-6.
- 30 Anka Gajentaan and Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom.*, 5:165–185, 1995. doi:10.1016/0925-7721(95)00022-2.
- 31 Omer Gold and Micha Sharir. Improved bounds for 3SUM, k -SUM, and linear degeneracy. In *25th Annual European Symposium on Algorithms, ESA 2017*, pages 42:1–42:13, 2017. doi:10.4230/LIPIcs.ESA.2017.42.
- 32 Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles. *J. ACM*, 65(4):22:1–22:25, 2018. Also in Proceedings of FOCS’14. doi:10.1145/3185378.
- 33 Daniel M. Kane, Shachar Lovett, and Shay Moran. Near-optimal linear decision trees for k -SUM and related problems. *J. ACM*, 66(3):16:1–16:18, 2019. Also in Proceedings of STOC’18. doi:10.1145/3285953.
- 34 D. T. Lee and Franco P. Preparata. Location of a point in a planar subdivision and its applications. *SIAM J. Comput.*, 6(3):594–606, 1977. doi:10.1137/0206043.
- 35 Andrea Lincoln, Virginia Vassilevska Williams, Joshua R. Wang, and R. Ryan Williams. Deterministic time-space trade-offs for k -SUM. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016*, pages 58:1–58:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.58.
- 36 Jiří Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10:157–182, 1993. doi:10.1007/BF02573972.
- 37 Jiří Matoušek and Zuzana Patáková. Multilevel polynomial partitions and simplified range searching. *Discrete Comput. Geom.*, 54(1):22–41, 2015. doi:10.1007/s00454-015-9701-2.
- 38 Stefan Meiser. Point location in arrangements of hyperplanes. *Inf. Comput.*, 106(2):286–303, 1993. doi:10.1006/inco.1993.1057.
- 39 Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010*, pages 1065–1075. SIAM, 2010. doi:10.1137/1.9781611973075.86.
- 40 Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction*. Texts and Monographs in Computer Science. Springer, 1985. doi:10.1007/978-1-4612-1098-6.
- 41 Edward M. Reingold. On the optimality of some set algorithms. *J. ACM*, 19(4):649–659, 1972. doi:10.1145/321724.321730.
- 42 Neil Sarnak and Robert Endre Tarjan. Planar point location using persistent search trees. *Commun. ACM*, 29(7):669–679, 1986. doi:10.1145/6138.6151.
- 43 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.
- 44 J. Michael Steele and Andrew Chi-Chih Yao. Lower bounds for algebraic decision trees. *J. Algorithms*, 3(1):1–8, 1982. doi:10.1016/0196-6774(82)90002-5.
- 45 Virginia Vassilevska Williams. On some fine-grained complexity questions in algorithms and complexity. *Proceedings of the International Congress of Mathematicians, ICM 2018*, pages 3447–3487, 2018. doi:10.1142/9789813272880_0188.
- 46 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM ’79*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.

Constant-Hop Spanners for More Geometric Intersection Graphs, with Even Smaller Size

Timothy M. Chan  

Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA

Zhengcheng Huang 

Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA

Abstract

In SoCG 2022, Conroy and Tóth presented several constructions of sparse, low-hop spanners in geometric intersection graphs, including an $O(n \log n)$ -size 3-hop spanner for n disks (or fat convex objects) in the plane, and an $O(n \log^2 n)$ -size 3-hop spanner for n axis-aligned rectangles in the plane. Their work left open two major questions: (i) can the size be made closer to linear by allowing larger constant stretch? and (ii) can near-linear size be achieved for more general classes of intersection graphs?

We address both questions simultaneously, by presenting new constructions of constant-hop spanners that have *almost* linear size and that hold for a *much larger* class of intersection graphs. More precisely, we prove the existence of an $O(1)$ -hop spanner for arbitrary *string graphs* with $O(n\alpha_k(n))$ size for any constant k , where $\alpha_k(n)$ denotes the k -th function in the inverse Ackermann hierarchy. We similarly prove the existence of an $O(1)$ -hop spanner for intersection graphs of d -dimensional fat objects with $O(n\alpha_k(n))$ size for any constant k and d .

We also improve on some of Conroy and Tóth's specific previous results, in either the number of hops or the size: we describe an $O(n \log n)$ -size 2-hop spanner for disks (or more generally objects with linear union complexity) in the plane, and an $O(n \log n)$ -size 3-hop spanner for axis-aligned rectangles in the plane.

Our proofs are all simple, using separator theorems, recursion, shifted quadtrees, and shallow cuttings.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Hop spanners, geometric intersection graphs, string graphs, fat objects, separators, shallow cuttings

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.23

Related Version *Full Version*: <https://arxiv.org/abs/2303.16303>

Funding *Timothy M. Chan*: Work supported by NSF Grant CCF-2224271.

1 Introduction

Spanners – subgraphs of a given graph that preserve distances up to some multiplicative factor – have numerous applications and have been studied extensively in both the graph algorithms and the computational geometry literature [4, 37]. Traditionally, in computational geometry, the focus has been on Euclidean spanners or metric spanners (i.e., spanners for a weighted complete graph defined by n points, where the edge weights are Euclidean distances or distances under some metric).

Recently, spanners for geometric intersection graphs have gained more attention. A *geometric intersection graph* is an unweighted, undirected graph formed by n geometric objects, where the vertices are the objects, and we place an edge between two objects iff they intersect. Such graphs are popularly studied in computational geometry (e.g., see [11, 12, 21, 22, 31]), and arise naturally in applications to wireless communication.



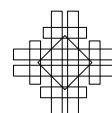
© Timothy M. Chan and Zhengcheng Huang;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 23; pp. 23:1–23:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Formally, in the unweighted setting, a t -hop spanner of a graph G is a subgraph \widehat{G} of G , such that for each edge $uv \in E(G)$, there is a path of at most t edges in \widehat{G} from u to v . (It is sometimes just called a t -spanner, but the term “hop” emphasizes that we are considering unweighted graph distances here.) The parameter t is called the *hop stretch*. For an arbitrary unweighted graph with n vertices, it is known [5] that there exists a t -hop spanner with $O(n^{1+1/\lceil t/2 \rceil})$ size (i.e., number of edges) for any constant integer $t \geq 3$; this bound is tight assuming the Erdős girth conjecture [27]. Our goal is to obtain better bounds in the setting of geometric intersection graphs.

Previous results. Several papers studied hop spanners in the case of *unit-disk graphs*, i.e., intersection graphs of unit disks in the plane: Yan et al. [38] obtained 15-hop spanners with $O(n \log n)$ size. Catusse et al. [13] obtained 5-hop spanners with $O(n)$ size (with improvements on the hidden constant factor in the size bound by Biniarz [10] and Dumitrescu et al. [26]). Dumitrescu et al. [26] also obtained 3-hop spanners with $O(n)$ size and 2-hop spanners with $O(n \log n)$ size. Finally, Conroy and Toth [24] obtained 2-hop spanners with $O(n)$ size.

Conroy and Toth [24] also initiated the study of hop spanners for other families of geometric intersection graphs. They obtained:

- 2-hop spanners for fat rectangles¹ (e.g., squares) in the plane with $O(n \log n)$ size. (In fact, they proved a nearly matching lower bound of $\Omega(n \log n / \log \log n)$ for squares, or for homothets of any fixed convex object in the plane.)
- 3-hop spanners for fat convex objects (e.g., disks) in the plane with $O(n \log n)$ size.
- 3-hop spanners for arbitrary rectangles in the plane with $O(n \log^2 n)$ size.

Main questions. Conroy and Tóth’s work represented significant progress on hop spanners in geometric intersection graphs, but it also raised a number of intriguing questions:

1. Can the size of hop spanners be made closer to linear for the classes of graphs they considered? Their bounds for arbitrary disks, rectangles, etc. all have extra logarithmic factors. At the end of their paper, Conroy and Tóth explicitly asked: “is there a constant $t \in \mathbb{N}$ for which every intersection graph of n disks or rectangles admits a t -hop spanner with $O(n)$ edges?”
2. Ignoring logarithmic factors, can near-linear size hop spanners be obtained for larger classes of geometric intersection graphs than the ones they considered? In particular, no $O(n \text{ polylog } n)$ size bounds were known for arbitrary line segments or arbitrary triangles in the plane, or arbitrary balls in \mathbb{R}^d for $d \geq 3$. At the end of their paper, Conroy and Tóth wrote: “it would be interesting to see other classes of intersection graphs (e.g., for strings or convex sets in \mathbb{R}^2 , set systems with bounded VC-dimension or semi-algebraic sets in \mathbb{R}^d) for which the general bound of $O(n^{1+1/\lceil t/2 \rceil})$ edges for t -hop spanners can be improved”.

To appreciate the difficulty of these questions, it is worth mentioning the connection to biclique cover size. A *biclique cover* of a graph G refers to a collection of bicliques $A_1 \times B_1, \dots, A_s \times B_s$, such that $E(G) = \bigcup_{i=1}^s (A_i \times B_i)$. The *size* of the cover refers to $M = \sum_{i=1}^s (|A_i| + |B_i|)$. Biclique covers are a standard technique closely related to range searching, and have many applications in computational geometry (e.g., see [2, 17, 19]). Most classes of geometric intersection graphs admit biclique covers with subquadratic size; in

¹ All rectangles, squares, and hypercubes are axis-aligned throughout this paper.

fact, for rectangles or axis-aligned boxes, there are standard constructions of biclique covers with $O(n \text{polylog } n)$ size (similar to the construction of range trees [3, 25]). Given a biclique cover of size M , it is easy to build a 3-hop spanner of size $O(M)$, as noted by Conroy and Tóth [24], by just keeping two stars per biclique. In particular, Conroy and Tóth's 3-hop, $O(n \log^2 n)$ -size spanners for rectangles were obtained essentially by using range-tree-style divide-and-conquer.

However, biclique cover constructions typically require multiple logarithmic factor. This makes the first question challenging for rectangles. For non-axis-aligned objects, the biclique cover size is even larger; for example, for line segments, the known upper bound is near $n^{4/3}$ (e.g., see [19]). Thus, this would not yield better bounds than for general graphs for hop stretch $t \geq 5$. The situation is even worse for *string graphs*, i.e., intersection graphs of curves (which could have large description complexity) in the plane. New ideas are needed to address the second question.

Main new results. We make progress towards both of the above questions at once, by obtaining the following result:

- $O_k(1)$ -hop spanners with $O(n\alpha_k(n))$ size for arbitrary string graphs.

Here, subscripts in the O notation indicate variables that are assumed to be constant; the hidden constant factor may depend on such variables. The function $\alpha_k(\cdot)$ denotes the k -th function in the inverse Ackermann hierarchy: $\alpha_0(n) = n/2$, $\alpha_1(n) = \log n$, $\alpha_2(n) = \log^* n$ (the iterated logarithm), $\alpha_3(n) = \log^{**} n$ (the iterated iterated logarithm), etc. Since these functions are extremely slow-growing as k increases, we thus get constant-hop spanners with *almost* linear size. Although inverse Ackermann has arisen in some past work on Euclidean spanners before (namely, on the trade-off between size and hop-diameter [7, 33]), its appearance here for hop spanners in geometric intersection graphs is still surprising.

String graphs include intersection graphs of arbitrary regions enclosed by closed curves in the plane (e.g., see [34, Lemma 4]). Thus, our result is very general, encompassing arbitrary line segments and triangles in \mathbb{R}^2 for the first time, and also including all the previous types of geometric objects considered by Conroy and Tóth, such as disks, rectangles, and fat convex objects in \mathbb{R}^2 – our result shows that fatness is not needed in \mathbb{R}^2 .

We obtain a similar result also for *higher*-dimensional fat objects:

- $O_k(1)$ -hop spanners with $O_{d,k}(n\alpha_k(n))$ size for intersection graphs of fat objects in \mathbb{R}^d .

In particular, this includes the case of arbitrary balls in \mathbb{R}^d , for which there were no prior results for $d \geq 3$. (In $d = 2$ dimensions, compared to Conroy and Tóth's previous result on fat objects, we work with a different definition of fatness that does not require convexity.)

More new results. The above new results improve previous size bounds for sufficiently large hop stretch, but do not necessarily improve Conroy and Tóth's 2-hop and 3-hop spanners. We have additional new constructions that directly improve some of their specific results. Notably, we obtain:

- 2-hop spanners with $O(n \log n)$ size for objects with linear union complexity in the plane.

Classes of objects with linear union complexity include arbitrary disks, pseudodisks, and fat rectangles in \mathbb{R}^2 . Thus, our result strictly improves the hop stretch in Conroy and Tóth's 3-hop, $O(n \log n)$ -size spanners for the case of disks, and also generalizes their 2-hop,

$O(n \log n)$ -size spanners for the case of fat rectangles. So, our result significantly enlarges the class of geometric intersection graphs that admit sparse 2-hop spanners.

In addition, we obtain:

- 3-hop spanners with $O(n \log n)$ size for rectangles in the plane.

This is a logarithmic-factor improvement over Conroy and Tóth's previous result.

A summary of our results can be found in Table 1.

Techniques. Our proofs use interesting techniques. For string graphs, our approach (see Section 2) is based on divide-and-conquer via *graph separators*. Separator theorems for string graphs have been developed in a series of papers [28, 36, 34], but sublinear-size separators exist only if the string graph is not too dense. On the other hand, if the graph is dense, there exist high-degree vertices, whose neighborhoods form large stars. The key is to realize that each such star can be viewed as a single object, since a connected union of strings can be regarded as a new string. Besides recursion in the parts produced by the separator, we use an extra recursive call to handle these stars. (Luckily, the separator bound for string graphs is not influenced by how complicated the strings are.) This double recursion eventually leads to inverse Ackermann complexity. The overall construction is simple.

Curiously, even if we are only interested in the very special case of vertical/horizontal line segments (or rectangles), it is still important to generalize to strings with the above approach. (In fact, we started this research more modestly with the case of vertical/horizontal segments, using more traditional divide-and-conquer, but the above string-graph separator approach wins out at the end.)

For fat objects in \mathbb{R}^d , the approach is similar, except that we use *shifted quadtrees* [9, 14, 16] and tree partitioning [30] to do divide-and-conquer (see Section 3). The key is to view a union of fat objects containing a common point as a new fat object. Again, we get a double recursion leading to inverse Ackermann.

For our 2-hop spanners for objects with linear union complexity in \mathbb{R}^2 , we use logarithmically many layers of *shallow cuttings* (see Section 4). Shallow cuttings [35] have many applications, for example, to static data structures for halfspace range searching [15] and orthogonal range searching [1], dynamic geometric data structures [18], levels in arrangements [15], incidences [20], epsilon-nets [35], and geometric set cover [23]. Interestingly, our work adds one more (unexpected) application to the list. Given the shallow cutting lemma, our proof is simple, this time, not even needing recursion; in fact, it is simpler than Conroy and Tóth's previous proofs for their 2-hop spanners for fat rectangles, as well as their 3-hop spanners for fat convex objects in \mathbb{R}^2 .

Our 3-hop spanners for rectangles in \mathbb{R}^2 (see Section 5) is perhaps the least exciting. It is similar to Conroy and Tóth's previous proof, using straightforward range-tree-style divide-and-conquer, but exploiting known spanners in one dimension (namely, points and intervals on the real line) as a base case.

2 String Graphs

Our spanner constructions for string graphs will use a separator theorem by Lee [34] (which was an improvement over previous versions by Fox and Pach [28] and Matoušek [35]).

► **Lemma 1** (String-graph separator [34]). *For every string graph G with n vertices and m edges, there exists a partition of $V(G)$ into subsets V_1, V_2, X with $|V_1|, |V_2| \leq 2n/3$, $|X| = O(\sqrt{m})$, such that there are no edges between V_1 and V_2 .*

■ **Table 1** Our results on $O(1)$ -hop spanners for different classes of geometric intersection graphs.

	hop stretch	size
String graphs	3	$O(n \log^3 n)$
	$O_k(1)$	$O(n \alpha_k(n))$
Fat objects in \mathbb{R}^d	3	$O(n \log n)$
	$O_k(1)$	$O_{d,k}(n \alpha_k(n))$
Objects with union complexity $\mathcal{U}(\cdot)$	2	$O(\mathcal{U}(n) \log n)$
Rectangles in \mathbb{R}^2	3	$O(n \log n)$

We first warm up by describing a 3-hop spanner with $O(n \log^3 n)$ size, and a 7-hop spanner with $O(n \log \log n)$ size, before generalizing it to a larger-hop spanner with inverse Ackermann complexity.

2.1 3-Hop Spanner with $O(n \log^3 n)$ Size

String Graph Construction I

1. Repeatedly pick a vertex with degree larger than Δ , for a parameter Δ to be chosen later, and remove the vertex along with its neighborhood, until there are no vertices with at most degree Δ in the remaining graph. Let G' be the remaining graph. A vertex and its neighborhood forms a star, and at most n/Δ such stars are removed. Add the edges of these stars ($O(n)$ in total) to the output spanner \widehat{G} .
2. For each vertex $u \in V(G)$ and for each star removed in step 1 that contains a vertex adjacent to u , add an edge between u and an arbitrary such vertex in the star to \widehat{G} . At most $O(n \cdot n/\Delta)$ edges are added this way.
3. Apply Lemma 1 to G' to obtain V_1, V_2, X . Since G' has $O(\Delta n)$ edges, $|X| = O(\sqrt{\Delta n})$. Recursively construct a 3-hop spanner for the subgraph induced by $V_1 \cup X$ and for the subgraph induced by $V_2 \cup X$. Add all their edges to \widehat{G} .

Hop stretch. For any edge uv , if both u and v belong to G' , then u and v are connected by 3 hops by induction. Otherwise, one of its vertices, say, v , belongs to a star removed in step 1. By step 2, the spanner \widehat{G} connects u to some vertex v' in the same star as v . The star connects v' and v by 2 hops, so u and v are connected by 3 hops (see Figure 1).

Sparsity. The size $S(n)$ of the spanner follows the recurrence

$$S(n) \leq \max_{n_1, n_2 \leq 2n/3: n_1 + n_2 \leq n} (S(n_1 + O(\sqrt{\Delta n})) + S(n_2 + O(\sqrt{\Delta n})) + O(n \cdot n/\Delta)).$$

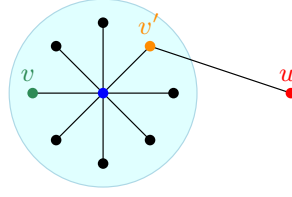
By setting $\Delta = n/\log^2 n$, the recurrence solves to $S(n) = O(n \log^3 n)$.

► **Theorem 2.** *Every string graph with n vertices admits a 3-hop spanner of size $O(n \log^3 n)$.*

2.2 7-Hop Spanner with $O(n \log \log n)$ Size

To obtain hop spanners with still smaller size, we need a generalized version of Theorem 1 that partitions into multiple subsets (analogous to “ r -divisions” in planar graphs [29]):

► **Lemma 3** (String-graph separator: multiple-subsets version). *Given parameters r and Δ with $\Delta = o(r)$, for every string graph G with n vertices and maximum degree Δ , there exist $O(n/r)$ subsets $\overline{V}_i \subset V(G)$ of at most r vertices each, such that $E(G) \subset \bigcup_i (\overline{V}_i \times \overline{V}_i)$, and the boundary complexity, defined as $\sum_i |\overline{V}_i| - |V(G)|$, is at most $O(\sqrt{\Delta n}/\sqrt{r})$.*



■ **Figure 1** The light blue circle represents the star containing v . The blue dot at the center represents the center of the star.

Proof. Apply Lemma 1 to obtain V_1, V_2, X , with $|X| = O(\sqrt{\Delta n})$, and recursively generate subsets for the subgraph induced by $V_1 \cup X$ and for the subgraph induced by $V_2 \cup X$. When a subgraph has fewer than r vertices, output its vertex set.

Let $B(n)$ count the boundary complexity of the subsets produced by the above division procedure on an n -vertex string graph. We have the recurrence

$$B(n) \leq \begin{cases} \max_{n_1, n_2 \leq 2n/3: n_1 + n_2 \leq n} (B(n_1 + O(\sqrt{\Delta n})) + B(n_2 + O(\sqrt{\Delta n})) + O(\sqrt{\Delta n})) & \text{if } n \geq r \\ 0 & \text{if } n < r \end{cases}$$

The recurrence solves to $B(n) = O(\sqrt{\Delta n}/\sqrt{r})$. ◀

String Graph Construction II

1. Follow step 1 of Construction I.
2. For each vertex $u \in V(G)$ that is adjacent to a vertex of at least one star, add an edge between u and an arbitrary such vertex in such a star to \widehat{G} . At most $O(n)$ edges are added this way.
3. For each pair of stars removed in step 1 such that there is a 2-hop path between them, add an arbitrary such 2-hop path to \widehat{G} . At most $O((n/\Delta)^2)$ edges are added this way.
4. Apply Lemma 3 to G' to obtain the subsets \overline{V}_i . Recursively construct a 7-hop spanner for the subgraph induced by each \overline{V}_i . Add all their edges to \widehat{G} .

Hop stretch. For any edge uv , if both u and v belong to G' , then u and v are connected by 7 hops by induction. Otherwise, one of its vertices, say, v belongs to a star removed in step 1. By step 2, the spanner \widehat{G} connects u to some vertex v' in a possibly different star. By step 3, these two stars are connected by 2 hops in \widehat{G} . Inside a star, 2 hops suffice. Thus, u and v are connected by 7 hops (see Figure 2).

Sparsity. The size of the spanner follows the recurrence

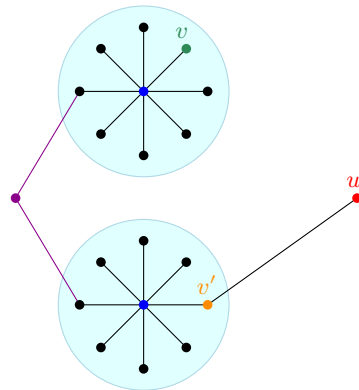
$$S(n) \leq \max_{n_1, n_2, \dots \leq r: \sum_i n_i \leq n + O(\sqrt{\Delta n}/\sqrt{r})} \left(\sum_i S(n_i) + O(n + (n/\Delta)^2) \right).$$

By choosing $\Delta = \sqrt{n}$ and $r = n^{0.9}$, the recurrence solves to $S(n) = O(n \log \log n)$.

► **Theorem 4.** *Every string graph with n vertices admits a 7-hop spanner of size $O(n \log \log n)$.*

2.3 $O_k(1)$ -Hop Spanner with $O(n\alpha_k(n))$ Size

Let $t_1 = 3$ and $t_k = 5t_{k-1} + 3$ for all $k > 1$. This implies $t_k = \frac{3}{4}(5^k - 1)$. We next modify the preceding construction to obtain a t_k -hop spanner. The key is the following observation:



■ **Figure 2** If $uv \in E(G)$, then \widehat{G} connects u and v by 7 hops. The purple paths represent the arbitrary 2-hop path added in step 3.

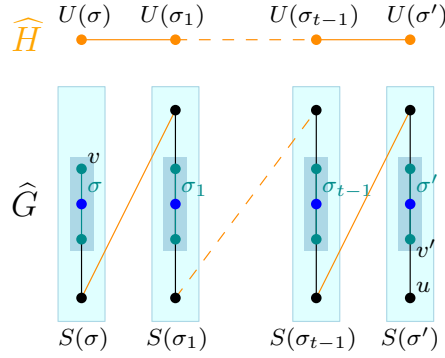
► **Observation 5.** *The union C of strings that form a connected string graph may be viewed as a string.*

Proof. We can first eliminate cycles from C by removing infinitesimally small arcs, and can then take an Euler traversal of the resulting tree to obtain a noncrossing path. (The observation becomes even more obvious if one defines a string as any connected set in the plane, as some authors did [28].) ◀

String Graph Construction III

1. Follow step 1 of Construction I.
2. Apply Lemma 3 to G' to obtain the subsets \overline{V}_i . Let B be the boundary vertices, i.e., vertices that are in at least two subsets \overline{V}_i . Recursively construct a t_k -hop spanner for the subgraph induced by $\overline{V}_i \setminus B$ for each \overline{V}_i . Add all their edges to \widehat{G} . Also, for each vertex in B , create a star of size 1 (i.e., a singleton) and remove it from G' . The number of stars is now $O(n/\Delta + \sqrt{\Delta n}/\sqrt{r})$.
3. For each vertex u that is adjacent to a vertex of at least one star, add an edge between u and an arbitrary such vertex in such a star to \widehat{G} ; we say that u is *assigned* to this star. At most $O(n)$ edges are added this way.
4. For each star σ , define its *extended star* $S(\sigma)$ to be the set of all vertices that are in σ or assigned to σ , and define the new object $U(\sigma)$ to be the union of all the strings in $S(\sigma)$. Recursively construct a t_{k-1} -hop spanner \widehat{H} for these new objects, which can be viewed as strings by Observation 5. For each edge $U(\sigma)U(\sigma')$ in the spanner \widehat{H} , add an edge ww' to \widehat{G} , where $w \in S(\sigma)$ and $w' \in S(\sigma')$ are intersecting strings chosen arbitrarily.

Hop stretch. For any edge uv , if both u and v belong to G' , then u and v are connected by t_k hops by induction. Otherwise, one of its vertices, say, v belongs to a star σ removed in step 1 or 2. By step 3, the spanner \widehat{G} connects u to some vertex v' in a possibly different star σ' . Since u is in $S(\sigma')$ and v is in σ , the two objects $U(\sigma)$ and $U(\sigma')$ intersect and, by induction, are connected by t_{k-1} hops in the spanner \widehat{H} . Inside an extended star, 4 hops suffice. Thus, u and v are connected by $5t_{k-1} + 3$ hops in \widehat{G} (see Figure 3).



■ **Figure 3** $v \in S(\sigma)$ and $v' \in S(\sigma')$. If \widehat{H} connects $U(\sigma)$ and $U(\sigma')$ by t hops, then \widehat{G} connects v and v' by $5t + 3$ hops. The smaller boxes represent stars in \widehat{G} ; the larger boxes represent extended stars. Blue dots at the center of the boxes represent centers of the stars.

Sparsity. The size of the t_k -hop spanner follows the recurrence

$$S_k(n) \leq \max_{n_1, n_2, \dots, \sum_i n_i \leq n} \left(\sum_i S_k(n_i) + S_{k-1}(O(n/\Delta + \sqrt{\Delta n}/\sqrt{r})) + O(n) \right).$$

For the base case, we have $S_1(n) = O(n \log^3 n)$ by Theorem 2. For $k = 2$, by choosing $\Delta = \log^3 n$ and $r = \Delta^3$, the recurrence gives $S_2(n) = O(n \log^* n)$. For $k > 2$, we choose $\Delta = c_0 \alpha_{k-1}(n)$ and $r = \Delta^3$ for a sufficiently large constant c_0 . It is straightforward to show by induction that $S_k(n) \leq c_0 n \alpha_k(n)$.

▶ **Theorem 6.** *Every string graph with n vertices admits a $\frac{3}{4}(5^k - 1)$ -hop spanner of size $O(n \alpha_k(n))$ for any $k \geq 2$.*

Remarks. We have not attempted to optimize the hop stretch in the above theorem. Since the constant factor in the above size bound does not depend on k , we can also choose $k = \alpha(n)$ and obtain an $O(5^{\alpha(n)})$ -hop spanner with $O(n)$ size.

Although we have cited Lee's string-graph separator theorem [34], the weaker separator bound by Fox and Pach [28] is actually sufficient to prove Theorems 4–6 (although for Theorem 2, the bound would have more logarithmic factors).

3 Fat Objects in \mathbb{R}^d

In this section, we turn our attention to the case of fat objects. We will use the following definition of fatness [16]. Here, the *side length* of an object refers to the side length of its smallest enclosing hypercube.

▶ **Definition 7.** *A collection of objects is c -fat if for every hypercube γ with side length ℓ , there exist c points hitting all objects that intersect γ and have side length at least ℓ .*

Our spanner construction will use quadtrees together with a known “shifting lemma” [16] (based on an earlier work [14]).

▶ **Definition 8.** *A quadtree cell is a hypercube of the form $[i_1/2^k, (i_1 + 1)/2^k) \times \dots \times [i_d/2^k, (i_d + 1)/2^k)$ for integers i_1, \dots, i_d, k .*

An object u of side length ℓ is C -aligned if it is contained in a quadtree cell with side length at most $C\ell$.

► **Lemma 9** (Quadtree shifting [16]). *Fix an odd number $d^* > d$. Let $\tau^{(j)} = (j/d^*, \dots, j/d^*) \in \mathbb{R}^d$. For any object $u \subset [0, 1]^d$, the shifted object $u + \tau^{(j)}$ is $(2d^*)$ -aligned for all but at most d indices $j \in \{0, \dots, d^* - 1\}$.*

Choose $d^* = 2d + 1$. By rescaling, we may assume that all input objects are in $[0, 1]^d$. Then for any pair of objects u and v , there exists at least one index $j \in \{0, \dots, d^* - 1\}$ such that $u + \tau^{(j)}$ and $v + \tau^{(j)}$ are both $(2d^*)$ -aligned. For each j , it suffices to construct a hop spanner for the subset of all objects u such that $u + \tau^{(j)}$ is $(2d^*)$ -aligned; we can then output the union of these d^* spanners.

Thus, from now on, we may assume that all given objects are $(2d^*)$ -aligned.

We warm up by describing a 3-hop spanner with $O(n \log n)$ size, before describing a larger-hop spanner with inverse Ackermann complexity.

3.1 3-Hop Spanner of $O(n \log n)$ Size

Our 3-hop spanner will use the following lemma, which follows directly by taking a tree centroid in the quadtree:

► **Lemma 10** (Quadtree centroid [8, 14]). *For any set of n points in \mathbb{R}^d , there exists a quadtree cell such that the number of points inside and the number of points outside are both at most $\frac{2^d}{2^d+1}n$.*

Fat Object Construction I

1. Apply Lemma 10 to the leftmost points of the objects to obtain a quadtree cell γ . Recursively construct a 3-hop spanner for the objects completely inside γ and for the objects completely outside γ . Add all their edges to \widehat{G} .
2. Let P_γ be a set of points hitting all objects that intersect $\partial\gamma$ and have side length at least $\ell_\gamma/(2d^*)$, where ℓ_γ denotes the side length of γ . A hitting set of size $|P_\gamma| = O_d(c)$ exists by definition of c -fatness (since $\partial\gamma$ can be covered by $(2d^*)^d$ hypercubes of side length $\ell_\gamma/(2d^*)$). For each point $p \in P_\gamma$, build a star $S(p)$ connecting all objects hit by p , with the center chosen arbitrarily. Add the edges of these stars to \widehat{G} . At most $O_d(cn)$ edges are added this way.
3. For each object u and for each star $S(p)$ that contains an object intersecting u , add an edge between u and an arbitrary such object in $S(p)$ to \widehat{G} . At most $O(n)$ edges are added this way.

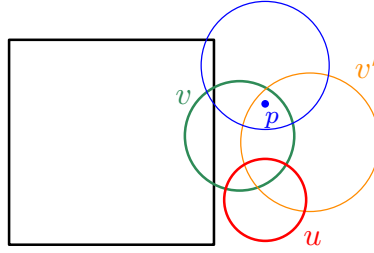
Hop stretch. For any edge uv , if the objects u and v are both inside γ or both outside γ , then u and v are connected by 3 hops by induction. Otherwise, one of the objects, say, v , intersects $\partial\gamma$ (see Figure 4). Observe that v has side length at least $\ell_\gamma/(2d^*)$, since v is $(2d^*)$ -aligned. Thus, v belongs to a star $S(p)$ from step 2. By step 3, the spanner \widehat{G} connects u to some object v' in the same star $S(p)$. Since v' and v are connected by 2 hops in \widehat{G} , u and v are connected by 3 hops.

Sparsity. The size $S(n)$ of the spanner follows the recurrence

$$S(n) \leq \max_{n_1, n_2 \leq 2^d n / (2^d + 1): n_1 + n_2 \leq n} (S(n_1) + S(n_2) + O_d(cn)).$$

The recurrence solves to $S(n) = O_d(cn \log n)$.

► **Theorem 11.** *The intersection graph of n fat objects in \mathbb{R}^d admits a 3-hop spanner of size $O(n \log n)$.*



■ **Figure 4** $uv' \in \widehat{G}$, and v' is connected to v via the center of star $S(p)$ (the blue disk).

3.2 $O_k(1)$ -Hop Spanner with $O_{d,k}(n\alpha_k(n))$ Size

To obtain hop spanners of still smaller size, we need a generalized version of Lemma 10 that partitions into multiple subsets:

► **Definition 12.** A generalized quadtree cell γ refers to either a quadtree cell or the difference of an outer quadtree cell γ^+ and an inner quadtree cell γ^- (in the former case, we let $\gamma^+ = \gamma$ and $\gamma^- = \emptyset$).

► **Lemma 13** (Quadtree partitioning). Given parameter r , for any set of n points in $[0, 2)^d$, there exists a partition of $[0, 2)^d$ into $O(n/r)$ generalized quadtree cells, each containing at most r points.

Proof. This follows directly by applying the tree partitioning scheme by Frederickson [30] to the quadtree, or alternatively by applying Lemma 10 recursively (stopping when cells have at most r points each, and with further splitting to ensure each generalized cell has at most one inner quadtree cell – e.g., see [8]). ◀

► **Observation 14.** Given a collection of c -fat C -aligned objects in \mathbb{R}^d , a union of a subset of objects all hit by a common point can be viewed as a $(4^d c)$ -fat C -aligned object.

Proof. Consider a hypercube γ with side length ℓ . Expand γ into a hypercube $\widehat{\gamma}$ with side length 2ℓ , keeping the same center. There exists a set P of $4^d c$ points hitting all objects that intersect $\widehat{\gamma}$ and have side length at least $\ell/2$.

Now take a subset S of objects containing a common point p_0 . Let U be the union of the objects in S . Suppose that U intersects γ and has side length at least ℓ .

■ Case 1: $p_0 \in \widehat{\gamma}$. Some object $u \in S$ has side length at least $\ell/2$. Since u contains p_0 and thus intersects $\widehat{\gamma}$, we know that u is hit by P , and so U is hit by P .

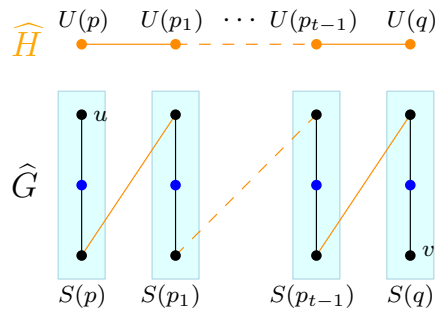
■ Case 2: $p_0 \notin \widehat{\gamma}$. Some object $u \in S$ intersects γ , and u must have side length at least $\ell/2$. Thus, u is hit by P , and so U is hit by P .

This proves $(4^d c)$ -fatness of U . The C -alignedness of U follows from the C -alignedness of the individual objects in S . ◀

Let $t_1 = 3$ and $t_k = 3t_{k-1} + 3$. This implies $t_k = \frac{11}{9}3^k - \frac{2}{3}$. We now describe a construction of a t_k -hop spanner.

Fat Object Construction II

1. Apply Lemma 13 to the leftmost points of the objects to obtain a set Γ of $O(n/r)$ generalized quadtree cells. For each $\gamma \in \Gamma$, construct a t_k -hop spanner recursively for the objects completely inside γ .



■ **Figure 5** Assume that $u \in S(p)$ and $v \in S(q)$ for some $p, q \in P$. If \widehat{H} connects $U(p)$ and $U(q)$ by t hops, then \widehat{G} connects u and v by $3t + 2$ hops. Stars in \widehat{G} are shown as blue boxes. The blue dots represent the centers of these stars.

2. For each $\gamma \in \Gamma$, let P_{γ^+} be a set of $O_d(c)$ points hitting all objects that intersect $\partial\gamma^+$ and have side length at least $\ell_{\gamma^+}/(2d^*)$, where ℓ_{γ^+} denotes the side length of γ^+ . Similarly, let P_{γ^-} be a set of $O_d(c)$ points hitting all objects that intersect $\partial\gamma^-$ and have side length at least $\ell_{\gamma^-}/(2d^*)$, where ℓ_{γ^-} denotes the side length of γ^- . Let $P_\gamma = P_{\gamma^+} \cup P_{\gamma^-}$. For each object u completely inside γ and each point $p \in P_\gamma$, add an edge between u and an arbitrary object that is hit by p and intersects u (if exists) to \widehat{G} . At most $O_d(cn)$ edges are added this way.
3. Let $P = \bigcup_{\gamma \in \Gamma} P_\gamma$. Assign each object u that is hit by P to an arbitrary $p \in P$ that hits u . For each $p \in P$, build a star $S(p)$ connecting all objects assigned to p , with the center chosen arbitrarily. Add the edges of these stars to \widehat{G} . At most $O(n)$ edges are added this way.
4. For each $p \in P$, define the new object $U(p)$ to be the union of the objects in $S(p)$. Recursively construct a t_{k-1} -hop spanner \widehat{H} for these new objects, which are $(4^d c)$ -fat and $(2d^*)$ -aligned by Observation 14. For each edge $U(p)U(p')$ in the spanner \widehat{H} , add an edge ww' to \widehat{G} , where $w \in S(p)$ and $w' \in S(p')$ are intersecting objects chosen arbitrarily.

Hop stretch. For any edge uv , if both u and v are completely inside a generalized quadtree cell in Γ , then u and v are connected by t_k hops by induction.

Otherwise, consider the case where neither u nor v are completely inside a generalized quadtree cell in Γ . Then u intersects $\partial\gamma$ and v intersects $\partial\gamma'$ for some $\gamma, \gamma' \in \Gamma$. Observe that u has side length at least $\ell_{\gamma^+}/(2d^*)$ (resp. $\ell_{\gamma^-}/(2d^*)$) if u intersects $\partial\gamma^+$ (resp. $\partial\gamma^-$), because u is $(2d^*)$ -aligned. So, u is hit by P_γ . Similarly, v is hit by $P_{\gamma'}$. Thus, u and v belong to two stars $S(p)$ and $S(q)$ from step 3 for some $p, q \in P$. The two objects $U(p)$ and $U(q)$ intersect and, by induction, are connected by t_{k-1} hops in the spanner \widehat{H} . Inside a star, 2 hops suffice. Thus, u and v are connected by $3t_{k-1} + 2$ hops in \widehat{G} (see Figure 5).

Lastly, consider the case when exactly one of the objects, say, u , is completely inside a generalized quadtree cell γ in Γ , and the other object v intersects $\partial\gamma$. Then, v is hit by some point $q \in P_\gamma$. By step 2, the spanner \widehat{G} connects u to some object v' that is hit by the same point q . Then v and v' belong to two stars $S(p)$ and $S(p')$ from step 3 for some $p, p' \in P$. The two objects $U(p)$ and $U(p')$ intersect. By the same argument in the previous case, v and v' are connected by $3t_{k-1} + 2$ hops in \widehat{G} , and so u and v are connected by $3t_{k-1} + 3$ hops.

23:12 Constant-Hop Spanners for More Geometric Intersection Graphs

Sparsity. The size of the t_k -hop spanner for c -fat ($2d^*$)-aligned objects follows the recurrence

$$S_{k,c}(n) \leq \max_{n_1, n_2, \dots \leq r: \sum_i n_i \leq n} \left(\sum_i S_{k,c}(n_i) + S_{k-1, 4^d c}(O_d(cn/r)) + O_d(cn) \right).$$

For the base case, we have $S_{1,c}(n) = O_d(cn \log n)$ by Theorem 11. For $k > 1$, we choose $r = \alpha_{k-1}(n)$. It is straightforward to verify by induction that $S_{k,c}(n) = O_{d,k,c}(n\alpha_k(n))$.

► **Theorem 15.** *The intersection graph of n c -fat objects in \mathbb{R}^d admits a $(\frac{11}{9}3^k - \frac{2}{3})$ -hop spanner of size $O_{d,k,c}(n\alpha_k(n))$ for any $k \geq 1$.*

Remarks. Because the fatness parameter c grows as a function of k during recursion, the constant factor in the above size bound depends on k .

Like in the previous section, it is also possible to obtain an intermediate result, namely, a 6-hop spanner with $O(n \log \log n)$ size.

4 Objects with (Near) Linear Union Complexity in \mathbb{R}^2

In this section, we describe a different approach to construct hop spanners, using the shallow cutting lemma introduced by Matoušek [35]. The variant below can be found in [23].

► **Lemma 16** (Shallow cutting). *Consider a family of well-behaved² objects in \mathbb{R}^2 , such that the union of any n objects has complexity at most $\mathcal{U}(n)$, assuming that $\mathcal{U}(n)/n$ is nondecreasing. Given a set of n objects in this family and parameters r and k , there exists a collection of $O((rk/n + 1)^2 \mathcal{U}(n/k))$ cells, such that (i) each cell intersects the boundaries of at most n/r objects, and (ii) the cells cover all points of depth at most k . Here, the depth of a point p is the number of objects that contain p .*

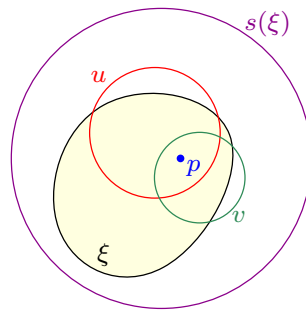
Construction via Shallow Cuttings

1. For each $i = 1, \dots, \log n$, apply Lemma 16 with $k = 2^i$ and $r = n/2^{i-2}$ to obtain a collection Ξ_i of $O(\mathcal{U}(n/2^i))$ cells. We may assume that each cell Ξ_i contains at least one point of depth at most 2^i (otherwise, the cell may be removed).
2. For each i and for each $\xi \in \Xi_i$ such that there exists an object $s(\xi)$ that contains ξ completely, build a star centered at $s(\xi)$ connecting all objects that intersect ξ . Add the edges of these stars to \widehat{G} . Since there are at most 2^i objects that contain ξ and 2^{i-2} objects whose boundaries intersect ξ , the number of edges added is $O(\sum_{i=1}^{\log n} \mathcal{U}(n/2^i) \cdot 2^i) = O(\mathcal{U}(n) \log n)$.

Hop stretch. For any edge uv , pick an arbitrary point p in the intersection of u and v , and let k be the depth of p . Let i be the number such that $2^{i-1} \leq k < 2^i$. Let ξ be the cell in Ξ_i that contains p . At least 2^{i-1} objects contain p , but at most 2^{i-2} objects have boundaries intersecting ξ . Thus, there must exist an object $s(\xi)$ that completely contains ξ . Then, \widehat{G} contains the edges $s(\xi)u$ and $s(\xi)v$, and so u and v are connected by 2 hops (see Figure 6).

► **Theorem 17.** *Consider a family of well-behaved objects in \mathbb{R}^2 , such that the union of any n objects has complexity at most $\mathcal{U}(n)$, assuming that $\mathcal{U}(n)/n$ is nondecreasing. The intersection graph of n objects in this family admits a 2-hop spanner of size $O(\mathcal{U}(n) \log n)$.*

² See [23] for a precise definition. Most families of objects in \mathbb{R}^2 , such as disks, pseudodisks, etc. are well-behaved.



■ **Figure 6** Objects u and v intersect, p is an arbitrary point in $u \cap v$, and $\xi \in \Xi_i$ is the cell containing p . Our counting argument shows that there must exist an object $s(\xi)$ completely containing ξ , and thus \widehat{G} must contain the path $u \rightarrow s(\xi) \rightarrow v$.

For example, it is known that $\mathcal{U}(n) = O(n)$ for disks and pseudodisks in the plane [32], and $\mathcal{U}(n) = O(n \log^* n)$ for fat triangles in the plane [6]. So, we obtain 2-hop spanners of size $O(n \log n)$ for disks and pseudodisks, and size $O(n \log n \log^* n)$ for fat triangles.

Remarks. It is possible to reduce the size bound to $O(\mathcal{U}(n) \log \log n)$ with 5 hops (by using fewer shallow cuttings, with $k = 2^{(1+\delta)^i}$, for $i = 1, \dots, O(\log \log n)$), but this approach does not appear to yield further improvement for larger hop stretch.

5 Axis-Aligned Rectangles in \mathbb{R}^2

In this section, we describe a 3-hop spanner for the case where the input objects consist of horizontal line segments H and vertical line segments V in the plane. Spanners for the more general case of axis-aligned rectangles will then follow.

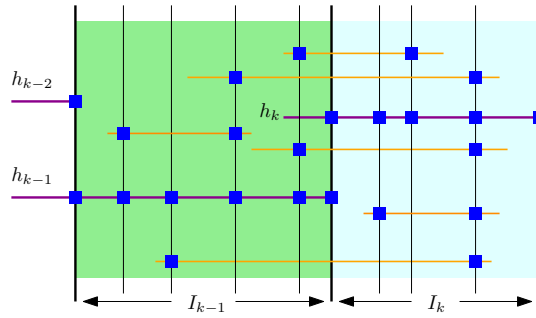
We first consider the special case where all vertical segments are lines. This problem is 1-dimensional in the sense that the y -coordinates of the segments are irrelevant. Borrowing Conroy and Tóth’s technique [24] for 1D interval graphs, we divide the x -axis into disjoint intervals $\mathcal{I} = \{I_1, \dots, I_\ell\}$ as follows (see Figure 7):

1. $I_0 = \{x_0\}$ is the interval containing only the x -coordinate of the leftmost endpoint among all horizontal segments.
2. For integers $k \geq 1$, $I_k = (x_{k-1}, x_k]$, where x_{k-1} is the right boundary of I_{k-1} , and x_k is the largest number for which there exists a line segment $h_k = [x'_k, x_k] \in H$ such that $x'_k \leq x_{k-1}$. We say that h_k is the *covering segment* of I_k .

► **Lemma 18.** *The intersection graph G of n horizontal segments and vertical lines admits a 3-hop spanner \widehat{G} with $O(n)$ edges.*

Proof. For each interval $I = (x_L, x_R]$, let h_I be the covering segment. Keep all intersections that involve h_I in the slab $I \times \mathbb{R}$. Finally, for every segment $h \in H$, keep the intersections with the leftmost and the rightmost vertical line that intersects h , denoted $v_L(h)$ and $v_R(h)$ respectively. Let \widehat{G} be the subgraph that includes an edge for each intersection we keep.

Hop stretch. Consider $h \in H$ and $v \in V$ that intersect. Either $v_L(h)$ or $v_R(h)$ is in the same interval $I \in \mathcal{I}$ as v ; say it is $v_R(h)$. Both v and $v_R(h)$ intersect the covering segment h_I . Thus, \widehat{G} contains the 3-hop path $h \rightarrow v_R(h) \rightarrow h_I \rightarrow v$.



■ **Figure 7** Illustration of the disjoint intervals I_{k-1} and I_k , and the covering segments h_{k-2}, h_{k-1}, h_k (drawn in purple). The blue squares indicate which intersections are kept.

Sparsity. For each covering segment h_I , we have only kept intersections in the interval I , so we have kept $O(n)$ intersections over all intervals I . For each $h \in H$ that is not a covering segment, we have kept only two intersections involving h . ◀

Using Lemma 18, the standard binary divide-and-conquer along the y -axis gives us a 3-hop spanner with $O(n \log n)$ edges for the case of horizontal and vertical line segments. Given a horizontal slab σ , we construct the 3-hop spanner as follows:

1. Construct a 3-hop spanner according to Lemma 18 to handle the intersections between horizontal segments and *long* vertical segments, i.e., vertical segments that cross the entire slab σ . Then remove the long vertical segments.
2. Divide σ into two horizontal subslabs, each containing half the number of horizontal segments. For each of the two subslabs, construct a 3-hop spanner recursively.

Each segment, whether horizontal or vertical, appears in $O(\log n)$ of the recursive calls. Therefore, the total number of edges in the spanner is bounded by $O(n \log n)$. Thus, we have proved the following:

► **Lemma 19.** *The intersection graph of n horizontal/vertical segments admits a 3-hop spanner of size $O(n \log n)$.*

We can extend the results for axis-aligned line segments to axis-aligned rectangles by replacing each rectangle with four line segments, each being one side of the rectangle. We build a spanner for these line segments. If two rectangles intersect, then either their sides intersect, or one rectangle contains the other. The first case reduces to segment intersection. For the case of containment, Conroy and Tóth [24] have shown that using $O(n \log n)$ edges, there is a 2-hop spanner for the subgraph that includes only “corner intersections”, i.e., intersections where one rectangle contains a corner of the other rectangle.

► **Theorem 20.** *The intersection graph of n axis-aligned rectangles in \mathbb{R}^2 admits a 3-hop spanner of size $O(n \log n)$.*

6 Open Questions

Although we have obtained almost linear size bounds for hop spanners in string graphs and fat-object intersection graphs, a remaining question is whether these upper bounds could be further improved to linear, or whether an inverse-Ackermann-type lower bound could be proved.

Another question is whether near-linear bounds are possible for other intersection graphs not addressed here, e.g., for simplices in \mathbb{R}^3 . Here, one might want to start more modestly with any upper bound better than for general graphs.

References

- 1 Peyman Afshani and Konstantinos Tsakalidis. Optimal deterministic shallow cuttings for 3-d dominance ranges. *Algorithmica*, 80(11):3192–3206, 2018. doi:10.1007/s00453-017-0376-3.
- 2 Pankaj K. Agarwal, Noga Alon, Boris Aronov, and Subhash Suri. Can visibility graphs be represented compactly? *Discret. Comput. Geom.*, 12:347–365, 1994. doi:10.1007/BF02574385.
- 3 Pankaj K. Agarwal and Jeff Erickson. Geometric range searching and its relatives. In *Advances in Discrete and Computational Geometry*, volume 223 of *Contemporary Mathematics*, pages 1–56. AMS Press, 1999. URL: <http://jeffe.cs.illinois.edu/pubs/survey.html>.
- 4 Abu Reyan Ahmed, Greg Bodwin, Faryad Darabi Sahneh, Keaton Hamm, Mohammad Javad Latifi Jebelli, Stephen G. Kobourov, and Richard Spence. Graph spanners: A tutorial review. *Comput. Sci. Rev.*, 37:100253, 2020. doi:10.1016/j.cosrev.2020.100253.
- 5 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discret. Comput. Geom.*, 9:81–100, 1993. doi:10.1007/BF02189308.
- 6 Boris Aronov, Mark de Berg, Esther Ezra, and Micha Sharir. Improved bounds for the union of locally fat objects in the plane. *SIAM J. Comput.*, 43(2):543–572, 2014. doi:10.1137/120891241.
- 7 Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel H. M. Smid. Euclidean spanners: Short, thin, and lanky. In *Proc. 27th Annual ACM Symposium on Theory of Computing (STOC)*, pages 489–498, 1995. doi:10.1145/225058.225191.
- 8 Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998. doi:10.1145/293347.293348.
- 9 Marshall W. Bern. Approximate closest-point queries in high dimensions. *Inf. Process. Lett.*, 45(2):95–99, 1993. doi:10.1016/0020-0190(93)90222-U.
- 10 Ahmad Biniaz. Plane hop spanners for unit disk graphs: Simpler and better. *Comput. Geom.*, page 101622, 2020. doi:10.1016/j.comgeo.2020.101622.
- 11 Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, André Nusser, and Zahra Parsaeian. Towards sub-quadratic diameter computation in geometric intersection graphs. In *Proc. 38th International Symposium on Computational Geometry (SoCG)*, pages 21:1–21:16, 2022. doi:10.4230/LIPIcs.SoCG.2022.21.
- 12 Sergio Cabello and Miha Ježič. Shortest paths in intersection graphs of unit disks. *Comput. Geom.*, 48(4):360–367, 2015. doi:10.1016/j.comgeo.2014.12.003.
- 13 Nicolas Catusse, Victor Chepoi, and Yann Vaxès. Planar hop spanners for unit disk graphs. In *Proc. 6th International Workshop on Algorithms for Sensor Systems, Wireless Ad Hoc Networks, and Autonomous Mobile Entities (ALGOSENSORS)*, pages 16–30, 2010. doi:10.1007/978-3-642-16988-5_2.
- 14 Timothy M. Chan. Approximate nearest neighbor queries revisited. *Discret. Comput. Geom.*, 20(3):359–373, 1998. doi:10.1007/PL00009390.
- 15 Timothy M. Chan. Random sampling, halfspace range reporting, and construction of ($\leq k$)-levels in three dimensions. *SIAM J. Comput.*, 30(2):561–575, 2000. doi:10.1137/S0097539798349188.
- 16 Timothy M. Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46(2):178–189, 2003.
- 17 Timothy M. Chan. Dynamic subgraph connectivity with geometric applications. *SIAM J. Comput.*, 36(3):681–694, 2006. doi:10.1137/S009753970343912X.
- 18 Timothy M. Chan. Dynamic geometric data structures via shallow cuttings. *Discret. Comput. Geom.*, 64(4):1235–1252, 2020. doi:10.1007/s00454-020-00229-5.

- 19 Timothy M. Chan. Finding triangles and other small subgraphs in geometric intersection graphs. In *Proc. 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023. To appear. URL: <https://arxiv.org/abs/2211.05345>.
- 20 Timothy M. Chan and Sarel Har-Peled. On the number of incidences when avoiding an induced biclique in geometric settings. In *Proc. 34th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2023. To appear. URL: <https://arxiv.org/abs/2112.14829>.
- 21 Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. *J. Comput. Geom.*, 10(1):27–41, 2019. doi:10.20382/jocg.v10i1a2.
- 22 Timothy M. Chan and Dimitrios Skrepetos. Approximate shortest paths and distance oracles in weighted unit-disk graphs. *J. Comput. Geom.*, 10(2):3–20, 2019. doi:10.20382/jocg.v10i2a2.
- 23 Chandra Chekuri, Kenneth L. Clarkson, and Sarel Har-Peled. On the set multicover problem in geometric settings. *ACM Trans. Algorithms*, 9(1), December 2012. doi:10.1145/2390176.2390185.
- 24 Jonathan B. Conroy and Csaba D. Tóth. Hop-spanners for geometric intersection graphs. In *38th International Symposium on Computational Geometry (SoCG)*, pages 30:1–30:17, 2022. doi:10.4230/LIPIcs.SoCG.2022.30.
- 25 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2008. URL: <https://www.worldcat.org/oclc/227584184>.
- 26 Adrian Dumitrescu, Anirban Ghosh, and Csaba D. Tóth. Sparse hop spanners for unit disk graphs. *Comput. Geom.*, page 101808, 2022. doi:10.1016/j.comgeo.2021.101808.
- 27 Paul Erdős. Extremal problems in graph theory. In *Proc. Symp. on Graph Theory, Smolenice, Acad. C.S.S.R.*, pages 29–36, 1963.
- 28 Jacob Fox and János Pach. A separator theorem for string graphs and its applications. *Combinatorics, Probability and Computing*, 19(3):371–390, 2010.
- 29 Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal on computing*, 16(6):1004–1022, 1987.
- 30 Greg N. Frederickson. Ambivalent data structures for dynamic 2-edge-connectivity and k smallest spanning trees. *SIAM Journal on Computing*, 26(2):484–538, 1997.
- 31 Jie Gao and Li Zhang. Well-separated pair decomposition for the unit-disk graph metric and its applications. *SIAM J. Comput.*, 35(1):151–169, 2005. doi:10.1137/S0097539703436357.
- 32 Klara Kedem, Ron Livne, János Pach, and Micha Sharir. On the union of jordan regions and collision-free translational motion amidst polygonal obstacles. *Discret. Comput. Geom.*, 1:59–70, 1986. doi:10.1007/BF02187683.
- 33 Hung Le, Lazar Milenkovic, and Shay Solomon. Sparse euclidean spanners with tiny diameter: A tight lower bound. In *Proc. 38th International Symposium on Computational Geometry (SoCG)*, pages 54:1–54:15, 2022. doi:10.4230/LIPIcs.SoCG.2022.54.
- 34 James R. Lee. Separators in region intersection graphs. In *Proc. 8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 1:1–1:8, 2017. doi:10.4230/LIPIcs.ITCS.2017.1.
- 35 Jirí Matoušek. Reporting points in halfspaces. *Computational Geometry*, 2(3):169–186, 1992.
- 36 Jirí Matoušek. Near-optimal separators in string graphs. *Comb. Probab. Comput.*, 23(1):135–139, 2014. doi:10.1017/S0963548313000400.
- 37 Giri Narasimhan and Michiel H. M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- 38 Chenyu Yan, Yang Xiang, and Feodor F. Dragan. Compact and low delay routing labeling scheme for unit disk graphs. In *Proc. Algorithms and Data Structures, 11th International Symposium (WADS)*, pages 566–577, 2009. doi:10.1007/978-3-642-03367-4_49.

Minimum L_∞ Hausdorff Distance of Point Sets Under Translation: Generalizing Klee’s Measure Problem

Timothy M. Chan  

Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA

Abstract

We present a (combinatorial) algorithm with running time close to $O(n^d)$ for computing the minimum directed L_∞ Hausdorff distance between two sets of n points under translations in any constant dimension d . This substantially improves the best previous time bound near $O(n^{5d/4})$ by Chew, Dor, Efrat, and Kedem from more than twenty years ago. Our solution is obtained by a new generalization of Chan’s algorithm [FOCS’13] for Klee’s measure problem.

To complement this algorithmic result, we also prove a nearly matching conditional lower bound close to $\Omega(n^d)$ for combinatorial algorithms, under the Combinatorial k -Clique Hypothesis.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Hausdorff distance, geometric optimization, Klee’s measure problem, fine-grained complexity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.24

Related Version *Full Version*: <https://arxiv.org/abs/2303.09122>

Funding *Timothy M. Chan*: Work supported in part by NSF Grant CCF-2224271.

1 Introduction

This paper is about the following problem:

► **Problem 1.** (L_∞ Translational Hausdorff) *Given a set P of n points and a set Q of m points in \mathbb{R}^d , compute the minimum directed L_∞ Hausdorff distance from P to Q under translation, i.e., compute $\min_{v \in \mathbb{R}^d} \vec{h}_\infty(P+v, Q)$ where $\vec{h}_\infty(P, Q) := \max_{p \in P} \min_{q \in Q} \|p - q\|_\infty$.*

The problem has been extensively studied in computational geometry in the 1990s. The analogous problem for *undirected* Hausdorff distance (defined as $h_\infty(P, Q) = \max\{\vec{h}_\infty(P, Q), \vec{h}_\infty(Q, P)\}$) is reducible [27] to the directed version if $m = \Theta(n)$. The motivation lies in measuring the resemblance between two geometric objects represented as point clouds; furthermore, a connection with an even more fundamental problem, Klee’s measure problem (see next page), provides added theoretical interest (and is what attracted this author’s attention in the first place). Huttenlocher and Kedem [22] introduced the problem and presented the first algorithms for $d = 2$ (a subsequent paper [23] also examined variants in L_2). Chew and Kedem [17] gave an improved algorithm with running time $O(mn \log^2(mn))$ for $d = 2$ (in L_∞), and generalized the algorithm to any constant dimension d with running time $O((mn)^{d-1} \log^2(mn))$. Chew, Dor, Efrat, and Kedem [15] described further improved algorithms in higher dimensions: in the main case $m = n$, their time bounds were $O(n^3 \log^2 n)$ for $d = 3$, $O(n^{(4d-2)/3} \log^2 n)$ for $4 \leq d \leq 7$, and $O(n^{5d/4} \log^2 n)$ for any constant $d \geq 8$. The exponent $5d/4$ looks peculiar, and naturally raises the question of whether further improvements are still possible, but none has been found in the intervening two decades (except in the logarithmic factors [12, 14]).

Many other variants of Problem 1 have also been considered, for example, using other metrics such as L_2 (as already mentioned above), allowing rotation and/or scaling besides translation, handling other objects besides points, allowing approximations, etc. (e.g., see



© Timothy M. Chan;

licensed under Creative Commons License CC-BY 4.0

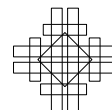
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 24; pp. 24:1–24:13

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Table 1** Previous upper bounds [15] and new upper bounds for Problem 1 in the $m = n$ case, ignoring polylogarithmic factors.

dimension	2	3	4	5	6	7	8	9	10	11	12	...
prev. bound	n^2	n^3	$n^{4.66\dots}$	n^6	$n^{7.33\dots}$	$n^{8.66\dots}$	n^{10}	$n^{11.25}$	$n^{12.5}$	$n^{13.75}$	n^{15}	...
new bound			n^4	n^5	n^6	n^7	n^8	n^9	n^{10}	n^{11}	n^{12}	...

[16, 4, 21, 24, 19, 18, 3]). Several other alternatives to the Hausdorff distance have also been popularly studied in computational geometry, such as the Earth mover distance and the Fréchet distance. We will ignore all these variants in the present paper, focusing only on exact directed L_∞ Hausdorff distance for point sets under translation.

Our new result is an algorithm for Problem 1 running in $O(n^d(\log \log n)^{O(1)})$ time (using randomization) for any constant $d \geq 3$ in the main $m = n$ case (or with one extra logarithmic factor if randomization is not allowed). The exponent d is thus a *substantial* improvement over Chew et al.’s previous exponents for *every* $d \geq 4$; see Table 1. In the general case, the running time of our algorithm is $O((mn)^{d/2}(\log \log(mn))^{O(1)})$.

Connection to a generalized Klee’s measure problem. It suffices to focus on the decision problem: deciding whether the minimum is at most a given value r . The original problem reduces to the decision problem, at the expense of one extra logarithmic factor in the running time by a well-known technique of Frederickson and Johnson [20] (in fact, when $d \geq 4$, a standard binary search suffices, since the optimal value lies in a universe of $O((mn)^2)$ possible values which we can explicitly enumerate). In some cases, the extra logarithmic factor can even be eliminated by a randomized optimization technique [12].

Equivalently, we want to decide whether there exists a vector $v \in \mathbb{R}^d$ with $P + v \subseteq Q + [-r, r]^d$ (where “+” denotes the Minkowski sum when it is clear from the context). Assume (by rescaling) that $r = 1/2$. Let \mathcal{Q} be the set of unit hypercubes $\{q + [-1/2, 1/2]^d : q \in \mathcal{Q}\}$, and let $S^* := \bigcup_{B \in \mathcal{Q}} B$. The condition is equivalent to $P + v \subseteq S^*$, i.e., $v \in \bigcap_{p \in P} (S^* - p)$. Thus, the decision problem is equivalent to the following:

► **Problem 2.** (L_∞ Translational Hausdorff Decision) *Given a set P of n points and a set \mathcal{Q} of m unit hypercubes¹ in \mathbb{R}^d , decide whether $\bigcap_{p \in P} (S^* - p) = \emptyset$, where $S^* := \bigcup_{B \in \mathcal{Q}} B$.*

For each $B \in \mathcal{Q}$ and $p \in P$, create a new unit hypercube $B - p$ and give this hypercube the color p . Problem 2 then immediately reduces to the following problem on $N = mn$ colored unit hypercubes: decide whether $\bigcap_\chi S_\chi = \emptyset$, where $S_\chi := \bigcup_{B \in \mathcal{B} \text{ with color } \chi} B$. (In other words, we want to decide whether there exists a “colorful” point that lies in hypercubes of all colors.)

The unit hypercube case in turn reduces to the case of *orthants* (i.e., d -sided boxes which are unbounded in one direction along each axis): we can build a uniform grid of unit-side length and solve the subproblem inside each grid cell, but inside a grid cell, a unit hypercube is identical to an orthant. (We can ignore grid cells that do not intersect hypercubes of all colors.) Since a unit hypercube intersects only $O(1)$ grid cells, these subproblems have total input size $O(N)$. All this motivates the definition of the following problem(s) on colored orthants:

¹ Throughout this paper, all hypercubes and boxes are axis-aligned.

► **Problem 3** (Generalized Klee's Measure Problem). *Given a set \mathcal{B} of N colored orthants in \mathbb{R}^d ,*

- (a) *decide whether $\bigcap_{\chi} S_{\chi} = \emptyset$,*
 - (b) *or more generally, compute a point of maximum or minimum depth among the S_{χ} 's (i.e., a point in the most or least number of regions S_{χ}),*
 - (c) *or alternatively, compute the volume of $\bigcap_{\chi} S_{\chi}$,*
- where $S_{\chi} := \bigcup_{B \in \mathcal{B} \text{ with color } \chi} B$.

To recap, if Problem 3(a) can be solved in $T(N)$ time, then Problem 2 can automatically be solved in $O(T(N)) = O(T(mn))$ time (assuming superadditivity of $T(\cdot)$).

Problem 3(a) is a generalization of the *box coverage problem*: determine whether the union of a set of N boxes in \mathbb{R}^d covers the entire space. This is because $\bigcap_{\chi} S_{\chi} = \emptyset$ iff $\bigcup_{\chi} \overline{S_{\chi}} = \mathbb{R}^d$, and a box can be expressed as the complement of a union of at most $2d$ orthants (we use \overline{S} to denote the complement of a set S). Similarly, Problem 3(b) is a generalization of the *box depth problem*: determine the minimum or maximum depth among N boxes in \mathbb{R}^d . Problem 3(c) is a generalization of *Klee's measure problem*: compute the volume of the union of a set of N boxes in \mathbb{R}^d . This generalization allows us to compute the volume of the union of more general shapes, so long as each shape can be expressed as the complement of a union of orthants. (Note that we can clip to a bounding box to ensure that the volume is finite.)

The original Klee's measure problem has been extensively studied in computational geometry [2, 8, 13, 14, 28, 30]. The best known algorithm by Chan [14] for Klee's measure problem runs in $O(N^{d/2})$ time, based on a clever but simple divide-and-conquer. The box coverage and box depth problem can be solved by similar algorithms, and in fact with slightly lower time bounds by polylogarithmic factors using table lookup and bit packing tricks [14].

For $d \leq 3$, a union of orthants has linear combinatorial complexity [7] and can be constructed in near linear time. Thus, a straightforward way to solve Problem 3 is to first construct all the regions S_{χ} explicitly, decompose each $\overline{S_{\chi}}$ as a union of disjoint boxes, and then run a known algorithm for Klee's measure problem on the resulting $O(N)$ boxes. With this approach, Problem 3 can be solved in $O(N \log N)$ time for $d = 2$, and $O(N^{3/2})$ time for $d = 3$; consequently, Problem 2 can be solved in $O((mn) \log(mn))$ time for $d = 2$, and $O((mn)^{3/2})$ time for $d = 3$. This was essentially how the previous known 2D and 3D algorithms by Chew and Kedem [17] and Chew et al. [15] were designed.

However, for $d \geq 4$, a union of N orthants may have $\Theta(N^{\lfloor d/2 \rfloor})$ combinatorial complexity in the worst case [7]. So, a two-stage approach that explicitly constructs all the regions S_{χ} and then invokes an algorithm for Klee's measure problem would be too slow! Chew et al. [15] adapted Overmars and Yap's algorithm for Klee's measure problem [28] in a nontrivial way to obtain an $O(N^{5d/8} \log N)$ -time algorithm for Problem 3, and consequently an $O(n^{5d/4} \log n)$ -time algorithm for Problem 2 when $m = n$.

We present a new algorithm that solves Problem 3(c) in $O(N^{d/2} \log^{d/2} N)$ time, matching the known time bound for the original Klee's measure problem up to logarithmic factors. In fact, for Problem 3(a,b), the polylogarithmic factor can be lowered to poly-log log N factors using table lookup and bit packing tricks. Consequently, we obtain an $O(n^d (\log \log n)^{O(1)})$ time bound for Problem 2 when $m = n$, or $O((mn)^{d/2} (\log \log(mn))^{O(1)})$ in general. Our result is obtained by directly modifying Chan's divide-and-conquer algorithm for Klee's measure problem [14]. The adaptation is not straightforward and uses interesting new ideas. As mentioned, we cannot afford to separate into two stages. Instead, within a single recursive process, we will handle two types of objects simultaneously, (i) the input orthants, and (ii) features of the regions S_{χ} that have been found during the process. The analysis of the recurrence is more delicate (though the overall algorithm remains simple).

Conditional lower bounds. In the other direction, recently in SoCG’21, Bringmann and Nusser [10] proved an $\Omega((mn)^{1-\delta})$ conditional lower bound for Problem 1–2 for $d = 2$ for an arbitrarily small constant $\delta > 0$, under the Orthogonal Vectors (OV) Hypothesis [29] (in particular, it holds under the Strong Exponential-Time Hypothesis (SETH) [29]). This showed that Chew and Kedem’s upper bound for $d = 2$ is likely near optimal [17]. However, Bringmann and Nusser did not obtain any lower bound in higher dimensions.

As observed by Chan [13], Klee’s measure problem and the box coverage problem have an $\Omega(N^{d/2-\delta})$ lower bound for combinatorial algorithms under the *Combinatorial k -Clique Hypothesis*, which states that there is no $O(n_0^{k-\delta})$ -time combinatorial algorithm for detecting a k -clique in a graph with n_0 vertices, for any constant $k \geq 3$. The notion of “combinatorial” algorithms is not mathematically well-defined, but intuitively it refers to algorithms that avoid the use of fast matrix multiplication (such as Strassen’s algorithm); all algorithms in this paper and in Chew et al.’s previous paper fulfill this criterion. (Recently, Künnemann [26] obtained new lower bounds for arbitrary, noncombinatorial algorithms for Klee’s measure problem under the “ k -Hyperclique Hypothesis”, but his bounds are not tight for $d \geq 4$. See also [11, 6, 5] for conditional lower bounds for other related geometric problems. See [25] for a recent example of the usage of the Combinatorial k -Clique Hypothesis in computational geometry, and [1, 9] for other examples involving the Combinatorial k -Clique Hypothesis outside of geometry.)

Since our algorithms for Problem 3 have near $N^{d/2}$ running time, they are near optimal for combinatorial algorithms under the Combinatorial k -Clique Hypothesis. However, this does not necessarily imply optimality of our algorithms for Problem 1 or 2.

In the second part of this paper, we prove that Problems 1–2 have a conditional lower bound of $\Omega(n^{d-\delta})$ for $m = n$, or $\Omega((mn)^{d/2-\delta})$ for $m = n^\gamma$ for any fixed $\gamma \leq 1$, for combinatorial algorithms under the Combinatorial k -Clique Hypothesis. This shows that our combinatorial algorithm for Problem 1 is also conditionally near optimal. While the previous conditional lower bound for Klee’s measure problem by Chan [13] was obtained by reduction from d -clique, we will reduce from clique of arbitrarily large constant size. Our new reduction is more challenging and more interesting, but still simple.

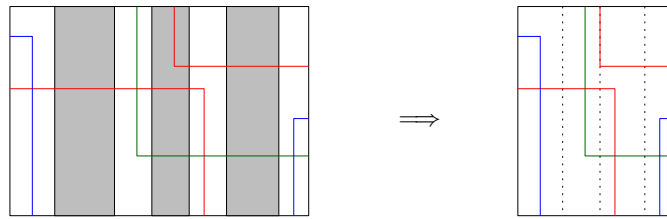
2 Algorithm

In this section, we present our new algorithm for the generalized Klee’s measure problem (Problem 3) for any constant dimension $d \geq 4$. From this result, new algorithms for Problems 1–2 will immediately follow.

To solve Problem 3(c), we solve a generalization, where we are given a box “cell” γ and an extra set \mathcal{E} of boxes, and we want to compute the volume of $\bigcap_{\chi} S_\chi \cap \bigcap_{E \in \mathcal{E}} \overline{E} \cap \gamma$. Initially, $\gamma = \mathbb{R}^d$ and $\mathcal{E} = \emptyset$. We assume that the coordinates of the input have been pre-sorted (this requires only an initial $O(N \log N)$ cost).

Call an orthant or a box *short* if some of its $(d - 2)$ -faces intersect γ ’s interior, *long* if it intersects γ ’s interior but is not short, and *trivial* if it does not intersect γ ’s interior or it completely contains γ .

Our algorithm is inspired by Chan’s divide-and-conquer algorithm [14] for the original Klee’s measure problem, with many similarities (for example, in how we use weighted medians to divide a cell) but also major new innovation (in how we reduce the number of long objects, and how we “convert” some objects of \mathcal{B} into new objects in \mathcal{E} during recursion). The analysis of our algorithm requires a new charging argument and recurrence, causing some extra logarithmic factors.



■ **Figure 1** Shrinking the width of the shaded slabs to 0, to eliminate long boxes in \mathcal{E} .

Defining weights. Consider a $(d - 2)$ -face f of a short orthant of \mathcal{B} such that f intersects γ 's interior. If f is orthogonal to the i -th and the j -th axes, assign f a weight of $2^{(i+j)/d}$. Note that this weight is $\Theta(1)$, and so each short orthant of \mathcal{B} contributes a total weight of $\Theta(1)$.

Similarly, consider a $(d - 2)$ -face f of a box of \mathcal{E} such that f intersects γ 's interior. If f is orthogonal to the i -th and the j -th axes, assign f a weight of $2^{(i+j)/d}/t$, where $t \geq 1$ is a parameter to be set later. Note that this weight is $\Theta(1/t)$, and so each short box of \mathcal{E} contributes a total weight of $\Theta(1/t)$.

Let $T(N_{\text{long}}, W_{\text{short}})$ denote the time complexity of the problem, where N_{long} denotes the total number of long and trivial orthants in \mathcal{B} and long and trivial boxes in \mathcal{E} , and W_{short} denotes the total weight of all short orthants in \mathcal{B} and short boxes in \mathcal{E} . Note that the total number of orthants in \mathcal{B} and boxes in \mathcal{E} is upper-bounded by $O(N_{\text{long}} + tW_{\text{short}})$.

Reducing the number of long objects. First, the trivial orthants and boxes can be easily eliminated: We can remove all orthants of \mathcal{B} and boxes of \mathcal{E} that do not intersect γ 's interior. If an entire color class of \mathcal{B} does not intersect γ 's interior, or if some box of \mathcal{E} completely contains γ , we can return 0 as the answer. If there is an orthant of \mathcal{B} completely containing γ , we can remove its color class from \mathcal{B} .

For each color χ , consider the long orthants of \mathcal{B} with color χ ; the union of these long orthants are defined by at most $2d$ orthants (since it is the complement of a box with at most $2d$ sides). Keep these $O(1)$ long orthants per color, and remove the rest. If there is a long orthant with color χ but no short orthant with that color, then $\overline{S_\chi} \cap \gamma$ is a box – add this box to \mathcal{E} and remove the color class from \mathcal{B} (in other words, we have “converted” an entire color class in \mathcal{B} into a single box in \mathcal{E}). This step increases W_{short} by at most N_{long}/t . After this step, each remaining long orthant can be “charged” to a short orthant of the same color, and so the number of remaining long orthants of \mathcal{B} is bounded by $O(1)$ times the number of short orthants, which is $O(W_{\text{short}})$.

Next, for each $i \in \{1, \dots, d\}$, consider the long boxes of \mathcal{E} having $(d - 1)$ -faces intersecting γ that are orthogonal to the i -th axis. Compute the union of these boxes by a linear scan after sorting, since this corresponds to computing the union of 1D intervals when projected to the i -th axis. The union forms a disjoint collection of slabs. Readjust all the i -th coordinates to shrink the width of these slabs to 0, without altering the volume of $\bigcap_\chi S_\chi \cap \bigcap_{E \in \mathcal{E}} \overline{E} \cap \gamma$, as illustrated in Figure 1. After doing this successively for every $i \in \{1, \dots, d\}$, all long boxes of \mathcal{E} are eliminated.

After this process, there are $O(W_{\text{short}})$ remaining long orthants of \mathcal{B} and zero long boxes of \mathcal{E} . Thus, N_{long} is reduced to $O(W_{\text{short}})$. The weight of the short orthants of \mathcal{B} may increase to at most $W_{\text{short}} + N_{\text{long}}/t$. We then have the following, for some constant c :

$$T(N_{\text{long}}, W_{\text{short}}) \leq T(cW_{\text{short}}, W_{\text{short}} + N_{\text{long}}/t) + O(N_{\text{long}} + tW_{\text{short}}). \tag{1}$$

Divide-and-conquer. Next, compute the weighted median m of the first coordinates of the $(d-2)$ -faces of \mathcal{B} and \mathcal{E} intersecting γ 's interior that are orthogonal to the first axis. Divide γ into two subcells γ_L and γ_R by the hyperplane $\{(x_1, \dots, x_d) : x_1 = m\}$. Renumber the coordinate axes $1, 2, \dots, d$ to $2, \dots, d, 1$, and recursively solve the problem for γ_L and for γ_R .

To analyze the algorithm, consider a $(d-2)$ -face f of \mathcal{B} (resp. \mathcal{E}) orthogonal to the i -th and j -th axes with $i, j \neq 1$. After the axis renumbering, its weight changes from $2^{(i+j)/d}$ to $2^{(i-1+j-1)/d}$ (resp. from $2^{(i+j)/d}/t$ to $2^{(i-1+j-1)/d}/t$), i.e., the weight decreases by a factor of $2^{2/d}$.

Next consider a $(d-2)$ -face f of \mathcal{B} (resp. \mathcal{E}) orthogonal to the first and the j -th axes with $j \neq 1$. After the axis renumbering, its weight changes from $2^{(1+j)/d}$ to $2^{(d+j-1)/d}$ (resp. from $2^{(1+j)/d}/t$ to $2^{(d+j-1)/d}/t$), i.e., the weight increases by a factor of $2^{(d-2)/d}$. But when γ is divided into subcells γ_L and γ_R , the weight within each subcell decreases by a factor of 2; the net decrease in weight is thus a factor of $2^{2/d}$.

Hence, W_{short} decreases by a factor of $2^{2/d}$ in either subcell. (On the other hand, N_{long} may not necessarily decrease.) We then have

$$T(N_{\text{long}}, W_{\text{short}}) \leq 2T(N_{\text{long}}, W_{\text{short}}/2^{2/d}) + O(N_{\text{long}} + tW_{\text{short}}). \quad (2)$$

Putting it together. By combining (2) and (1) and letting $T(N) := T(cN, N)$, we obtain

$$\begin{aligned} T(N) &\leq 2T(cN, N/2^{2/d}) + O(tN) \leq 2T(cN/2^{2/d}, N/2^{2/d} + cN/t) + O(tN) \\ &\leq 2T\left(\frac{1+2c/t}{2^{2/d}}N\right) + O(tN). \end{aligned}$$

For the base case, we have $T(O(1)) = O(t^{d/2})$: when $W_{\text{short}} = O(1)$, there are $O(1)$ orthants of \mathcal{B} and $O(t)$ boxes of \mathcal{E} , and so the problem can be solved by running a known algorithm for Klee's measure problem on $O(t)$ boxes [14].

By the master theorem, the solution to the recurrence is

$$T(N) = O(t^{d/2} N^{1/\log_2(2^{2/d}/(1+2c/t))}) = O(t^{d/2} N^{d/2+O(1/t)}).$$

Choosing $t = \log N$ yields $T(N) = O(N^{d/2} \log^{d/2} N)$. This completes the description and analysis of the main algorithm.

Shaving logs by bit packing. For Problem 3(a), we can obtain a minor (but not-very-practical) improvement in the polylogarithmic factors by using more technical but standard bit-packing tricks, as we now briefly explain (see [14] for more details on these kinds of tricks): The main observation is that actual coordinate values do not matter here, only their relative order, so we can replace them with their ranks in the sorted list. Thus, the $O(N_{\text{long}} + tW_{\text{short}})$ input objects can be represented by $O((N_{\text{long}} + tW_{\text{short}}) \log(N_{\text{long}} + tW_{\text{short}}))$ bits and packed in $O(((N_{\text{long}} + tW_{\text{short}}) \log(N_{\text{long}} + tW_{\text{short}}))/w)$ words, assuming a w -bit word RAM model of computation. The $O(N_{\text{long}} + tW_{\text{short}})$ cost for various linear scans during recursion can be reduced to $O(((N_{\text{long}} + tW_{\text{short}}) \log^2(N_{\text{long}} + tW_{\text{short}}))/w)$, since sorting k b -bit numbers can be done in $O((kb \log k)/w)$ time by a packed version of mergesort. Thus, the recurrence changes to $T(N) \leq 2T(\frac{1+2c/t}{2^{2/d}}N) + O(1 + (tN \log^2(tN))/w)$.

For the base case, we can use a bit-packed version of Chan's algorithm for the box coverage problem on $O(t)$ boxes, which runs in $T(O(1)) = O(1 + (t/w)^{d/2} \log^{O(1)} w)$ time [14, Section 3.1]. The recurrence solves to $T(N) = O(N^{d/2+O(1/t)} \cdot (1 + (t/w)^{d/2} \log^{O(1)} w))$. Choosing $t = \log N$ yields $T(N) = O(N^{d/2} \cdot (1 + ((\log N)/w)^{d/2} \log^{O(1)} w))$.

The above may require nonstandard operations on w -bit words. By choosing $w = \delta \log N$ for a sufficiently small constant $\delta > 0$, such operations can be simulated in constant time via table lookup after preprocessing in $2^{O(w)} = N^{O(\delta)}$ time. Hence, we obtain the final time bound of $O(N^{d/2}(\log \log N)^{O(1)})$.

Problem 3(b) can be solved similarly, with some modification to the steps to reduce the number of long boxes in \mathcal{E} , but this is identical to the modification of Chan's algorithm for the original box depth problem [14, Section 3.1]. To summarize, we have obtained the following theorem:

► **Theorem 1.** *For any constant $d \geq 4$, Problem 3(c) can be solved in $O(N^{d/2} \log^{d/2} N)$ time, and Problems 3(a,b) can be solved in $O(N^{d/2}(\log \log N)^{O(1)})$ time.*

► **Corollary 2.** *For any constant $d \geq 4$, Problem 2 can be solved in $O((mn)^{d/2}(\log \log(mn))^{O(1)})$ time. Problem 1 can be solved in $O((mn)^{d/2} \log(mn)(\log \log(mn))^{O(1)})$ time deterministically, or in $O((mn)^{d/2}(\log \log(mn))^{O(1)})$ expected time with randomization.*

Proof. As mentioned in Section 1, Problem 1 reduces to Problem 2 by Frederickson and Johnson's technique [20] or by ordinary binary search, with an extra logarithmic factor.

Chan [12, Section 4.2] has described how to apply his randomized optimization technique to reduce the following problem to its decision problem without losing a logarithmic factor:

Given N colored points in \mathbb{R}^d , find the smallest hypercube that contains points of all colors.

As noted in [12], Problem 1 reduces to this problem. On the other hand, as noted in Section 1, the decision version of this problem (equivalent to finding a point that is inside unit hypercubes of all colors) reduces to Problem 3(a), which we have just solved. ◀

3 Conditional Lower Bound

In this section, we prove a nearly matching conditional lower bound for Problems 1–2 for combinatorial algorithms under the Combinatorial k -Clique Hypothesis. We first introduce an intermediate problem which is more convenient to work with. Roughly speaking, Problem 2 considers the intersection of translates of a *single* shape (the shape being a union of unit hypercubes), whereas the problem below considers the intersection of translates of *multiple* shapes (each shape being a union of orthants).

► **Problem 4.** *Let \mathcal{Z} be a set of shapes, where each shape is a union of orthants in \mathbb{R}^d . Let m be the total number of orthants over all shapes of \mathcal{Z} . Given a set \mathcal{S} of n objects where each object is a translate of some shape in \mathcal{Z} , decide whether $\bigcap_{S \in \mathcal{S}} S = \emptyset$.*

► **Lemma 3.** *Problem 4 reduces to Problem 2 on $O(n)$ points and $O(m)$ unit hypercubes.*

Proof. Assume (by rescaling) that the coordinates of all orthants of \mathcal{Z} and all translation vectors used in \mathcal{S} are in $[0, 1/2]$. In particular, if $\bigcap_{S \in \mathcal{S}} S$ is nonempty, it must contain a point in $[0, 1]^d$. Inside $[0, 1]^d$, each orthant of \mathcal{Z} may be replaced by an equivalent unit hypercube.

Let Z_1, \dots, Z_ℓ be the shapes of \mathcal{Z} . Let \mathcal{B}_i be the unit hypercubes corresponding to the orthants defining Z_i (so that $(Z_i + t) \cap [0, 1]^d = \bigcup_{B \in \mathcal{B}_i} (B + t) \cap [0, 1]^d$ for any $t \in [0, 1/2]^d$).

We construct an instance of Problem 2 as follows: For each $B \in \mathcal{B}_i$, add the shifted unit hypercube $B + u_i$ to \mathcal{Q} where $u_i := (4i, 0, \dots, 0) \in \mathbb{R}^d$. (This operation distributes objects in different classes \mathcal{B}_i to different parts of space, since the vectors u_i are at least 4 units apart

from each other along the first axis.) For each object $S \in \mathcal{S}$, if S is the translate $Z_i + t$, add the point $u_i - t$ to P . Lastly, add two auxiliary unit hypercubes $[0, 1]^d$ and $u_{\ell+1} + [0, 1]^d$ to \mathcal{Q} , and two auxiliary points u_0 and $u_{\ell+1}$ to P .

We solve Problem 2 on these points of P and these unit hypercubes of \mathcal{Q} , to determine whether $\bigcap_{p \in P} (S^* - p) = \emptyset$, where $S^* := \bigcup_{B \in \mathcal{Q}} B$. For correctness, we just observe that $\bigcap_{p \in P} (S^* - p)$ is identical to $\bigcap_{S \in \mathcal{S}} S$ inside $[0, 1]^d$. This is because for each $B \in \mathcal{B}_i$, $(B + u_i) - (u_i - t)$ may intersect $[0, 1]^d$ only if $i = i'$ (since u_i and $u_{i'}$ are far apart if $i \neq i'$), assuming $t \in [0, 1/2]^d$. ◀

We now prove hardness of Problem 4 by reduction from the clique problem for graphs. We first warm up with two simpler reductions yielding weaker lower bounds, before presenting the final reduction in Lemma 6. (Readers who do not need intuition building may go straight to Lemma 6's proof.)

First attempt. First observe that the box coverage problem (deciding whether n boxes in \mathbb{R}^d cover the entire space, i.e., deciding whether the intersection of the complements of n boxes is empty) easily reduces to Problem 4 with n orthants and n objects in \mathbb{R}^d , since the complement of a box is the union of $O(1)$ orthants. By Lemma 3, we immediately obtain an $\Omega(n^{d/2-\delta})$ conditional lower bound for Problem 4, since the box coverage problem has an $\Omega(n^{d/2-\delta})$ lower bound under the Combinatorial k -Clique Hypothesis [13].

In the following lemma, we directly modify the (very simple) known reduction from clique to the box coverage problem [13], to show the same lower bound even when the number of orthants m is $O(1)$:

► **Lemma 4.** *Detecting a d -clique in a graph with n_0 vertices reduces to Problem 4 with $m = O(1)$ orthants and $n = O(n_0^2)$ objects in \mathbb{R}^d .*

Proof. Let $G = (V, E)$ be the given graph, with $V = [n_0] = \{0, \dots, n_0 - 1\}$. We will construct a set \mathcal{S} of objects whose intersection is

$$\{(x_1, \dots, x_d) \in [0, n_0]^d : \{[x_1], \dots, [x_d]\} \text{ is a } d\text{-clique of } G\}. \quad (3)$$

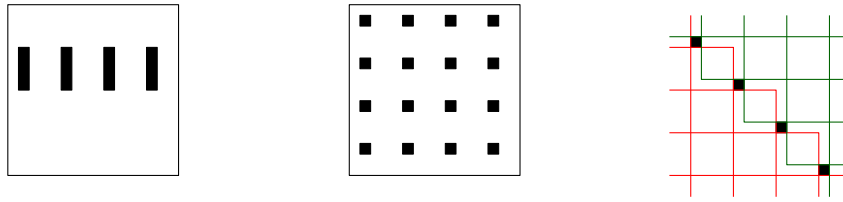
It would then follow that the intersection is nonempty iff a d -clique exists.

The construction is very simple: for each $\alpha, \beta \in \{1, \dots, d\}$ with $\alpha \neq \beta$ and for each $u, v \in [n_0]$ with $uv \notin E$, add the complement of the box

$$B_{\alpha, \beta, u, v} := \{(x_1, \dots, x_d) : [x_\alpha] = u, [x_\beta] = v\}$$

to \mathcal{S} . (Note that if $u = v$, we consider $uv \notin E$.) These boxes are unit squares when projected to the α -th and β -th axes, and are thus translates of $O(1)$ fixed boxes, and the complement of each such fixed box can obviously be expressed as a union of $O(1)$ orthants and can be added to \mathcal{Z} . Lastly, add the $O(1)$ halfspaces bounding $[0, n_0]^d$ to \mathcal{S} . Then \mathcal{S} has a total of $O(n_0^2)$ translates and clearly satisfies the desired property (3). ◀

In combination with Lemma 3, the above lemma indeed implies an $\Omega(n^{d/2-\delta})$ conditional lower bound for Problem 2: if Problem 2 for $m = O(1)$ has an $O(n^{d/2-\delta})$ -time combinatorial algorithm, then the d -clique detection problem for a graph with n_0 vertices has a combinatorial algorithm with running time $O((n_0^2)^{d/2-\delta}) = O(n_0^{d-2\delta})$, contradicting the Combinatorial k -Clique Hypothesis.



■ **Figure 2** (left) A region $Y_{\alpha,\beta,a,b,u,v}$ for $(a,b) = (0,1)$. (middle) A region $Y_{\alpha,\beta,a,b,u,v}$ for $(a,b) = (0,0)$. (right) A diagonal $D_{\alpha,\beta}$ (whose complement can be expressed as a union of the red and the green orthants).

Second attempt. We now improve the lower bound by reducing from clique of a large size $2d$. We use the following key idea: encode a pair of vertices in a single coordinate value.

► **Lemma 5.** *Detecting a $(2d)$ -clique in a graph G with n_0 vertices reduces to Problem 4 with $m = O(n_0)$ orthants and $n = O(n_0^3)$ objects in \mathbb{R}^d .*

Proof. Let $G = (V, E)$ be the given graph, with $V = [n_0]$. For any $x \in [0, n_0^2)$, let $\phi_0(x) = \lfloor x \rfloor \bmod n_0$ and $\phi_1(x) = \lfloor x/n_0 \rfloor$. We will construct a set \mathcal{S} of objects whose intersection is

$$\{(x_1, \dots, x_d) \in [0, n_0^2)^d : \{\phi_0(x_1), \phi_1(x_1), \dots, \phi_0(x_d), \phi_1(x_d)\} \text{ is a } (2d)\text{-clique in } G\}. \quad (4)$$

It would then follow that the intersection is nonempty iff a $(2d)$ -clique exists.

For each $\alpha, \beta \in \{1, \dots, d\}$ and $a, b \in \{0, 1\}$ with $(\alpha, a) \neq (\beta, b)$, and for each $u, v \in [n_0]$ with $uv \notin E$, define the region

$$Y_{\alpha,\beta,a,b,u,v} := \{(x_1, \dots, x_d) \in [0, n_0^2)^d : \phi_a(x_\alpha) = u, \phi_b(x_\beta) = v\}.$$

If $\alpha = \beta$, then $Y_{\alpha,\beta,a,b,u,v}$ is just a unit interval when projected to the α -th axis, and is thus a translate of one of $O(1)$ fixed boxes, and the complement of each such fixed box can be expressed as a union of $O(1)$ orthants and can be added to \mathcal{Z} . From now on, assume $\alpha \neq \beta$.

If $(a,b) = (1,1)$, then $Y_{\alpha,\beta,a,b,u,v}$ is just an $n_0 \times n_0$ square when projected to the α -th and β -th axes, and is thus a translate of one of $O(1)$ fixed boxes, and the complement of each such fixed box can be expressed as a union of $O(1)$ orthants and can be added to \mathcal{Z} .

If $(a,b) = (0,1)$ (or $(a,b) = (1,0)$), then $Y_{\alpha,\beta,a,b,u,v}$ is a union of n_0 rectangles of dimension $1 \times n_0$ (or $n_0 \times 1$) when projected to the α -th and β -th axes (see Figure 2(left)), and is thus a union of n_0 translates of one of $O(1)$ fixed boxes, and the complement of each fixed box can be expressed as a union of $O(1)$ orthants and can be added to \mathcal{Z} .

If $(a,b) = (0,0)$, then $Y_{\alpha,\beta,a,b,u,v}$ forms a $n_0 \times n_0$ grid pattern when projected to the α -th and β -th axes (see Figure 2(middle)). Although the complement of this region can't be expressed as a union of orthants, we can decompose the grid into subregions that can. The most obvious approach is to decompose into rows or columns, but this still doesn't work. Instead, we will decompose into "diagonals". More precisely, define

$$D_{\alpha,\beta} := \{(x_1, \dots, x_d) : \begin{aligned} & \lfloor x_\alpha \rfloor \bmod n_0 = 0, \lfloor x_\beta \rfloor \bmod n_0 = 0, \\ & \lfloor x_\alpha/n_0 \rfloor + \lfloor x_\beta/n_0 \rfloor = n_0, x_\alpha, x_\beta \geq 0 \}. \end{aligned}$$

Since $D_{\alpha,\beta,a,b}$ can be viewed as the region sandwiched between two staircases when projected to 2D, its complement $\overline{D_{\alpha,\beta,a,b}}$ can be expressed as a union of $O(n_0)$ orthants (see Figure 2(right)). Add the shape $D_{\alpha,\beta,a,b}$ to \mathcal{Z} . The region $Y_{\alpha,\beta,a,b,u,v}$ can be expressed as a union of $O(n_0)$ translates of $D_{\alpha,\beta}$ when clipped to $[0, n_0^2)^d$.

24:10 Minimum L_∞ Hausdorff Distance of Point Sets Under Translation

In any case, $\overline{Y_{\alpha,\beta,a,b,u,v}} \cap [0, n_0^2]^d$ can be expressed as the intersection of $[0, n_0^2]^d$ with $O(n_0)$ translates of shapes from \mathcal{Z} . Add these translates to \mathcal{S} , for each $\alpha, \beta \in \{1, \dots, d\}$ and $a, b \in [g]$ with $(\alpha, a) \neq (\beta, b)$, and for each $u, v \in [n_0]$ with $uv \notin E$. Lastly, add the $O(1)$ halfspaces bounding $[0, n_0^2]^d$ to \mathcal{S} . Then \mathcal{S} has a total of $O(n_0^3)$ translates and satisfies the desired property (4). \blacktriangleleft

The above lemma implies a larger $\Omega(n^{2d/3-\delta})$ conditional lower bound: if Problem 2 for $m = n^{1/3}$ has an $O(n^{2d/3-\delta})$ -time combinatorial algorithm, then the $(2d)$ -clique detection problem for a graph with n_0 vertices has a combinatorial algorithm with running time $O((n_0^3)^{2d/3-\delta}) = O(n_0^{2d-3\delta})$, contradicting the Combinatorial k -Clique Hypothesis.

Final reduction. We obtain our final lower bound by generalizing the idea further. We reduce from clique of still larger size and now encode g -tuples of vertices instead of pairs (incidentally, the idea of encoding tuples has also appeared recently in Künnemann’s conditional lower bound proofs for Klee’s measure problem [26]):

► **Lemma 6.** *Let g be any integer constant. Detecting a (dg) -clique in a graph G with n_0 vertices reduces to Problem 4 with $m = O(n_0^{g-1})$ orthants and $n = O(n_0^{g+1})$ objects in \mathbb{R}^d .*

More generally, for any given $m \leq n_0^{g-1}$, detecting a (dg) -clique in a graph G with n_0 vertices reduces to Problem 4 with m orthants and $n = O(n_0^{2g}/m)$ objects in \mathbb{R}^d .

Proof. Let $G = (V, E)$ be the given graph, with $V = [n_0]$. For any $x \in [0, n_0^g]$ and $a \in [g]$, let $\phi_a(x)$ be the $(a+1)$ -th least significant digit of $\lfloor x \rfloor$ in base n . We will construct a set \mathcal{S} of objects whose intersection is

$$\{(x_1, \dots, x_d) \in [0, n_0^g]^d : \{\phi_0(x_1), \dots, \phi_{g-1}(x_1), \dots, \phi_0(x_d), \dots, \phi_{g-1}(x_d)\} \text{ is a } (dg)\text{-clique in } G\}. \quad (5)$$

It would then follow that the intersection is nonempty iff a (dg) -clique exists.

For each $\alpha, \beta \in \{1, \dots, d\}$ and $a, b \in [g]$ with $(\alpha, a) \neq (\beta, b)$, and for each $u, v \in [n_0]$ with $uv \notin E$, define the region

$$\begin{aligned} Y_{\alpha,\beta,a,b,u,v} &:= \{(x_1, \dots, x_d) \in [0, n_0^d]^d : \phi_a(x_\alpha) = u, \phi_b(x_\beta) = v\} \\ &= \{(x_1, \dots, x_d) : x_\alpha \in [in_0^{a+1} + un_0^a, in_0^{a+1} + (u+1)n_0^a], \\ &\quad x_\beta \in [jn_0^{b+1} + vn_0^b, jn_0^{b+1} + (v+1)n_0^b] \\ &\quad \text{for some } i \in [n_0^{g-a-1}], j \in [n_0^{g-b-1}]\}. \end{aligned}$$

If $\alpha = \beta$, then $Y_{\alpha,\beta,a,b,u,v}$ is a union of at most $O(n_0^{g-1})$ unit intervals when projected to the α -th axis, and is thus a union of $O(n_0^{g-1})$ translates of $O(1)$ fixed boxes, and the complement of each fixed box can be expressed as a union of $O(1)$ orthants and can be added to \mathcal{Z} .

If $\alpha \neq \beta$, then $Y_{\alpha,\beta,a,b,u,v}$ forms a $O(n_0^{g-1}) \times O(n_0^{g-1})$ grid pattern when projected to the α -th and β -th axes. Define the “diagonal”

$$D_{\alpha,\beta,a,b} = \{(x_1, \dots, x_d) : x_\alpha \in [in_0^{a+1}, in_0^{a+1} + n_0^a], x_\beta \in [jn_0^{b+1}, jn_0^{b+1} + n_0^b] \text{ for some } i, j \in [m] \text{ with } i + j = m\}.$$

Since $D_{\alpha,\beta,a,b}$ can be viewed as the region sandwiched between two staircases when projected to 2D, its complement $\overline{D_{\alpha,\beta,a,b}}$ can be expressed as a union of $O(m)$ orthants. Add the shape $\overline{D_{\alpha,\beta,a,b}}$ to \mathcal{Z} . The region $Y_{\alpha,\beta,a,b,u,v}$ can be expressed as a union of $O(n_0^{2(g-1)}/m)$ translates of $D_{\alpha,\beta,a,b}$ when clipped to $[0, n_0^g]^d$.

In any case, $\overline{Y_{\alpha,\beta,a,b,u,v}} \cap [0, n_0^g]^d$ can be expressed as an intersection of $[0, n_0^g]^d$ with $O(n_0^{2(g-1)}/m)$ translates of shapes from \mathcal{Z} . Add all these translates to \mathcal{S} , for each $\alpha, \beta \in \{1, \dots, d\}$ and $a, b \in [g]$ with $(\alpha, a) \neq (\beta, b)$, and for each $u, v \in [n_0]$ with $uv \notin E$. Lastly, add the $O(1)$ halfspaces bounding $[0, n_0^g]^d$ to \mathcal{S} . Then \mathcal{S} has a total of $O(n_0^2 \cdot n_0^{2(g-1)}/m) = O(n_0^{2g}/m)$ translates and satisfies the desired property (5). ◀

The above lemma implies an $\Omega(n^{gd/(g+1)-\delta})$ conditional lower bound for any integer constant g : if Problem 2 for $m = n^{(g-1)/(g+1)}$ has an $O(n^{gd/(g+1)-\delta})$ -time combinatorial algorithm, then the (dg) -clique detection problem for a graph with n_0 vertices has a combinatorial algorithm with running time $O((n_0^{g+1})^{gd/(g+1)-\delta}) = O(n_0^{dg-(g+1)\delta})$, contradicting the Combinatorial k -Clique Hypothesis. The exponent $gd/(g+1) - \delta$ exceeds $d - 2\delta$, by picking a sufficiently large $g \geq d/\delta$.

More generally, for any constant $\gamma \leq (g-1)/(g+1)$, if Problem 2 for $m = n^\gamma$ has an $O((mn)^{d/2-\delta})$ -time combinatorial algorithm, then the (dg) -clique detection problem for a graph with n_0 vertices has a combinatorial algorithm with running time $O((n_0^{2g}/m) \cdot m)^{d/2-\delta} = O(n_0^{dg-2g\delta})$, contradicting the Combinatorial k -Clique Hypothesis.

▶ **Theorem 7.** *Under the Combinatorial k -Clique Hypothesis, Problem 1 or 2 for n points and n unit hypercubes in \mathbb{R}^d does not have an $O(n^{d-\delta})$ -time combinatorial algorithm for any constant $\delta > 0$.*

More generally, under the same hypothesis, for any fixed constant $\gamma \leq 1$, Problem 1 or 2 for $m = n^\gamma$ points and n unit hypercubes in \mathbb{R}^d does not have an $O((mn)^{d/2-\delta})$ -time combinatorial algorithm for any constant $\delta > 0$.

4 Final Remarks

To summarize, we have studied the L_∞ translational Hausdorff distance problem for point sets, a fundamental problem with a long history in computational geometry. We have obtained a substantially improved upper bound for this problem, and the first conditional lower bound in dimension 3 and higher, which nearly match the upper bound. Our technique for the upper bound is interesting, in that it implies a natural colored generalization of Klee’s measure problem can be solved in roughly the same time bound as the original Klee’s problem. Our lower bound proof is interesting, in that it adds to a growing body of recent work on fine-grained complexity in computational geometry, and more specifically illustrates the power of the Combinatorial Clique Hypothesis.

Our near- $O((mn)^{d/2})$ upper bound also applies to the variant of the problem for *undirected* Hausdorff distance, since the undirected version of Problem 1 can also be reduced to Problem 3(a) with $N = O(mn)$. However, more effort might be needed to adapt our lower bounds to the undirected problem (although we have not tried seriously).

For noncombinatorial algorithms, our reduction implies a lower bound of $\Omega((mn)^{d\omega/6-\delta})$, under the standard hypothesis that the k -clique problem for graphs with n_0 vertices requires $\Omega(n_0^{d\omega/3-\delta'})$ time, where $\omega \in [2, 2.373)$ denotes the matrix multiplication exponent. Proving better conditional lower bounds for noncombinatorial algorithms remains open. This might require further new techniques, as we currently do not have tight conditional lower bounds for the original Klee’s measure problem for noncombinatorial algorithms for $d \geq 4$ [26].

As mentioned, Bringmann and Nusser [10] proved a near- mn lower bound for $d = 2$ under the OV Hypothesis; their result is in some sense more robust (it holds for noncombinatorial algorithms) and applies also to the L_2 case (and L_p for any $1 \leq p \leq \infty$). However, the problem for L_2 probably has higher complexity than for L_∞ : the best upper bounds are near

n^3 for $d = 2$ and near n^5 for $d = 3$ [23], and near $n^{\lceil 3d/2 \rceil + 1}$ for $d \geq 4$ [15], in the $m = n$ case. (See Bringmann and Nusser’s paper for a 3SUM-based lower bound for the L_2 problem for $d = 2$ in the “unbalanced” case when m is constant.)

References

- 1 Amir Abboud, Arturs Backurs, and Virginia Vassilevska Williams. If the current clique algorithms are optimal, so is Valiant’s parser. *SIAM J. Comput.*, 47(6):2527–2555, 2018. Preliminary version in FOCS 2015. doi:10.1137/16M1061771.
- 2 Pankaj K. Agarwal. An improved algorithm for computing the volume of the union of cubes. In *Proc. 26th ACM Symposium on Computational Geometry (SoCG)*, pages 230–239, 2010. doi:10.1145/1810959.1811000.
- 3 Pankaj K. Agarwal, Sariel Har-Peled, Micha Sharir, and Yusu Wang. Hausdorff distance under translation for points and balls. *ACM Trans. Algorithms*, 6(4):71:1–71:26, 2010. Preliminary version in SoCG 2003. doi:10.1145/1824777.1824791.
- 4 Oswin Aichholzer, Helmut Alt, and Günter Rote. Matching shapes with a reference point. *Int. J. Comput. Geom. Appl.*, 7(4):349–363, 1997. Preliminary version in SoCG 1994. doi:10.1142/S0218195997000211.
- 5 Boris Aronov and Jean Cardinal. Geometric pattern matching reduces to k -SUM. In *Proc. 31st International Symposium on Algorithms and Computation (ISAAC)*, volume 181, pages 32:1–32:9, 2020. doi:10.4230/LIPIcs.ISAAC.2020.32.
- 6 Gill Barequet and Sariel Har-Peled. Polygon containment and translational min-Hausdorff-distance between segment sets are 3SUM-hard. *Int. J. Comput. Geom. Appl.*, 11(4):465–474, 2001. doi:10.1142/S0218195901000596.
- 7 Jean-Daniel Boissonnat, Micha Sharir, Boaz Tagansky, and Mariette Yvinec. Voronoi diagrams in higher dimensions under certain polyhedral distance functions. In *Proc. 11th Annual Symposium on Computational Geometry (SoCG)*, pages 79–88, 1995. doi:10.1145/220279.220288.
- 8 Karl Bringmann. An improved algorithm for Klee’s measure problem on fat boxes. *Comput. Geom.*, 45(5-6):225–233, 2012. Preliminary version in SoCG 2010. doi:10.1016/j.comgeo.2011.12.001.
- 9 Karl Bringmann, Allan Grønlund, and Kasper Green Larsen. A dichotomy for regular expression membership testing. In *Proc. 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 307–318, 2017. doi:10.1109/FOCS.2017.36.
- 10 Karl Bringmann and André Nusser. Translating Hausdorff is hard: Fine-grained lower bounds for Hausdorff distance under translation. In *Proc. 37th International Symposium on Computational Geometry (SoCG)*, pages 18:1–18:17, 2021. doi:10.4230/LIPIcs.SoCG.2021.18.
- 11 Sergio Cabello, Panos Giannopoulos, and Christian Knauer. On the parameterized complexity of d -dimensional point set pattern matching. *Inf. Process. Lett.*, 105(2):73–77, 2008. doi:10.1016/j.ipl.2007.08.003.
- 12 Timothy M. Chan. Geometric applications of a randomized optimization technique. *Discret. Comput. Geom.*, 22(4):547–567, 1999. Preliminary version in SoCG 1998. doi:10.1007/PL00009478.
- 13 Timothy M. Chan. A (slightly) faster algorithm for Klee’s measure problem. *Comput. Geom.*, 43(3):243–250, 2010. Preliminary version in SoCG 2008. doi:10.1016/j.comgeo.2009.01.007.
- 14 Timothy M. Chan. Klee’s measure problem made easy. In *Proc. 54th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 410–419, 2013. doi:10.1109/FOCS.2013.51.
- 15 L. Paul Chew, Dorit Dor, Alon Efrat, and Klara Kedem. Geometric pattern matching in d -dimensional space. *Discret. Comput. Geom.*, 21(2):257–274, 1999. Preliminary version in ESA 1995. doi:10.1007/PL00009420.

- 16 L. Paul Chew, Michael T. Goodrich, Daniel P. Huttenlocher, Klara Kedem, Jon M. Kleinberg, and Dina Kravets. Geometric pattern matching under Euclidean motion. *Comput. Geom.*, 7:113–124, 1997. doi:10.1016/0925-7721(95)00047-X.
- 17 L. Paul Chew and Klara Kedem. Getting around a lower bound for the minimum Hausdorff distance. *Comput. Geom.*, 10(3):197–202, 1998. Preliminary version in SWAT 1992. doi:10.1016/S0925-7721(97)00032-1.
- 18 Minkyong Cho and David M. Mount. Improved approximation bounds for planar point pattern matching. *Algorithmica*, 50(2):175–207, 2008. doi:10.1007/s00453-007-9059-9.
- 19 Alon Efrat, Piotr Indyk, and Suresh Venkatasubramanian. Pattern matching for sets of segments. *Algorithmica*, 40(3):147–160, 2004. Preliminary version in SODA 2001. doi:10.1007/s00453-004-1089-y.
- 20 Greg N. Frederickson and Donald B. Johnson. Generalized selection and ranking: Sorted matrices. *SIAM J. Comput.*, 13(1):14–30, 1984. doi:10.1137/0213002.
- 21 Michael T. Goodrich, Joseph S. B. Mitchell, and Mark W. Orletsky. Approximate geometric pattern matching under rigid motions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(4):371–379, 1999. Preliminary version in SoCG 1994. doi:10.1109/34.761267.
- 22 Daniel P. Huttenlocher and Klara Kedem. Computing the minimum Hausdorff distance for point sets under translation. In *Proc. 6th ACM Symposium on Computational Geometry (SoCG)*, pages 340–349, 1990. doi:10.1145/98524.98599.
- 23 Daniel P. Huttenlocher, Klara Kedem, and Micha Sharir. The upper envelope of Voronoi surfaces and its applications. *Discret. Comput. Geom.*, 9:267–291, 1993. Preliminary version in SoCG 1991. doi:10.1007/BF02189323.
- 24 Piotr Indyk, Rajeev Motwani, and Suresh Venkatasubramanian. Geometric matching under noise: Combinatorial bounds and algorithms. In *Proc. 10th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 457–465, 1999. URL: <http://dl.acm.org/citation.cfm?id=314500.314601>.
- 25 Ce Jin and Yinzhan Xu. Tight dynamic problem lower bounds from generalized BMM and OMv. In *Proc. 54th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1515–1528, 2022. doi:10.1145/3519935.3520036.
- 26 Marvin Künnemann. A tight (non-combinatorial) conditional lower bound for Klee’s Measure Problem in 3D. In *Proc. 63rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 555–566, 2022.
- 27 André Nusser. *Fine-Grained Complexity and Algorithm Engineering of Geometric Similarity Measures*. PhD thesis, Saarland University, Saarbrücken, Germany, 2021. URL: <https://publikationen.sulb.uni-saarland.de/handle/20.500.11880/33904>.
- 28 Mark H. Overmars and Chee-Keng Yap. New upper bounds in Klee’s measure problem. *SIAM J. Comput.*, 20(6):1034–1045, 1991. Preliminary version in FOCS 1988. doi:10.1137/0220065.
- 29 Virginia Vassilevska Williams. On some fine-grained questions in algorithms and complexity. In *Proceedings of the ICM*, volume 3, pages 3431–3472. World Scientific, 2018. URL: <https://people.csail.mit.edu/virgi/eccentri.pdf>.
- 30 Hakan Yildiz and Subhash Suri. Computing Klee’s measure of grounded boxes. *Algorithmica*, 71(2):307–329, 2015. Preliminary version in SoCG 2012. doi:10.1007/s00453-013-9797-9.

Meta-Diagrams for 2-Parameter Persistence

Nate Clause ✉

Ohio State University, Columbus, OH, USA

Tamal K. Dey ✉

Purdue University, West Lafayette, IN, USA

Facundo Mémoli ✉

Ohio State University, Columbus, OH, USA

Bei Wang ✉

University of Utah, Salt Lake City, UT, USA

Abstract

We first introduce the notion of meta-rank for a 2-parameter persistence module, an invariant that captures the information behind images of morphisms between 1D slices of the module. We then define the meta-diagram of a 2-parameter persistence module to be the Möbius inversion of the meta-rank, resulting in a function that takes values from signed 1-parameter persistence modules. We show that the meta-rank and meta-diagram contain information equivalent to the rank invariant and the signed barcode. This equivalence leads to computational benefits, as we introduce an algorithm for computing the meta-rank and meta-diagram of a 2-parameter module M indexed by a bifiltration of n simplices in $O(n^3)$ time. This implies an improvement upon the existing algorithm for computing the signed barcode, which has $O(n^4)$ time complexity. This also allows us to improve the existing upper bound on the number of rectangles in the rank decomposition of M from $O(n^4)$ to $O(n^3)$. In addition, we define notions of erosion distance between meta-ranks and between meta-diagrams, and show that under these distances, meta-ranks and meta-diagrams are stable with respect to the interleaving distance. Lastly, the meta-diagram can be visualized in an intuitive fashion as a persistence diagram of diagrams, which generalizes the well-understood persistence diagram in the 1-parameter setting.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Topology

Keywords and phrases Multiparameter persistence modules, persistent homology, Möbius inversion, barcodes, computational topology, topological data analysis

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.25

Related Version *Full Version:* <https://arxiv.org/abs/2303.08270> [11]

Funding *Nate Clause:* NC is partially supported by NSF CCF 1839356 and NSF DMS 1547357.

Tamal K. Dey: TD is partially supported by NSF CCF 2049010.

Facundo Mémoli: FM is partially supported by BSF 2020124, NSF CCF 1740761, NSF CCF 1839358, and NSF IIS 1901360.

Bei Wang: BW is partially supported by NSF IIS 2145499, NSF IIS 1910733, and DOE DE SC0021015.

1 Introduction

In the case of a 1-parameter persistence module, the persistence diagram (or barcode) captures its complete information up to isomorphism via a collection of intervals. The persistence diagram is represented as a multi-set of points in the plane, whose coordinates are the birth and death times of intervals, each of which encodes the lifetime of a topological feature. This compact representation of a persistence module enables its interpretability and facilitates



© Nate Clause, Tamal K. Dey, Facundo Mémoli, and Bei Wang;
licensed under Creative Commons License CC-BY 4.0

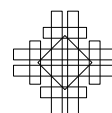
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 25; pp. 25:1–25:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



its visualization. When moving to the multiparameter setting, the situation becomes much more complex as a multiparameter persistence module may contain indecomposable pieces that are not entirely determined by intervals or do not admit a finite discrete description [9].

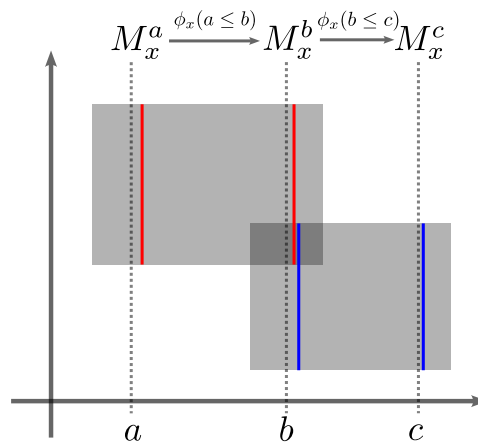
Such an increased complexity has led to the study of other invariants for multiparameter persistence modules. The first invariant is the *rank invariant* [9], which captures the information from the images of internal linear maps in a persistence module across all dimensions. Patel noticed that the persistence diagram in the 1-parameter setting is equivalent to the *Möbius inversion* [25] of the rank function [24]. He then defined the generalized persistence diagram as the Möbius inversion of a function defined on a subset of intervals of \mathbb{R} , denoted Dgm , with values in some abelian group.

The idea of Möbius inversion has been extended in many directions. Kim and Mémoli defined generalized persistence diagrams for modules on posets [12, 17]. Patel and McCleary extended Patel’s generalized persistence diagrams to work for persistence modules indexed over finite lattices [22]. Botnan et al. [6] implicitly studied the Möbius inversion of the rank function for 2-parameter modules, leading to a notion of a diagram with domain all rectangles in \mathbb{Z}^2 . Asashiba et al. used Möbius inversion on a finite 2D grid to define interval-decomposable approximations [1]. Morozov and Patel [23] defined a generalized persistence diagram in the 2-parameter setting via Möbius inversion of the birth-death function and provided an algorithm for its computation. Their algorithm has some similarity with ours: it utilizes the vineyards algorithm [13] to study a 2-parameter persistence module, by slicing it over 1D paths.

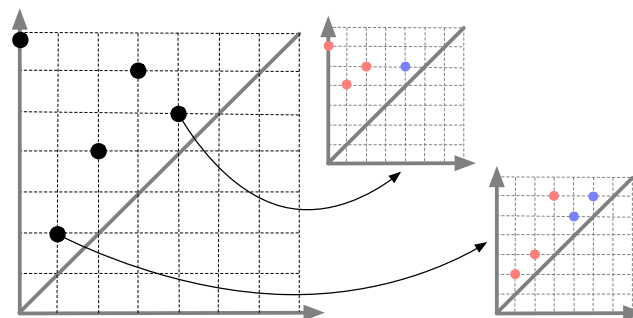
Our work also involves the idea of slicing a 2-parameter module. This idea of slicing appears in the fibered barcode [10, 20], which is equivalent to the rank function. To obtain insight into the structure of a 2-parameter persistence module M , Lesnick and Wright [20] explored a set of 1-parameter modules obtained via restricting M onto all possible lines of non-negative slope. Buchet and Escobar [8] showed that any 1-parameter persistence module with finite support could be found as a restriction of some indecomposable 2-parameter persistence module with finite support. Furthermore, Dey et al. [15] showed that certain zigzag (sub)modules of a 2-parameter module can be used to compute the generalized rank invariant, whose Möbius inversion is the generalized persistence diagram defined by Kim and Mémoli. Our work considers the images between slices of a 2-parameter module, which is related to the work by Bauer and Schmal [3].

In [7], Botnan et al. introduced the notion of *rank decomposition*, which is equivalent to the generalized persistence diagram formed by Möbius inversion of the rank function, under some additional conditions. Botnan et al. further demonstrated that the process of converting a module to a rank decomposition is stable with respect to the matching distance [18]. Additionally, they introduced a visualization of this rank decomposition via a *signed barcode*, which highlights the diagonals of rectangles appearing in the rank decomposition, along with their multiplicity. They visualized the value of the signed barcode with a 2-parameter persistence module generated by clustering a point cloud with a scale and a density parameter.

Unlike the previous results that perform Möbius inversion over a higher-dimensional poset such as \mathbb{Z}^2 , our work involves Möbius inversion over a finite subcollection of intervals of \mathbb{R} , which leads to a simpler inversion formula. In this work, we introduce the notion of *meta-rank* for a 2-parameter persistence module, which is a map from Dgm to isomorphism classes of persistence modules. Instead of looking at images of linear maps between vector spaces (as with the usual rank invariant), the meta-rank considers images of the maps between 1-parameter persistence modules formed by slicing a 2-parameter persistence module



■ **Figure 1** Slicing a 2-parameter module M along vertical lines yields 1-parameter modules, such as M_x^a , M_x^b , and M_x^c . There are morphisms between these 1-parameter modules induced by the internal morphisms of M , and the meta-rank captures the information about these morphisms. For example, if M is defined as the direct sum of the two interval modules given by the two shaded rectangles, then the meta-rank of M on $[a, b]$ is the image of $\phi_x(a \leq b)$, which has a barcode consisting of the red interval. The meta-rank of M on $[b, c]$ has a barcode consisting of the blue interval, and the meta-rank of M on $[a, c]$ is 0, as $\phi_x(a \leq c) = \phi_x(b \leq c) \circ \phi_x(a \leq b) = 0$.



■ **Figure 2** A meta-diagram viewed as a persistence diagram of signed diagrams (red and blue mean positive and negative signs respectively).

along vertical and horizontal lines, see Figure 1. We then define the meta-diagram as the Möbius inversion of the meta-rank, giving a map from Dgm to isomorphism classes of signed persistence modules. This contrasts Botnan et al.’s approach [7] of using Möbius inversion in 2D, as our Möbius inversion formula over Dgm is simpler and consists of fewer terms.

Contributions. The meta-rank and meta-diagram turn out to contain information equivalent to the rank invariant (Proposition 12) and signed barcode (Proposition 27) respectively. Therefore, both meta-rank and meta-diagram can be regarded as these known invariants seen from a different perspective. However, this different viewpoint brings forth several advantages as listed below that make the meta-rank and meta-diagram stand out on their own right:

1. The meta-rank and meta-diagram of a 2-parameter persistence module M induced by a bifiltration of a simplicial complex with n simplices can be computed in $O(n^3)$ time.
2. This immediately implies an improvement of the $O(n^4)$ algorithm of Botnan et al. [7] for computing the signed barcodes.

3. The $O(n^3)$ time algorithm for computing meta-rank and meta-diagram also implicitly improves the bound on the number of signed bars in the rank decomposition of M to $O(n^3)$ from the current known bound of $O(n^4)$. This addresses an open question whether the size of the signed barcode is bounded tightly by the number of rectangles or not.
4. The meta-diagram can be viewed as a persistence diagram of signed diagrams as illustrated in Figure 2. Such an intuitive visualization generalizes the classic persistence diagram – a known technique in topological data analysis (TDA) – to summarize persistent homology.
5. The meta-diagram also generalizes the concept of a sliced barcode well-known in TDA [20]. It ensembles sliced bars on a set of lines, but not forgetting the maps between slices induced by the module M being sliced.

For omitted proofs and further details, see the full version [11].

2 Preliminaries

We regard a poset (P, \leq) as a category, with objects the elements $p \in P$, and a unique morphism $p \rightarrow q$ if and only if $p \leq q$; this is referred to as the *poset category* for (P, \leq) . When it is clear from the context, we will denote the poset category by P .

Fix a field \mathbf{k} , and assume all vector spaces have coefficients in \mathbf{k} throughout this paper. Let \mathbf{vec} denote the category of finite-dimensional vector spaces with linear maps between them. A *persistence module*, or *module* for short, is a functor $M : P \rightarrow \mathbf{vec}$. For any $p \in P$, we denote the vector space $M_p := M(p)$, and for any $p \leq q \in P$, we denote the linear map $\varphi_M(p \leq q) := M(p \leq q)$. When M is apparent, we drop the subscript from φ_M . We call P the *indexing poset* for M . We focus on the cases when the indexing poset is \mathbb{R} or \mathbb{R}^2 , equipped with the usual order and product order, respectively. Definitions and statements we make follow analogously when the indexing poset is \mathbb{Z} or \mathbb{Z}^2 , which we will cover briefly in Section 5. If the indexing poset for M is $P \subseteq \mathbb{R}$, then M is a *1-parameter (or 1D) persistence module*. If the indexing poset for M is $P \subseteq \mathbb{R}^2$, with P not totally-ordered, then M is a *2-parameter (or 2D) persistence module*, or a *bimodule* for short.

Following [21], we require that persistence modules be *constructible*:

► **Definition 1.** A module $M : \mathbb{R} \rightarrow \mathbf{vec}$ is constructible if there exists a finite set $S := \{s_1 < \dots < s_n\} \subset \mathbb{R}$ such that:

- For $a < s_1$, $M(a) = 0$;
- For $s_i \leq a \leq b < s_{i+1}$, $\varphi_M(a \leq b)$ is an isomorphism;
- For $s_n \leq a \leq b$, $\varphi_M(a \leq b)$ is an isomorphism.

Similarly, a bimodule $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$ is constructible if there exists a finite set $S := \{s_1 < \dots < s_n\} \subset \mathbb{R}$ such that:

- If $x < s_1$ or $y < s_1$, then $M((x, y)) = 0$,
- For $s_i \leq x_1 \leq x_2 < s_{i+1}$ and $s_j \leq y_1 \leq y_2 < s_{j+1}$, $\varphi_M((x_1, y_1) \leq (x_2, y_2))$ is an isomorphism,
- If $x_1 \geq s_n$ or $y_1 \geq s_n$ and $(x_1, y_1) \leq (x_2, y_2)$, then $\varphi_M((x_1, y_1) \leq (x_2, y_2))$ is an isomorphism.

In either case, such a module is S -constructible.

If a module is S -constructible, unless otherwise stated, assume $S = \{s_1 < \dots < s_n\}$. If M is S -constructible, then M is S' -constructible for any $S' \supseteq S$. For the rest of the paper, we assume any given persistence module is constructible.

Of particular importance in the study of 1- and 2-parameter persistence modules are the notions of interval modules and interval decomposable modules. We state the definitions:

► **Definition 2.** For a poset (P, \leq) , an interval of P is a non-empty subset $I \subset P$ such that:

- (convexity) If $p, r \in I$ and $q \in P$ with $p \leq q \leq r$, then $q \in I$.
- (connectivity) For any $p, q \in I$, there is a sequence $p = r_0, r_1, \dots, r_n = q$ of elements of I , where for all $0 \leq i \leq n - 1$, either $r_i \geq r_{i+1}$ or $r_i \leq r_{i+1}$.

We denote the collection of all intervals of P as $\mathbf{Int}(P)$.

For $I \in \mathbf{Int}(P)$, the interval module \mathbf{k}^I is the persistence module indexed over P , with:

$$\mathbf{k}_p^I = \begin{cases} \mathbf{k} & \text{if } p \in I \\ 0 & \text{otherwise} \end{cases}, \quad \varphi_{\mathbf{k}^I}(p \leq q) = \begin{cases} \text{id}_{\mathbf{k}} & \text{if } p \leq q \in I \\ 0 & \text{otherwise} \end{cases}.$$

Given any $M, N : P \rightarrow \mathbf{vec}$, the direct sum $M \oplus N$ is defined point-wise at each $p \in P$. We say a nontrivial $M : P \rightarrow \mathbf{vec}$ is decomposable if M is isomorphic to $N_1 \oplus N_2$ for some non-trivial $N_1, N_2 : P \rightarrow \mathbf{vec}$, which we denote by $M \cong N_1 \oplus N_2$. Otherwise, M is indecomposable. Interval modules are indecomposable [5].

A persistence module $M : P \rightarrow \mathbf{vec}$ is interval decomposable if it is isomorphic to a direct sum of interval modules. That is, if there is a multiset of intervals $\text{barc}(M)$, such that:

$$M \cong \bigoplus_{I \in \text{barc}(M)} \mathbf{k}^I$$

If this multiset exists, we call it the barcode of M . If it exists, $\text{barc}(M)$ is well-defined as a result of the Azumaya-Krull-Remak-Schmidt theorem [2]. Thus, in the case where M is interval decomposable, $\text{barc}(M)$ is a complete descriptor of the isomorphism type of M .

Of particular importance in this work are right-open rectangles, which are intervals $R \subset \mathbb{R}^2$ of the form $R = [a_1, b_1) \times [a_2, b_2)$. If M can be decomposed as a direct sum of interval modules \mathbf{k}^R with R a right-open rectangle, then we say M is rectangle decomposable.

1-parameter persistence modules are particularly nice, as they are always interval decomposable [14]. As a result, the barcode is a complete invariant for 1-parameter persistence modules. On the other hand, bimodules do not necessarily decompose in this way. In fact, there is no complete discrete descriptor for bimodules [9].

A number of invariants have been proposed to study bimodules. One of the first and the most notable invariant is the rank invariant [9] recalled in Definition 3.

► **Definition 3** ([9]). For P a poset, define $\mathbf{D}(P) := \{(a, b) \in P \times P \mid a \leq b\}$. For $M : P \rightarrow \mathbf{vec}$, the rank invariant of M , $\text{rank}_M : \mathbf{D}(P) \rightarrow \mathbb{Z}_{\geq 0}$, is defined point-wisely as:

$$\text{rank}_M(a, b) := \text{rank}(\varphi_M(a \leq b))$$

For a bimodule, the rank invariant is inherently a 4D object, making it difficult to visualize directly. RIVET [20] visualizes the rank invariant indirectly through the fibered barcode. In [7], Botnan et al. defined the signed barcode based on the notion of a rank decomposition:

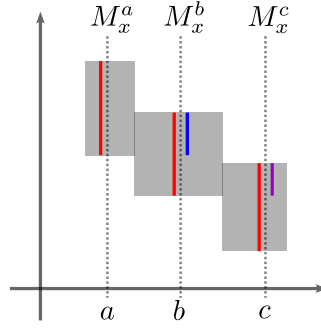
► **Definition 4** ([7]). Let $M : \mathbb{R}^n \rightarrow \mathbf{vec}$ be a persistence module with rank function rank_M . Suppose \mathcal{R}, \mathcal{S} are multisets of intervals from \mathbb{R}^n . Define $\mathbf{k}_{\mathcal{R}} := \bigoplus_{I \in \mathcal{R}} \mathbf{k}^I$, and similarly $\mathbf{k}_{\mathcal{S}}$. Then $(\mathcal{R}, \mathcal{S})$ is a rank decomposition for rank_M if as integral functions:

$$\text{rank}_M = \text{rank}_{\mathcal{R}} - \text{rank}_{\mathcal{S}}.$$

If \mathcal{R}, \mathcal{S} consist of right-open rectangles, then the pair is a rank decomposition by rectangles. We have:

► **Theorem 5** ([7], Theorem 3.3). Every finitely presented $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$ admits a unique minimal rank decomposition by rectangles.

Here minimality comes in the sense that $\mathcal{R} \cap \mathcal{S} = \emptyset$. The signed barcode then visualizes the rank function in \mathbb{R}^2 by showing the diagonals of the rectangles in \mathcal{R} and \mathcal{S} .



■ **Figure 3** An illustration of M and its barcode for some values of \mathbf{mrk}_M in Example 9.

3 Meta-Rank

In this section, we introduce the *meta-rank*. While the rank invariant captures the information of images between pairs of vector spaces in a persistence module, the meta-rank captures the information of images between two 1-parameter persistence modules obtained via slicing a bimodule. We begin with some preliminary definitions:

► **Definition 6.** Let $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$ be a bimodule. For $s \in \mathbb{R}$, define the vertical slice $M_x^s : \mathbb{R} \rightarrow \mathbf{vec}$ point-wise as $M_x^s(a) := M(s, a)$, and with morphisms from a to b as $\varphi_x^s(a \leq b) := \varphi((s, a) \leq (s, b))$. Analogously, define the horizontal slice $M_y^s : \mathbb{R} \rightarrow \mathbf{vec}$ by setting $M_y^s(a) := M(a, s)$ and $\varphi_y^s(a \leq b) := \varphi((a, s) \leq (b, s))$ for all $a \leq b \in \mathbb{R}$.

Define a morphism of 1-parameter persistence modules $\phi_x(s \leq t) : M_x^s \rightarrow M_x^t$ for $s \leq t \in \mathbb{R}$ by $\phi_x(s \leq t)(a) := \varphi((s, a) \leq (t, a))$. Analogously, define $\phi_y(s \leq t) : M_y^s \rightarrow M_y^t$ for $s \leq t \in \mathbb{R}$ by $\phi_y(s \leq t)(a) := \varphi((a, s) \leq (a, t))$.

Denote by \mathbf{Pvec} the isomorphism classes of persistence modules over \mathbb{R} . Each element of \mathbf{Pvec} can be uniquely represented by its barcode, which is what we do in practice. We recall the definition of \mathbf{Dgm} from [24], which serves as the domain for the meta-rank:

► **Definition 7** ([24]). Define \mathbf{Dgm} as the poset of all half-open intervals $[p, q) \subset \mathbb{R}$ for $p < q$, and all half-infinite intervals $[p, \infty) \subset \mathbb{R}$. The poset relation is inclusion.

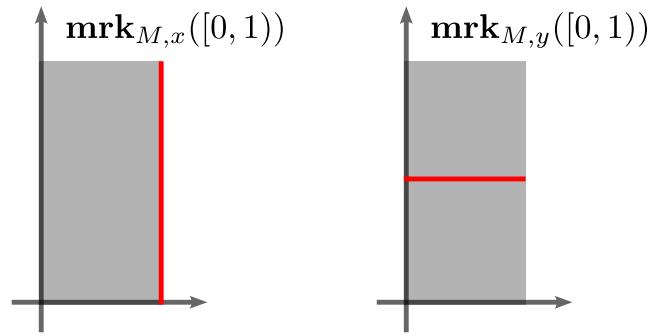
► **Definition 8.** Suppose $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$ is S -constructible. Define the horizontal meta-rank $\mathbf{mrk}_{M,x} : \mathbf{Dgm} \rightarrow \mathbf{Pvec}$ as follows:

- For $I = [s, s_i)$ with $s_i \in S$, $\mathbf{mrk}_{M,x}(I) := [\mathrm{im}(\phi_x(s \leq s_i - \delta))]$, for some $\delta > 0$ such that $s_i - \delta \geq s$ and $s_i - \delta \geq s_{i-1}$.
- For $I = [s, \infty)$, $\mathbf{mrk}_{M,x}(I) := [\mathrm{im}(\phi_x(s \leq s_n))]$.
- For all other $I = [s, t)$, $\mathbf{mrk}_{M,x}(I) := [\mathrm{im}(\phi_x(s \leq t))]$.

Analogously, define the vertical meta-rank, $\mathbf{mrk}_{M,y} : \mathbf{Dgm} \rightarrow \mathbf{Pvec}$ by replacing each instance of x above with y .

The results in this paper are stated in terms of the horizontal meta-rank, but hold analogously for the vertical meta-rank. To simplify notation, we henceforth denote $\mathbf{mrk}_{M,x}$ as \mathbf{mrk}_M . When there is no confusion, we drop the subscript from \mathbf{mrk}_M .

► **Example 9.** As illustrated in Figure 3, let I be the connected gray interval and define the bimodule $M := \mathbf{k}^I$. The barcodes for the 1-parameter modules M_x^a, M_x^b , and M_x^c are shown in red next to their corresponding vertical slices. The barcode for $\mathbf{mrk}_M([a, b))$ consists of the blue interval, which is the overlap of the bars in M_x^a and M_x^b , $\mathrm{barc}(M_x^a) \cap \mathrm{barc}(M_x^b)$.



■ **Figure 4** An illustration of M , depicting $\mathbf{mrk}_{M,x}([0, 1]) \neq \mathbf{mrk}_{M,y}([0, 1])$.

Similarly, $\mathbf{mrk}_M([b, c])$ has a barcode consisting of the purple interval, which is the overlap of the bars in M_x^b and M_x^c . As the bars in the barcodes for M_x^a and M_x^c have no overlap, $\text{im}(\phi_x(a \leq c)) = 0$, therefore $\mathbf{mrk}_M([a, c]) = 0$.

► **Remark 10.** In general, $\mathbf{mrk}_x \neq \mathbf{mrk}_y$. For example, consider the right-open rectangle R with the lower-left corner the origin, and the upper right corner $(1, 2)$, as in Figure 4. Let $M := \mathbf{k}^R$. As illustrated, $\mathbf{mrk}_{M,x}([0, 1]) = [0, 2] \neq [0, 1] = \mathbf{mrk}_{M,y}([0, 1])$.

The following Proposition 11 allows us to compute the meta-rank of a bimodule via the meta-ranks of its indecomposable summands:

► **Proposition 11.** For $M, N : \mathbb{R}^2 \rightarrow \mathbf{vec}$, we have:

$$\mathbf{mrk}_M \oplus \mathbf{mrk}_N = \mathbf{mrk}_{M \oplus N}$$

where $\mathbf{mrk}_M \oplus \mathbf{mrk}_N : \text{Dgm} \rightarrow \mathbf{Pvec}$ is defined as:

$$(\mathbf{mrk}_M \oplus \mathbf{mrk}_N)([s, t]) := [\mathbf{mrk}_M([s, t]) \oplus \mathbf{mrk}_N([s, t])].$$

For a finite $S \subseteq \mathbb{R}$, let $\bar{S} := S \cup \{\infty\}$. Define $\bar{S}_> : \mathbb{R} \cup \{\infty\} \rightarrow \bar{S}$ as $\bar{S}_>(t) := \min\{s \in \bar{S} \mid s > t\}$. For $M \in \mathbf{Pvec}$, $[b, d] \in \text{Dgm}$, let $\#[b, d] \in M$ denote the multiplicity of $[b, d] \in \text{barc}(M)$. The rank invariant and the meta-rank contain equivalent information:

► **Proposition 12.** For $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$, one can compute rank_M from \mathbf{mrk}_M and one can compute \mathbf{mrk}_M from rank_M . In particular, given $(s, y) \leq (t, y') \in \mathbb{R}^2$,

$$\text{rank}_M((s, y), (t, y')) = \#[b_i, d_i] \in \mathbf{mrk}_M([s, \bar{S}_>(t)]) \text{ s.t. } b_i \leq y \leq y' < d_i.$$

That is, the rank is the number of intervals in $\text{barc}(\mathbf{mrk}_M([s, \bar{S}_>(t)]))$ containing $[y, y']$.

The reason for needing $\bar{S}_>(t)$ for the right endpoint is that if $t \in S$, $\mathbf{mrk}_M([s, t])$ does not capture the information of the image of $\phi_x(s \leq t)$, only the image of $\phi_x(s \leq t - \delta)$.

Finally, we discuss the stability of the meta-rank. The meta-rank is stable with respect to a notion of erosion distance, based on that of Patel [24]. We introduce truncated barcode:

► **Definition 13.** For $\epsilon \geq 0$, and $I = [s, t] \in \text{Dgm}$, define $I[\epsilon :] := [s + \epsilon, t)$. For $M : \mathbb{R} \rightarrow \mathbf{vec}$ define: $\text{barc}_\epsilon(M) := \{I[\epsilon :] \mid I \in \text{barc}(M)\}$. If $I = [s, t] \in \text{barc}(M)$ has $t - s \leq \epsilon$, then I has no corresponding interval in $\text{barc}_\epsilon(M)$.

► **Definition 14.** For $M, N : \mathbb{R} \rightarrow \mathbf{vec}$, we say $M \preceq_\epsilon N$ if there exists an injective function on barcodes $\iota : \text{barc}_\epsilon(M) \hookrightarrow \text{barc}(N)$ such that for all $J \in \text{barc}_\epsilon(M)$, $J \subseteq \iota(J)$.

For $\epsilon \geq 0$, $M \in \mathbf{Pvec}$, let M^ϵ refer to the ϵ -shift of M [19], with $M^\epsilon(a) := M(a + \epsilon)$ and $\varphi_{M^\epsilon}(a \leq b) := \varphi_M(a + \epsilon \leq b + \epsilon)$. For $I = [s, t] \in \text{Dgm}$ and $a, b \in \mathbb{R}$, let $I_a^b := [s + a, t + b]$, with the convention $\infty + b := \infty$ for any $b \in \mathbb{R}$. We now define the erosion distance:

► **Definition 15.** Let $M, N : \mathbb{R}^2 \rightarrow \mathbf{vec}$. Define the erosion distance as follows:

$$d_E(\mathbf{mrk}_M, \mathbf{mrk}_N) := \inf\{\epsilon > 0 \mid \forall I \in \text{Dgm}, \mathbf{mrk}_M(I_{-\epsilon}^\epsilon) \preceq_{2\epsilon} \mathbf{mrk}_N(I) \text{ and} \\ \mathbf{mrk}_N(I_{-\epsilon}^\epsilon) \preceq_{2\epsilon} \mathbf{mrk}_M(I)\}$$

if the set we are infimizing over is empty, we set $d_E(\mathbf{mrk}_M, \mathbf{mrk}_N) := \infty$.

► **Proposition 16.** d_E as defined in Definition 15 is an extended pseudometric on the collection of meta-ranks of constructible bimodules $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$.

We compare bimodules M and N using the multiparameter interleaving distance [19]. The ϵ -shift and the truncation of the barcode in Definition 13 are necessary for stability, due to the interleaving distance being based on diagonal shifts of bimodules, whereas the meta-rank is based on horizontal maps instead of diagonal ones. We have the following:

► **Theorem 17.** For constructible $M, N : \mathbb{R}^2 \rightarrow \mathbf{vec}$, we have:

$$d_E(\mathbf{mrk}_M, \mathbf{mrk}_N) \leq d_I(M, N)$$

4 Meta-Diagram

We use the Möbius inversion formula from Patel [24] on the meta-rank function to get a *meta-diagram*. This formula involves negative signs, so we need a notion of signed persistence modules. Our ideas are inspired by the work of Betthausen et al. [4], where we consider breaking a function into positive and negative parts.

► **Definition 18.** A signed 1-parameter persistence module is an ordered pair (M, N) , where $M, N : \mathbb{Z} \rightarrow \mathbf{vec}$ are 1-parameter persistence modules. M is the positively signed module, and N is the negatively signed module.

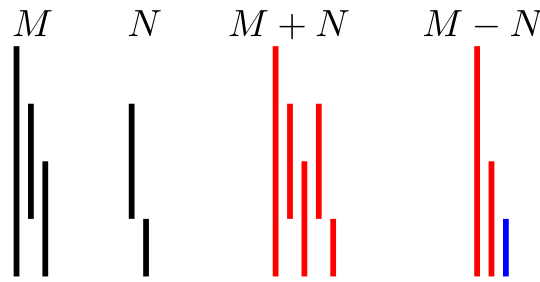
► **Definition 19.** View \mathbf{Pvec} as a commutative monoid with operation \oplus given by $[M] \oplus [N] := [M \oplus N]$, and identity element $[0]$. Define \mathbf{SPvec} to be the Grothendieck group of \mathbf{Pvec} .

Each element of \mathbf{SPvec} is an isomorphism class of ordered pairs $[[M^+], [M^-]]$. From the completeness of barcodes for 1-parameter persistence modules, we assume without loss of generality that each element M^+, M^- is given by $* := \bigoplus_{I \in \text{barc}(*)} \mathbf{k}^I$ and drop the internal equivalence class notation to write an element of \mathbf{SPvec} as $[(M^+, M^-)]$. Proposition 20 allows us to make a canonical choice of representative for each element of \mathbf{SPvec} :

► **Proposition 20.** Let $A \in \mathbf{SPvec}$. Then there is a unique representative $A = [(M^+, M^-)]$ with $\text{barc}(M^+) \cap \text{barc}(M^-) = \emptyset$.

As a result of Proposition 20, when convenient, we represent an element of \mathbf{SPvec} uniquely by the sum of barcodes of this special representative, as in the following example:

► **Example 21.** Consider $[(N^+, N^-)] \in \mathbf{SPvec}$ where $\text{barc}(N^+) = \{[0, 4], [1, 3], [2, 4]\}$ and $\text{barc}(N^-) = \{[1, 3], [3, 4]\}$. By Proposition 20, $[(N^+, N^-)]$ is uniquely represented by $[(M^+, M^-)]$ with $\text{barc}(M^+) = \{[0, 4], [2, 4]\}$ and $\text{barc}(M^-) = \{[3, 4]\}$. In practice, we will denote this element of \mathbf{SPvec} as $[0, 4] + [2, 4] - [3, 4] \in \mathbf{SPvec}$. If $M, N \in \mathbf{Pvec}$, denote by $M + N$ the element $[(M \oplus N, 0)] \in \mathbf{SPvec}$, and denote by $M - N$ the element $[(M, N)] \in \mathbf{SPvec}$. For an illustration, see Figure 5.



■ **Figure 5** Illustration of the barcodes for $M, N \in \mathbf{Pvec}$ and $M + N, M - N \in \mathbf{SPvec}$. For $M + N$ and $M - N$, a red interval is positively signed and a blue interval is negatively signed.

With this notion of signed persistence module in hand, we now use a modified version of the Möbius inversion formula from [24] to define a meta-diagram:

► **Definition 22.** Let $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$ be S -constructible. Define the horizontal meta-diagram to be the function $\mathbf{mdgm}_M : \mathbf{Dgm} \rightarrow \mathbf{SPvec}$ via the Möbius inversion formula:

$$\begin{aligned} \mathbf{mdgm}_{M,x}([s_i, s_j]) &:= \mathbf{mrk}_{M,x}([s_i, s_j]) - \mathbf{mrk}_{M,x}([s_i, s_{j+1})) \\ &\quad + \mathbf{mrk}_{M,x}([s_{i-1}, s_{j+1})) - \mathbf{mrk}_{M,x}([s_{i-1}, s_j]) \\ \mathbf{mdgm}_{M,x}([s_i, \infty)) &:= \mathbf{mrk}_{M,x}([s_i, \infty)) - \mathbf{mrk}_{M,x}([s_{i-1}, \infty)) \end{aligned}$$

where s_0 is any value $s_0 < s_1$ and s_{n+1} is any value $s_{n+1} > s_n$. For any other $[s, t) \in \mathbf{Dgm}$, set $\mathbf{mdgm}_{M,x}([s, t)) := 0$. Define the vertical meta-diagram by replacing each instance of x above with y .

We henceforth let \mathbf{mdgm} refer to the horizontal meta-diagram of M , dropping the subscript when there is no confusion. The following Möbius inversion formula describes the relation between the meta-rank and meta-diagram. It is the direct analogue of [24, Theorem 4.1].

► **Proposition 23.** For $[s, t) \in \mathbf{Dgm}$, we have:

$$\mathbf{mrk}([s, t)) = \sum_{\substack{I \in \mathbf{Dgm} \\ I \supseteq [s, t)}} \mathbf{mdgm}(I)$$

► **Proposition 24.** For $M, N : \mathbb{R}^2 \rightarrow \mathbf{vec}$, we have:

$$\mathbf{mdgm}_M \oplus \mathbf{mdgm}_N = \mathbf{mdgm}_{M \oplus N},$$

where $\mathbf{mdgm}_M \oplus \mathbf{mdgm}_N : \mathbf{Dgm} \rightarrow \mathbf{SPvec}$ is defined by

$$\begin{aligned} (\mathbf{mdgm}_M \oplus \mathbf{mdgm}_N)([s, t)) &:= [\mathbf{mdgm}_M([s, t))^+ \oplus \mathbf{mdgm}_N([s, t))^+, \\ &\quad \mathbf{mdgm}_M([s, t))^- \oplus \mathbf{mdgm}_N([s, t))^-]. \end{aligned}$$

Proposition 24 allows us to compute meta-diagrams straightforwardly if we have an indecomposable decomposition of a module. In particular, by Proposition 25, meta-diagrams are simply computable for rectangle decomposable modules.

► **Proposition 25.** Suppose $M = \mathbf{k}^R$ is an \mathbb{R}^2 -indexed interval module supported on the right-open rectangle R , with lower-left corner (s, t) and upper-right corner (s', t') . We have:

$$\mathbf{mdgm}_M([a, b)) = \begin{cases} [t, t') & \text{if } a = s \text{ and } b = s'; \\ 0 & \text{otherwise.} \end{cases}$$

► **Corollary 26.** *Let $M = \bigoplus_{R \in \text{barc}(M)} \mathbf{k}^R$ be rectangle decomposable. Then the interval $[t, t')$ appears in $\mathbf{mdgm}([s, s'])$ with multiplicity n if and only if the right-open rectangle with lower-left corner (s, t) and upper right corner (s', t') appears in $\text{barc}(M)$ with multiplicity n .*

4.1 Equivalence With Rank Decomposition via Rectangles

For $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$, the rank decomposition by rectangles contains the same information as the rank invariant, which by Proposition 12 contains the same information as the meta-rank. We now show one can directly go from the meta-diagram to the rank decomposition:

► **Proposition 27.** *Let $M : \mathbb{R}^2 \rightarrow \mathbf{vec}$ be constructible. Define as follows:*

$$\mathcal{R} := \bigcup_{I \in \text{Dgm}} \left(\bigcup_{[a,b] \in \mathbf{mdgm}_M(I)} I \times [a, b) \right),$$

$$\mathcal{S} := \bigcup_{I \in \text{Dgm}} \left(\bigcup_{-[a,b] \in \mathbf{mdgm}_M(I)} I \times [a, b) \right),$$

where all unions are the multiset union. Then $(\mathcal{R}, \mathcal{S})$ is a rank decomposition for M .

Proof. It suffices to show that for all $w_1 := (x_1, y_1) \leq w_2 := (x_2, y_2) \in \mathbb{R}^2$, $\text{rank}_M(w_1, w_2) = \text{rank}_{\mathbf{k}_{\mathcal{R}}}(w_1, w_2) - \text{rank}_{\mathbf{k}_{\mathcal{S}}}(w_1, w_2)$. Suppose $w_1 \leq w_2 \in \mathbb{R}^2$ as above. By Proposition 12,

$$\text{rank}_M(w_1, w_2) = \#[b_i, d_i] \in \mathbf{mrk}_M([x_1, x'_2]) \quad \text{s.t.} \quad b_i \leq y_1 \leq y_2 < d_i,$$

where for notational simplicity, $x'_2 := \bar{S}_>(x_2)$.

Now fix $[b, d]$ such that $b \leq y_1 \leq y_2 < d$. By Proposition 23, we have:

$$\begin{aligned} \#[b, d] \in \mathbf{mrk}_M([x_1, x'_2]) &= \#[b, d] \in \sum_{\substack{I \in \text{Dgm} \\ I \supseteq [x_1, x'_2]}} \mathbf{mdgm}_M(I) \\ &= \left(\#[b, d] \in \sum_{\substack{I \in \text{Dgm} \\ I \supseteq [x_1, x'_2]}} \mathbf{mdgm}_M^+(I) \right) - \left(\#[b, d] \in \sum_{\substack{I \in \text{Dgm} \\ I \supseteq [x_1, x'_2]}} \mathbf{mdgm}_M^-(I) \right) \end{aligned}$$

By Proposition 25 and Corollary 26, the term $\#[b, d] \in \sum_{\substack{I \in \text{Dgm} \\ I \supseteq [x_1, x'_2]}} \mathbf{mdgm}^+(I)$ is the number of

times $I \times [b, d]$ appears in \mathcal{R} across all $I \supseteq [x_1, x'_2]$, and the term $\#[b, d] \in \sum_{\substack{I \in \text{Dgm} \\ I \supseteq [x_1, x'_2]}} \mathbf{mdgm}^-(I)$

is the number of times $I \times [b, d]$ appears in \mathcal{S} across all $I \supseteq [x_1, x'_2]$.

Thus, we see that $\text{rank}_M(w_1, w_2)$ is equal to the number of rectangles in \mathcal{R} containing w_1 and w_2 minus the number of rectangles in \mathcal{S} containing w_1 and w_2 . From the definition of rectangle module and the fact that rank commutes with direct sums, the first term is $\text{rank}(\mathbf{k}_{\mathcal{R}})(w_1, w_2)$ and the second term is $\text{rank}(\mathbf{k}_{\mathcal{S}})(w_1, w_2)$, and so we get:

$$\text{rank}_M(w_1, w_2) = \text{rank}_{\mathbf{k}_{\mathcal{R}}}(w_1, w_2) - \text{rank}_{\mathbf{k}_{\mathcal{S}}}(w_1, w_2) \quad \blacktriangleleft$$

4.2 Stability of Meta-Diagrams

We now show a stability result for meta-diagrams. We need to modify the notion of erosion distance to do so, as meta-diagrams have negatively signed parts. We proceed by adding the positive part of one meta-diagram to the negative part of the other. This idea stems from Betthausen et al.'s work [4], and was also used in the stability of rank decompositions in [7].

► **Definition 28.** For $M, N : \mathbb{R}^2 \rightarrow \mathbf{vec}$, define $\text{PN}(M, N) : \text{Dgm} \rightarrow \mathbf{vec}$ as

$$\text{PN}(M, N)([s, t]) := \mathbf{mdgm}_M^+([s, t]) + \mathbf{mdgm}_N^-([s, t])$$

$\text{PN}(M, N)$ is a non-negatively signed 1-parameter persistence module for all $[s, t] \in \text{Dgm}$, allowing us to make use of the previous notion of \preceq_ϵ (Definition 14) to define an erosion distance for meta-diagrams. Unlike meta-ranks which have a continuous support, a meta-diagram is only supported on $(\bar{S})^2$ for some finite $S \subset \mathbb{R}$. As a result, we first modify the notion of erosion distance to fit the discrete setting.

Define maps $\bar{S}_\geq, \bar{S}_\leq : \mathbb{R} \cup \{\infty\} \rightarrow \bar{S}$ by $\bar{S}_\geq(x) := \min\{s \in \bar{S} \mid x \geq s\}$ and $\bar{S}_\leq(x) := \max\{s \in \bar{S} \mid x \leq s\}$, or some value less than s_1 if this set is empty. We say S is *evenly-spaced* if there exists $c \in \mathbb{R}$ such that $s_{i+1} - s_i = c$ for all $1 \leq i \leq n - 1$. In the following, fix an evenly-spaced finite $S \subset \mathbb{R}$.

► **Definition 29.** For S -constructible $M, N : \mathbb{R}^2 \rightarrow \mathbf{vec}$, define the erosion distance:

$$\begin{aligned} d_E^S(\mathbf{mdgm}_M, \mathbf{mdgm}_N) &:= \inf\{\epsilon \geq 0 \mid \forall s \leq t \in \bar{S}, \\ &\quad \text{PN}(M, N)([\bar{S}_\leq(s - \epsilon), \bar{S}_\geq(s + \epsilon)]^\epsilon) \preceq_{2\epsilon} \text{PN}(N, M)([s, t]) \text{ and} \\ &\quad \text{PN}(N, M)([\bar{S}_\leq(s - \epsilon), \bar{S}_\geq(s + \epsilon)]^\epsilon) \preceq_{2\epsilon} \text{PN}(M, N)([s, t])\} \end{aligned}$$

We have the following stability result for meta-diagrams,

► **Theorem 30.** For S -constructible $M, N : \mathbb{R}^2 \rightarrow \mathbf{vec}$, with S evenly-spaced, we have

$$d_E^S(\mathbf{mdgm}_M, \mathbf{mdgm}_N) \leq d_I(M, N).$$

For a stability result when S is not evenly-spaced, see the full version [11].

5 Algorithms

In this section, we provide algorithms for computing meta-ranks and meta-diagrams. The input to these algorithms is a simplex-wise bifiltration:

► **Definition 31.** Let $n \in \mathbb{Z}$, and $[n]$ denote the poset $\{1, \dots, n\}$ with the usual order. Let K be a simplicial complex, and $\text{sub}(K)$ denote all subsets of K which are themselves simplicial complexes. A filtration is a function $F : [n] \rightarrow \text{sub}(K)$ such that for $a \leq b$, $F(a) \subseteq F(b)$. We say a filtration is simplex-wise if for all $1 \leq a \leq n - 1$, either $F(a + 1) = F(a)$ or $F(a + 1) = F(a) \cup \{\sigma\}$ for some $\sigma \in K \setminus F(a)$. In the latter case, we denote this with $F(a) \xrightarrow{\sigma} F(a + 1)$. We say $\sigma \in \text{sub}(K)$ arrives at a if $\sigma \in F(a)$ and $\sigma \notin F(a - 1)$.

Define $P_n := [n] \times [n]$ equipped with the product order. A bifiltration is a function $F : P_n \rightarrow \text{sub}(K)$. We say a bifiltration is simplex-wise if for all $(a, b) \in P_n$, for $(x, y) = (a + 1, b)$ or $(a, b + 1)$, if $(x, y) \in P_n$, then either $F((x, y)) = F((a, b))$, or $F((a, b)) \xrightarrow{\sigma} F((x, y))$ for some $\sigma \notin F((a, b))$.

Applying homology to a bifiltration yields a bimodule defined on P_n . Our theoretical background in previous sections focused on the case of bimodules defined over \mathbb{R}^2 . The same ideas and major results follow similarly for a module defined over P_n . We quickly highlight the differences in definitions when working with modules defined on P_n . The following definitions are re-phrasings of the horizontal meta-rank and horizontal meta-diagram for modules indexed over P_n , but as before, the statements are directly analogous in the vertical setting. Let $\mathbf{Int}([n])$ refer to all intervals of $[n]$, which consists of $\{[a, b] \mid a \leq b, a, b \in [n]\}$.

► **Definition 32.** For $M : P_n \rightarrow \mathbf{vec}$, define the meta-rank, $\mathbf{mrk}_M : \mathbf{Int}([n]) \rightarrow \mathbf{Pvec}$ by

$$\mathbf{mrk}_M([s, t]) := [\mathrm{im}(\phi_x(s \leq t))]$$

► **Definition 33.** For $M : P_n \rightarrow \mathbf{vec}$, define the meta-diagram, $\mathbf{mdgm}_M : \mathbf{Int}([n]) \rightarrow \mathbf{SPvec}$ as follows: if $1 < s \leq t < n$, define:

$$\begin{aligned} \mathbf{mdgm}_M([s, t]) &:= \mathbf{mrk}_M([s, t]) - \mathbf{mrk}_M([s, t+1]) \\ &\quad + \mathbf{mrk}_M([s-1, t+1]) - \mathbf{mrk}_M([s-1, t]), \\ \mathbf{mdgm}_M([s, n]) &:= \mathbf{mrk}_M([s, n]) - \mathbf{mrk}_M([s-1, n]), \\ \mathbf{mdgm}_M([1, t]) &:= \mathbf{mrk}_M([1, t]) - \mathbf{mrk}_M([1, t+1]), \text{ and} \\ \mathbf{mdgm}_M([1, n]) &:= \mathbf{mrk}_M([1, n]). \end{aligned}$$

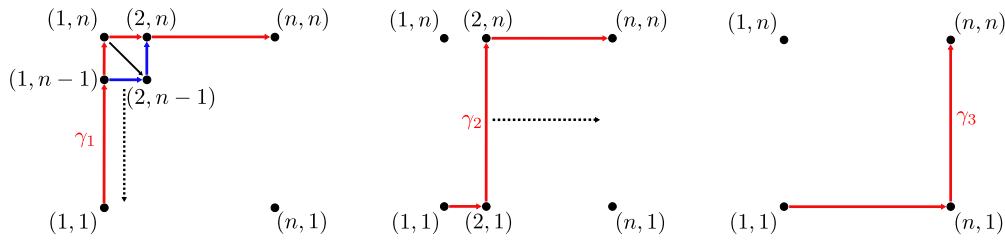
5.1 Overview of the Algorithm

Henceforth, assume $F : P_n \rightarrow \mathbf{sub}(K)$ is a simplex-wise bifiltration, and M is the result of applying homology to F . Our algorithm to compute the meta-rank relies on the vineyards algorithm from [13]. The algorithm starts with F as the input. Define γ_1 to be the path in P going from $(1, 1) \rightarrow (1, n) \rightarrow (n, n)$, i.e. the path along the top-left boundary of P . We compute the $D = RU$ decomposition for the interval decomposition of the persistence module given by the 1-parameter filtration found by slicing F over γ_1 , which we denote F_{γ_1} . This decomposition gives us all the persistence intervals and persistence pairs (σ_i, σ_j) and unpaired simplices corresponding to each interval, the former corresponding to a finite interval and the latter an infinite interval. To simplify notation, for every unpaired simplex corresponding to an infinite interval, we pair it with an implicit simplex arriving in an extended F at $(n+1, n+1)$. We store the persistence intervals in an ordered list, which we denote `intervals`. All intervals in `intervals` restricted to $[1, n]$ constitute together the 1-parameter persistence module M_x^1 , which is precisely $\mathbf{mrk}_M([1, 1])$. We then store $\mathbf{mrk}_M([1, 1])$ as a list, with the same ordering as `intervals`, leaving an empty placeholder whenever an interval does not intersect $[1, n]$.

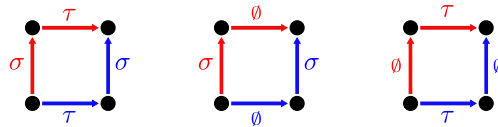
We sweep γ through P , over one square at a time, going down through the first column, until we reach γ_2 , the path $(1, 1) \rightarrow (2, 1) \rightarrow (2, n) \rightarrow (n, n)$. From there, we repeat the process column-by-column until we reach γ_n , the path $(1, 1) \rightarrow (n, 1) \rightarrow (n, n)$; see Figure 6.

After each change of a single vertex in our intermediary paths γ stemming from swapping the upper-left boundary of a single square to the lower-right one, the resulting filtration F_γ either remains the same, or changes in one of the ways illustrated in Figure 7.

After passing through each square, we update each interval in `intervals` in-place. If F_γ remains the same, then there is no change to `intervals`. If F_γ changes by altering the arrival time of a single simplex, then the pairings do not change, and the interval corresponding to the shifted simplex either extends by one or shrinks by one. If a transposition occurs, see Figure 7 (left), then we use the transposition update process from the vineyards algorithm.



■ **Figure 6** We start with γ_1 on the left, and then push γ_1 through the square to track along the lower-right corner of the square (in blue). We repeat this process, descending down each square in the first column until we reach γ_2 (middle). Then we repeat this process column-by-column until we've reached γ_n (right).



■ **Figure 7** Three possible ways in which F_γ can change via being pushed through a one-by-one square. In our algorithm, γ always starts along the red path, then shifts to the blue path.

If we start at γ_1 , then when we reach F_{γ_2} , we can restrict each interval in `intervals` to $[2, n + 1]$ and shift it back down one, and this corresponds to $\mathbf{mrk}_M([2, 2])$, which we store using the same rules as we did with $\mathbf{mrk}_M([1, 1])$.

Since we are storing all intervals in meta-ranks in this ordered fashion, we can take any interval in $\mathbf{mrk}_M([2, 2])$, and see where it came from in $\mathbf{mrk}_M([1, 1])$, which would be the interval stored at the same index in both lists. By taking the intersection, we get the corresponding interval which we put into this location in the list $\mathbf{mrk}_M([1, 2])$. We repeat the process of modifying γ one vertex at a time to get the paths γ_i from $(1, 1) \rightarrow (i, 1) \rightarrow (i, n) \rightarrow (n, n)$ as above, updating `intervals` and getting $\mathbf{mrk}_M([i, i])$ by taking appropriate intersections and shifts. Since every list of intervals we store maintains this ordering, we can take any interval in $\mathbf{mrk}_M([i, i])$, and see the corresponding interval it was previously (if any) in $\mathbf{mrk}_M([k, i - 1])$ for all $1 \leq k \leq i - 1$. Then by intersecting the interval in $\mathbf{mrk}_M([i, i])$ with its corresponding interval in $\mathbf{mrk}_M([k, i - 1])$, we get a new corresponding interval in $\mathbf{mrk}_M([k, i])$. We repeat this process iteratively with i going from 1 to n , which at the end computes all of $\mathbf{mrk}_M : \mathbf{Int}([n]) \rightarrow \mathbf{Pvec}$.

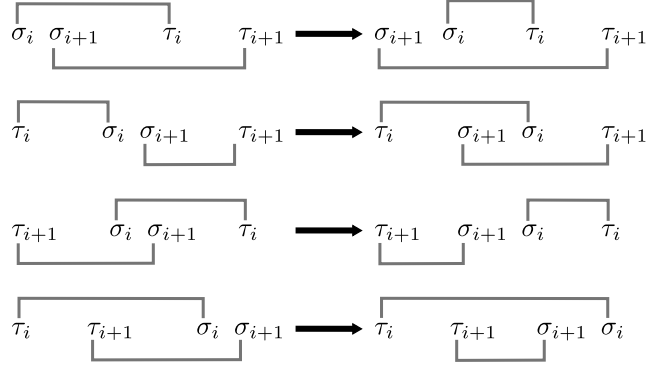
We now describe what can happen to the intervals as we pass over a single square in which a transposition occurs, swapping σ_i and σ_{i+1} . From the analysis in [13, Section 3], if the pairing function changes, then the intervals themselves do not change. If the pairing function remains the same, then two of the persistence intervals will change. Suppose σ_i is paired with τ_i and σ_{i+1} is paired with τ_{i+1} . There are four possibilities, see Figure 8.

We describe the algorithm in Algorithm 1. The output of Algorithm 1 will be \mathbf{mrk}_M , stored as a collection of lists of the barcodes $\mathbf{mrk}_M([s, t])$ for all $s \leq t \in [n]$.

We now prove the correctness of Algorithm 1.

► **Proposition 34.** *For $i \in [n]$, $\mathbf{mrk}_M([i, i])$ is found by taking each interval in the barcode for F_{γ_i} , shifting it down by $i - 1$, and then taking the intersection with $[1, n]$.*

► **Proposition 35.** *Let $1 < i \leq n$, and suppose we know $\mathbf{mrk}_M([i, i])$ and $\mathbf{mrk}_M([k, i - 1])$ for all $1 \leq k \leq i - 1$, and that these lists of intervals are stored in the ordered fashion previously described. From this information, we can compute $\mathbf{mrk}_M([k, i])$.*



■ **Figure 8** Four cases in which intervals change after a transposition. Observe that in each case, both intervals change, and this change is in exactly one coordinate.

2 ■ **Algorithm 1** METARANK(F).

-
- 3 ■ Step 1. Compute $D = RU$ for F_{γ_1} , getting the ordered list **intervals** and the pairing
4 for each interval.
- 5 ■ Step 2. for each interval in **intervals**, intersect the interval with $[1, n]$, and store the
6 result in the ordered list **mrk** $([1, 1])$.
- 7 ■ Step 3. For $i := 1$ to $n - 1$, do
8 ■ Step 3.1. For $j := n$ down to 2, do
9 * update D , R , U , and **intervals** via the vineyards algorithm, as γ sweeps through
10 the square with upper-left corner (i, j) and lower-right corner $(i + 1, j - 1)$.
- 11 ■ Step 3.2. For each interval in **intervals**, shift the interval down by $i - 1$, and intersect
12 the interval with $[1, n]$, storing the result in the ordered list **mrk** $([i, i])$.
- 13 ■ Step 3.3. For $k := 1$ to $i - 1$, do
14 * For each interval in **mrk** $([i, i])$, intersect with the corresponding interval in
15 **mrk** $([k, i - 1])$. Store this intersection in the ordered list **mrk** $([k, i])$.
-

► **Theorem 36.** *Algorithm 1 correctly computes the meta-rank for the bimodule M induced by homology of the input bifiltration F , and runs in time $O(n^3)$. As a result, the number of rectangles in the rank decomposition for M is also $O(n^3)$.*

Proof. By Proposition 34, we can compute **mrk** $_M([1, 1])$, and further **mrk** $_M([i, i])$ for all $i \in [n]$. Then we can use Proposition 35 iteratively to fill in **mrk** $_M([k, i])$ for all $1 \leq k < i \leq n$, and we are done.

For the runtime analysis, first observe that the initial $D = RU$ computation in Step 1 takes $O(n^3)$ time, and **intervals** can be computed from the decomposition in linear time. The loop in Step 2 also takes linear time, as the size of **intervals** is $O(n)$ which is fixed throughout. Step 3 consists of a for loop with $O(n)$ iterations. Step 3.1 consists of a for loop with $O(n)$ iterations, and each loop inside performs an update over a square using the vineyards approach. A single update takes $O(n)$ time in the worst case, so Step 3.1 takes $O(n^2)$ time. Step 3.2 runs in linear time for the same reason as Step 2. Step 3.3 consists of a for loop with $O(n)$ iterations, with each iteration taking $O(n)$ operations as the size of each **mrk** $_M([k, i])$ is the same as **intervals**. Hence, Step 3.3 has total runtime $O(n^2)$. Thus, each loop in Step 3 consists of substeps that run in $O(n^2)$ time, $O(n)$ time, and $O(n^2)$ time respectively, incurring a total cost of $O(n^3)$ over $O(n)$ iterations.

To summarize, we have a step with $O(n^3)$ cost, followed by a step with $O(n)$ cost, followed by a step with $O(n^3)$ cost, so the algorithm runs in $O(n^3)$ time.

By Definition 33, we can compute \mathbf{mdgm}_M from \mathbf{mrk}_M in $O(n^3)$ time, implying the number of non-zero intervals in \mathbf{mdgm}_M is $O(n^3)$. By Proposition 27, each non-zero interval in \mathbf{mdgm}_M corresponds uniquely to a single rectangle in the rank decomposition of M , and so the number of such rectangles is likewise $O(n^3)$. ◀

6 Discussion

We conclude with some open questions. First, we would like to extend our approach to the d -parameter setting. We expect that a proper extension would satisfy relationships with the rank invariant and rank decompositions similar to Proposition 12 and Proposition 27. Such an extension would also lead to a “recursive” formulation of the persistence diagram of diagrams illustrated in Figure 2. Next, Theorem 36 implies that the number of rectangles needed in a rank decomposition for a bimodule is bounded above by $O(n^3)$. It is not known whether this bound is tight. Lastly, there have been multiple recent works that use algorithmic ideas from 1-parameter persistence to compute invariants in the multiparameter setting [15, 16, 23]. We wish to explore in what ways these approaches can create new algorithms or improve upon existing ones for computing the invariants of multi-parameter persistence modules.

References

- 1 Hideto Asashiba, Emerson G Escolar, Ken Nakashima, and Michio Yoshiwaki. On approximation of 2 d persistence modules by interval-decomposables. *arXiv preprint arXiv:1911.01637*, 2019.
- 2 Gorô Azumaya. Corrections and supplementaries to my paper concerning krull-remak-schmidt’s theorem. *Nagoya Mathematical Journal*, 1:117–124, 1950.
- 3 Ulrich Bauer and Maximilian Schmah. Efficient computation of image persistence. *arXiv preprint arXiv:2201.04170*, 2022.
- 4 Leo Betthausen, Peter Bubenik, and Parker B Edwards. Graded persistence diagrams and persistence landscapes. *Discrete & Computational Geometry*, 67(1):203–230, 2022.
- 5 Magnus Botnan and Michael Lesnick. Algebraic stability of zigzag persistence modules. *Algebraic & geometric topology*, 18(6):3133–3204, 2018.
- 6 Magnus Bakke Botnan, Vadim Lebovici, and Steve Oudot. On rectangle-decomposable 2-parameter persistence modules. *Discrete & Computational Geometry*, pages 1–24, 2022.
- 7 Magnus Bakke Botnan, Steffen Oppermann, and Steve Oudot. Signed barcodes for multi-parameter persistence via rank decompositions. In *38th International Symposium on Computational Geometry (SoCG 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.
- 8 Mickaël Buchet and Emerson G. Escolar. Every 1D persistence module is a restriction of some indecomposable 2D persistence module. *Journal of Applied and Computational Topology*, 4:387–424, 2020.
- 9 Gunnar Carlsson and Afra Zomorodian. The theory of multidimensional persistence. *Discrete & Computational Geometry*, 42(1):71–93, 2009.
- 10 Andrea Cerri, Barbara Di Fabio, Massimo Ferri, Patrizio Frosini, and Claudia Landi. Betti numbers in multidimensional persistent homology are stable functions. *Mathematical Methods in the Applied Sciences*, 36(12):1543–1557, 2013.
- 11 Nate Clause, Tamal K. Dey, Facundo Mémoli, and Bei Wang. Meta-diagrams for 2-parameter persistence. *arXiv preprint arXiv:2303.08270*, 2023.
- 12 Nate Clause, Woojin Kim, and Facundo Memoli. The discriminating power of the generalized rank invariant. *arXiv preprint arXiv:2207.11591*, 2022.

25:16 Meta-Diagrams for 2-Parameter Persistence

- 13 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 119–126, 2006.
- 14 William Crawley-Boevey. Decomposition of pointwise finite-dimensional persistence modules. *Journal of Algebra and its Applications*, 14(05):1550066, 2015.
- 15 Tamal K. Dey, Woojin Kim, and Facundo Mémoli. Computing generalized rank invariant for 2-parameter persistence modules via zigzag persistence and its applications. In *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPICs*, pages 34:1–34:17, 2022.
- 16 Abigail Hickok. Computing persistence diagram bundles. *arXiv preprint arXiv:2210.06424*, 2022.
- 17 Woojin Kim and Facundo Mémoli. Generalized persistence diagrams for persistence modules over posets. *Journal of Applied and Computational Topology*, 5(4):533–581, 2021.
- 18 Claudia Landi. The rank invariant stability via interleavings. In *Research in computational topology*, pages 1–10. Springer, 2018.
- 19 Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015.
- 20 Michael Lesnick and Matthew Wright. Interactive visualization of 2-D persistence modules. *arXiv preprint arXiv:1512.00180*, 2015.
- 21 Alex McCleary and Amit Patel. Bottleneck stability for generalized persistence diagrams. *Proceedings of the American Mathematical Society*, 148(733), 2020.
- 22 Alexander McCleary and Amit Patel. Edit distance and persistence diagrams over lattices. *SIAM Journal on Applied Algebra and Geometry*, 6(2):134–155, 2022.
- 23 Dmitriy Morozov and Amit Patel. Output-sensitive computation of generalized persistence diagrams for 2-filtrations. *arXiv preprint arXiv:2112.03980*, 2021.
- 24 Amit Patel. Generalized persistence diagrams. *Journal of Applied and Computational Topology*, 1(3):397–419, 2018.
- 25 Gian-Carlo Rota. On the foundations of combinatorial theory i. theory of möbius functions. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 2(4):340–368, 1964.

Algorithms for Length Spectra of Combinatorial Tori

Vincent Delecroix ✉

Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France

Matthijs Ebbens ✉

Institut Fourier, CNRS, Université Grenoble Alpes, France

Francis Lazarus ✉

G-SCOP/Institut Fourier, CNRS, Université Grenoble Alpes, France

Ivan Yakovlev ✉

Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400 Talence, France

Abstract

Consider a weighted, undirected graph cellularly embedded on a topological surface. The function assigning to each free homotopy class of closed curves the length of a shortest cycle within this homotopy class is called the marked length spectrum. The (unmarked) length spectrum is obtained by just listing the length values of the marked length spectrum in increasing order.

In this paper, we describe algorithms for computing the (un)marked length spectra of graphs embedded on the torus. More specifically, we preprocess a weighted graph of complexity n in time $O(n^2 \log \log n)$ so that, given a cycle with ℓ edges representing a free homotopy class, the length of a shortest homotopic cycle can be computed in $O(\ell + \log n)$ time. Moreover, given any positive integer k , the first k values of its unmarked length spectrum can be computed in time $O(k \log n)$.

Our algorithms are based on a correspondence between weighted graphs on the torus and polyhedral norms. In particular, we give a weight independent bound on the complexity of the unit ball of such norms. As an immediate consequence we can decide if two embedded weighted graphs have the same marked spectrum in polynomial time. We also consider the problem of comparing the unmarked spectra and provide a polynomial time algorithm in the unweighted case and a randomized polynomial time algorithm otherwise.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Mathematics of computing → Graphs and surfaces; Mathematics of computing → Enumeration

Keywords and phrases graphs on surfaces, length spectrum, polyhedral norm

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.26

Related Version *Full Version:* <https://arxiv.org/abs/2303.08036>

Funding *Francis Lazarus:* This work is supported by the project TOFU of the LabEx PERSYVAL-Lab (ANR-11-LABX-0025-01) funded by the French program Investissement d'avenir.

Acknowledgements We would like to thank Marcos Cossarini who suggested to look at the norm point of view and Bruno Grenet who enlightened us about the Polynomial Identity Testing problem. We would also like to thank the referees for their helpful comments, in particular for suggesting the complexity improvement in the case of unweighted graphs.

1 Introduction

Combinatorial surfaces are well-studied in computational topology and are usually represented as graphs cellularly embedded on a topological surface. Given a combinatorial surface S with underlying graph G , many algorithms exist for computing the length of its shortest homotopically non-trivial closed walk [23, 10, 13, 4, 3, 9, 2]. Here, the length of a walk is the sum of the weights of its edges if the edges are weighted, or the number of edges if not.



© Vincent Delecroix, Matthijs Ebbens, Francis Lazarus, and Ivan Yakovlev;
licensed under Creative Commons License CC-BY 4.0

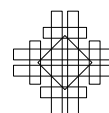
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 26; pp. 26:1–26:16

Leibniz International Proceedings in Informatics

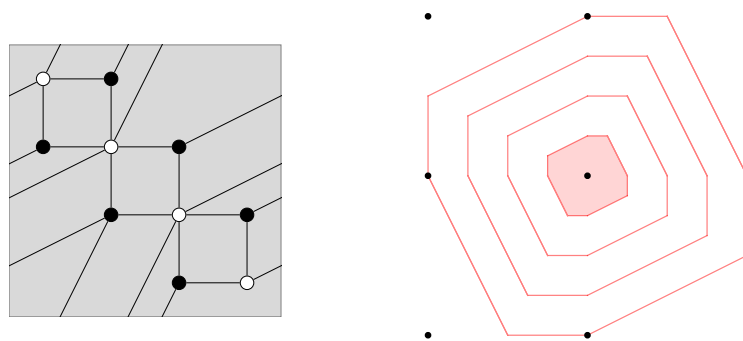


Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



However, relatively little is known about how to compute the *second shortest* non-trivial closed walk, the *third shortest*, etc. More precisely, for every closed walk c in G , we can compute the length of the shortest closed walk freely homotopic to c on S . Obviously, this length only depends on the free homotopy class of c . The ordered sequence of lengths of all free homotopy classes of closed walks is called the *length spectrum* of S with respect to its (weighted) graph G , while the mapping between free homotopy classes of curves and their lengths is called the *marked length spectrum*. The marked length spectrum thus records for every length in the sequence from which free homotopy class it comes from. These notions are well studied in the realm of hyperbolic or Riemannian surfaces [15, 1, 16]. A striking result in that respect is that the marked length spectrum of a non-positively curved surface entirely determines the geometry of the surface [15]. In other words, one may learn the geometry of a surface by just looking at the length of its curves. However, the unmarked length spectrum does not determine the surface even in constant curvature [25].

Analogously, Schrijver [19, Th. 1] proved that embedded graphs that are minor-minimal among graphs with the same marked length spectrum, which he calls *kernels*, are determined by their marked length spectrum up to simple transformations. In [20] Schrijver restricts to unweighted graphs on the torus and notices that the marked length spectrum extends to an integer norm in \mathbb{R}^2 , i.e. a norm taking integer values at integer vectors. See figure 1. Moreover, its dual unit ball is a finite polygon with integer vertices (see [17, 18]). This allows



■ **Figure 1** Left, an unweighted graph on the torus. Right, four dilates of the unit ball of the associated norm. Note that the vertical and horizontal generators of the torus have length four as can be seen from the right diagram.

him to reconstruct for every integer norm a graph whose marked length spectrum is given by this norm.

The aim of our paper is threefold. We first extend the results of Schrijver [20] to weighted graphs on the torus. There are good reasons to focus on the torus. For instance, the marked length spectrum of a graph embedded on the torus being a norm is due to the equivalence between homotopy and homology, which is not true for higher genus surfaces. For weighted graphs, the marked length spectrum function still extends to a norm on \mathbb{R}^2 , that we denote by $N_{G,w}$, but not necessarily to an integer norm. However, we show that it is a polyhedral norm for any choice of weights w , i.e. that the unit ball $B_{G,w} := \{\alpha \in \mathbb{R}^2 \mid N_{G,w}(\alpha) \leq 1\}$ is always a polygon. We also prove that the number of extremal points of this polygon is bounded by a linear function of the number of vertices of G , independent of the weights w .

► **Theorem 1.** *Let (G, w) be a weighted graph with $|V|$ vertices cellularly embedded on the torus. Then $N_{G,w}$ is a polyhedral norm. Moreover, its unit ball $B_{G,w}$ is a polygon with no more than $4|V| + 5$ extremal points, and the ratio of the coordinates of each extremal point is rational.*

We also extend Schrijver's reconstruction of a toroidal graph from an integer norm [20] to the weighted case for non-integer polyhedral norms. See Theorem 7 in Section 3.

Our second goal is to provide algorithms to compute the unit ball $B_{G,w}$ and to compute the length spectrum. Our complexity estimates assume the standard RAM model of computation or the standard real-RAM model for non-integer weights supporting constant time arithmetic operations. We denote by n the complexity of G , i.e. its total number of edges and vertices.

► **Theorem 2.** *The unit ball $B_{G,w}$ can be computed in $O(n^2 \log \log n)$ time.*

This allows us to compute the length of the shortest closed walk freely homotopic to an input closed walk of ℓ edges in $O(\ell + \log n)$ time. It is a priori not obvious to sort the values of the length spectrum from their homotopy classes. However, by decomposing the unit ball into unimodular sectors, i.e., sectors generated by the columns of unimodular matrices, we are able to compute efficiently the first k values of the length spectrum.

► **Theorem 3.** *Let (G, w) be a weighted graph of complexity n cellularly embedded on the torus and let k be a positive integer. After $O(n^2 \log \log n)$ preprocessing time, the first k values of the length spectrum of (G, w) can be computed in $O(k \log n)$ time.*

Recently, Ebbens and Lazarus [7] used shortest path computations in the universal cover of the torus to determine the length spectrum. They compute the first k values of the length spectrum in time $O(kn^2 \log(kn))$. This is to be compared to $O(k \log n)$ in our submission.

Finally, we provide algorithms to check whether two weighted graphs have the same (un)marked length spectrum. In the unweighted case it takes the following simple form.

► **Theorem 4.** *The equality of marked and unmarked spectra of two unweighted graphs G and G' embedded on tori can be tested in time $O(n^2)$ and $O(n^3)$, respectively.*

Our algorithm for the marked length spectrum is also polynomial in the weighted case (Theorem 18). However, to compute the unmarked length spectrum we reduce the equality of length spectra to polynomial identity testing (PIT). See Theorem 19. It becomes deterministic polynomial in the unweighted case as stated in Theorem 4 above.

In contrast with [19], we provide in the full version an example of isospectral toroidal graphs whose associated unit balls are not related by any linear transformation. Hence, they cannot have the same marked spectrum even after applying a homeomorphism on the torus. Also, in the full version on arXiv, we show that in Theorems 2 and 3 the $\log \log n$ factor can be omitted if G is unweighted.

Organization of the paper

We start by discussing some preliminaries in Section 2. We next prove Theorem 1 in Section 3. Theorem 2 is the object of Sections 4 and 5, while Theorem 3 is proved in Section 6. The equality of length spectra is finally discussed in Section 7.

2 Preliminaries

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E . We allow G to have loop and multiple edges. We denote by $n := |V| + |E|$ the complexity of G . A *weight function* for G is a map $w : E \rightarrow \mathbb{R}_+$. The positive value $w(e)$ is the *weight* (or *length*) of the edge $e \in E$. We write (G, w) for a graph G with weight function w . A *walk* is a finite alternating sequence of vertices and edges, starting and ending with a vertex, such that any two successive elements in the sequence are incident. We also use *path* as a synonym for walk.

The *length* $w(c)$ of a walk c is the sum of the weights of its edges, counted with multiplicity. A walk is *closed* when its first and last vertices coincide. This vertex is the *basepoint* of the closed walk. A closed walk without repeated vertices is also called a *simple cycle*.

Throughout this paper, we will use S to denote a topological surface and T to denote the topological orientable surface of genus 1, i.e., a torus. In this paper we assume that G is *cellularly embedded* on S , which means that the complement $S \setminus G$ is a collection of open disks. This embedding can be represented using one of the standard representations, e.g., the incidence graph of flags [8] or rotation systems [14]. A surface together with a cellular embedding of a weighted graph is called a *combinatorial surface*.

Homotopy

Two walks of G are said *homotopic* if they are homotopic as curves in S , i.e., one can be continuously deformed into the other on S while keeping the endpoints fixed. Similarly, two closed walks are *freely homotopic* if they are so as curves in S . Here, we do not require the basepoint to stay fixed during the homotopy. Closed walks (freely) homotopic to a walk reduced to a vertex are said *trivial*. Homotopy is an equivalence relation between walks. The set of homotopy classes of closed walks with fixed basepoint v defines a group under concatenation. It is called the *fundamental group* of S , and denoted by $\pi_1(S, v)$. The fundamental group of the torus is Abelian and isomorphic to \mathbb{Z}^2 . See e.g. [22]. $\pi_1(T, v)$ is thus in bijection with its set of conjugacy classes, hence with the set of free homotopy classes.

A closed walk is *tight* if it is shortest in its free homotopy class. Note that a homotopy class may contain more than one tight closed walk. Let \mathcal{C} denotes the set of free homotopy classes of S . The map $\mathcal{C} \rightarrow \mathbb{R}_+$ that associates to every free homotopy class the length of a tight closed walk in the class is the *marked length spectrum* of S with respect to (G, w) . The *unmarked length spectrum* is the list containing in increasing order the lengths of the non-trivial free homotopy classes of G , counted with multiplicity: if two homotopy classes have the same length, then this length will appear twice in the list.

Homology

Let F be the set of faces of the cellular embedding of G in S . We also call a face, an edge or a vertex, a k -cell for $k = 2, 1, 0$, respectively. The group of *2-chains*, C_2 , is the group of formal linear combinations of faces with integer coefficients with the obvious addition as group operation. A typical element of C_2 has the form $\sum_{f \in F} n_f f$ with $n_f \in \mathbb{Z}$. Likewise, the group C_1 of 1-chains and the group C_0 of 0-chains are the groups of formal linear combinations of edges and vertices, respectively. Cells are assumed to be oriented and a cell multiplied by -1 represents the same cell with opposite orientation.

For $k = 1, 2$, the boundary operator $\partial_k : C_k \rightarrow C_{k-1}$ is the linear extension of the map that sends a k -cell to the formal sum of its boundary facets, where the coefficient of a facet in the sum is 1 if its orientation is induced by the orientation of the k -cell and -1 otherwise. The kernel of ∂_k is denoted by Z_k . Its elements are called *k-cycles*, not to be confused with cycles in the graph theoretical sense. The image of ∂_k is denoted by B_{k-1} . The *first homology group* of S with respect to the coefficients \mathbb{Z} is the group $H_1(S; \mathbb{Z}) := \ker \partial_1 / \text{Im } \partial_2$. From homology theory, $H_1(S; \mathbb{Z})$ does not depend on the specific cell decomposition induced by the cellular embedding of G . We can similarly define the first homology group with real coefficients $H_1(S; \mathbb{R})$. Since the 1-chains only depend on the graph G , we also write $Z_1(G; \mathbb{Z})$ for the group of 1-cycles. The Hurewicz theorem states that the map $\pi_1(S, v) \rightarrow H_1(S; \mathbb{Z})$ that sends (the homotopy class of) a closed walk to the (homology class of the) formal sum

of its oriented edges is onto with kernel the commutator subgroup of $\pi_1(S, v)$. In the case of the torus, $\pi_1(T, v)$ is commutative, so that the above map is an isomorphism. From now on we will identify homotopy and first homology classes on the torus. We will denote by the same letter a closed walk on G and the corresponding 1-cycle in $Z_1(G; \mathbb{Z})$. The homotopy or homology class of a closed walk or 1-cycle c will be indifferently denoted by $[c]$.

Intersection numbers

Given two closed oriented curves c, d on S (endowed with an orientation) with transverse intersections, we may assign a sign to each intersection according to whether the tangents of c and d at the intersection form a positively oriented basis. The sum of the signs over all intersections is called the *algebraic intersection number*. It is a classical result that this number only depends on the homology classes $[c]$ and $[d]$ and that it defines an antisymmetric, nondegenerate bilinear form on $H_1(S; \mathbb{Z})$, denoted by the *pairing* $\langle [c], [d] \rangle$. Of course the total number of intersections of c and d is at least $|\langle [c], [d] \rangle|$.

The universal cover of the torus

We can form a torus by identifying the opposite sides of a square. Equivalently, we can see a torus as the quotient of the plane \mathbb{R}^2 by the action of the group of translations generated by $(1, 0)$ and $(0, 1)$, which we identify with the lattice \mathbb{Z}^2 . Hence, T is identified with $\mathbb{R}^2/\mathbb{Z}^2$ and we have a quotient map $q: \mathbb{R}^2 \rightarrow T$. The plane \mathbb{R}^2 , with the map q , is called the *universal cover* of T . Given a curve c with source point v on T , and a point $\tilde{v} \in q^{-1}(v)$, there is a unique curve \tilde{c} in the plane with source \tilde{v} that *projects* to c , i.e., such that $q(\tilde{c}) = c$. The curve \tilde{c} is a *lift* of c . If c is a closed curve, then the vector from the source to the target of \tilde{c} has integer coordinates and only depends on $[c]$. Hence, each homotopy class can be identified with a lattice translation. Such translations are called *covering transformations* (or translations). A curve is freely homotopic to a simple curve if and only if the coordinates of the corresponding covering translation are coprime [22, Sec. 6.2.2]. By the identification between \mathbb{Z}^2 , $\pi_1(T, v)$ and $H_1(T; \mathbb{Z})$, any pair (α, β) of homology classes that generates $H_1(T; \mathbb{Z})$ must correspond to an invertible integer transformation, hence to a unimodular matrix. Equivalently, $\langle \alpha, \beta \rangle = \pm 1$. (α, β) is a *positively oriented basis* when $\langle \alpha, \beta \rangle = 1$.

Integer and intersection norms

Let $N: \mathbb{Z}^d \rightarrow \mathbb{R}_{\geq 0}$ satisfy the norm axioms:

- $N(\alpha + \beta) \leq N(\alpha) + N(\beta)$ (subadditivity)
- $N(k\alpha) = |k|N(\alpha)$ (absolute homogeneity)
- $N(\alpha) = 0 \implies \alpha = 0$ (separation)

Then N extends to \mathbb{Q}^d using homogeneity, and can be extended to \mathbb{R}^d so that it is continuous. It can be shown that this indeed provides a well-defined norm over \mathbb{R}^d [24]. Such a function N , or its real extension, is called an *integer norm* if $N(\mathbb{Z}^d) \subseteq \mathbb{Z}_{\geq 0}$. Integer norms are *polyhedral*, i.e. their unit ball is a centrally symmetric polytope, and their dual unit ball is a centrally symmetric polytope with integer vertices [24, 20, 17]. See also [5, Sec. 6.0.4]. Integer norms naturally arise as length functions defined over homology classes of curves on surfaces. There are several ways to define curves and their lengths with respect to a graph G embedded on a surface S . One can consider continuous curves on S and define their length as the number of crossings with G . Schrijver [20] applies this framework when S is a torus and shows that this indeed defines a norm. He also considers a framework where the curves are in general position with respect to G , thus avoiding its vertices, and G is required to be 4-regular.

In [19] Schrijver shows that the first framework reduces to the second by considering the medial graph of G . In turn, the second framework reduces to our framework by duality, in the special case where the faces are quadrilaterals and the edges are unweighted.

3 Length spectrum and polyhedral norms on homology

In this section, given a weighted graph (G, w) embedded on a torus T , we introduce a norm on the first homology group of the torus that will be used throughout the article. A correspondence between graphs on the torus and polyhedral norms has been known for some time [20]. But, as far as we know, it has been studied only in the unweighted case and furthermore never analyzed from a computational point of view. For $\alpha \in H_1(T; \mathbb{Z})$ let

$$N_{G,w}(\alpha) := \inf \left\{ \sum_{e \in E(G)} |x_e| w(e) : \sum_{e \in E(G)} x_e e \in Z_1(G; \mathbb{Z}) \text{ and } \left[\sum_{e \in E(G)} x_e e \right] = \alpha \right\}. \quad (1)$$

In the full version we show that $N_{G,w}$ satisfies the norm axioms. Hence, as explained in the subsection ‘‘Integer and intersection norms’’ of Section 2, $N_{G,w}$ extends to a norm on $H_1(T; \mathbb{R})$ (because $H_1(T; \mathbb{Z})$ is naturally a lattice in $H_1(T; \mathbb{R})$). The next lemma asserts that $N_{G,w}$ is indeed the marked length spectrum of T with respect to (G, w) .

► **Lemma 5.** *For every $\alpha \in H_1(T; \mathbb{Z})$ we have*

$$N_{G,w}(\alpha) = \inf \left\{ \sum_{i \in I} x_i \cdot w(c_i) : \left[\sum_{i \in I} x_i \cdot c_i \right] = \alpha \text{ and } x_i \in \mathbb{Z}_{\geq 0} \text{ for } i \in I \right\}, \quad (2)$$

where $\{c_i\}_{i \in I}$ is the (finite) set of all simple cycles in G . The infimum in (2) is attained. Furthermore, for every $\alpha \in H_1(T; \mathbb{Z})$, $N_{G,w}(\alpha)$ is the length of a shortest closed walk c in G with $[c] = \alpha$. In other words, $N_{G,w}$ is the marked length spectrum of T with respect to (G, w) .

In the proof of Theorem 1 below we show that the extremal points of the unit ball $B_{G,w} = \{\alpha \in H_1(T; \mathbb{R}) \mid N_{G,w}(\alpha) \leq 1\}$ of $N_{G,w}$ correspond to homology classes that can be represented by simple cycles in G . The next lemma gives a bound on their number.

For a subset X of a real vector space let $\text{conv}(X)$ denote the convex hull of X . Note that $H_1(T; \mathbb{Z})$ is naturally a subset of the real vector space $H_1(T; \mathbb{R})$.

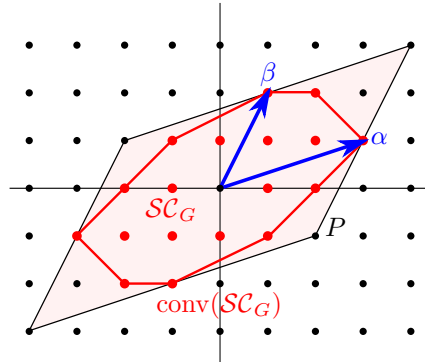
► **Lemma 6.** *Let G be a graph with $|V|$ vertices cellularly embedded on the torus T , and let $\mathcal{SC}_G \subset H_1(T; \mathbb{Z})$ be the set of homology classes of curves that can be represented as simple cycles in G . Then, in $H_1(T; \mathbb{R})$ we have $|\text{conv}(\mathcal{SC}_G) \cap H_1(T; \mathbb{Z})| \leq 4|V| + 5$.*

Proof. Identify $H_1(T; \mathbb{Z})$ with \mathbb{Z}^2 via an arbitrary positively oriented basis. $H_1(T; \mathbb{R})$ is then identified with \mathbb{R}^2 , and the algebraic intersection pairing is given by $\langle (x, y), (x', y') \rangle = xy' - x'y$, whose absolute value is the Euclidean area of the parallelogram generated by these vectors.

Let $\alpha, \beta \in \mathcal{SC}_G$ be represented by simple cycles c_α, c_β in G . On the one hand, the number of intersections between c_α and c_β is bounded by $|V|$, since each intersection corresponds to at least one vertex of G , and all these vertices must be different. On the other hand, it is bounded below by the algebraic intersection number $|\langle \alpha, \beta \rangle|$. Hence, $|\langle \alpha, \beta \rangle| \leq |V|$.

Denote by $\|\cdot\|$ the Euclidean norm on \mathbb{R}^2 and by $\text{dist}(\cdot, \cdot)$ the Euclidean distance, and consider $\alpha, \beta \in \mathcal{SC}_G$ such that $|\langle \alpha, \beta \rangle|$ is maximal. Then for any $\gamma \in \mathcal{SC}_G$, we have $|\langle \gamma, \alpha \rangle| \leq |\langle \alpha, \beta \rangle|$ and $|\langle \gamma, \beta \rangle| \leq |\langle \alpha, \beta \rangle|$. Note that, since these numbers are the areas of the corresponding parallelograms, $|\langle \gamma, \alpha \rangle| = \|\alpha\| \cdot \text{dist}(\gamma, \mathbb{R}\alpha)$, $|\langle \gamma, \beta \rangle| = \|\beta\| \cdot \text{dist}(\gamma, \mathbb{R}\beta)$ and

$|\langle \alpha, \beta \rangle| = \|\alpha\| \cdot \text{dist}(\beta, \mathbb{R}\alpha) = \|\beta\| \cdot \text{dist}(\alpha, \mathbb{R}\beta)$, where $\mathbb{R}\alpha, \mathbb{R}\beta$ denote the one-dimensional \mathbb{R} -subspaces generated by α, β respectively. It follows that $\text{dist}(\gamma, \mathbb{R}\alpha) \leq \text{dist}(\beta, \mathbb{R}\alpha)$ and $\text{dist}(\gamma, \mathbb{R}\beta) \leq \text{dist}(\alpha, \mathbb{R}\beta)$, and so \mathcal{SC}_G is contained in the parallelogram P with vertices $\pm\alpha \pm \beta$, see Figure 2. Clearly, the area $A(P)$ of P is $4|\langle \alpha, \beta \rangle|$. At the same time, by Pick's



■ **Figure 2** The elements of \mathcal{SC}_G are contained in a parallelogram P of area at most $4|V|$.

theorem $A(P) = I + B/2 - 1$, where I is the number of integer points strictly inside P and B is the number of integer points on its boundary. Since α and β are homology classes represented by simple cycles, their corresponding vectors in \mathbb{Z}^2 have coprime coordinates, i.e. the only integer points on the vectors α and β are their endpoints. It follows that the only integer points on the boundary of P are $\pm\alpha, \pm\beta, \pm\alpha \pm \beta$ and so $B = 8$.

Finally, since $\mathcal{SC}_G \subset P$, we have $\text{conv}(\mathcal{SC}_G) \subset P$ as well, and so

$$|\text{conv}(\mathcal{SC}_G) \cap \mathbb{Z}^2| \leq I + B = A(P) + B/2 + 1 = 4|\langle \alpha, \beta \rangle| + 5 \leq 4|V| + 5. \quad \blacktriangleleft$$

In the full version, it is shown that the order of the bound in Lemma 6 is optimal. We now pass to the proof of Theorem 1. In the unweighted case, the polyhedrality of the norm follows from its integrality [20]. However, this argument does not apply in the weighted case.

Proof of Theorem 1. We refer to the full version for a proof that $N_{G,w}$ satisfies the norm axioms. Hence, as explained in Section 2, $N_{G,w}$ extends to a norm on $H_1(T; \mathbb{R})$. To prove the polyhedrality of this norm, we show that

$$B_{G,w} = \text{conv} \left(\left\{ \frac{[c_i]}{w(c_i)} \mid i \in I \right\} \right), \tag{3}$$

where $\{c_i\}_{i \in I}$ is the (finite) set of all oriented simple cycles in G , as in Lemma 5.

Denote the right-hand side of (3) by $B'_{G,w}$. Clearly, for every $i \in I$ we have $N_{G,w}([c_i]) \leq w(c_i)$, so $B'_{G,w} \subset B_{G,w}$. Conversely, take any homology class $\alpha \in H_1(T; \mathbb{Z})$. By Lemma 5, we have $N_{G,w}(\alpha) = \sum_{i \in I} x_i \cdot w(c_i)$ for some $x_i \in \mathbb{Z}_{\geq 0}$ such that $\alpha = [\sum_{i \in I} x_i \cdot c_i]$. Then

$$\frac{\alpha}{N_{G,w}(\alpha)} = \frac{\sum_{i \in I} x_i \cdot [c_i]}{\sum_{i \in I} x_i \cdot w(c_i)} = \sum_{i \in I} \frac{x_i w(c_i)}{\sum_{j \in I} x_j \cdot w(c_j)} \cdot \frac{[c_i]}{w(c_i)}$$

is a representation of $\frac{\alpha}{N_{G,w}(\alpha)}$ as a convex combination of $\frac{[c_i]}{w(c_i)}, i \in I$. Hence $B_{G,w} \subset B'_{G,w}$ and we get (3). By definition, the $[c_i]$ can be represented by simple cycles in G . By Lemma 6 their number is at most $4|V| + 5$, and so the number of extremal points of $B_{G,w}$ is also at most $4|V| + 5$. The slopes of $[c_i]/w(c_i)$ are rational since the $[c_i]$ belong to $H_1(T; \mathbb{Z})$. \blacktriangleleft

Finally, we show how to reconstruct a weighted graph (G, w) embedded on the torus T from a polyhedral norm on \mathbb{R}^2 .

► **Theorem 7.** *Let $N : \mathbb{R}^2 \rightarrow \mathbb{R}$ be a polyhedral norm all of whose extremal points have rational slopes. Let $\{\pm(p_i, q_i)\}_{i=1, \dots, n}$ be the set of non-zero integral vectors closest to the origin on the rays issued from the origin in the direction of the extremal points of the unit ball $\{v \in \mathbb{R}^2 : N(v) \leq 1\}$. Then there exists a weighted 4-valent graph (G, w) embedded on the torus T with $\sum_{1 \leq i < j \leq n} |p_i q_j - p_j q_i|$ vertices so that $N_{(G, w)} = N$.*

4 Good short basis

Our computation of the length spectrum and of its unit ball relies on the initial computation of a *good short basis*. By a *short basis* we mean a pair of tight simple cycles (a, b) in G such that a is a shortest non-trivial closed walk and b is a shortest non-trivial closed walk satisfying $\langle [a], [b] \rangle = 1$. We say that (a, b) is a *good basis* if $([a], [b])$ is a positively oriented basis of $H_1(T; \mathbb{Z})$ and a and b intersect along a connected path, possibly reduced to a vertex.

► **Lemma 8.** *Let (G, w) be a weighted graph of complexity n cellularly embedded on the torus. A good short basis can be computed in $O(n \log n)$ time.*

Sketch of proof. We first compute a shortest non-trivial closed walk a in $O(n \log n)$ time following Kutz [13, Th. 1]. This closed walk must be a tight simple cycle as otherwise it could be decomposed into shorter non-trivial closed walks. We claim that among all shortest non-trivial closed walks b satisfying $\langle [a], [b] \rangle = 1$ there is one that intersects a along a connected path. See full version. Cutting T along a yields an annulus A with two copies a' and a'' of a as boundary components. By the above claim, b intersects A in a shortest path connecting two copies of the same vertex respectively on a' and a'' . We find this shortest path using the multiple-source shortest path algorithm of Klein; see [12] and [2, Th. 3.8]. This algorithm builds a data structure in $O(n \log n)$ time that allows to query for the distance between any vertex on a' and any other vertex in A in $O(\log n)$ time. We need to query for the $O(n)$ pairs of copies of vertices of a and retain a pair (u', u'') that minimizes the distance. In order to find an explicit representative of b , we can in a second step run Dijkstra's algorithm with source u' in A . Finally, to ensure that b intersects a along a connected path, we can replace the subpath between u' and the last occurrence of a vertex on a' by a subpath of a' with the same length and do similarly on a'' . The total running time is $O(n \log n)$. We obtain b by gluing back the two copies of a . ◀

We shall always express a homology class in the basis (a, b) and identify the class with a vector in \mathbb{Z}^2 . Hence, $[a]$ and $[b]$ are identified with $(1, 0)$ and $(0, 1)$, respectively.

5 Computing the unit ball

Here, we provide an algorithm for computing the unit ball $B_{G, w}$ of $N_{G, w}$ corresponding to the weighted graph (G, w) . Let $\mathcal{TSC}_{G, w} \subset H_1(T; \mathbb{Z})$ be the set of homology classes that admit a tight and simple representative in G . Of course, $\mathcal{TSC}_{G, w} \subseteq \mathcal{SC}_G$, and the homology classes of a and b computed in Section 4 are in $\mathcal{TSC}_{G, w}$ by construction. In Section 3, we proved that $B_{G, w}$ is the convex hull of a set $\{\alpha/N_{G, w}(\alpha)\}_{\alpha \in \mathcal{SC}_G}$ containing $O(|V|)$ classes. We shall compute a subset H of $\mathcal{TSC}_{G, w}$ whose normalized vectors, $\{\alpha/N_{G, w}(\alpha)\}_{\alpha \in H}$, include all the extremal points of $B_{G, w}$. Since the coordinates of each element of $\mathcal{TSC}_{G, w}$ must be

coprime, the set of directions defined by the elements of $\mathcal{TS}C_{G,w}$ are pairwise distinct and naturally ordered angularly. We search for H by exploring the whole set of directions using dichotomy together with a simple pruning strategy. Suppose we need to explore the angular sector $\angle(\alpha, \beta)$, where (α, β) forms a basis of $H_1(T; \mathbb{Z})$. The dichotomy consists in splitting the sector into the sectors $\angle(\alpha, \gamma)$ and $\angle(\gamma, \beta)$ with $\gamma := \alpha + \beta$. Note that (α, γ) and (γ, β) are again bases of $H_1(T; \mathbb{Z})$. In particular, the coordinates of γ are coprime. Since for any nonzero $\eta \in H_1(T; \mathbb{Z})$, the vector $\eta/N_{G,w}(\eta)$ lies on the boundary of the unit ball, it follows by convexity of $B_{G,w}$ that the segment $[\frac{\alpha}{N_{G,w}(\alpha)}, \frac{\beta}{N_{G,w}(\beta)}]$ is a subset of a supporting line of $B_{G,w}$ whenever $\gamma/N_{G,w}(\gamma)$ lies on this segment. This last condition has a simple certificate.

▷ **Claim 9.** $\frac{\gamma}{N_{G,w}(\gamma)}$ lies on the segment $[\frac{\alpha}{N_{G,w}(\alpha)}, \frac{\beta}{N_{G,w}(\beta)}]$ if and only if $N_{G,w}(\alpha + \beta) = N_{G,w}(\alpha) + N_{G,w}(\beta)$.

It follows from the previous discussion that if $N_{G,w}(\alpha + \beta) = N_{G,w}(\alpha) + N_{G,w}(\beta)$, then the interior of the sector $\angle(\alpha, \beta)$ cannot contain any extremal point and we can prune this whole sector in our search. This leads to the pseudo-code of Algorithm 1 for computing H . In the sequel, we say that a pair of closed walks in G is **good** if they are simple and tight cycles, if their homology classes form a basis of $H_1(T; \mathbb{Z})$, and if they moreover intersect along a connected path, possibly reduced to a vertex.

■ **Algorithm 1** Compute H .

Require: A weighted graph (G, w) cellularly embedded on the torus

Ensure: A short basis (a, b) and a sorted list $H = [(x_i, y_i), c_i, w(c_i)]$ where c_i is a simple tight cycle in G , $(x_i, y_i) \in \mathbb{Z}^2$ represents its homology class $[c_i] = x_i[a] + y_i[b]$, and $w(c_i) = N_{G,w}([c_i])$. Also, the extremal points of $B_{G,w}$ are contained in the set of vectors $\{[c_i]/w(c_i) : i \in \{0, \dots, size(H) - 1\}\}$.

- 1: Compute a good short basis (a, b) as explained in Section 4
- 2: $h_1 := ((1, 0), a, w(a))$ {Note that $N_{G,w}([a]) = w(a)$ }
- 3: $h_2 := ((0, 1), b, w(b))$ {and that $N_{G,w}([b]) = w(b)$ }
- 4: $\bar{h}_1 := ((-1, 0), \bar{a}, w(a))$
- 5: $H := \{h_1, h_2\}$ {Initialise H .}
- 6: $S := \{\angle(h_1, h_2), \angle(h_2, \bar{h}_1)\}$ {Initialise a set of sectors to explore with the upper quadrants.}
- 7: **while** $S \neq \emptyset$ **do**
- 8: Extract and remove from S a sector $\angle(h, h')$
- 9: $(x, y), c, \ell := h$ {Note that $N_{G,w}([c]) = \ell$.}
- 10: $(x', y'), c', \ell' := h'$ {Similarly $N_{G,w}([c']) = \ell'$.}
- Require:** (c, c') is a good pair
- 11: Compute a tight representative c'' of $\gamma'' := [c] + [c']$ with its norm $\ell'' := N_{G,w}(\gamma'') = w(c'')$
- 12: **if** $\ell'' < \ell + \ell'$ **then**
- 13: $h'' := ((x + x', y + y'), c'', \ell'')$
- 14: Insert h'' in H between h and h'
- 15: $S := S \cup \{\angle(h, h''), \angle(h'', h')\}$
- 16: **end if**
- 17: **end while**
- 18: $H := H \cup \bar{H}$ {Add the symmetric of H w.r.t. the origin.}

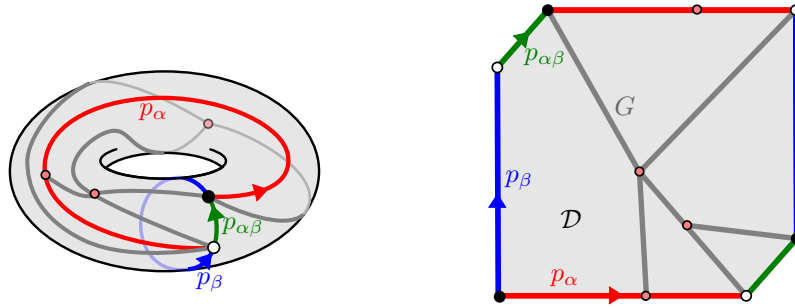
By subadditivity of the norm, the test in Line 12 of Algorithm 1 may only fail when $N_{G,w}([c'']) = N_{G,w}([c]) + N_{G,w}([c'])$. It then follows from Claim 9 and the preceding discussion on our pruning strategy that we are not missing any direction of extremal points in the upper plane when adding homology classes in Line 14. Moreover, Line 18 and the central symmetry of the unit ball ensure that the above algorithm indeed computes a set of homology classes

26:10 Algorithms for Length Spectra of Combinatorial Tori

whose normalized vectors contains the extremal points of $B_{G,w}$. It remains to explain how to perform the computation in Line 11 and to analyse the complexity of Algorithm 1.

► **Lemma 10.** *Let (α, β) be a homology basis such that α and β have representatives, respectively c_α and c_β , forming a good pair. We can compute a tight representative $c_{\alpha+\beta}$ for $\alpha + \beta$ in $O(n \log \log n)$ time.*

Sketch of proof. By hypothesis, c_α and c_β intersect along a connected path $p_{\alpha\beta}$. The path $p_{\alpha\beta}$ may be oriented the same way or not in c_α and c_β . We consider the case where it is oriented the same way (see the full version for the other case). Then $c_\alpha = p_\alpha \cdot p_{\alpha\beta}$ and $c_\beta = p_\beta \cdot p_{\alpha\beta}$ for some paths p_α, p_β in G . We cut T along $c_\alpha \cup c_\beta$, viewed as a subgraph of G . We obtain a hexagonal plane domain \mathcal{D} with sides $p_\beta, p_{\alpha\beta}, p_\alpha, \bar{p}_\beta, \bar{p}_{\alpha\beta}, \bar{p}_\alpha$ in the clockwise order around the boundary of \mathcal{D} . See Figure 3. The universal cover of T is tessellated by

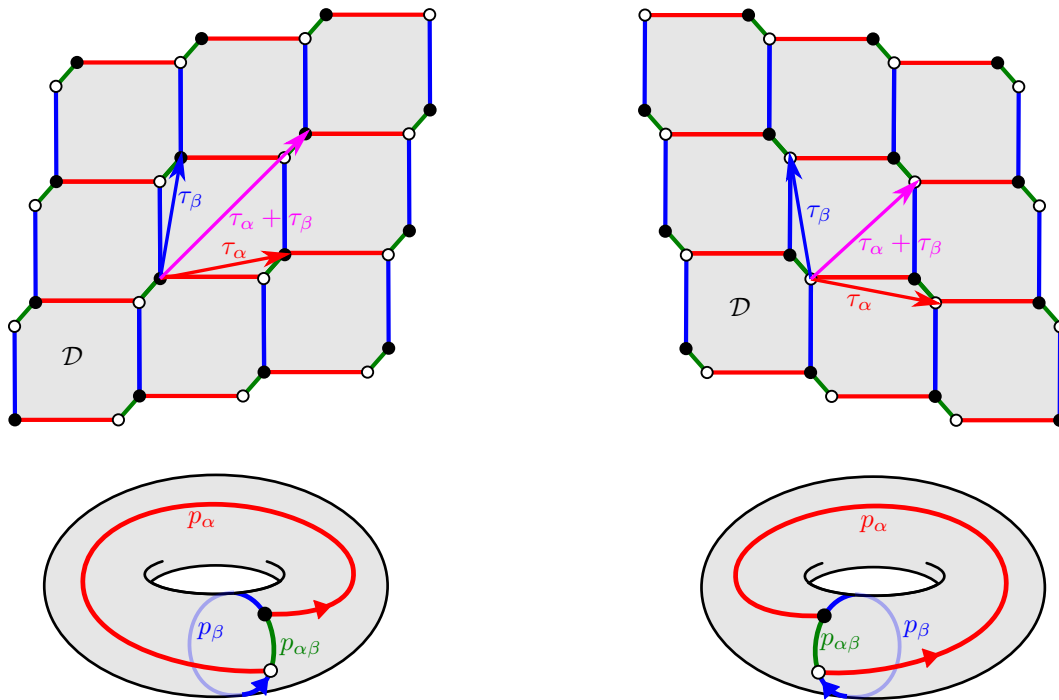


■ **Figure 3** Cutting T along $c_\alpha \cup c_\beta$.

translated copies of \mathcal{D} glued along their sides so that the side p of a domain is glued to the side \bar{p} of the adjacent domain. See Figure 4. As before, let $\gamma = \alpha + \beta$. Since (α, γ) is a positively oriented basis, we know that $\langle \alpha, \gamma \rangle = 1$. Hence, any representative of γ must cross c_α . Let c_γ be a tight representative of γ with a lift \tilde{c}_γ in the universal cover starting from a vertex \tilde{v} on the side p_α or $p_{\alpha\beta}$ of a domain \mathcal{D}_0 and ending at the vertex $\tilde{w} := \tilde{v} + \tau_\alpha + \tau_\beta$, where τ_α and τ_β are the covering translations corresponding to α and β respectively. There are two situations according to whether \tilde{v} lies on the side p_α or $p_{\alpha\beta}$ of \mathcal{D}_0 . See Figure 5.

- If \tilde{v} lies on the side p_α of \mathcal{D}_0 then \tilde{w} belongs to the side \bar{p}_α of $\mathcal{D}_1 := \mathcal{D}_0 + \tau_\alpha$. We claim that $\mathcal{D}_0 \cup \mathcal{D}_1$ is convex, i.e., any two vertices in $\mathcal{D}_0 \cup \mathcal{D}_1$ can be joined by a shortest path contained in $\mathcal{D}_0 \cup \mathcal{D}_1$. Indeed, since c_α is tight, any bi-infinite concatenation of its lifts is a geodesic line in the weighted lift of G in the universal cover of T . Similarly, any bi-infinite concatenation of lifts of c_β is a geodesic line and thus delimits two convex half-planes. See the dotted and broken lines in Figure 5. It follows that $\mathcal{D}_0 \cup \mathcal{D}_1$ is the intersection of four half-planes, hence is convex. We can thus assume that c_γ has a lift in $\mathcal{D}_0 \cup \mathcal{D}_1$ with endpoints \tilde{v} and \tilde{w} on the boundary of $\mathcal{D}_0 \cup \mathcal{D}_1$. We can glue the p_α side of \mathcal{D}_0 with the \bar{p}_α side of \mathcal{D}_1 and search for the shortest generating cycle of the resulting annulus in $O(n \log n)$ time as in [6, Prop. 2.7(e)], or, more efficiently, in time $O(n \log \log n)$ as in [11, Theorem 7].
 - If \tilde{v} lies on the side $p_{\alpha\beta}$ of \mathcal{D}_0 we can compute \tilde{c}_γ in linear time. See the full version.
- In all cases we may compute a tight representative of γ in $O(n \log \log n)$ time. ◀

► **Lemma 11.** *The tight representative $c_{\alpha+\beta}$ computed in Lemma 10 can be modified in $O(n)$ time in order to satisfy the additional following property (P): The intersection of any lift of $c_{\alpha+\beta}$ in the universal cover with any line is either empty or a common connected subpath.*



■ **Figure 4** The universal cover of T . Left: $p_{\alpha\beta}$ is oriented consistently with both c_α and c_β . Right: $p_{\alpha\beta}$ has opposite orientation in c_α and c_β .

► **Lemma 12.** *Let (α, β) be a homology basis such that α and β have representatives, respectively c_α and c_β , forming a good pair. If $N_{G,w}(\alpha + \beta) < N_{G,w}(\alpha) + N_{G,w}(\beta)$, then the tight representative $c_{\alpha+\beta}$ computed in Lemma 11 is such that $(c_\alpha, c_{\alpha+\beta})$ and $(c_{\alpha+\beta}, c_\beta)$ are good pairs.*

Since in Algorithm 1 we only add $\alpha + \beta$ to H at line 14 when α, β are already in H with $N_{G,w}(\alpha + \beta) < N_{G,w}(\alpha) + N_{G,w}(\beta)$, Lemma 12 immediately implies

► **Corollary 13.** $H \subset \mathcal{TSC}_{G,w} \subset \mathcal{SC}_G$

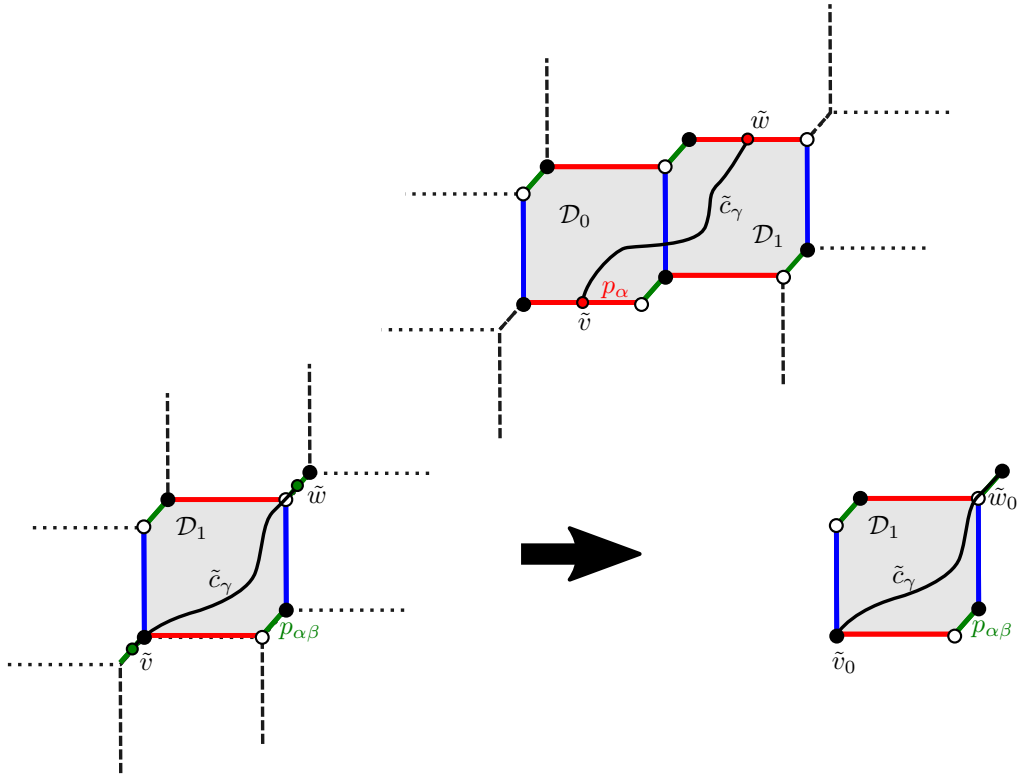
► **Corollary 14.** *The number of iterations in the while loop of Algorithm 1, from Line 7 to 17, is bounded by twice the size of $\mathcal{TSC}_{G,w}$.*

► **Proposition 15.** *Algorithm 1 runs in $O(n^2 \log \log n)$ time.*

Proof. From Corollary 14, Algorithm 1 enters at most $2|\mathcal{TSC}_{G,w}|$ times the while loop between Lines 7 and 17. This is $O(n)$ iterations by Lemma 6. Each iteration takes $O(n \log \log n)$ time for executing Line 11 by Lemmas 10 and 11. Lemma 12 ensures that only good pairs are stored at Line 15, so that the requirement for executing Line 11 is always satisfied. Line 18 moreover takes time $O(|H|) = O(n)$. Since every other line takes constant time to execute, the total time for running Algorithm 1 is $O(n \cdot n \log \log n + n) = O(n^2 \log \log n)$. ◀

We are now ready to prove that $B_{G,w}$ can be computed in $O(n^2 \log \log n)$ time.

Proof of Theorem 2. Proposition 15 states that we can compute in $O(n^2 \log \log n)$ time a list H of $O(n)$ vectors, with their norms, that contains the directions of the extremal points of the unit ball $B_{G,w}$. After normalising the vectors we compute their convex hull in $O(n \log n)$ time with any classical convex hull algorithm. Overall, this leads to an $O(n^2 \log \log n)$ time algorithm for computing $B_{G,w}$. ◀



■ **Figure 5** Top: if c_γ crosses p_α , then c_γ has a lift in $\mathcal{D}_0 \cup \mathcal{D}_1$. The dotted and broken lines are supporting geodesics for the considered regions. Bottom: when c_γ crosses $p_{\alpha\beta}$ it has a lift contained in the union of \mathcal{D}_1 with two lifts of $p_{\alpha\beta}$. We can shift the origin of the lift to \tilde{v}_0 .

For further reference, we establish a useful property of the ordered set of elements in H . Namely, two consecutive cycles in H define a *unimodular cone*.

► **Lemma 16.** *The sorted list $H = [(x_i, y_i), c_i, w(c_i)]$ computed by Algorithm 1 is such that the rays $\mathbb{R}_{\geq 0}c_i$ are ordered cyclically by angle, and $\langle [c_i], [c_{i+1}] \rangle = 1$ for all i . In particular, the half-open cones $C_i = \mathbb{Z}_{\geq 0}[c_i] + \mathbb{Z}_{> 0}[c_{i+1}]$ constitute a partition of $H_1(T; \mathbb{Z})$.*

6 Computing the length spectrum

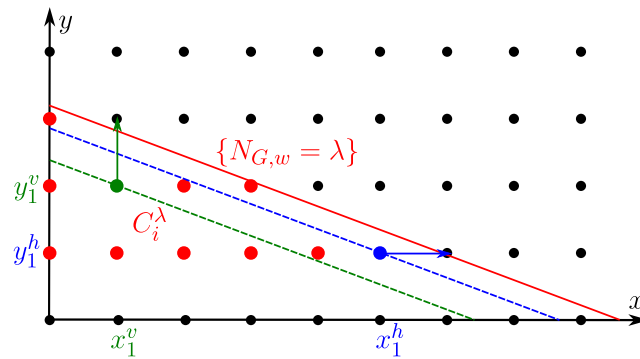
We now give a proof of Theorem 3.

Proof. First, as described in Sections 4 and 5, we compute in time $O(n^2 \log \log n)$ a short basis (a, b) and a list H of triples $((x_i, y_i), c_i, w(c_i))$, where c_i is a simple tight cycle in G , $(x_i, y_i) \in \mathbb{Z}^2$ represents its homology class $[c_i] = x_i[a] + y_i[b]$ and $w(c_i) = N_{G,w}([c_i])$. From Lemma 16, this provides a partition of $H_1(T; \mathbb{Z})$ into half-open cones $C_i = \mathbb{Z}_{\geq 0}[c_i] + \mathbb{Z}_{> 0}[c_{i+1}]$. Note that by construction $N_{G,w}$ is linear over each C_i .

We then compute the values of the length spectrum iteratively, in increasing order, storing them into a list Λ , initially empty. Intuitively, the algorithm consists in sweeping $H_1(T; \mathbb{Z})$ by increasing the radius of the λ -ball $\lambda B_{G,w}$ from $\lambda = 0$. Each time a lattice point is swept, its norm λ is added to Λ . We actually sweep the cones C_i in parallel.

By Lemma 16, the half-open cones C_i decompose the ball $\lambda B_{G,w}$ into sectors $C_i^\lambda := \lambda B_{G,w} \cap C_i$. For each sweeping value λ and each i we store two ordered subsets of C_i^λ into dequeues (double-ended queues) F_i^h and F_i^v corresponding to the *horizontal* and *vertical*

sweeping front, respectively. Formally $F_i^h = ((x_1^h, y_1^h), \dots, (x_{i_h}^h, y_{i_h}^h))$, where each $(x_j^h, y_j^h) \in \mathbb{Z}_{\geq 0} \times \mathbb{Z}_{> 0}$ is such that $x_j^h[c_i] + y_j^h[c_{i+1}] \in C_i^\lambda$ and $(x_j^h + 1)[c_i] + y_j^h[c_{i+1}] \notin \lambda B_{G,w}$. Moreover, the homology classes in F_i^h are ordered by their norms in increasing order. Similarly, $F_i^v = ((x_1^v, y_1^v), \dots, (x_{i_v}^v, y_{i_v}^v))$ contains the list (ordered by norm in increasing order) of coordinates of the homology classes contained in $\lambda B_{G,w}$ but whose translates by $[c_{i+1}]$ have norms larger than λ . Initially F_i^h is the empty dequeue and F_i^v contains the coordinates $(0, 0)$ of the zero class. See Figure 6



■ **Figure 6** Here, the point (x, y) represents the class $x[c_i] + y[c_{i+1}]$. The solid red line represents the points whose norm is λ . The dashed lines correspond to the norms of (x_1^h, y_1^h) and (x_1^v, y_1^v) .

▷ **Claim 17.** The coordinates in the $([c_i], [c_{i+1}])$ basis of the homology class in $C_i \setminus C_i^\lambda$ with the smallest norm is either $(x_1^h + 1, y_1^h)$ or $(x_1^v, y_1^v + 1)$.

We are now ready to describe the sweeping algorithm. We store the indices i of the sectors C_i^λ in a (balanced) binary search tree \mathcal{S} allowing minimum extraction, deletion and insertion in logarithmic time. The key of sector C_i^λ used for comparisons in the tree is the minimum norm of a homology class in $C_i \setminus C_i^\lambda$. From the previous claim it can be computed in constant time from F_i^h and F_i^v as

$$\begin{aligned} \text{key}[i] &= \min(N_{G,w}([(x_1^h + 1)c_i + y_1^h c_{i+1}]), N_{G,w}([x_1^v c_i + (y_1^v + 1)c_{i+1}])) \\ &= \min((x_1^h + 1)w(c_i) + y_1^h w(c_{i+1}), x_1^v w(c_i) + (y_1^v + 1)w(c_{i+1})) \end{aligned}$$

Suppose we have computed the m first values of the length spectrum, i.e., $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$, and we want to compute λ_{m+1} . We extract and remove from \mathcal{S} the sector $C_i^{\lambda_m}$, with $i = \mathcal{S}.\text{min}()$, i.e., with a non-swept homology class α of minimal norm. Hence, we have $\lambda_{m+1} = \text{key}[i]$. We update F_i^h and F_i^v as follows. If $\alpha = (x_1^h + 1)[c_i] + y_1^h[c_{i+1}]$, then we remove (x_1^h, y_1^h) from the bottom of F_i^h and push $(x_1^h + 1, y_1^h)$ on its top. Likewise, if $\alpha = x_1^v[c_i] + (y_1^v + 1)[c_{i+1}]$, we remove (x_1^v, y_1^v) from the bottom of F_i^v and push $(x_1^v, y_1^v + 1)$ on its top. We do both if $\alpha = (x_1^h + 1)[c_i] + y_1^h[c_{i+1}] = x_1^v[c_i] + (y_1^v + 1)[c_{i+1}]$. Clearly, the updated dequeues contain the required lattice points with respect to the sweeping value $\lambda = \lambda_{m+1}$. We then update \mathcal{S} by inserting i with its new key resulting from the updates of F_i^h and F_i^v . We finally add λ_{m+1} to Λ . By Corollary 13 and Theorem 1, \mathcal{S} contains $O(n)$ items. The running time for a sweeping step is thus $O(\log n)$ time for interacting with \mathcal{S} plus constant time for updating F_i^h, F_i^v and Λ . We can thus compute the first k values of the length spectrum in $O(n^2 \log \log n + k \log n)$ time. ◀

7 Deciding equality of length spectra

We now present an application of Algorithm 1 to the following decision problem: given two weighted graphs (G, w) and (G', w') embedded on tori do they have the same length spectra? This question actually covers two problems: the equality of the marked and of the unmarked length spectrum. As we show, the former reduces to the linear equivalence of polyhedral norms which has a straightforward quadratic time solution. In contrast the latter reduces to the problem of polynomial identity testing which is only known to be in the co-RP complexity class [21]. In particular, this problem is in co-NP. For unweighted graphs however, the complexity is deterministic polynomial.

We first aim to compare the length spectrum as maps from $H_1(T; \mathbb{Z}) \rightarrow \mathbb{R}$. We are given two weighted graphs (G, w) and (G', w') embedded on tori T and T' respectively. We say that (G, w) and (G', w') have the same *marked spectrum* if there exists a homeomorphism $\phi : T \rightarrow T'$ such that for all $\gamma \in H_1(T; \mathbb{Z})$ we have $N_{G', w'}(\phi_*(\gamma)) = N_{G, w}$ where $\phi_*(\gamma)$ denotes the class $[\phi(c)]$ where c is a curve representative of γ .

► **Theorem 18.** *Let (G, w) and (G', w') be two weighted graphs cellularly embedded on tori T and T' , each with complexity bounded by n . Then there is an algorithm that answers whether (G, w) and (G', w') have the same marked spectrum in time $O(n^2 \log \log n)$.*

Now we consider the more delicate question of comparing unmarked spectra. That is, we want to decide whether the list of values $\{N_{G, w}(\alpha) : \alpha \in H_1(T; \mathbb{Z})\}$ and $\{N_{G', w'}(\alpha') : \alpha' \in H_1(T'; \mathbb{Z})\}$ coincide where each value comes with multiplicity according to the number of homology classes that realize this length. This equality of unmarked length spectra is always decidable and we show that it belongs to the co-RP complexity class, i.e. our algorithm can detect if the unmarked spectra of (G, w) and (G', w') are different in random polynomial time. For this specific test, we need to have access to all integral linear relations between the weights at once. That is, our algorithm needs to have access to the \mathbb{Q} -vector space $\{(x_e)_{e \in E(G)} \in \mathbb{Q}^{E(G)} : \sum_e x_e w_e = 0\}$. We assume that the weights are given in the following form : we are given r real numbers $o = (o_1, o_2, \dots, o_r)$ that do not satisfy any integral linear relations, and for each edge $e \in E(G)$ its weight is given as a linear combination of these real numbers with integral coefficients $w_e = w_{e,1}o_1 + \dots + w_{e,r}o_r$. We call *complexity* of these weights the sum $\|w\|_o := \sum_{e \in E(G)} \sum_{i=1}^r |w_{e,i}|$. Note that this complexity depends on the choice of the numbers o_1, \dots, o_r and not only on the values w_e as real numbers.

► **Theorem 19.** *Let (G, w) and (G', w') be two weighted graphs cellularly embedded on tori T and T' , each with complexity bounded by n , where each weight is specified as $w_e = w_{e,1}o_1 + \dots + w_{e,r}o_r$ with $w_{e,i} \in \mathbb{Z}$ and o_1, \dots, o_r are r given real numbers. There is an algorithm to decide whether (G, w) and (G', w') have different (unmarked) spectra that runs in random polynomial time in the total input size $n + \log(\|w\|_o + \|w'\|_o)$. Moreover, for fixed r , there is a deterministic algorithm that runs in time $O(n^2 \cdot (\|w\|_o + \|w'\|_o)^r)$.*

Let us explain how to deduce Theorem 4 from Theorems 18 and 19. We emphasize that Theorem 19 allows us to deduce deterministic polynomial time only in the unweighted case. Even with rational weights we are not aware of a deterministic polynomial time algorithm.

Proof of Theorem 4. The case of marked spectrum is simply a particular case of Theorem 18, where we use the fact that the unit ball can be computed in quadratic time in the unweighted case. For the equality of unmarked spectrum, we have $r = 1$ and $o_1 = 1$. The second part of Theorem 19 hence gives deterministic polynomial time in $O(n^2 \cdot (\sum_{e \in E(G) \cup E(G')} 1)) = O(n^3)$. ◀

References

- 1 Peter Buser. *Geometry and Spectra of Compact Riemann Surfaces*. Birkhäuser, 1992.
- 2 Sergio Cabello, Erin W. Chambers, and Jeff Erickson. Multiple-source shortest paths in embedded graphs. *SIAM Journal on Computing*, 42(4):1542–1571, 2013.
- 3 Sergio Cabello, Éric Colin de Verdière, and Francis Lazarus. Algorithms for the edge-width of an embedded graph. *Computational Geometry*, 45(5-6):215–224, 2012.
- 4 Sergio Cabello and Bojan Mohar. Finding shortest non-separating and non-contractible cycles for topologically embedded graphs. *Discrete & Computational Geometry*, 37(2):213–235, 2007.
- 5 Marcos Cossarini. *Discrete surfaces with length and area and minimal fillings of the circle*. PhD thesis, Instituto de Matemática Pura e Aplicada (IMPA), 2018.
- 6 Éric Colin de Verdière and Jeff Erickson. Tightening nonsimple paths and cycles on surfaces. *SIAM Journal on Computing*, 39(8):3784–3813, 2010.
- 7 Matthijs Ebbens and Francis Lazarus. Computing the length spectrum of combinatorial graphs on the torus. 38th International Symposium on Computational Geometry: Young Researchers Forum, 2022.
- 8 David Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 599–608, 2003.
- 9 Jeff Erickson. Combinatorial optimization of cycles and bases. In Afra Zomorodian, editor, *Advances in Applied and Computational Topology*, Proceedings of Symposia in Applied Mathematics, 2012.
- 10 Jeff Erickson and Sarel Har-Peled. Optimally cutting a surface into a disk. *Discrete & Computational Geometry*, 31(1):37–59, 2004.
- 11 Giuseppe F. Italiano, Yahav Nussbaum, Piotr Sankowski, and Christian Wulff-Nilsen. Improved algorithms for min cut and max flow in undirected planar graphs. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 313–322, New York, NY, USA, 2011. Association for Computing Machinery. doi:10.1145/1993636.1993679.
- 12 Philip N Klein. Multiple-source shortest paths in planar graphs. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 146–155, 2005.
- 13 Martin Kutz. Computing Shortest Non-Trivial Cycles on Orientable Surfaces of Bounded Genus in Almost Linear Time. In *22nd Annual ACM Symposium on Computational Geometry*, pages 430–437, 2006.
- 14 Bojan Mohar and Carsten Thomassen. *Graphs on surfaces*, volume 10. JHU press, 2001.
- 15 Jean-Pierre Otal. Le spectre marqué des longueurs des surfaces à courbure négative. *Annals of Mathematics*, 131(1):151–162, 1990.
- 16 Hugo Parlier. Interrogating surface length spectra and quantifying isospectrality. *Mathematische Annalen*, 370(3):1759–1787, 2018. URL: <https://math.uni.lu/parlier/Papers/Isospectral12016-11-07.pdf>.
- 17 Mikael de la Salle. On norms taking integer values on the integer lattice. *Comptes Rendus Mathématique*, 354(6):611–613, 2016.
- 18 Abdoul Karim Sane. Intersection norms and one-faced collections. *Comptes Rendus Mathématique*, 358(8):941–956, 2020.
- 19 Alexander Schrijver. On the uniqueness of kernels. *Journal of Combinatorial Theory, Series B*, 55(1):146–160, 1992.
- 20 Alexander Schrijver. Graphs on the torus and geometry of numbers. *Journal of Combinatorial Theory, Series B*, 58(1):147–158, 1993.
- 21 Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.*, 27:701–717, 1980. doi:10.1145/322217.322225.
- 22 John Stillwell. *Classical topology and combinatorial group theory*, volume 72. Springer Science & Business Media, 1993.
- 23 Carsten Thomassen. Embeddings of graphs with no short noncontractible cycles. *Journal of Combinatorial Theory, Series B*, 48(2):155–177, 1990.

26:16 Algorithms for Length Spectra of Combinatorial Tori

- 24 William P. Thurston. A norm for the homology of 3-manifolds. In *Mem. Amer. Math. Soc.*, volume 339, pages 99–130. AMS, 1986.
- 25 Marie-France Vignéras. Isospectral and non-isometric Riemannian manifolds. *Ann. Math. (2)*, 112:21–32, 1980. doi:10.2307/1971319.

Computing a Dirichlet Domain for a Hyperbolic Surface

Vincent Despré ✉ 🏠

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Benedikt Kolbe ✉ 🏠

Hausdorff Center for Mathematics, Universität Bonn, Germany

Hugo Parlier ✉ 🏠

Department of Mathematics, University of Luxembourg, Luxembourg

Monique Teillaud ✉ 🏠

Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Abstract

This paper exhibits and analyzes an algorithm that takes a given closed orientable hyperbolic surface and outputs an explicit Dirichlet domain. The input is a fundamental polygon with side pairings. While grounded in topological considerations, the algorithm makes key use of the geometry of the surface. We introduce data structures that reflect this interplay between geometry and topology and show that the algorithm runs in polynomial time, in terms of the initial perimeter and the genus of the surface.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Theory of computation → Computational geometry

Keywords and phrases Hyperbolic geometry, Topology, Voronoi diagram, Algorithm

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.27

Funding This work was partially supported by grant ANR-17-CE40-0033 of the French National Research Agency ANR and INTER/ANR/16/11554412/SoS of the Luxembourg National Research fund FNR. Website of the SoS project: <https://SoS.loria.fr/>.

Benedikt Kolbe: This work was done while this author was working at Université de Lorraine, CNRS, Inria, LORIA, F-54000 Nancy, France

Acknowledgements The authors wish to thank the anonymous reviewer who suggested this simpler version of Section 4.

1 Introduction and motivation

Hyperbolic surfaces and their moduli spaces play an ubiquitous role in mathematics, namely, through relationships with other areas including Riemannian geometry, number theory, geometric group theory and mathematical physics. Algorithms for surface groups, as combinatorial or topological objects, have a rich history dating back to Dehn. Recently, in part motivated by applications in other sciences [1, 17], there has been a push to understand hyperbolic structures on surfaces from the point of view of computational geometry.

Dealing with hyperbolic surfaces necessarily involves describing them – or even visualizing them – meaningfully. A fundamental domain (in the hyperbolic plane) with a side pairing is one way to determine a hyperbolic metric on the surface. Lengths of curves in a pants decomposition and their associated pasting parameters (so-called Fenchel-Nielsen coordinates) are another. No matter which construction or parameter set used, it is always interesting to know to which extent two different constructions output the “same” surface, where “same” can have different meanings. However, these representations, either by a fundamental



© Vincent Despré, Benedikt Kolbe, Hugo Parlier, and Monique Teillaud;
licensed under Creative Commons License CC-BY 4.0

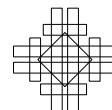
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 27; pp. 27:1–27:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



domain or a set of Fenchel-Nielsen coordinates, are not unique, and determining a canonical representation is challenging for either option. In this paper, we tackle this question for fundamental domains, by computing a so-called Dirichlet domain.

Roughly speaking, a Dirichlet domain of a hyperbolic surface is a fundamental polygon in the hyperbolic plane, with a special point where distances to that point in the polygon correspond to distances on the surface. Another way of thinking of them is that it is a Voronoi cell associated to a lift of a single point of the surface to its universal cover \mathbb{H}^2 . A more precise definition is given in the next section. Note that, any given surface has infinitely many Dirichlet domains up to isometry. This is namely due to the fact that (closed) hyperbolic surfaces are not homogeneous, and generically any choice for a basepoint of the Dirichlet domain will give a different domain. This is in strong contrast to, for example, flat tori. Nonetheless, when describing a surface via fundamental domains, the prize for the most relevant geometric domain undoubtedly goes to Dirichlet domains because they visualize the distance function for a given point. As far as we know, there is only one algorithm in the literature that computes a Dirichlet domain for a hyperbolic surface and a given point [19]. In the case of arithmetic surfaces, this algorithm has subsequently been improved and experiments indicate a significant speed up [18]. Unfortunately, the complexity of the algorithm is not studied and an analysis seems complicated.

The contribution of this paper is an algorithm that computes a Dirichlet domain efficiently, and its analysis. The point defining the domain is not given as input, but it is part of the output. The Dirichlet domain of a given input point can then be computed with a complexity that only depends of the genus of the surface [10]. Our main result is the following:

► **Theorem 1.** *Let S be a closed orientable hyperbolic surface of genus g given by a fundamental polygon of perimeter L and side pairings. A Dirichlet domain for S can be computed in $O((gL)^{6g-4})$ time.*

A key ingredient is the use of *Delaunay triangulations* on hyperbolic surfaces, an area of research that has recently gained traction, both from an experimental and a theoretical perspective [3, 7, 4, 15, 8, 12]. Recently, it has been shown that the well-known flip algorithm that computes the Delaunay triangulation of a set of points in the Euclidean plane \mathbb{E}^2 also works on a hyperbolic surface; the complexity result announced in Theorem 1 crucially depends on the only known upper bound on the complexity of this Delaunay flip algorithm [11]. The algorithm subsumes the real RAM model; studying the algebraic numbers involved in the computations goes beyond the scope of this paper.

The paper is structured as follows: In Section 3, we give an overview of the algorithm and we present the data structure. Sections 4 and 5 explain in detail the main two steps of the algorithm, which output a geometric triangulation of the surface having only one vertex. Finally, in Section 6 we conclude, using known results, the proof of Theorem 1 with the last step of the algorithm.

2 Preliminaries

We begin by recalling a collection of facts and setting notations, and we refer to [2, 5, 14] for details. The surfaces studied in this paper are assumed to be closed, orientable, and of genus $g \geq 2$. We begin with a topological surface and endow it with a hyperbolic metric to obtain a hyperbolic surface, denoted by S . A hyperbolic surface is locally isometric to its universal covering space, the hyperbolic plane \mathbb{H}^2 . Such surfaces can always be obtained by considering the quotient of \mathbb{H}^2 under the action of Γ , a discrete subgroup of isometries of \mathbb{H}^2 isomorphic to the fundamental group $\pi_1(S)$.

Let $S := \mathbb{H}^2/\Gamma$ be a hyperbolic surface of genus g and fundamental group Γ . The projection map is denoted as $\rho: \mathbb{H}^2 \rightarrow S = \mathbb{H}^2/\Gamma$. We denote by $\tilde{x} \in \rho^{-1}(x)$ one of the lifts, to \mathbb{H}^2 , of an object x on S . More generally, objects in \mathbb{H}^2 are denoted with $\tilde{\cdot}$.

A *fundamental domain* \mathcal{F} for the action of Γ is defined as a closed domain, i.e., $\overline{\text{int}(\mathcal{F})} = \mathcal{F}$, such that $\Gamma\mathcal{F} = \mathbb{H}^2$ and the interiors of different copies of \mathcal{F} under Γ are disjoint.

For a point $\tilde{x} \in \mathbb{H}^2$, the *Dirichlet domain* $\mathcal{D}_{\tilde{x}}$ is defined as the Voronoi cell containing \tilde{x} , of the Voronoi diagram associated to the point set $\Gamma\tilde{x}$. In other words,

$$\mathcal{D}_{\tilde{x}} = \{ \tilde{y} \in \mathbb{H}^2 \mid d_{\mathbb{H}^2}(\tilde{x}, \tilde{y}) \leq d_{\mathbb{H}^2}(\tilde{x}, \Gamma\tilde{y}) \} = \{ \tilde{y} \in \mathbb{H}^2 \mid d_{\mathbb{H}^2}(\tilde{x}, \tilde{y}) \leq d_{\mathbb{H}^2}(\Gamma\tilde{x}, \tilde{y}) \},$$

where the equality is true since Γ acts by isometries with respect to $d_{\mathbb{H}^2}$. The Dirichlet domain is a compact convex fundamental domain for Γ with finitely many geodesic sides [2, §9.4] and is generally considered a canonical choice of fundamental domain. A property of Dirichlet domains, of interest for the conception of algorithms, is that, by the triangle inequality,

$$\text{diam}(\mathcal{D}_{\tilde{x}}) \leq 2 \text{diam}(S) \leq 2 \text{diam}(\mathcal{D}_{\tilde{x}}),$$

where $\text{diam}(\cdot)$ denotes the diameter.

2.1 The Poincaré disk model

As they play an important part in our investigations, for completeness, we quickly recall the Poincaré disk model of the hyperbolic plane and its isometries. In this model, the hyperbolic plane \mathbb{H}^2 is represented as the open unit disk in the complex plane \mathbb{C} and the unit circle consists of the points at infinity. The geodesics for the hyperbolic metric are either diameters of the unit disk or circle arcs meeting the unit circle orthogonally.

The representations of the fundamental group of an orientable surface we're interested in consist of orientation-preserving isometries of \mathbb{H}^2 , represented by either of the two matrices

$$\pm \begin{pmatrix} a & b \\ \bar{b} & \bar{a} \end{pmatrix}$$

with $a, b \in \mathbb{C}$ such that $|a|^2 - |b|^2 = 1$. The image of a point of \mathbb{H}^2 represented by $z \in \mathbb{C}$ is the point represented by $\frac{az + b}{bz + \bar{a}}$. Composition of isometries corresponds to multiplication of either of their representing matrices.

All isometries in the fundamental group of a surface are hyperbolic translations, which are characterized by having two distinct fixed points at infinity. An isometry is a hyperbolic translation if and only if the absolute value of the trace of its matrix is larger than 2.

2.2 Curves, paths, and loops

A closed curve is the image of \mathbb{S}^1 under a continuous map and it is said to be non-trivial (or essential) if it is not freely homotopic to a point. Similarly, a path is a continuous image of the interval $[0, 1]$, and the images of 0 and 1 are referred to as its endpoints. A loop is a path whose endpoints are equal; this endpoint is referred to as its basepoint. More generally we refer to [14] for basic topological notions concerning curves on surfaces.

For a closed curve or loop c , we will denote by $[c]$ its free homotopy class, and, if c is a loop based in a point p , by $[c]_p$ its homotopy class of loops based in p . For a path c between points p and q , we denote by $[c]_{p,q}$ the homotopy class of the path with fixed endpoints. We

will readily make use of the fact that if c is a closed non-trivial curve on a hyperbolic surface, then there is a unique closed geodesic in the homotopy class $[c]$. Similarly, if c is a loop based in p , in $[c]_p$ there is a unique geodesic loop, and if c is a path between p and q , in $[c]_{p,q}$ there is a unique geodesic path. If c is a simple closed curve then the closed geodesic in $[c]$ is also simple, but this is no longer necessarily the case for loops or paths with basepoints.

The intersection number $i(c, c')$ between homotopy classes of curves c and c' is defined as the minimal intersection among its representatives. Note that closed geodesics on a hyperbolic surface always intersect minimally. The situation for paths is slightly different. The unique geodesic representatives of paths (with fixed endpoints) might not intersect minimally. For instance, take four points close together on a surface and such that their distance paths intersect. While the geodesics intersect it is possible to move one of them (keeping the end points fixed) by a homotopy such that they no longer intersect. This subtlety plays a key technical role in our story.

2.3 Fundamental polygon

Let S be a (closed) hyperbolic surface of genus g and fundamental group Γ . A polygon $P \subset \mathbb{H}^2$ (i.e., a circular sequence of geodesic edges) bounding a fundamental domain for Γ (as defined in the introduction) is called a fundamental polygon. Poincaré's theorem implies that Γ is generated by the side pairings on P [2, §9.8]. The edges and vertices of P project to a graph G_P on S ; the region enclosed by P projects to the unique face of G_P .

The numbers n_G of vertices and m_G of edges of G_P satisfy Euler's relation $n_G - m_G + 1 = 2 - 2g$, as there is only one face. It follows that if G_P only has one vertex, then that vertex is incident to the $m_G = 2g$ edges, which are actually all loops. The number of vertices is maximal when they all have degree 3 (then there are no loops); in this case $3n_G = 2m_G$, so, $m_G = 6g - 3$ and $n_G = 4g - 2$. More generally, the number $2m_G$ of edges and vertices of P lies between the two extreme cases: $4g \leq 2m_G \leq 12g - 6$. Some vertices of P project to the same vertex of G_P , i.e., they belong to the same orbit under Γ . The graph G_P has a loop for each edge of P whose vertices are in the same orbit; then the projected point on S is incident to that loop twice.

3 Algorithm overview

Let S be a (closed) hyperbolic surface of genus g and fundamental group Γ .

We propose the algorithm sketched below to compute a Dirichlet fundamental domain of S . The output of Step 1 will be denoted with primes; it will be used as input for Step 2, whose output will be denoted with double primes.

1. Find a system $\beta'_0, \dots, \beta'_{2g-1}$ of simple topological loops based at a common point b' that cuts S into a disk. Construct a topological polygon Π' from lifts of these loops, together with side pairings. This step is described precisely in Section 4.
2. Construct a point b'' so that the topological polygon Π' leads to an embedded polygon Π'' when based at a lift \tilde{b}'' of b'' (Section 5).
3. Construct the Dirichlet domain of \tilde{b}'' (Section 6).

Obviously, the complexity of the algorithm heavily depends on the data structure used to store the objects involved in the constructions. As the algorithm actually operates in the universal covering space \mathbb{H}^2 of S , it is natural to present the data structure in \mathbb{H}^2 . We assume that, as input, we are given a fundamental polygon $\Pi \subset \mathbb{H}^2$ for Γ , together with side pairings, as in Section 2.3. The data structure described below is actually equivalent to a

combinatorial map [16, Section 3.3] on S , enriched with geometric information. In particular, for each vertex x of G_Π (the projection of Π onto S , as in Section 2.3), the sequence of edges around x is ordered (edges that correspond to a loop appear twice).

Description of the input

Let a representative $\tilde{e}_i, i = 0, \dots, m - 1$ be chosen for each couple of paired edges of Π and denote as $\gamma_0, \dots, \gamma_{m-1} \in \Gamma$ the corresponding side pairings in Π : the other edge of the couple is $\gamma_i^{-1}\tilde{e}_i$, where γ_i^{-1} is the inverse of γ_i . We denote the set of the $2m$ edges of Π as E_Π and the set of its $2m$ vertices as V_Π . We choose a representative $\tilde{v}_j, j = 0, \dots, n - 1$ for each orbit of vertices of Π ; n is the number of vertices of G_Π .

Each element of Γ can be represented as a word on the alphabet $\mathcal{A}_\Gamma = \{\mathbb{1}, \gamma_0, \dots, \gamma_{m-1}, \gamma_0^{-1}, \dots, \gamma_{m-1}^{-1}\}$, where $\mathbb{1}$ denotes the identity in Γ . Here, letters of \mathcal{A}_Γ and the corresponding generators in Γ are denoted by the same symbol; this should not cause any confusion.

The core of the data structure is a doubly linked circular list of edges of Π , which stores the combinatorial information. Additional information is necessary to store the geometry (i.e., the positions of the vertices of Π in \mathbb{H}^2) and the side pairings. The data stored for each edge and vertex is constant, so the size of the data structure is $O(g)$ (we do not try to shave constants in the $O()$). Notation introduced below is illustrated by Figure 1.

Concretely, for each edge $\tilde{x} \in E_\Pi$, the data structure stores:

- two pointers $\text{prev}(\tilde{x})$ and $\text{next}(\tilde{x})$ that give access to the previous and next edges in Π , respectively (in counterclockwise order);
- two pointers $\text{source}(\tilde{x})$ and $\text{target}(\tilde{x})$ that give access to the source and target vertices of \tilde{x} in Π , respectively (in counterclockwise order); when $\rho\tilde{x}$ is a loop in G_Π , $\text{source}(\tilde{x})$ and $\text{target}(\tilde{x})$ lie in the same orbit under Γ ;
- a pointer to the paired edge $\text{pair}(\tilde{x})$ in Π ;
- a letter $w(\tilde{x}) \in \mathcal{A}_\Gamma$ that encodes the relation between \tilde{x} and $\text{pair}(\tilde{x})$:

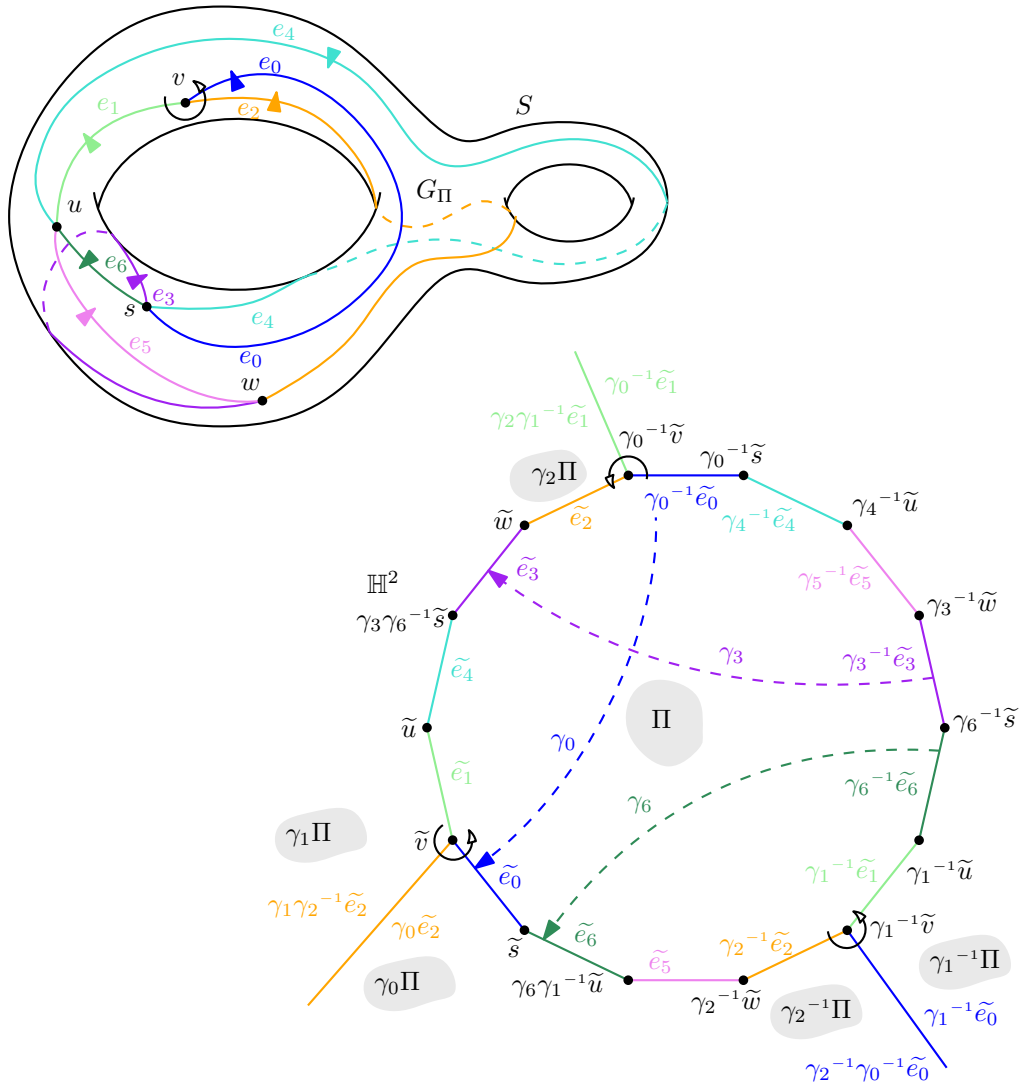
$$w(\tilde{x}) = \begin{cases} \mathbb{1} & \text{if } \tilde{x} = \tilde{e}_i \\ \gamma_i & \text{if } \tilde{x} = \gamma_i^{-1}\tilde{e}_i \end{cases} \text{ for some } i \in \{0, \dots, m - 1\}.$$

$$\text{By definition, } \text{pair}(\tilde{x}) = \begin{cases} \gamma_i^{-1}\tilde{x} & \text{when } w(\tilde{x}) = \mathbb{1} \ (\tilde{x} = \tilde{e}_i) \\ \gamma_i\tilde{x} & \text{when } w(\tilde{x}) = \gamma_i \end{cases}.$$

For each vertex $\tilde{y} \in V_\Pi$, the data structure stores:

- $\text{point}(\tilde{y})$, which is the representative point of its orbit: $\text{point}(\tilde{y}) = \tilde{v}_j$ for some $j \in \{0, \dots, n - 1\}$;
- a word $\tilde{\gamma}_y$ on \mathcal{A}_Γ (equivalently, $\tilde{\gamma}_y \in \Gamma$), which specifies the precise position $\tilde{\gamma}_y\text{point}(\tilde{y})$ of \tilde{y} in \mathbb{H}^2 .

The graph G_Π lifts in the universal cover \mathbb{H}^2 to the (infinite) graph $\rho^{-1}G_\Pi = \Gamma\Pi$. In particular, the sequence of edges of $\Gamma\Pi$ incident to a given vertex $\tilde{v} \in \rho^{-1}v$ is a sequence of lifts of the edges incident to v in G_Π . Each of these lifts is the image by an element of Γ of an edge of Π . The precise positions in \mathbb{H}^2 of all vertices of Π in the orbit $\rho^{-1}v$ can be computed using the information $\text{point}(\cdot)$ and $w(\cdot)$ stored in the data structure. Relations in the finitely presented group [6, Chapter 5.5] Γ can be deduced by comparing the two sequences of edges of $\Gamma\Pi$ – clockwise and counterclockwise – around each vertex.



■ **Figure 1** (Top) The graph G_Π . The arrow around vertex v shows its incident edges. (Bottom) The fundamental polygon Π . Vertices $\tilde{s}, \tilde{u}, \tilde{v}$, and \tilde{w} of Π are chosen as representatives of the orbits of s, u, v , and w , respectively. The arrows show the combinatorics of the tiling $\Gamma\Pi$ at the three vertices of Π in the orbit $\rho^{-1}v$: $\tilde{v}, \gamma_0\tilde{v} = \gamma_2\gamma_1^{-1}\tilde{v}$, and $\gamma_1^{-1}\tilde{v} = \gamma_2^{-1}\gamma_0\tilde{v}$.

4 Constructing the initial system of simple loops

Step 1 of the algorithm described in Section 3 is reminiscent of a construction that is quite common in the topology literature: it consists in computing a spanning tree \mathcal{T} of G_Π , then the edges of \mathcal{T} are contracted, so that each vertex of \mathcal{T} is merged into the root, and each edge of G_Π that is not an edge of \mathcal{T} is transformed into a loop based at the root.

Instead, our construction is performed in the dual of G_Π , which allows for a simpler presentation. Note also that topology is not enough in this work. We will actually compute the geometry of each loop by constructing lifted vertices in \mathbb{H}^2 and side pairings.

The main result of this section is as follows:

► **Proposition 2.** *Let S be a closed orientable surface of genus g and Π a fundamental polygon of S with $2m$ edges and side pairings as described in Section 3. A system of loops based at a common point on S , whose lifts form a topological polygon Π' in \mathbb{H}^2 , together with side pairings, can be constructed in time $O(g)$. The total geodesic length of this system of loops is $O(gL)$, where L denotes the perimeter of Π .*

The topological polygon Π' is given by a circular list of vertices in \mathbb{H}^2 ; its edges are homotopic with fixed endpoints to paths forming a fundamental domain. Π' is not a fundamental polygon as in Section 2.3: the geodesic segments between its vertices (i.e., endpoints of these paths) will intersect in general, so they do not bound a fundamental domain. Section 5 will present the construction of a fundamental polygon from this topological polygon (Step 2 of the algorithm).

The algorithm that constructs Π' proceeds in three phases:

- (i) Compute a spanning tree \mathcal{T} of G_Π .
- (ii) Compute a set of loops that topologically cuts the surface into a disk.
- (iii) Construct a topological polygon Π' by lifting the loops of the previous phase. The side pairings in Π' are also computed.

The rest of this section is devoted to proving Proposition 2, by detailing the construction.

Proof. The notation in this proof is the same as in Section 3.

Phase (i) is performed by a standard construction of a spanning tree \mathcal{T} in time $O(m)$, i.e., $O(g)$. The tree has $n - 1$ edges.

Let us denote as G_Π^* the dual of G_Π . For phase (ii), we use a tree-cotree decomposition $(\mathcal{T}, \mathcal{C}, \mathcal{X})$, which is a partition of the edges of G_Π : \mathcal{C} consists of the edges of $G_\Pi \setminus \mathcal{T}$ that are dual to the edges of a spanning tree of G_Π^* ; \mathcal{X} contains the $2g$ remaining edges. Here, \mathcal{C} is empty since G_Π has only one face, so $\mathcal{X} = G_\Pi \setminus \mathcal{T}$. Completing the $2g$ edges of \mathcal{X} with the unique paths in \mathcal{T} between their endpoints would lead to a basis of the homology [13]. Instead of considering \mathcal{X} , we actually consider the dual \mathcal{X}^* : since \mathcal{C}^* is empty, the edges of \mathcal{X}^* are already loops based at a single point and they form a basis of the homology.

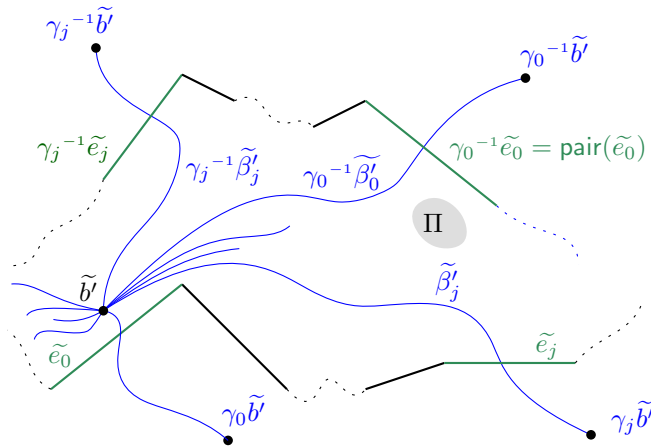
Phase (iii) computes realizations in \mathbb{H}^2 of the $2g$ loops of \mathcal{X}^* . We first choose a point \tilde{b}' in the interior of Π . The loops will be obtained by constructing $4g$ paths incident to \tilde{b}' .

Let us re-index the edges of Π so that $\tilde{\mathcal{X}} = \{\tilde{e}_i, i = 0, \dots, 2g-1\} \cup \{\gamma_i^{-1}\tilde{e}_i, i = 0, \dots, 2g-1\}$; the associated side pairings in Π are re-indexed accordingly. For each $i \in \{0, \dots, 2g-1\}$, we take a path $\tilde{\beta}'_i$ with endpoints \tilde{b}' and $\gamma_i\tilde{b}'$; the path $\gamma_i^{-1}\tilde{\beta}'_i$ has endpoints \tilde{b}' and $\gamma_i^{-1}\tilde{b}'$. The two paths $\tilde{\beta}'_i$ and $\gamma_i^{-1}\tilde{\beta}'_i$ project onto S on the same loop β'_i based at b' , and which lies in the homotopy class of the dual edge e_i^* of $e_i \in \mathcal{X}$ (Figure 2).

The order in which $\beta'_0, \dots, \beta'_{2g-1}$ appear around b' on S is simply given by the order of the elements of $\tilde{\mathcal{X}}$ around Π , which is obtained in $O(g)$ operations by traversing the boundary of Π (as mentioned in Section 2.3, each loop appears twice in the sequence). However, in general the geodesic segments between the endpoints of each path may intersect and not appear in the right order around \tilde{b}' .

To construct Π' , we will walk in the faces of the graph $\Gamma\Pi$ and construct $4g$ lifts of $\beta'_0, \dots, \beta'_{2g-1}$ and their vertices in the orbit $\Gamma b'$, which will form the ordered sequence of edges of the topological polygon Π' . Along the construction, we will create a table \mathfrak{t} that stores for each vertex \tilde{x} of Π' the element $\gamma \in \Gamma$ (as a word) such that $\tilde{x} = \gamma\tilde{b}'$. We describe the walk in the next paragraph.

Without loss of generality, we start with $\tilde{\beta}'_0$ as a first edge of Π' , incident to \tilde{b}' . Its other vertex is $\gamma_0\tilde{b}'$; we store $\mathbb{1}$ and γ_0 in \mathfrak{t} for the first two vertices of Π' . The next edge of Π' is incident to the endpoint $\gamma_0\tilde{b}'$ of $\tilde{\beta}'_0$. It is given by the dual of the first edge of Π – when



■ **Figure 2** Edges of Π , which are geodesics in \mathbb{H}^2 , are represented as straight lines. Lifts of e_0 and e_j , which belong to \mathcal{X} , are shown in green, while edges of Π that belong to \mathcal{T} are black. Lifts incident to \tilde{b}' of the loops $\beta_0, \dots, \beta_{2g-1}$ corresponding to edges of $\tilde{\mathcal{X}}$ appear as blue curves.

following Π in counterclockwise order – that is an element of $\tilde{\mathcal{X}}$, which is found in the data structure presented in Section 3 as $\text{next}(\text{next}(\dots \text{next}(\text{pair}(\tilde{e}_0))) \dots)$ (see Figure 3). Here the iteration on the edges of Π stops as soon as it encounters an element of $\tilde{\mathcal{X}}$; it is an edge \tilde{e}_j or $\gamma_j^{-1} \tilde{e}_j$, for some $j \in 0, \dots, 2g - 1$. The edge of Π' following $\tilde{\beta}'_0$ after $\gamma_0 \tilde{b}'$ is thus $\gamma_0 \tilde{\beta}'_j$ or $\gamma_0 \gamma_j^{-1} \tilde{\beta}'_j$ and its next vertex is $\gamma_0 \gamma_j \tilde{b}'$ or $\gamma_0 \gamma_j^{-1} \tilde{b}'$. We store either $\gamma_0 \gamma_j$ or $\gamma_0 \gamma_j^{-1}$ in \mathfrak{t} . We repeat the same operations for each edge of Π' . The total number of accesses to $\text{next}(\cdot)$ during the process is exactly the number of edges of Π , and there are a constant number of other operations for each edge, so, the boundary of Π' is constructed in $O(g)$ time.

The side pairing between any two paired sides is easily obtained by comparing the elements of \mathfrak{t} that yield their respective vertices when applied to \tilde{b}' .

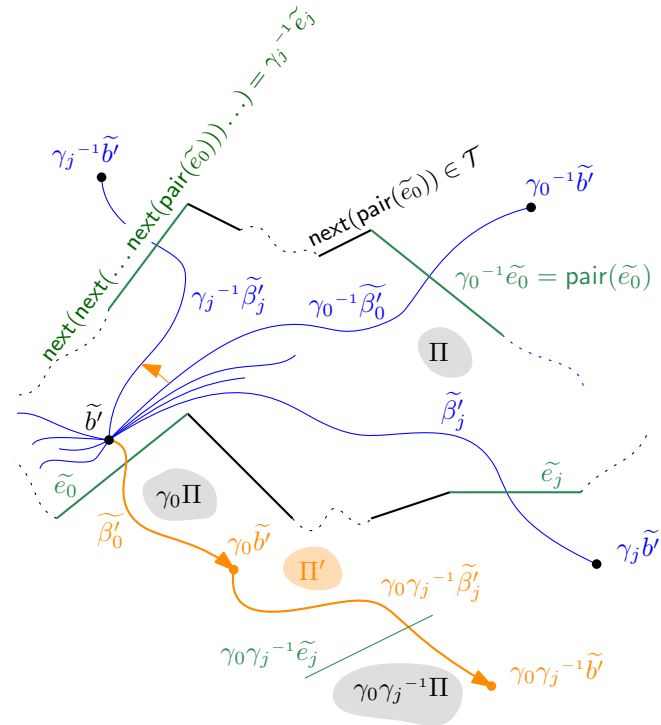
The homotopy class of each loop β'_i contains the path formed by the projection onto S of the path \tilde{c}'_i formed by the shortest path in Π between \tilde{b}' and a point on \tilde{e}_i (e.g., the midpoint), followed by the shortest path in $\gamma_i \Pi$ between this (mid)point and the other vertex $\gamma_i \tilde{b}'$ of $\tilde{\beta}'_i$ (see Figure 4). These two shortest paths are contained respectively in Π and $\gamma_i \Pi$, so, each of them is not longer than the diameter of Π , which is bounded by half the perimeter L of Π . Thus the geodesic length of β'_i , for $i = 0, \dots, 2g - 1$, is not larger than L . ◀

► **Remark 3.** The topological polygon Π' is obtained as a circular sequence of images of some lifts $\tilde{\beta}'_i, i = 0, \dots, 2g - 1$ by elements of Γ . The shape formed by the sequence of the images of all shortest paths \tilde{c}'_i (see the last paragraph of the proof) by the same elements of Γ contains a lift of the tree \mathcal{T} that is a tree in \mathbb{H}^2 .

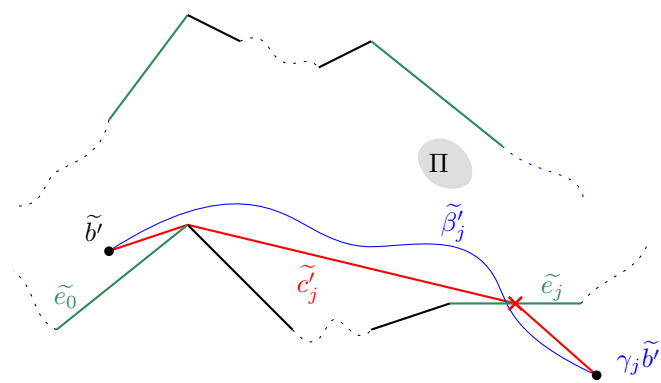
It is crucial to notice that the point \tilde{b}' from which the topological polygon Π' is constructed can be any point in the interior of Π . The following section shows how to choose a point that yields a polygon embedded in \mathbb{H}^2 .

5 Finding an embedded system of loops

We want to find a collection of geodesic loops on a hyperbolic surface S , all based in a single point and disjoint otherwise, such that the complementary region of the loops is a convex hyperbolic polygon. What we show is that in fact we can retain the choice of topological loops $\beta'_0, \dots, \beta'_{2g-1}$ made in Section 4 by moving the basepoint appropriately to ensure that their geodesic realizations satisfy the desired properties.



■ **Figure 3** Construction of the edges of Π' , colored in orange.



■ **Figure 4** Bounding the geodesic length of β'_j .

27:10 Computing a Dirichlet Domain for a Hyperbolic Surface

Consider the set of topological loops $\beta'_0, \dots, \beta'_{2g-1}$ all based at point b' constructed in the previous section. We choose a pair that intersects minimally exactly once which, up to reordering, we suppose are β'_0 and β'_1 . For future reference we set $L_0 := \max\{\ell(\beta'_0), \ell(\beta'_1)\}$, where ℓ denotes the length.

► **Remark 4.** While all of the loops are based at the same point, they each correspond to free homotopy classes of closed curves that may or may not intersect. We can actually fix *any* loop to be β'_0 and find a loop β'_1 (essentially) intersecting it exactly once (meaning that the corresponding closed curves intersect exactly once). Indeed, the set $\beta'_0, \dots, \beta'_{2g-1}$ contains curves that pairwise intersect at most once, and are all non-separating and thus homologically non-trivial. As it generates homotopy, it also generates homology and in particular every curve must be intersected by at least one other curve. As they can intersect at most once, they intersect exactly once.

We begin by taking the unique geodesic loops, based in b' , in the free homotopy classes of β'_0 and β'_1 , and we replace the curves with these geodesic representatives (we keep the same notation for convenience). Now we further consider the unique simple closed geodesic representatives in the free homotopy class of β'_0 and β'_1 , which we denote β''_0 and β''_1 , respectively. By hypothesis, they intersect in a single point b'' , which will be our new basepoint.

We now define a path between b'' and b' as follows. We consider a single lift $\widetilde{\beta}'_0$ of β'_0 . Its endpoints both correspond to distinct lifts of b' which are related by a unique translation g_0 in Γ . The copies of $\widetilde{\beta}'_0$ by iterates of g_0 form a piecewise geodesic line $\hat{\beta}'_0$ with the same endpoints at infinity as the geodesic axis of g_0 . This line $\hat{\beta}'_0$ separates \mathbb{H}^2 into two half-spaces, only one of which is convex – unless β'_0 is smooth at b' , in which case the two half-spaces are convex. We now choose an endpoint of $\hat{\beta}'_0$ and consider the lift of β'_1 that lies in the convex half-space and shares the same endpoint; in the case when the two half-spaces are convex, we can take any of them. This lift we denote by $\widetilde{\beta}'_1$ and, as before, we consider the corresponding translation g_1 in Γ and its geodesic axis and its corresponding piecewise geodesic line $\hat{\beta}'_1$. Now, we obtain \widetilde{b}'' as the intersection of the axes of g_0 and g_1 . We consider the unique geodesic path \widetilde{c} between \widetilde{b}' and \widetilde{b}'' and its projection c on S .

We observe that the axis of g_0 must lie in an R neighborhood of $\hat{\beta}'_0$ where $R < \ell(\beta'_1)$. In particular, the axis of g_1 intersects $\widetilde{\beta}'_0$ (see Figure 5). Similarly, the axis of g_0 intersects $\widetilde{\beta}'_1$. By following an arc of $\widetilde{\beta}'_0$ from \widetilde{b}' and then a segment on the axis g_1 , we obtain a path between b' to b'' .

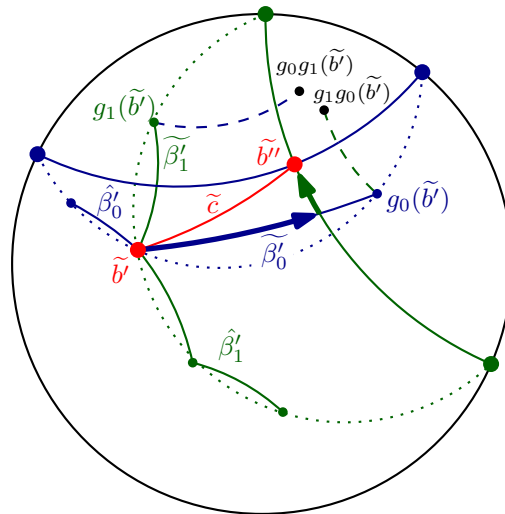
Observe that for $i = 0, 1$, β''_i , based in b'' , is freely homotopic to $c^{-1} \cdot \beta'_i \cdot c$ and that there is a homeomorphism of S , isotopic to the identity, that takes b' to b'' and that sends (the homotopy class of) β'_i to β''_i . This homeomorphism is often referred to as the point pushing map (see for instance [14, Section 4.2] for details).

We can apply this same homeomorphism to the remaining curves. For $i = 0, \dots, 2g - 1$ we set the homotopy class of loop β''_i to be:

$$[\beta''_i]_{b''} = [c^{-1} \cdot \beta'_i \cdot c]_{b''}. \quad (5.1)$$

As we have just moved the basepoint by a homeomorphism, the homotopy classes $[\beta''_i]_{b''}$ all have simple representatives and can be realized disjointly outside of b'' . The following lemma implies that their unique geodesic representatives enjoy this same property. It is well known to specialists, but we include a proof sketch for completeness.

► **Lemma 5.** *Let Σ be a hyperbolic surface with piecewise-geodesic boundary such that the interior angles on the singular points s_0, \dots, s_{k-1} of the boundary are cone points of angle $\leq \pi$. If $[\alpha]_{p_i, q_i}, [\alpha']_{p_j, q_j}$ are simple homotopy classes of paths (with endpoints p_i, p_j, q_i, q_j*



■ **Figure 5** c is homotopic on S to the projection of the concatenation of the bold arc of $\tilde{\beta}'_0$ from \tilde{b}' and the bold segment of the axis of g_1 .

in the set s_0, \dots, s_{k-1}), and disjoint except for possibly in their endpoints, then the unique geodesic representatives are also simple and disjoint.

Sketch of proof. We consider $\tilde{\Sigma}$, the universal cover of Σ , which we view as a (geodesically convex) subset of \mathbb{H}^2 . We lift $\partial\Sigma$ to $\tilde{\Sigma}$ and representatives of $[\alpha]_{p_i, q_i}$ and $[\alpha']_{p_j, q_j}$, which are simple and disjoint, to the universal cover. Observe that being simple and disjoint is equivalent to all individual lifts in \mathbb{H}^2 being simple and pairwise disjoint. Now take two individual lifts of either α or α' , and their unique geodesic representatives. We will see that they are also disjoint. Note that in general, given two simple disjoint paths in the hyperbolic (or Euclidean) plane, the unique geodesics between their endpoints might intersect (as already mentioned in Section 2.2). However:

Observation: Let $C \subset \mathbb{H}^2$ be a convex with non-empty boundary, and $p_0, q_0, p_1, q_1 \in \partial C$. Let $\alpha_1 : [0, 1] \rightarrow C$ and $\alpha_2 : [0, 1] \rightarrow C$ be simple paths, disjoint in their interior, with $\alpha_0(0) = p_0, \alpha_0(1) = q_0$ and $\alpha_1(0) = p_1, \alpha_1(1) = q_1$. Then the unique geodesic between p_0 and q_0 and the unique geodesic between p_1 and q_1 are disjoint in their interior as well.

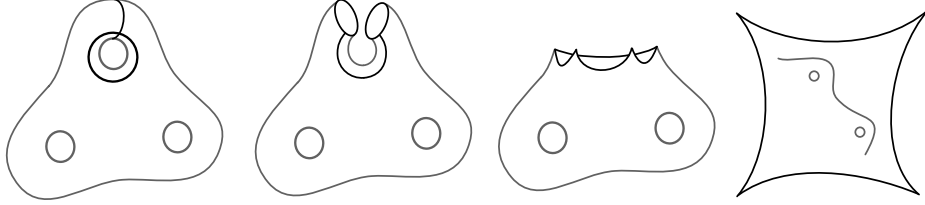
A key point is that, thanks to the angle condition on the cone points, $\tilde{\Sigma}$ is a convex region of \mathbb{H}^2 . (This is just a slightly more sophisticated observation than the elementary fact that a polygon with all interior angles less than π is convex.) The observation now implies that the lifts of geodesics corresponding to α and α' are disjoint in their interior if and only if there are representatives of $[\alpha]_{p_i, q_i}$ and $[\alpha']_{p_j, q_j}$ that are, too, which, by hypothesis, is the case. ◀

We can now apply Lemma 5 to the geodesic representatives of $[\beta''_i]_{b''}$. For simplicity we denote by β''_i the unique geodesic loop in the corresponding homotopy class.

► **Theorem 6.** Let $\beta''_0, \dots, \beta''_{2g-1}$ be a set of topological loops based in b'' that cuts a surface S into a disk. Assume that β''_0 and β''_1 are closed geodesics. Then, the geodesic loops homotopic to $\beta''_0, \dots, \beta''_{2g-1}$ are simple and pairwise disjoint in their interiors. Furthermore, by cutting S along those geodesics and lifting to \mathbb{H}^2 , one obtains a convex hyperbolic polygon with $4g$ edges.

27:12 Computing a Dirichlet Domain for a Hyperbolic Surface

Proof. As β_0'' and β_1'' are closed geodesics, they form 4 angles in b'' , and the opposite ones are equal. These angles thus satisfy $2\theta + 2\theta' = 2\pi$ so in particular both θ and θ' are strictly less than π . Thus by cutting along β_0'' and β_1'' , we obtain a genus $g - 1$ surface with a boundary consisting of 4 geodesic segments, and with 4 cone point singularities of angles $< \pi$ (see Figure 6).



■ **Figure 6** A visualization of the cutting along β_0'' and β_1'' .

We now proceed inductively for $i \geq 2$ and consider the unique geodesic path β_i'' , which by virtue of Lemma 5, has disjoint interior from the previous geodesic segments. Furthermore, as each segment further splits the angles, the angles are all less than π .

The end result is a polygon with all interior angles less than π which, by elementary hyperbolic geometry, is convex. ◀

► **Proposition 7.** *Let S be hyperbolic of genus g and Π' a topological fundamental polygon of S with $4g$ edges and side pairings as described at the end of Section 4. A convex fundamental polygon Π'' with its side pairing and whose vertices project to a single vertex on S , can be constructed in $O(g)$ time. The perimeter of Π'' is $O(gL)$, where L denotes the perimeter of Π .*

Proof. We need to compute the output convex polygon Π'' i.e., $4g$ lifts of b'' and $2g$ side pairings $\gamma_0'', \dots, \gamma_{2g-1}''$. As homotopy classes of β_i'' and β_i' are conjugates for $i = 0, \dots, 2g - 1$ (Equation 5.1), the side pairing γ_i'' is equal to γ_i' for each i .

The key point here is the computation of a lift of b'' . The first step consists in finding the loops β_0' and β_1' satisfying $i(\beta_0', \beta_1') = 1$. As shown in Remark 4, we can choose any loop for β_0' . We also fix \tilde{b}_0' to be an endpoint of one of the two paired sides of Π' that are lifts of β_0' . We know the ordered sequence of loops around b' (see the proof of Proposition 2); recall that each loop $\beta_0', \dots, \beta_{2g-1}'$ appears twice in the sequence (Section 2.3). We take as β_1' one of the loops that alternate with β_0' in the sequence, and choose for $\tilde{\beta}_1'$ one of its two lifts that are incident to \tilde{b}_0' .

The second step consists in finding the free geodesics in the homotopy classes of β_0' and β_1' , respectively. Using τ , we find the word g_0 on $\{\gamma_0', \dots, \gamma_{2g-1}'\}$ representing the translation that sends \tilde{b}_0' to the other endpoint of $\tilde{\beta}_0'$ (see Figure 7). The sequences $g_0^n(\tilde{b}_0')$ and $g_0^{-n}(\tilde{b}_0')$ converge in \mathbb{C} to two points on the unit circle: these points are the two (infinite in \mathbb{H}^2) fixed points of the translation g_0 , i.e., the two solutions of equation $g_0(z) = z$ in \mathbb{C} . The axis of g_0 , i.e., the geodesic between these two points, projects onto S to the free geodesic in $[\beta_0']$.

We repeat the same process with $\tilde{\beta}_1'$ and find the geodesic in \mathbb{H}^2 that projects to the free geodesic in $[\beta_1']$. The point \tilde{b}_0'' comes as the intersection point of the two geodesics in \mathbb{H}^2 . We now define β_0'' as the geodesic segment between \tilde{b}_0'' and $g_0(\tilde{b}_0'')$, and $\tilde{\beta}_1''$ in the same way. This step is performed in constant time.

We can now compute the $4g$ lifts of b'' that are the vertices of Π'' by applying the elements of τ to \tilde{b}_0'' . This last step has complexity $O(g)$.

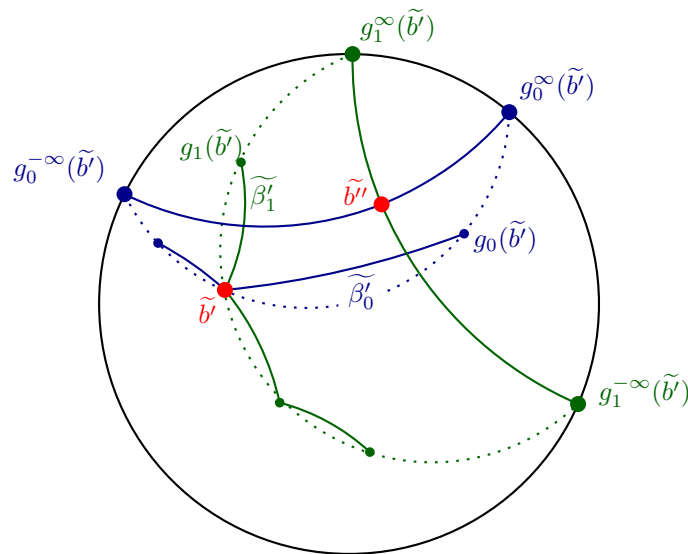


Figure 7 The computation of b'' from b' .

Recall that the upper bound on the geodesic perimeter of Π' given by Proposition 2 holds for the geodesic perimeter of Π' constructed from any basepoint b' in the interior of Π . So, it also holds for the new basepoint b'' and gives a bound on the perimeter of Π'' . ◀

6 Finding a Dirichlet domain from an embedded system of loops

We first summarize what we have obtained so far. We started with a polygon Π of perimeter L and we obtained a convex polygon Π'' of total length $O(gL)$. Additionally, all vertices of Π'' project on a single vertex b'' on S . This construction has complexity $O(g)$ by Propositions 2 and 7.

For the sake of clarity, the two steps 1 and 2 of the algorithm (Section 3) were distinct and were presented in different sections (respectively, Sections 4 and 5). However, we can remark that they could have been merged by placing the point b' , which is left undefined in Section 4, directly at b'' as defined in Section 5. This is probably what would be done in practice.

Now it is easy to compute a Dirichlet domain. Indeed, we can triangulate Π'' easily since it is convex and, thus, we obtain a geometric triangulation T , on to which the Delaunay flip algorithm can be applied [11]. The complexity of this algorithm depends on the diameter of T , for which the perimeter of Π'' is an upper bound.

► Remark 8. Remark 3 can be expressed in a much simpler way at this point: Π'' itself contains a lift $\tilde{\mathcal{T}}$ of \mathcal{T} . As Π'' is convex, the geodesic segments dual to the edges of $\tilde{\mathcal{T}}$ are contained in Π'' and induce a partition of Π'' . We can actually use these segments to triangulate Π'' . In this way, in the case when the input polygon Π is already a Dirichlet domain, we already have a Delaunay tessellation, and no flip is performed by the algorithm.

The output of the flip algorithm is a Delaunay triangulation DT of S with the single vertex b'' computed in Section 5. To obtain a Dirichlet domain from DT , we just have to compute the triangles of \tilde{DT} incident to a lift \tilde{b}'' of b'' and their dual: we compute the circumcenter of each triangle to get the vertices of the Dirichlet domain and we put a geodesic between vertices that correspond to adjacent triangles around \tilde{b}'' . This step is also clearly done in $O(g)$ operations. Putting all together we obtain the following theorem:

► **Theorem 9.** *Let S be a closed orientable hyperbolic surface of genus g given by a fundamental polygon of perimeter L and side pairings. A Dirichlet domain of S can be computed in time $O(f(gL) + g)$ where $f(\Delta)$ is the complexity of the flip algorithm for a triangulation of diameter Δ with a single vertex.*

Using the best known bound $O(\Delta^{6g-4})$ for the flip algorithm so far [11], we obtain Theorem 1 stated in the introduction as a corollary. Note that the constant in the $O()$ depends on the metric on S . However, there are experimental and theoretical insights suggesting that the actual complexity of the flip algorithm may be much better [9].

References

- 1 N.L. Balazs and A. Voros. Chaos on the pseudosphere. *Physics Reports*, 143(3):109–240, 1986. doi:10.1016/0370-1573(86)90159-6.
- 2 Alan F. Beardon. *The Geometry of Discrete Groups*. Springer-Verlag, 1983.
- 3 Mikhail Bogdanov, Olivier Devillers, and Monique Teillaud. Hyperbolic Delaunay complexes and Voronoi diagrams made practical. *Journal of Computational Geometry*, 5:56–85, 2014. doi:10.20382/jocg.v5i1a4.
- 4 Mikhail Bogdanov, Monique Teillaud, and Gert Vegter. Delaunay triangulations on orientable surfaces of low genus. In Sándor Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry (SoCG 2016)*, volume 51 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:17, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2016.20.
- 5 Peter Buser. *Geometry and spectra of compact Riemann surfaces*. Modern Birkhäuser classics. Birkhäuser, Boston, Mass., 2nd edition, 2010.
- 6 H. S. M. Coxeter and W. O. J. Moser. *Generators and Relations for Discrete Groups*. Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1957.
- 7 Jason DeBlois. The centered dual and the maximal injectivity radius of hyperbolic surfaces. *Geometry and Topology*, 19(2):953–1014, 2015. doi:10.2140/gt.2015.19.953.
- 8 Jason DeBlois. The Delaunay tessellation in hyperbolic space. *Mathematical Proceedings of the Cambridge Philosophical Society*, 164(1):15–46, 2018. doi:10.1017/S0305004116000827.
- 9 Vincent Despré, Loïc Dubois, Benedikt Kolbe, and Monique Teillaud. Experimental analysis of Delaunay flip algorithms on genus two hyperbolic surfaces. Preprint, INRIA, May 2021. URL: <https://hal.inria.fr/hal-03462834/>.
- 10 Vincent Despré, Benedikt Kolbe, and Monique Teillaud. Representing infinite hyperbolic periodic Delaunay triangulations using finitely many Dirichlet domains. Preprint, INRIA, July 2021. URL: <https://hal.inria.fr/hal-03045921>.
- 11 Vincent Despré, Jean-Marc Schlenker, and Monique Teillaud. Flipping geometric triangulations on hyperbolic surfaces. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry (SoCG 2020)*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 35:1–35:16, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2020.35.
- 12 Matthijs Ebbens, Hugo Parlier, and Gert Vegter. Minimal Delaunay triangulations of hyperbolic surfaces. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:16, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2021.31.
- 13 David Eppstein. Dynamic generators of topologically embedded graphs. In *Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 599–608, 2003.
- 14 Benson Farb and Dan Margalit. *A Primer on Mapping Class Groups (PMS-49)*. Princeton University Press, 2012. URL: <http://www.jstor.org/stable/j.ctt7rkjw>.

- 15 Iordan Iordanov and Monique Teillaud. Implementing Delaunay triangulations of the Bolza surface. In Boris Aronov and Matthew J. Katz, editors, *33rd International Symposium on Computational Geometry (SoCG 2017)*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:15, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2017.44.
- 16 Bojan Mohar and Carsten Thomassen. *Graphs on Surfaces*. Johns Hopkins University Press, Baltimore, 2001.
- 17 Nikolai C Passler, Xiang Ni, Guangwei Hu, Joseph R Matson, Giulia Carini, Martin Wolf, Mathias Schubert, Andrea Alù, Joshua D Caldwell, Thomas G Folland, et al. Hyperbolic shear polaritons in low-symmetry crystals. *Nature*, 602(7898):595–600, 2022. doi:10.1038/s41586-021-04328-y.
- 18 James Rickards. Improved computation of fundamental domains for arithmetic Fuchsian groups. *Mathematics of Computation*, 91:2929–2954, 2022. doi:10.1090/mcom/3777.
- 19 John Voight. Computing fundamental domains for Fuchsian groups. *Journal de Théorie des Nombres de Bordeaux*, 21(2):467–489, 2009. doi:10.5802/jtnb.683.

The Parameterized Complexity of Coordinated Motion Planning

Eduard Eiben  

Department of Computer Science, Royal Holloway, University of London, Egham, UK

Robert Ganian  

Algorithms and Complexity Group, TU Wien, Austria

Iyad Kanj  

School of Computing, DePaul University, Chicago, IL, USA

Abstract

In Coordinated Motion Planning (CMP), we are given a rectangular-grid on which k robots occupy k distinct starting gridpoints and need to reach k distinct destination gridpoints. In each time step, any robot may move to a neighboring gridpoint or stay in its current gridpoint, provided that it does not collide with other robots. The goal is to compute a schedule for moving the k robots to their destinations which minimizes a certain objective target – prominently the number of time steps in the schedule, i.e., the makespan, or the total length traveled by the robots. We refer to the problem arising from minimizing the former objective target as CMP-M and the latter as CMP-L. Both CMP-M and CMP-L are fundamental problems that were posed as the computational geometry challenge of SoCG 2021, and CMP also embodies the famous $(n^2 - 1)$ -puzzle as a special case.

In this paper, we settle the parameterized complexity of CMP-M and CMP-L with respect to their two most fundamental parameters: the number of robots, and the objective target. We develop a new approach to establish the fixed-parameter tractability of both problems under the former parameterization that relies on novel structural insights into optimal solutions to the problem. When parameterized by the objective target, we show that CMP-L remains fixed-parameter tractable while CMP-M becomes para-NP-hard. The latter result is noteworthy, not only because it improves the previously-known boundaries of intractability for the problem, but also because the underlying reduction allows us to establish – as a simpler case – the NP-hardness of the classical Vertex Disjoint and Edge Disjoint Paths problems with constant path-lengths on grids.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases coordinated motion planning, multi-agent path finding, parameterized complexity, disjoint paths on grids

Digital Object Identifier 10.4230/LIPIcs.SocG.2023.28

Funding *Robert Ganian*: Project No. Y1329 of the Austrian Science Fund (FWF), Project No. ICT22-029 of the Vienna Science Foundation (WWTF).

Iyad Kanj: DePaul URC Grants 606601 and 350130.

1 Introduction

Who among us has not struggled through solving the 15-puzzle? Given a small square board, tiled with 15 tiles numbered $1, \dots, 15$, and a single hole in the board, the goal of the puzzle is to slide the tiles in order to reach the final configuration in which the tiles appear in (sorted) order; see Figure 1 for an illustration. The 15-puzzle has been generalized to an $n \times n$ square-board, with tiles numbered $1, \dots, n^2 - 1$. Unsurprisingly, this generalization is called the $(n^2 - 1)$ -puzzle. Whereas deciding whether a solution to an instance of the $(n^2 - 1)$ -puzzle exists (i.e., whether it is possible to sort the tiles starting from an initial configuration) is in P [20], determining whether there is a solution that requires at most $\ell \in \mathbb{N}$ tile moves has been shown to be NP-hard [9, 26].



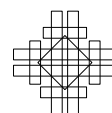
© Eduard Eiben, Robert Ganian, and Iyad Kanj;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

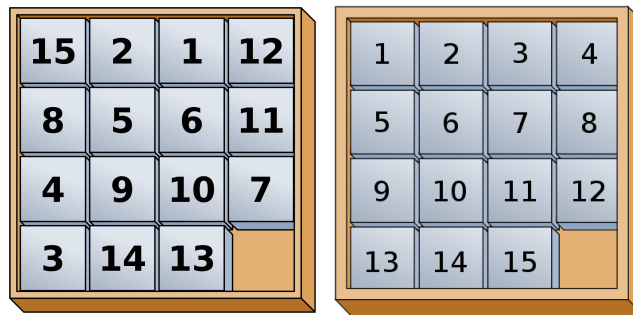
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 28; pp. 28:1–28:16

Leibniz International Proceedings in Informatics



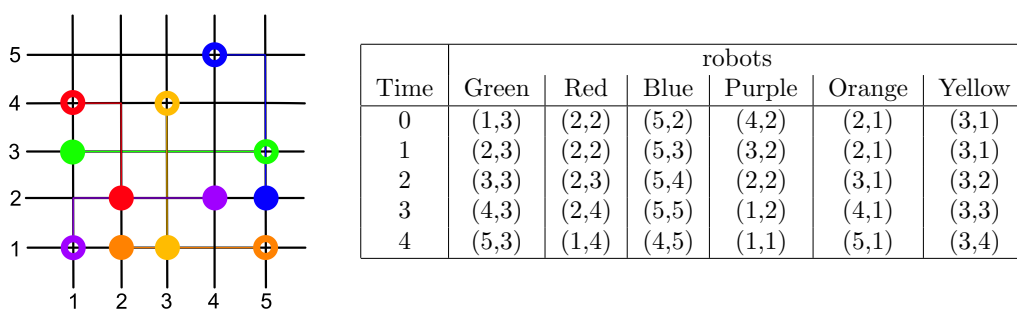
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** The left figure shows an initial configuration of the 15-puzzle and the right figure shows the desirable final configuration. Source: https://en.wikipedia.org/wiki/15_puzzle.

Deciding whether an $(n^2 - 1)$ -puzzle admits a solution is a special case of Coordinated Motion Planning (CMP), a prominent task originating from robotics which has been extensively studied in the fields of Computational Geometry and Artificial Intelligence (where it is often referred to as Multi-Agent Path Finding). In CMP, we are given an $n \times m$ rectangular-grid on which k robots occupy k distinct starting gridpoints and need to reach k distinct destination gridpoints. Robots may move simultaneously at each time step, and at each time step, a robot may move to a neighboring gridpoint or stay in its current gridpoint provided that (in either case) it does not collide with any other robots; two robots collide if they are occupying the same gridpoint at the end of a time step, or if they are traveling along the same grid-edge (in opposite directions) during the same time step. We are also given an objective target, and the goal is to compute a schedule for moving the k robots to their destination gridpoints which satisfies the specified target. The two objective targets we consider here are (1) the number of time steps used by the schedule (i.e., the makespan), and (2) the total length traveled by all the robots (also called the “total energy”, e.g., in the SoCG 2021 Challenge [1]); the former gives rise to a problem that we refer to as CMP-M, while we refer to the latter as CMP-L. An illustration is provided in Figure 2.



■ **Figure 2** An illustration (left) of an instance of CMP-M with six robots, indicated using distinct colors (blue, green, yellow, red, orange, purple), and a makespan $\ell = 4$. The starting points are marked using a disk shape (filled circle) and destination points using an annular shape. A schedule indicating each of the robot’s position at each of the four time steps is shown in the table (right).

In this paper, we settle the parameterized complexity of CMP-M and CMP-L with respect to their two most fundamental parameters: the number k of robots, and the objective target. In particular, we obtain fixed-parameter algorithms for both problems when parameterized

by k and for CMP-L when parameterized by the target, but show that CMP-M remains NP-hard even for fixed values of the target. Given how extensively CMP has been studied in the literature (see the related work below), we consider it rather surprising that fundamental questions about the problem’s complexity have remained unresolved. We believe that one aspect contributing to this gap in our knowledge was the fact that, even though the problems seem deceptively easy, it was far from obvious how to obtain exact and provably optimal algorithms in the parameterized setting. Furthermore, en route to the aforementioned intractability result, we establish the NP-hardness of the classical VERTEX DISJOINT PATHS and EDGE DISJOINT PATHS problems on grids when restricted to bounded-length paths.

1.1 Related Work

CMP has been extensively studied by researchers in the fields of computational geometry, AI/Robotics, and theoretical computer science in general. In particular, CMP-M and CMP-L were posed as the Third Computational Geometry Challenge of SoCG 2021, which took place during the Computational Geometry Week in 2021 [1]. The CMP problem generalizes the $(n^2 - 1)$ -puzzle, which was shown to be NP-hard as early as 1990 by Ratner and Warmuth [26]. A simpler NP-hardness proof was given more recently by Demaine et al. [9]. Several recent papers studied the complexity of CMP with respect to optimizing various objective targets, such as: the makespan, the total length traveled, the maximum length traveled (over all robots), and the total arrival time [2, 8, 15, 32]. The continuous geometric variants of CMP, in which the robots are modeled as geometric shapes (e.g., disks) in a Euclidean environment, have also been extensively studied [3, 8, 12, 25, 27]. Finally, we mention that there is a plethora of works in the AI and Robotics communities dedicated to variants of the CMP problem, for both the continuous and the discrete settings [4, 17, 28, 29, 31, 33, 34].

The fundamental vertex and edge disjoint paths problems have also been thoroughly studied, among others due to their connections to graph minors theory. The complexity of both problems on grids was studied as early as in the 1970’s motivated by its applications in VLSI design [13, 21, 24, 30], with more recent results focusing on approximation [5, 6].

1.2 High-Level Overview of Our Results and Contributions

As our first set of results, we show that CMP-M and CMP-L are fixed-parameter tractable (FPT) parameterized by the number k of robots, i.e., can be solved in time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f and input size n . Both results follow a two-step approach for solving each of these problems. In the first step, we obtain a structural result revealing that every YES-instance of the problem has a *canonical* solution in which the number of “turns” (i.e., changes in direction) made by any robot-route is upper bounded by a function of the parameter k ; this structural result is important in its own right, and we believe that its applications extend beyond this paper. This first step of the proof is fairly involved and revolves around introducing the notion of “slack” to partition the robots into two types, and then exploiting this notion to reroute the robots so that their routes form a canonical solution. In the second step, we show that it is possible to find such a canonical solution (or determine that none exists) via a combination of delicate branching and solving subinstances of Integer Linear Programming (ILP) in which the number of variables is upper bounded by a function of the parameter k ; fixed-parameter tractability then follows since the latter can be solved in FPT-time thanks to Lenstra’s result [14, 16, 19].

Next, we consider the other natural parameterization of the problem: the objective target. For CMP-L, this means parameterizing by the total length traveled, and there we establish fixed-parameter tractability via exhaustive branching. The situation becomes much more

intriguing for CMP-M, where we show that the problem remains NP-hard even when the target makespan is a fixed constant. As a by-product of our reduction, we also establish the NP-hardness of the classical Vertex and Edge Disjoint Paths problems on grids when restricted to bounded-length paths.

The contribution of our intractability results are twofold. First, the NP-hardness of CMP with constant makespan is the first result showing its NP-hardness in the case where one of the parameters is a fixed constant. As such, it refines and strengthens several existing NP-hardness results for CMP [2, 8, 15]. It also answers the open questions in [15] about the complexity of the problem in restricted settings where the optimal path of each robot passes through a constant number of starting/destination points, or where the overlap between any two optimal paths is upper bounded by a constant, by directly implying their NP-hardness. Second, the NP-hardness results for the bounded-length vertex and edge disjoint paths problems on grids also refine and deepen several intractability results for these problems. All previous NP-hardness (and APX-hardness) results for the vertex and edge disjoint paths problems on grids [2, 6, 8, 9, 15, 21, 24, 26] yield instances in which the path length is unbounded. Last but not least, we believe that the NP-hardness results we derive are of independent interest, and have the potential of serving as a building block in NP-hardness proofs for problems in geometric and topological settings, where it is very common to start from a natural problem whose restriction to instances embedded on a grid remains NP-hard.

2 Preliminaries and Problem Definition

We use standard terminology for graph theory [10] and assume basic familiarity with the parameterized complexity paradigm including, in particular, the notions of *fixed-parameter tractability* and *para-NP-hardness* [7, 11]. For $n \in \mathbb{N}$, we write $[n]$ for the set $\{1, \dots, n\}$.

Let G be an $n \times m$ rectangular grid, where $n, m \in \mathbb{N}$. Let $\{R_i \mid i \in [k]\}$, $k \in \mathbb{N}$, be a set of robots that will move on G . Each R_i , $i \in [k]$, is associated with a starting gridpoint s_i and a destination gridpoint t_i in $V(G)$, and hence can be specified as the pair $R_i = (s_i, t_i)$; we assume that all the s_i 's are pairwise distinct and that all the t_i 's are pairwise distinct, and we denote by $\mathcal{R} = \{(s_i, t_i) \mid i \in [k]\}$ the set of all robots. At each time step, a robot may either stay at the gridpoint it is currently on, or move to an adjacent gridpoint, and robots may move simultaneously. We reference the sequence of moves of the robots using a time frame $[0, t]$, $t \in \mathbb{N}$, and where in time step $x \in [0, t]$ each robot remains stationary or moves.

Let a *route* for R_i be a tuple $W_i = (u_0, \dots, u_t)$ of vertices in G such that (i) $u_0 = s_i$ and $u_t = t_i$ and (ii) $\forall j \in [t]$, either $u_{j-1} = u_j$ or $u_{j-1}u_j \in E(G)$. Intuitively, W_i corresponds to a “walk” in G , with the exception that consecutive vertices in W_i may be identical (representing waiting time steps), in which R_i begins at its starting point at time step 0, and is at its destination point at time step t . Two routes $W_i = (u_0, \dots, u_t)$ and $W_j = (v_0, \dots, v_t)$, where $i \neq j \in [k]$, are *non-conflicting* if (i) $\forall r \in \{0, \dots, t\}$, $u_r \neq v_r$, and (ii) $\nexists r \in \{0, \dots, t-1\}$ such that $v_{r+1} = u_r$ and $u_{r+1} = v_r$. Otherwise, we say that W_i and W_j *conflict*. Intuitively, two routes conflict if the corresponding robots are at the same vertex at the end of a time step, or go through the same edge (in opposite directions) during the same time step.

A *schedule* \mathcal{S} for \mathcal{R} is a set of routes W_i , $i \in [k]$, during a time interval $[0, t]$, that are pairwise non-conflicting. The integer t is called the *makespan* of \mathcal{S} . The (*traveled*) *length* of a route (or its associated robot) within \mathcal{S} is the number of time steps j such that $u_j \neq u_{j+1}$, and the *total traveled length* of a schedule is the sum of the lengths of its routes.

We are now ready to define the problems under consideration.

COORDINATED MOTION PLANNING WITH MAKESPAN MINIMIZATION (CMP-M)

Given: An $n \times m$ rectangular grid G , where $n, m \in \mathbb{N}$, and a set $\mathcal{R} = \{(s_i, t_i) \mid i \in [k]\}$ of pairs of gridpoints of G where the s_i 's are distinct and the t_i 's are distinct; $k, \ell \in \mathbb{N}$.

Question: Is there a schedule for \mathcal{R} of makespan at most ℓ ?

The COORDINATED MOTION PLANNING WITH LENGTH MINIMIZATION problem (CMP-L) is defined analogously but with the distinction being that, instead of ℓ , we are given an integer λ and are asked for a schedule of total traveled length at most λ . For an instance \mathcal{I} of CMP-M or CMP-L, we say that a schedule is *valid* if it has makespan at most ℓ or has total traveled length at most λ , respectively. We remark that even though both CMP-M and CMP-L are stated as decision problems, all the algorithms provided in this paper are constructive and can output a valid schedule (when it exists) as a witness.

We will assume throughout the paper that $k \geq 2$; otherwise, both problems can be solved in linear time. Furthermore, we remark that the inputs can be specified in $\mathcal{O}(k \cdot (\log n + \log m) + \log \ell)$ (or $+\log \lambda$) bits, and our fixed-parameter algorithms work seamlessly even if the inputs are provided in such concise manner. On the other hand, the lower-bound result establishes “strong” NP-hardness of the problem (i.e., also applies to cases where the input contains a standard encoding of G as a graph).

For two gridpoints $p = (x_p, y_p)$ and $q = (x_q, y_q)$, the Manhattan distance between p and q , denoted $\Delta(p, q)$, is $\Delta(p, q) = |x_p - x_q| + |y_p - y_q|$. For two robots $R_i, R_j \in \mathcal{R}$ and a time step $x \in \mathbb{N}$, denote by $\Delta_x(R_i, R_j)$ the Manhattan distance between the grid points at which R_i and R_j are located at time step x . The following notion will be used in several of our algorithms:

► **Definition 1.** Let $(G, \mathcal{R}, k, \bullet)$ be an instance of CMP-M or CMP-L and let $T = [t_1, t_2]$ for $t_1, t_2 \in \mathbb{N}$. For a robot R_i with corresponding route W_i , let u_p and u_q be the gridpoints in W_i at time steps t_1 and t_2 , respectively. Define the *slack* of R_i w.r.t. T , denoted $\mathbf{slack}_T(R_i)$, as $(t_2 - t_1) - \Delta(u_p, u_q)$ (alternatively, $(q - p) - \Delta(u_p, u_q)$).

Observe that the slack measures the amount of time (i.e., number of time steps) that robot R_i “wastes” when going from u_p to u_q relative to the shortest time needed to get from u_p to u_q . For a robot R_i with route W_i , for convenience we write $\mathbf{slack}_T(W_i)$ for $\mathbf{slack}_T(R_i)$. When dealing with CMP-M, we write $\mathbf{slack}(R_i)$ as shorthand for $\mathbf{slack}_{[0, \ell]}(R_i)$, and when dealing with CMP-L, we write $\mathbf{slack}(R_i)$ as shorthand for $\mathbf{slack}_{[0, \lambda]}(R_i)$.

3 CMP Parameterized by the Number of Robots

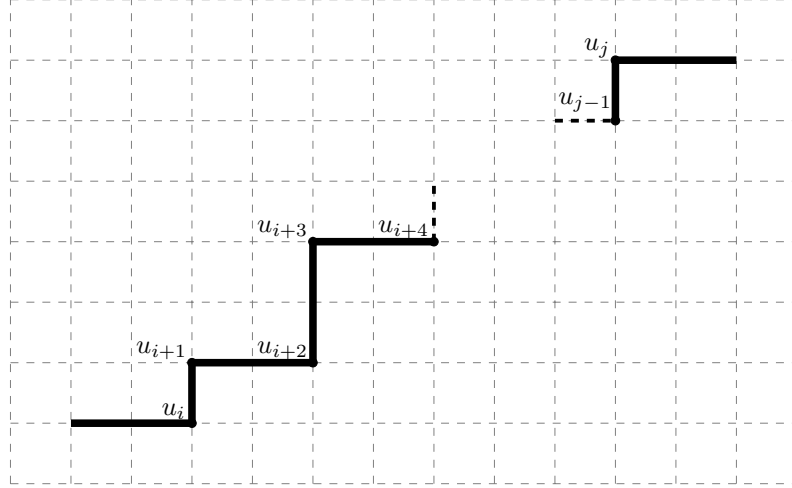
In this section, we establish the fixed-parameter tractability of CMP-M and CMP-L parameterized by the number k of robots. Both results follow the two-step approach outlined in Subsection 1.2: showing the existence of a canonical solution, and then reducing the problem via branching to a tractable fragment of Integer Linear Programming. These two steps are described for CMP-M in Subsections 3.1 and 3.2, while Subsection 3.3 shows how the same technique is used to establish the fixed-parameter tractability of CMP-L.

3.1 Canonical Solutions for CMP-M

We begin with a few definitions that formalize some intuitive notions such as “turns”.

Let $W = (u_0, \dots, u_\ell)$, where $\ell > 2$, be a route in an $n \times m$ grid G , where $n, m \in \mathbb{N}$. We say that W makes a *turn* at $u_i = (x_i, y_i)$, where $i \in \{1, \dots, \ell - 1\}$, if the two vectors $\overrightarrow{u_{i-1}u_i}$ and $\overrightarrow{u_iu_{i+1}}$ have different orientations (i.e., either one is horizontal and the other is vertical,

or they are parallel but have opposite directions). We write $\langle u_{i-1}, u_i, u_{i+1} \rangle$ for the turn at u_i . A turn $\langle u_{i-1}, u_i, u_{i+1} \rangle$ is a *U-turn* if $\overrightarrow{u_{i-1}u_i} = -\overrightarrow{u_iu_{i+1}}$; otherwise, it is a *non U-turn*. The *number of turns* in W , denoted $\nu(W)$, is the number of vertices in W at which it makes turns. A sequence $M = [u_i, \dots, u_j]$ of consecutive turns is said to be *monotone* if all the turns in each of the two alternating sequences $[u_i, u_{i+2}, u_{i+4}, \dots]$ and $[u_{i+1}, u_{i+3}, u_{i+5}, \dots]$, in which M can be partitioned, have the same direction (see Figure 3).



■ **Figure 3** Illustration of a monotone sequence of consecutive turns.

Let $T = [t_1, t_2] \subseteq [0, \ell]$. We say that a route W_i for R_i has *no slack* in T if $\text{slack}_T(R_i) = 0$; that is, robot R_i does not “waste” any time and always progresses towards its destination during T . The following observation is straightforward:

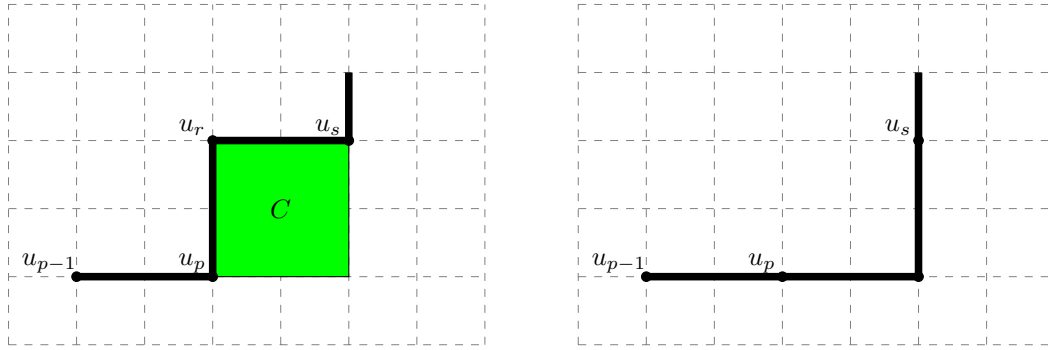
► **Observation 2.** Let W_i be a route for R_i and $T \subseteq [0, \ell]$ be a time interval such that $\text{slack}_T(R_i) = 0$. The sequence of turns that W_i makes during T is a monotone sequence (and in particular does not include any *U-turns*).

Let $W_i = (s_i = u_0, \dots, u_t = t_i)$ be a route for R_i in a valid schedule \mathcal{S} of a YES-instance of CMP-M or CMP-L, and let $W = (u_q, u_{q+1}, \dots, u_r)$ be the subroute of W_i during a time interval $T \subseteq [0, t]$. We say that a route $W' = (v_q, \dots, v_r)$ is *equivalent* to W if: (i) $v_q = u_q$ and $v_r = u_r$ (i.e., both routes have the same starting and ending points); (ii) $|W| = |W'|$; and (iii) replacing W_i in \mathcal{S} with the route $(s_i = u_0, \dots, u_{q-1}, v_q, \dots, v_r, u_{r+1}, \dots, u_t = t_i)$ still yields a valid schedule of the instance.

► **Definition 3.** Let $\mathcal{I} = (G, \mathcal{R}, k, \bullet)$ be a YES-instance of CMP-M or CMP-L. A valid schedule \mathcal{S} for $(G, \mathcal{R}, k, \bullet)$ is *minimal* if the sum of the number of turns made by all the routes in \mathcal{S} is minimum over all valid schedules of \mathcal{I} .

The following lemma is the building block for the crucial Lemma 5, which will establish the existence of a canonical solution (for a YES-instance) in which the number of turns made by “small-slack” robots is upper bounded by a function of the parameter. This is achieved by a careful application of a “cell flattening” operation depicted in Figure 4.

More specifically, we show that if in a solution a robot has no slack during a time interval but its route makes a “large” number of turns, then there exists a “cell” corresponding to a turn in its route that can be flattened, resulting in another (valid) solution with fewer turns.



■ **Figure 4** Illustration of a *cell* in a route (left) and its *flattening* (right).

► **Lemma 4.** *Let \mathcal{S} be a minimal (valid) schedule for a YES-instance of CMP-M. Let W_i be a route in \mathcal{S} and $T_i \subseteq [0, \ell]$ be a time interval during which W_i has no slack. Then there is an equivalent route, W'_i , to W_i such that the number of turns that W'_i makes during T_i , $\nu_{T_i}(W'_i)$, satisfies $\nu_{T_i}(W'_i) \leq 3k^k$.*

By carefully subdividing a time interval into roughly $\sigma(k)$ subintervals, for a function $\sigma(k)$ that upper bounds the slack of a robot, and applying Lemma 4 to each of these subintervals, we can extend the result in Lemma 4 to robots whose slack is upper bounded by $\sigma(k)$:

► **Lemma 5.** *Let $(G, \mathcal{R}, k, \ell)$ be a YES-instance of CMP-M, and let $T_i \subseteq [0, \ell]$. Then $(G, \mathcal{R}, k, \ell)$ has a minimal schedule such that, for each $R_i, i \in [k]$, satisfying $\text{slack}_{T_i}(W_i) \leq \sigma(k)$ for an arbitrary function σ , its route W_i satisfies $\nu_{T_i}(W_i) \leq \tau(k)$, where $\tau(k) = 3k^k(\sigma(k) + 1) + \sigma(k)$.*

Lemma 5 already provides us with the property we need for “small-slack” robots: their number of turns can be upper-bounded by a function of the parameter. We still need to deal with the more complicated situation of “large-slack” robots. Our next course of action will be establishing the existence of a sufficiently large time interval during which the “large-slack” robots are far from the “small-slack” ones. We begin with an observation linking the slack of two robots that “travel together”.

► **Observation 6.** *Let $R, R' \in \mathcal{R}$ and let $T = [t_1, t_2] \subseteq [0, \ell]$. Let u, u' be the gridpoints at which R and R' are located at time step t_1 , respectively, and v, v' those at which R and R' are located at time t_2 , respectively. Suppose that $\Delta(u, u') \leq d(k)$ and $\Delta(v, v') \leq d(k)$, for some function $d(k)$. Then $\text{slack}_T(R') \leq \text{slack}_T(R) + 2d(k)$.*

Intuitively speaking, the above observation implies that a robot with a large slack in some time interval cannot be close to a robot with a small slack for the whole interval (otherwise, both robots would be moving at “comparable speeds”, which would contradict that one of them has a small slack and the other a large-slack).

Next, we observe that either the slack of all the robots can be upper-bounded by a function h , or there is a sufficiently large multiplicative gap between the slack of some robots. This will allow us to partition the set of robots into those with small or large slack. For any function h , let $h^{(j)} = \underbrace{h \circ \dots \circ h}_{j \text{ times}}$ denote the composition of h with itself j times.

► **Lemma 7.** *Let $(G, \mathcal{R}, k, \ell)$ be an instance of CMP-M and let $T \subseteq [0, \ell]$. Let $h(k)$ be any computable function satisfying $h^{(p)}(k) \leq h^{(q)}(k)$ for $p \leq q \in [k]$. Then either $\text{slack}_T(R_i) \leq h^{(k)}(k)$ for every $i \in [k]$, or there exists $j \in \mathbb{N}$ with $2 \leq j \leq k$, such that \mathcal{R} can be partitioned into $(\mathcal{R}_S, \mathcal{R}_L)$ where $\mathcal{R}_L \neq \emptyset$, $\text{slack}_T(R) \leq h^{(j-1)}(k)$ for every $R \in \mathcal{R}_S$, and $\text{slack}_T(R') > h^{(j)}(k)$ for every $R' \in \mathcal{R}_L$.*

The next definition yields a time interval with the property that small-slack robots are sufficiently far from large-slack ones during that interval. Such an interval will be useful, since within it we will be able to re-route the large-slack robots (which are somewhat flexible) to reduce the number of turns they make, while avoiding collision with small-slack robots.

► **Definition 8.** Let $\sigma(k), \gamma(k), d(k)$ be functions such that $\sigma(k) < \gamma(k)$. An interval $T = [t_1, t_2] \subseteq [0, \ell]$ is a $[\sigma, \gamma]$ -good interval w.r.t. $d(k)$ if \mathcal{R} can be partitioned into \mathcal{R}_S and \mathcal{R}_L such that: (i) every $R \in \mathcal{R}_S$ satisfies $\text{slack}_T(R) \leq \sigma(k)$ and every $R' \in \mathcal{R}_L$ satisfies $\text{slack}_T(R') \geq \gamma(k)$; (ii) for every time step $t \in T$, $\Delta_t(R, R') \geq d(k)$ for every $R \in \mathcal{R}_S$ and every $R' \in \mathcal{R}_L$; and (iii) there exists a robot $R_i \in \mathcal{R}_L$ such that $\nu_T(W_i) > 3k^k(\sigma(k)+1)+\sigma(k)$. If the function $d(k)$ is specified or clear from the context, we will simply say that T is a $[\sigma, \gamma]$ -good interval (and thus omit writing “w.r.t. $d(k)$ ”).

The following key lemma asserts the existence of a good interval assuming the solution contains a robot that makes a large number of turns:

► **Lemma 9.** *Let $(G, \mathcal{R}, k, \ell)$ be a YES-instance of CMP-M and let \mathcal{S} be a minimal schedule for $(G, \mathcal{R}, k, \ell)$. If there exists $R' \in \mathcal{R}$ with route W' such that $\nu(W') > 3^{k^3+1} \cdot (3k^k \cdot (3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}} + 1) + 3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}})$, then there exists a $[\sigma, \gamma]$ -good interval $T \subseteq [0, \ell]$ w.r.t. a function $d(k)$ such that $k^{13^{k-1}} \leq \sigma(k) \leq 3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}}$, and $d(k) = \gamma(k) = \sigma^{13}(k)$.*

Once we fix a good interval T , we can finally formalize/specify what it means for a robot to have small or large slack within T :

► **Definition 10.** Let $T = [t_1, t_2] \subseteq [0, \ell]$ be a $[\sigma, \gamma]$ -good interval with respect to some function $d(k)$, where $\sigma(k) < \gamma(k)$ are two functions, and let $R_i \in \mathcal{R}$. We say that R_i is a T -large slack robot if $\text{slack}_T(R_i) \geq \gamma(k)$; otherwise, $\text{slack}_T(R_i) \leq \sigma(k)$ and we say that R_i is a T -small slack robot.

At this point, we are finally ready to prove Lemma 11, which is the core tool that establishes the existence of a solution with a bounded number of turns (w.r.t. the parameter), even in the presence of large-slack robots: for each solution with too many turns, we can produce a different one with strictly less turns. Note that if one simply replaces the routes of large-slack robots so as to reduce their number of turns, then the new routes may bring the large-slack robots much closer to the small-sack robots and hence may lead to collisions. Therefore, the desired rerouting scheme needs to be carefully designed, and it exploits the properties of a good interval: property (i) is used to reorganize and properly reroute these robots, while property (ii) is used to avoid collisions.

► **Lemma 11.** *Let $(G, \mathcal{R}, k, \ell)$ be a YES-instance of CMP-M and let \mathcal{S} be a minimal schedule for $(G, \mathcal{R}, k, \ell)$. Let $T = [t_1, t_2] \subseteq [0, \ell]$ be a $[\sigma, \gamma]$ -good interval with respect to $d(k)$, where $k^{13^{k-1}} \leq \sigma(k) \leq 3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}}$, and $d(k) = \gamma(k) = \sigma^{13}(k)$. For every T -large-slack robot R_i , there is a route W'_i that is equivalent to W_i and such that $\nu_T(W'_i)$ is at most $3k^3$ and W'_i is identical to W_i in $[0, \ell] \setminus T$.*

We now establish the canonical-solution result that forms the culmination of this section.

► **Theorem 12.** *Let $(G, \mathcal{R}, k, \ell)$ be an instance of CMP-M such that at least one dimension of the grid G is lower bounded by $2k \cdot (3k^k(3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}} + 1) + 3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}}) + 4k$. If $(G, \mathcal{R}, k, \ell)$ is a YES-instance, then it has a valid schedule \mathcal{S} in which each route makes at most $\rho(k) = 3^{k^3+1} \cdot (3k^k(3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}} + 1) + 3^{13^{2k^2-2k}} \cdot k^{13^{2k^2-2k}})$ turns.*

Proof. Suppose that $(G, \mathcal{R}, k, \ell)$ is a YES-instance of CMP-M. We proceed by contradiction. Let \mathcal{S} be a minimal schedule for $(G, \mathcal{R}, k, \ell)$ and assume that \mathcal{S} has a route W_i for R_i that makes more than $\rho(k)$ turns. By Lemma 9, there exists a $[\sigma, \gamma]$ -good interval $T \subseteq [0, \ell]$ such that $\nu_T(W_i) > 3k^k(\sigma(k) + 1) + \sigma(k)$, where σ and γ are the function specified in Lemma 9. By Lemma 11, there is an equivalent route W'_i to W_i that agrees with W_i outside of T and such that $\nu_T(W'_i) \leq 3k^3 < 3k^k(\sigma(k) + 1) + \sigma(k)$, which contradicts the minimality of \mathcal{S} . ◀

3.2 Finding Canonical Solutions

Having established the existence of canonical solutions with a bounded number of turns, we can proceed to describe the proof of the FPT result. In the proof, we identify a “combinatorial snapshot” of a solution whose size is upper-bounded by a function of the parameter k . We then branch over all possible combinatorial snapshots and, for each such snapshot, we reduce the problem of determining whether there exists a corresponding solution to an instance of Integer Linear Programming in which the number of variables is upper-bounded by a function of the parameter, which can be solved in FPT-time by existing algorithms [14, 16, 19].

In particular, the aforementioned combinatorial snapshot will be a tuple $(G_{snap}, \mathcal{R}_{snap}, W_{snap}, \iota)$ where G_{snap} is a bounded-size subgrid, \mathcal{R}_{snap} is a tuple of k pairs of starting and ending vertices in G_{snap} , W_{snap} specifies a set of routes connecting the individual starting and ending vertices, and ι contains information about the order in which vertices are visited by the routes in W_{snap} . For each snapshot, we construct an ILP instance with variables that capture (1) the amount of “expansion” necessary to go from the snapshot to the full input grid, and (2) the amount of waiting a robot performs at certain “critical” junctions in the route. Constraints are then used to ensure that each robot arrives in time, that the routes correspond to the information in ι and do not lead to conflicts, and finally that the amount of expansion needed matches the size of the input grid.

► **Theorem 13.** *CMP-M is FPT parameterized by the number of robots.*

3.3 Minimizing the Total Traveled Length

In this subsection, we discuss how the strategy for establishing the fixed-parameter tractability of CMP-M parameterized by the number k of robots can be used for CMP-L.

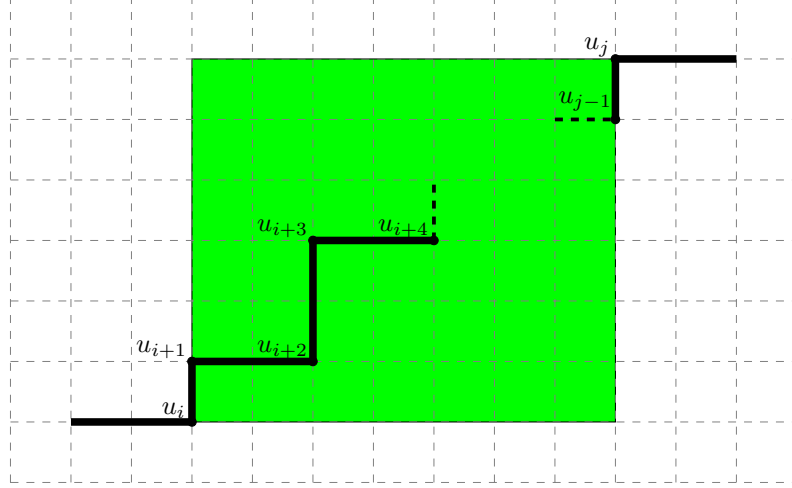
The main difference between the two problems can be intuitively stated as follows: for CMP-M “time matters” but travel length could be lax, whereas for CMP-L “travel length matters” but time can be lax. The key tool we use to handle the complications arising in CMP-L when showing the existence of a canonical solution is a result that exhibits a schedule for any instance of CMP-L whose travel length is within a quadratic additive factor in k from any length-optimal solution. Denote by dist_{min} the sum of the Manhattan distances, over all the robots, between the starting point of the robot and its destination point. We have:

► **Theorem 14.** *Let $\mathcal{I} = (G, \mathcal{R}, k, \lambda)$ be a YES-instance of CMP-L. There is a schedule \mathcal{S} for \mathcal{I} satisfying that the total travel length of \mathcal{S} is at most $\text{dist}_{min} + c(k)$, where $c(k) = \mathcal{O}(k^2)$ is a computable function, and in which the number of turns made by each robot is $\mathcal{O}(k)$.*

28:10 The Parameterized Complexity of Coordinated Motion Planning

The above theorem is then exploited for showing that if a robot makes a large number of turns, then we can find a time interval and a large rectangle of the grid such that, during that time interval, all the robots that are present in that rectangle behave “nicely”. We formalize these notions in the following definitions:

Let $M = [u_i, \dots, u_j]$ be a monotone sequence of turns made by a robot $R \in \mathcal{R}$ during some time interval. The *rectangle* of M , denoted $\text{rectangle}(M)$, is the rectangle with diagonally-opposite vertices u_i and u_j . We refer to Figure 5 for illustration.



■ **Figure 5** $\text{rectangle}(M)$ (the green-shaded area) for a monotone sequence $M = [u_1, \dots, u_j]$.

► **Definition 15.** Let W be a subroute of a robot $R \in \mathcal{R}$ during some time interval T such that the sequence M of turns in W is monotone. Let $\sigma(k)$ be a function to be specified later. We say that $\text{rectangle}(M)$ is *good* w.r.t. $\sigma(k)$ and a time subinterval $T' \subseteq T$ if: (i) the set of robots present in $\text{rectangle}(M)$ is the same during each time step of T' ; (ii) each robot R_i present in $\text{rectangle}(M)$ during T' satisfies $\text{slack}_{T'}(R_i) \geq \sigma(k)$; (iii) each robot R_i present in $\text{rectangle}(M)$ during T' is traveling in the same direction as (the directions of the turns in) M ; and (iv) each robot R_i present in $\text{rectangle}(M)$ during T' satisfies $\nu_{T'}(W_i) \geq \sigma(k)$.

Next, we show that if a robot makes a large number of turns, then a good rectangle exists:

► **Lemma 16.** Let $\mathcal{I} = (G, \mathcal{R}, k, \lambda)$ be a YES-instance of CMP-L, let \mathcal{S} be a valid schedule for \mathcal{I} , and assume that $\lambda < \text{dist}_{\min} + c(k)$, where $c(k) = \mathcal{O}(k^2)$ is the computable function in Theorem 14. Let $\sigma(k) = 4k^2$ and $\tau(k) = 3k^k(\sigma(k) + 1) + \sigma(k)$. Let R be a robot such that the walk W of R during the time interval T spanning \mathcal{S} satisfies $\nu(W) = \Omega(\tau(k)^{2k+1})$. Then there exists a subwalk W' for R and a time interval $T' \subseteq T$ such that the sequence of turns M' in W' corresponding to T' is monotone and $\text{rectangle}(M')$ is good w.r.t. $\sigma(k)$ and T' .

Using Theorem 14 and Lemma 16, we can prove that, given a good rectangle, we can reroute the robots that are present in that rectangle during a certain time interval so as to reduce the number of turns they make, which leads to the existence of a canonical solution:

► **Theorem 17.** If $\mathcal{I} = (G, \mathcal{R}, k, \lambda)$ is a YES-instance of CMP-L, then \mathcal{I} has a valid schedule \mathcal{S} in which each route makes at most $\mathcal{O}(\tau(k)^{2k+1})$ turns, where $\tau(k) = 3k^k(\sigma(k) + 1) + \sigma(k)$, and $\sigma(k) = 4k^2$.

At this point, we can turn to the second step of our approach, notably checking whether an instance of CMP-L admits a solution in which the number of turns is upper-bounded by a function of the parameter. Luckily, here the proof of Theorem 13 can be reused almost as-is, with only a single change in the ILP encoding at the end.

► **Theorem 18.** *CMP-L is FPT parameterized by the number of robots.*

4 CMP Parameterized by the Objective Target

Having resolved the parameterization by the number k of robots, we now turn our attention to the second fundamental measure in CMP problems, notably the objective target. Unlike the case where we parameterize by the number k of robots, here the complexity of the problem strongly depends on the considered variant. We begin by establishing the fixed-parameter tractability of CMP-L parameterized by λ via an exhaustive branching algorithm. The rest of this section then deals with the significantly more complicated task of establishing the intractability of CMP-L parameterized by ℓ .

► **Theorem 19.** *CMP-L is FPT parameterized by the objective target λ .*

4.1 Intractability of CMP-M with Small Makespans

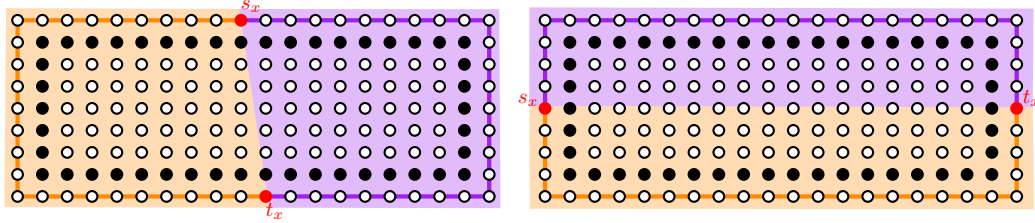
The aim of this subsection is to establish that CMP-M is NP-hard even when the makespan ℓ is upper bounded by a constant. Before we proceed to show this NP-hardness result for CMP-M, we will establish the NP-hardness of d -BOUNDED LENGTH VERTEX DISJOINT PATHS on grids, as well as its edge variant d -BOUNDED LENGTH EDGE DISJOINT PATHS, which can be seen as a stepping stone for the para-NP-hardness proof for CMP-M. In fact, the NP-hardness result for these two classical disjoint paths problems on grids with constant path lengths is significant in its own right, as discussed earlier in the paper.

All our reductions start from 4-BOUNDED PLANAR 3-SAT, a problem which is known to be NP-complete [18, 22]. The *incidence graph* of a CNF formula is the graph whose vertices are the variables and clauses of the formula, and in which two vertices are adjacent if and only if one is a variable, the other is a clause, and the variable-vertex occurs either as a positive or a negative literal in the clause-vertex. In 4-BOUNDED PLANAR 3-SAT, we are asked to evaluate a CNF formula whose incidence graph is planar and in which each clause contains exactly 3 distinct literals and each variable occurs in at most 4 clauses. On the other hand, in the aforementioned d -BOUNDED LENGTH VERTEX (resp. EDGE) DISJOINT PATHS problems, we are given a graph with a set of vertex-pairs (called *requests*), and are asked to determine if there is a set of vertex (resp. edge) disjoint paths containing an s - t path of length at most $d \in \mathbb{N}$ for every $(s, t) \in R$.

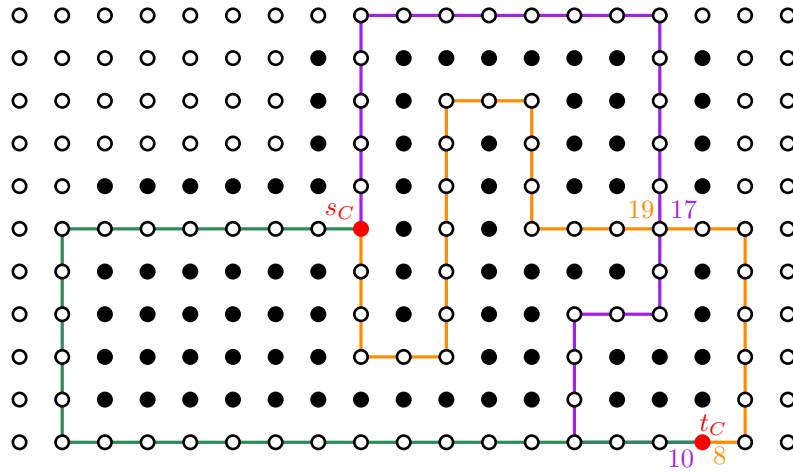
For all three reductions, consider an instance φ of 4-BOUNDED PLANAR 3-SAT and let G_φ be its incidence graph. We start with an orthogonal drawing Ω of G_φ in a polynomial-size grid. Our first goal is to show how to encode the satisfiability of φ as an instance of d -BOUNDED LENGTH VERTEX DISJOINT PATHS on grids; the reduction for d -BOUNDED LENGTH EDGE DISJOINT PATHS is almost the same, and both can be seen as a stepping stone towards CMP-M. We encode variable assignment and clause satisfaction using bounded-length path requests that conform to the drawing Ω . To model a variable-assignment, we create a variable gadget with a single request between two vertices, s and t , on this gadget such that this request can be fulfilled by selecting one of the two s - t paths in this gadget, each of length 27. Selecting one of the two paths corresponds to assigning the variable a truth value; an illustration is provided in Figure 6. We model clause-satisfaction by creating, for each clause,

28:12 The Parameterized Complexity of Coordinated Motion Planning

a clause-gadget, where a clause-gadget for a clause C contains two vertices, s_C and t_C , with a request between them that can be fulfilled in one of three ways, each corresponding to choosing a length-27 path between s_C and t_C in the gadget (see Figure 7).



■ **Figure 6** Variable Gadget examples. In both cases, there is a request (s_x, t_x) . Each of the full black circles is a request (v, v) forcing only two different paths of length at most 27 between s_x and t_x . Examples of a left-right variable gadget (left) and a top-bottom variable gadget (right).

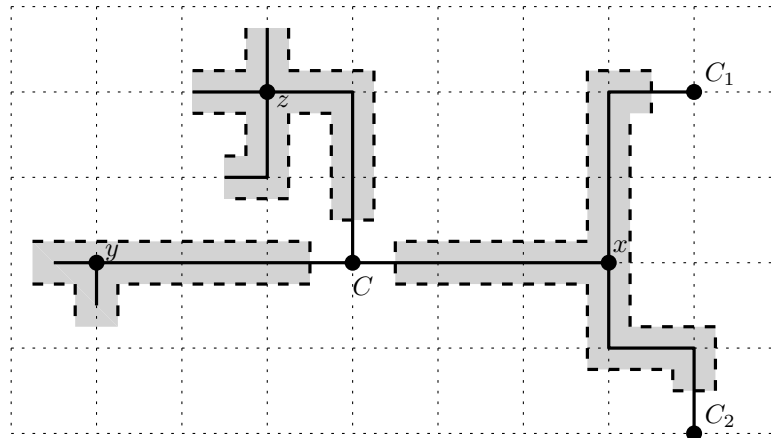


■ **Figure 7** Clause gadget example. There is a request (s_C, t_C) . Each of the full black circles is a request (v, v) . There are three possible ways to leave s_C . Choosing to go left forces us to take the green path of length 27. The orange path going down reaches the intersection point with the purple path (going up) after 19 steps on the orange path, but only 17 on the purple. Hence, the purple can choose between going down and taking 10 steps to reach t_C , or going right and taking 8 more steps, but the orange is forced to go right and reach t_C in 8 steps from the intersection point.

To implement the above idea, we needed to overcome several issues. First, the position of a variable in the embedding could be very far from the position of the clauses that it is incident to, hence prohibiting us from using bounded-length requests to encode the variable-clause incidences. Second, due to planarity constraints, embedding the three paths corresponding to a clause-gadget such that each intersects a different variable gadget, is only possible if two of the clause-paths intersect, which could create shortcuts (i.e., paths that do not intersect the variable gadgets). Third, requests may use grid paths that are not part of the embedding.

To handle the first issue, instead of using a single variable-gadget per variable, we use a “cycle” of copies of variable gadgets such that a variable assignment in any gadget of this cycle forces the same variable assignment in all copies, thus ensuring assignment consistency. The clause gadget for C is placed around the position of the vertex corresponding to C in Ω , whereas the cycle corresponding to a variable x is placed around the edges of Ω joining the

position of x in Ω to that of C ; see Figure 8. To fit all the variable cycles around a clause gadget in the embedding, we use a connection gadget, which is a path of copies of variable gadgets propagating the same variable assignment as in the corresponding variable cycle.



■ **Figure 8** Part of an orthogonal drawing of G_φ . Clause C contains variables x, y, z . The variable x is also in clauses C_1 and C_2 . The dashed lines represent the variable cycles.

To model clause-satisfaction for a clause C , each of the three s_C - t_C paths in the clause gadget of C overlaps with a copy of a variable gadget corresponding to one of the variables whose literal occurs in C . If an assignment to variable x whose literal occurs in C does not satisfy C , then the path corresponding to this assignment in the copies of the variable gadgets for x intersects the s_C - t_C path corresponding to x in the clause-gadget of C , thus prohibiting the simultaneous choice of these clause-path and variable path.

To handle the second issue, we prevent any shortcuts from being taken by making each created shortcut longer than the prescribed upper bound on the path length (i.e., 27).

Finally, to handle the third issue, when dealing with vertex disjoint paths we can artificially place an obstacle on a vertex v in the grid to “block” that vertex (i.e., to prevent it from being used by any path other than (v, v)) by adding the request (v, v) , thus forcing the set of possible paths between s and t for every request (s, t) , where $s \neq t$, to be chosen from the paths prescribed by the encoding of the instance of 4-BOUNDED PLANAR 3-SAT. A slight extension of this idea also works for edge disjoint paths. This allows us to establish:

► **Theorem 20.** *d -BOUNDED LENGTH VERTEX DISJOINT PATHS and d -BOUNDED LENGTH EDGE DISJOINT PATHS are NP-hard even when restricted to instances where $d = 27$ and G is a grid-graph.*

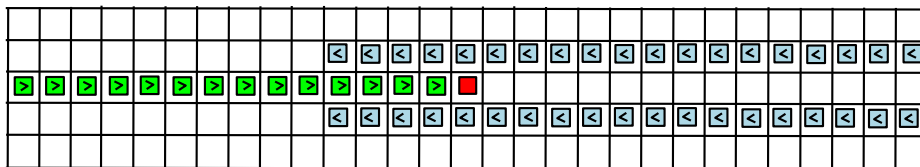
These high-level ideas are then used to obtain the targeted NP-hardness proof of CMP-M for a fixed makespan of 26, by having an (s, t) path request correspond to routing some robot from its starting gridpoint s to its destination gridpoint t . However, the way we force robots to follow the prescribed paths here is completely different and presents the main difficulty when going from d -BOUNDED LENGTH VERTEX DISJOINT PATHS on grids to CMP-M; in particular, it is no longer possible to block certain points on the grid by creating “dummy requests”. To ensure that the prescribed paths are followed, we block certain regions of the embedding by adding a large number of auxiliary non-stationary robots, and coordinating their motion so that they block the desired regions while still allowing the original robots to follow the set of paths prescribed by the encoding; this task turns out to be highly technical.

28:14 The Parameterized Complexity of Coordinated Motion Planning

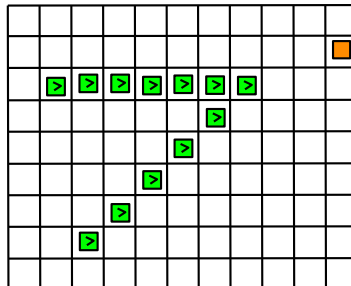
We introduce a set of new gadgets whose role is to force the main robots in the reduction to follow the paths prescribed by the embedding. Those gadgets are dynamic, as opposed to the “static blocker gridpoints” used in the d -BOUNDED LENGTH VERTEX DISJOINT PATHS on grids reduction. The two main new gadgets employed are a gadget simulating a “stream” of robots and a gadget simulating an “arrow” of robots.

The stream gadget consists of a relatively large number of robots, all moving along the same line, such that each needs to move precisely the makespan many steps in the same direction, and hence cannot afford to waste a single time step. The robots in the stream gadget will be used to either push the main robots in a certain direction, or to prevent them from taking shorter paths than the prescribed ones. See Figure 9 for an illustration. In the figure, the main red robot is pushed right by the green stream and forced to move right along the same horizontal line by the two blue streams sandwiching it.

Figure 10 shows an example of an arrow gadget. In this gadget, there is an orange robot whose destination is 26 steps somewhere down and to the left. The gadget is again a “stream” of robots that force the orange robot to select one of the two directions towards its destination in the first step, and then to stick to this selection for a number of steps that depends on the size of the arrow. For example, in Figure 10, there is a “right arrow” of green robots that all want to go 26 steps right. Since the orange robot has a slack of 0, the right arrow forces it to either take the first 5 steps all to the left, or the first 7 steps all down.



■ **Figure 9** Example of streams.



■ **Figure 10** An example of an arrow.

Other gadgets are needed to ensure that robots in the stream and arrow gadgets do not collide with anything. Using such enforcement gadgets, we can simulate the gadgets constructed in the reduction for d -BOUNDED LENGTH VERTEX DISJOINT PATHS on grids, thus encoding the instance of 4-BOUNDED PLANAR 3-SAT as an instance of CMP-M.

► **Theorem 21.** *CMP-M is NP-hard even when restricted to instances where $\ell = 26$.*

5 Conclusion

In this work, we settled the parameterized complexity of both CMP-M and CMP-L with respect to their two most fundamental parameters: the number of robots, and the objective target. Along the way, we established the NP-hardness of the classical Vertex Disjoint Paths

and the Edge Disjoint Paths problem with constant path-lengths on grids, strengthening the existing lower bounds for these problems as well. Our results reveal structural insights into the properties of optimal solutions that may also prove useful in contexts that lie outside of this work. We conclude by stating two open questions that arise from our work.

1. What is the parameterized complexity of other variants of CMP, such as the ones where the objective is to minimize the maximum length traveled or the total arrival time?
2. Can the fixed-parameter tractability of CMP-M or CMP-L parameterized by the number k of robots be lifted to grids with obstacles/holes, or more generally to planar graphs? It is worth noting that neither the structural results developed in this paper, nor other known techniques [23], seem to be applicable to these more general settings.

References

- 1 Computational Geometry Week: The Third Computational Geometry Challenge. <https://cse.buffalo.edu/socg21/challenge.html>.
- 2 Jacopo Banfi, Nicola Basilico, and Francesco Amigoni. Intractability of time-optimal multirobot path planning on 2D grid graphs with holes. *IEEE Robotics and Automation Letters*, 2(4):1941–1947, 2017.
- 3 Bahareh Banyassady, Mark de Berg, Karl Bringmann, Kevin Buchin, Henning Fernau, Dan Halperin, Irina Kostitsyna, Yoshio Okamoto, and Stijn Slot. Unlabeled multi-robot motion planning with tighter separation bounds. In *SoCG*, volume 224 of *LIPICs*, pages 12:1–12:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- 4 Eli Boyarski, Ariel Felner, Roni Stern, Guni Sharon, David Tolpin, Oded Betzalel, and Solomon Eyal Shimony. ICBS: Improved conflict-based search algorithm for multi-agent pathfinding. In *IJCAI*, pages 740–746, 2015.
- 5 Julia Chuzhoy and David H. K. Kim. On Approximating Node-Disjoint Paths in Grids. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPICs)*, pages 187–211, 2015.
- 6 Julia Chuzhoy, David Hong Kyun Kim, and Rachit Nimavat. Almost polynomial hardness of node-disjoint paths in grids. *Theory of Computing*, 17(6):1–57, 2021.
- 7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.
- 8 Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Henk Meijer, and Christian Scheffer. Coordinated motion planning: Reconfiguring a swarm of labeled robots with bounded stretch. *SIAM Journal on Computing*, 48(6):1727–1762, 2019.
- 9 Erik D. Demaine and Mikhail Rudoy. A simple proof that the $(n^2 - 1)$ -puzzle is hard. *Theoretical Computer Science*, 732:80–84, 2018.
- 10 Reinhard Diestel. *Graph Theory, 4th Edition*, volume 173 of *Graduate texts in mathematics*. Springer, 2012.
- 11 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
- 12 Adrian Dumitrescu. Motion planning and reconfiguration for systems of multiple objects. In Sascha Kolski, editor, *Mobile Robots*, chapter 24. IntechOpen, Rijeka, 2007.
- 13 András Frank. Disjoint paths in a rectilinear grid. *Combinatorica*, 2:361–371, 1982.
- 14 András Frank and Éva Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.
- 15 Tzvika Geft and Dan Halperin. Refined hardness of distance-optimal multi-agent path finding. In *AAMAS*, pages 481–488, 2022.
- 16 Jr. H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

- 17 Peter E. Hart, Nils J. Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- 18 Md. Manzurul Hasan, Debajyoti Mondal, and Md. Saidur Rahman. Positive planar satisfiability problems under 3-connectivity constraints. *Theoretical Computer Science*, 917:81–93, 2022. doi:10.1016/j.tcs.2022.03.013.
- 19 Ravi Kannan. Minkowski’s convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987. doi:10.1287/moor.12.3.415.
- 20 D. Kornhauser, G. Miller, and P. Spirakis. Coordinating pebble motion on graphs, the diameter of permutation groups, and applications. In *FOCS*, pages 241–250, 1984.
- 21 Mark Kramer. The complexity of wire-routing and finding minimum area layouts for arbitrary VLSI circuits. *Advances in Computing Research*, 2:129–146, 1984.
- 22 Jan Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Discrete Applied Mathematics*, 52(3):233–252, 1994. doi:10.1016/0166-218X(94)90143-0.
- 23 Daniel Lokshтанov, Pranabendu Misra, Michal Pilipczuk, Saket Saurabh, and Meirav Zehavi. An exponential time parameterized algorithm for planar disjoint paths. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1307–1316. ACM, 2020.
- 24 Dániel Marx. Eulerian disjoint paths problem in grid graphs is NP-complete. *Discrete Applied Mathematics*, 143(1):336–341, 2004.
- 25 Geetha Ramanathan and V.S. Alagar. Algorithmic motion planning in robotics: Coordinated motion of several disks amidst polygonal obstacles. In *ICRA*, volume 2, pages 514–522, 1985.
- 26 Daniel Ratner and Manfred Warmuth. The $(n^2 - 1)$ -puzzle and related relocation problems. *Journal of Symbolic Computation*, 10(2):111–137, 1990.
- 27 Jacob T. Schwartz and Micha Sharir. On the piano movers’ problem: III. coordinating the motion of several independent bodies: The special case of circular bodies moving amidst polygonal barriers. *The International Journal of Robotics Research*, 2:46–75, 1983.
- 28 Guni Sharon, Roni Stern, Ariel Felner, and Nathan R. Sturtevant. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence*, 219:40–66, 2015.
- 29 Irving Solis, James Motes, Read Sandström, and Nancy M. Amato. Representation-optimal multi-robot motion planning using conflict-based search. *IEEE Robotics Autom. Lett.*, 6(3):4608–4615, 2021.
- 30 Hitoshi Suzuki, Akira Ishiguro, and Takao Nishizeki. Edge-disjoint paths in a grid bounded by two nested rectangles. *Discrete Applied Mathematics*, 27(1):157–178, 1990.
- 31 Glenn Wagner and Howie Choset. Subdimensional expansion for multirobot path planning. *Artificial Intelligence*, 219:1–24, 2015.
- 32 Jingjin Yu. Constant factor time optimal multi-robot routing on high-dimensional grids. In Hadas Kress-Gazit, Siddhartha S. Srinivasa, Tom Howard, and Nikolay Atanasov, editors, *Robotics: Science and Systems XIV, Carnegie Mellon University, Pittsburgh, Pennsylvania, USA, June 26-30, 2018*, 2018.
- 33 Jingjin Yu and Steven M. LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In Marie desJardins and Michael L. Littman, editors, *AAAI*. AAAI Press, 2013.
- 34 Jingjin Yu and Steven M. LaValle. Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. *IEEE Transactions on Robotics*, 32(5):1163–1177, 2016.

Non-Crossing Hamiltonian Paths and Cycles in Output-Polynomial Time

David Eppstein ✉

Computer Science Department, University of California, Irvine, CA, USA

Abstract

We show that, for planar point sets, the number of non-crossing Hamiltonian paths is polynomially bounded in the number of non-crossing paths, and the number of non-crossing Hamiltonian cycles (polygonalizations) is polynomially bounded in the number of surrounding cycles. As a consequence, we can list the non-crossing Hamiltonian paths or the polygonalizations, in time polynomial in the output size, by filtering the output of simple backtracking algorithms for non-crossing paths or surrounding cycles respectively. To prove these results we relate the numbers of non-crossing structures to two easily-computed parameters of the point set: the minimum number of points whose removal results in a collinear set, and the number of points interior to the convex hull. These relations also lead to polynomial-time approximation algorithms for the numbers of structures of all four types, accurate to within a constant factor of the logarithm of these numbers.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Design and analysis of algorithms

Keywords and phrases polygonalization, non-crossing structures, output-sensitive algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.29

Related Version *Full Version*: <https://arxiv.org/abs/2303.00147>

Funding *David Eppstein*: Research supported in part by NSF grant CCF-2212129.

1 Introduction

In how many ways can we “connect the dots”, turning planar points into the vertices of a simple polygon? Despite heavy study, the answer is still unclear. Steinhaus proved in the 1960s that general-position points always have at least one polygonalization [32], but the condition is too strong: non-collinearity suffices. Points in convex position have only one polygonalization; other inputs can have exponentially many, but with upper and lower bounds that are far from matching [16, 31]. The complexity of counting polygonalizations is unknown [13, 22, 24]. Although all polygonalizations can be listed in singly-exponential time [35, 37], it was unknown (prior to our work) how to list them more quickly when there are few, for instance in polynomial time per output or in time polynomial in the output size.

Our main result is that both polygonalizations and a closely related structure, non-crossing Hamiltonian paths, can be listed in time polynomial in the output size by simple backtracking algorithms. These algorithms search spaces of easier-to-list structures, the *surrounding polygons* [37] and non-crossing paths, respectively. To prove these new results, following our work on monotone parameters of point sets [12], we relate the numbers of all four types of structures (polygonalizations, surrounding polygons, non-crossing Hamiltonian paths, and non-crossing paths) to two easily-computed parameters that depend only on the order-type of the points: the smallest number of points whose removal results in a collinear subset, and the number of points interior to the convex hull. These relations imply that the number of polygonalizations is at most polynomial in the number of surrounding polygons, and that the number of non-crossing Hamiltonian paths is at most polynomial in the number of non-crossing paths. Therefore, an algorithm for the easier-to-list structures



© David Eppstein;

licensed under Creative Commons License CC-BY 4.0

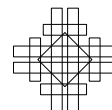
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 29; pp. 29:1–29:16

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



will take output-polynomial time when its output is filtered to generate only the harder-to-list structures. Our methods also provide a polynomial-time approximation algorithm for counting these structures, obtaining a constant approximation ratio with respect to the logarithm of the count.

We do not calculate explicitly the exponents of the polynomials relating these numbers of structures, as our upper bounds are quite imprecise. Instead, in a final section, we describe point sets whose exponential numbers of non-crossing structures can be calculated precisely. These examples demonstrate that the exponent of paths in terms of Hamiltonian paths can be at least $\log_2 3 \approx 1.585$ and that the exponent of surrounding polygons in terms of polygonalizations can be at least $\log_2((3 + \sqrt{5})/2) \approx 1.388$.

Output-polynomial time bounds, such as we prove, are not as good as a time bound that multiplies the output size by a polynomial of the input size, and even less good than polynomial delay for each output structure. Nevertheless, our results represent a significant improvement on previously known time bounds, which are singly exponential even for inputs whose output size is subexponential.

Because of space limitations, we omit many proofs from this proceedings version. A full version of this paper is available at arXiv:2303.00147.

1.1 Related work

A *planar straight line graph*, with given points as vertices, consists of line segments having the points as endpoints, with no line segment passing through a given point and no two line segments intersecting except at a shared endpoint. When the form of the resulting graph is constrained, one obtains non-crossing structures of various types, including triangulations (non-crossing maximal planar graphs), non-crossing spanning trees, and polygonalizations (non-crossing Hamiltonian cycles). Extensive research in discrete and computational geometry has sought upper and lower bounds for numbers of non-crossing structures, and studied algorithmic problems of counting or listing these structures for a given point set, or of finding a non-crossing structure that is optimal for some objective function [2–4, 8, 11, 15, 16, 19, 22, 29–31]. Many problems of this sort have algorithms for listing all graphs, with polynomial time per output, based on systems of local moves that link the state space into a connected structure [1, 5, 7, 20, 33, 35–37].

However, for polygonalizations and non-crossing Hamiltonian paths no such structure is known, and natural systems of local moves that change two or three edges at a time are known not to connect all polygonalizations [18]. Certain generalizations of polygonalizations have state spaces connected by local moves [9, 37], but this does not directly yield fast algorithms for polygonalizations, because of the many non-polygonalizations in these state spaces. As a 2011 survey of Welzl summarizes, “Basically nothing is known for related algorithmic questions (determining the number of simple polygonizations for a given point set, enumerating all simple polygonizations)” [34]. The shortest polygonalization is the NP-hard Euclidean traveling salesperson tour [28], and several other optimal polygonalizations are also NP-hard [10, 14]; the complexity of the longest polygonalization is another unknown [11].

Past work on non-crossing Hamiltonian paths includes approximation to the longest path [3, 11] and the existence of properly colored paths for colored point sets [6]. The number of non-crossing Hamiltonian paths can range from one, for collinear points, to exponentially large; for instance, $n \geq 2$ points in convex position have exactly $n2^{n-3}$ paths [27].

The surrounding polygons that we use for our polygonalization algorithm are another class of planar straight line graphs: they are the simple polygons that use a subset of the input points as vertices, and contain all of them. Yamanaka et al. [37] introduced these polygons,

and showed how to list them in polynomial time per polygon, from which they derived a singly-exponential time bound and polynomial space bound for listing polygonalizations. Despite this progress they were unable to obtain an output-sensitive time bound for polygonalizations. It is their algorithm that we follow here, with a new output-sensitive analysis. Yamanaka et al. also prove that, in the worst-case exponential bounds for numbers of polygonalizations and surrounding polygons, the bases of the exponentials are within 1 of each other; however, this analysis does not imply our stronger result, that on arbitrary instances the numbers of these two types of polygons are bounded by polynomials of each other. The examples from our final section confirm theoretically the empirical results of Yamanaka et al. that polygonalizations and surrounding polygons can grow at different rates.

2 Two simple backtracking algorithms

In this section we outline two simple algorithms (one a standard backtracking search, and the other the reverse search algorithm of Yamanaka et al. [37]) for listing non-crossing paths and surrounding cycles. These known algorithms will be the ones we use to prove our new time bounds for listing non-crossing Hamiltonian paths and polygonalizations. We state these bounds as theorems in this section, and defer the proofs to later sections.

► **Definition 1.** *Define a non-crossing path for a set of points S to be a non-self-intersecting polygonal curve P that passes through a subset of points of S , has endpoints in S , and turns only at points of S . Define the vertices of a non-crossing path to be the set $P \cap S$, counting a point of S as a vertex even when P passes straight through that point. Define a non-crossing path sequence to be the sequence of vertices in a non-crossing path of a given set of points in the plane, including also one-vertex sequences and the empty sequence. Let $|S|$ denote the number of points in S , let $\#\text{PATH}(S)$ denote the number of non-crossing paths with vertices in S , and let $\#\text{HAM}(S)$ denote the number of non-crossing Hamiltonian paths of S .*

Each non-crossing path corresponds to two sequences (one for each end-to-end order in which its vertices can be placed). We can form a rooted tree with the non-crossing path sequences as its nodes, in which the empty sequence is the root, by defining the parent of any other sequence to be the subsequence obtained by removing its last vertex. The non-crossing path sequences can be listed in polynomial time per sequence by performing a depth-first search of this tree. With a little care in listing the children of each node quickly, we can reduce this to linear time per sequence:

► **Subroutine 2** (listing children of a non-crossing path sequence). To find the children of a non-crossing path sequence σ , ending at point p :

- Apply the simple stack-based linear time algorithm of Lee to determine the visibility polygon V of the final vertex of the path, the region of the plane within which that vertex can be connected to another point by a segment that does not cross the existing path [21].
- Let U be the set of points not in σ , and find the radial ordering of U around p . When there are ties, keep only the closest of the tied points to p .
- Merge the radial orderings of U and V to determine, for each point u in U , the edge of V that is crossed by a ray from p through u .
- Whenever the merge finds a point u that is closer to p than the corresponding edge of V , make a child sequence by concatenating u to the end of σ .

The radially-sorted lists of all points around each of the given point can be precomputed and stored in $O(|S|^2)$ time; essentially, this is the same as the problem of constructing and storing an arrangement of lines dual to the points. With this precomputation, all steps of this algorithm take time $O(|S|)$.

29:4 Non-Crossing Hamiltonian Paths and Cycles in Output-Polynomial Time

► **Subroutine 3** (listing all non-crossing paths). To list all non-crossing paths of a given set of points, perform a depth-first search of the tree of non-crossing path sequences, using Subroutine 2 to list the children of each node in the tree. For each node that the search reaches, output it as a path whenever the starting vertex of the sequence has a smaller index than the ending vertex, so that we only output each non-crossing path once. For a point set S , we spend $O(|S|^2)$ preprocessing time and $O(|S|)$ time per path. The number of paths is always $\Omega(|S|^2)$, even for collinear point sets, because a Hamiltonian path always exists and contains that many paths within it, so the preprocessing time is dominated by the per-path time and the total time is $O(|S| \cdot \#\text{PATH}(S))$.

With these subroutines in hand, we can list all non-crossing Hamiltonian paths by the following very simple algorithm.

► **Algorithm 4** (listing non-crossing Hamiltonian paths). To list all non-crossing Hamiltonian paths in a point set S :

- List all non-crossing paths by Subroutine 3.
- Whenever a path uses all points of S , output it as a non-crossing Hamiltonian path.

► **Theorem 5.** *For a point set S (not assumed to be in general position), Algorithm 4 takes time $(|S| \cdot \#\text{HAM}(S))^{O(1)}$ to list all non-crossing Hamiltonian paths.*

Proof. This follows from Theorem 20, later in this paper, which states that

$$\#\text{PATH}(S) = (|S| \cdot \#\text{HAM}(S))^{O(1)}. \quad \blacktriangleleft$$

A very similar tree search can also be used for polygonalizations, instead of Hamiltonian paths.

► **Definition 6.** *Yamanaka et al. [37] define a surrounding polygon of a point set S to be a simple polygon having a subset of the points as its vertices, surrounding all of the vertices. These include the polygonalizations (in which the subset is all of the points) as well as other polygons; in particular, the convex hull is always a surrounding polygon. Define $\#\text{SURROUND}(S)$ to be the number of surrounding polygons in S , and $\#\text{POLY}(S)$ to be the number of polygonalizations in S .*

Yamanaka et al. define a tree structure on surrounding polygons in which the root is the convex hull, and the parent of any surrounding polygon is obtained by removing one vertex (in a canonically chosen way) from its cyclic sequence of vertices. It follows from a version of the two-ears theorem for polygons (often credited to G. H. Meisters, but used earlier by Max Dehn [17, 23]) that every polygon that is not the convex hull has a parent.

► **Subroutine 7** (listing surrounding polygons). As Yamanaka et al. [37] describe, the surrounding polygons of point set S can be listed in time $|S|^{O(1)}$ per polygon, and space $O(|S|)$, by a depth-first search of the tree of polygons. The algorithm uses the method of *reverse search* [5] to perform the depth-first search while only maintaining the identity of a bounded number of tree nodes.

Again, we can use this to list all polygonalizations, as was already done by Yamanaka et al. [37].

► **Algorithm 8** (listing polygonalizations). To list all polygonalizations of a point set S :

- List all surrounding polygons by Subroutine 7.
- Whenever a polygon uses all points of S output it as a polygonalization

Yamanaka et al. analyzed this algorithm as having singly-exponential time, but we instead prove that it is output-sensitive:

► **Theorem 9.** *For a point set S (not assumed to be in general position), Algorithm 8 takes time $(|S| \cdot \#POLY(S))^{O(1)}$ to list all polygonalizations.*

Proof. This follows from Theorem 27, later in this paper, which states that

$$\#SURROUND(S) = \#POLY(S)^{O(1)}. \quad \blacktriangleleft$$

3 Counting paths

The main result of this section is to prove a polynomial relation between the number of non-crossing paths and the number of non-crossing Hamiltonian paths, in any point set. This result combines a lower bound on the non-crossing Hamiltonian paths of a point set S , and an upper bound on the number of non-crossing paths, both expressed as a function of the following quantity.

► **Definition 10.** *Following [12], define $OFFLINE(S)$ to be the smallest k such that removing k points from S leaves a collinear subset.*

It will be convenient to have the following standard bound on logarithms of binomial coefficients. We use \log to denote the natural logarithm, but this choice of base makes no difference to the following lemma, because a change of base would only change the logarithm by a constant factor.

► **Lemma 11.** *For integers k and n with $k \leq n/2$,*

$$\log \binom{n}{k} = \Theta \left(k \log \frac{n}{k} \right).$$

Proof. By applying Stirling's formula, taking logarithms, and omitting terms that are $O(\log n)$, we obtain

$$\log \binom{n}{k} \approx \log \frac{n^n}{k^k (n-k)^{n-k}} = k \log \frac{n}{k} + (n-k) \log \frac{n}{n-k}.$$

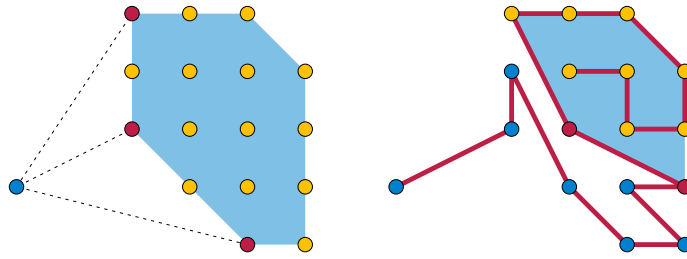
This expression is symmetric in k and $n-k$, and when $k \leq n/2$, the first term of this approximation is the larger of the two. ◀

3.1 Upper bound

We begin our bounds on non-crossing paths with the simplest one to prove, the upper bound. It is known that the number of paths is at most singly-exponential in the number n of vertices; we prove a tighter bound depending exponentially on the smaller number $OFFLINE(S)$ and only polynomially on n .

► **Lemma 12.** *Let S be a set of n points, with $OFFLINE(S) = k$. Then*

$$\log \#PATH(S) = O \left(\log n + k \left(\log \frac{n}{k+1} \right) \right).$$



■ **Figure 1** Left: A point p (blue), a point set S (red and yellow), and the visible vertices of S from p (the three red vertices). Note that points of S that lie within convex hull edges are not counted as vertices. Right: A maximal visible-vertex path (red edges) for $S \cup \{p\}$ starting from p , showing for one of its steps the hull (light blue) and visible vertices (red) of the remaining points.

Proof. We describe a method for encoding a non-crossing path using this many bits of information, so that each path is uniquely described by this encoding. For an encoding with b bits of information, the number of paths can be at most 2^b and the logarithm of this number is $O(b)$. Let K be a subset of S , with $|K| = k$, such that $S \setminus K$ lies on a line L , and let P be any given non-crossing path in S . Let $\ell = |L \cap S|$; because k is the minimum number of points that can be removed to form a collinear set, no point of K can lie on L , and $\ell = n - k$. To describe P , we combine the following pieces of information:

- The set Q of points of $L \cap S$ that belong to P , but for which zero or one of their neighbors in the path belong to L . Each such point is one of the two neighbors of a point in K , or one of the two ends of P , so $|Q| \leq 2k + 2$. Q can be encoded by specifying its size and the subset of $L \cap S$ of that size, out of $\binom{\ell}{|Q|}$ possibilities, so by Lemma 11 the number of bits needed to specify it is $O(\log n + k + k \log(n/k))$. (Both the $\log n$ term and the $+1$ in the statement of the lemma are included to handle the case when $k = 0$ but $|Q| > 0$. Lemma 11 applies only when $k \leq n/2$ but for larger k the bound to be proven is superlinear and the result is immediate.)
- For each point in Q , a specification of whether it has a neighbor in L , and if so in which direction. This takes $O(k)$ bits of information.
- The induced subgraph $P[K \cup Q]$, a linear forest using only the points in $K \cup Q$, and omitting the edges of P that lie entirely within L . As with any type of planar straight-line graph, the number of linear forests on $O(k)$ points is singly exponential in k [30], so $P[K]$ can be encoded with $O(k)$ bits of information.

Then P may be recovered by combining the induced subgraph $P[K \cup Q]$ with segments of L starting and ending at points of Q and continuing in the specified direction from each of these points. All pieces of this encoding add up to the stated bound on the number of bits needed to encode the entire path. ◀

3.2 Visible-vertex paths

Our lower bounds will greedily construct paths such that, at each step, the remaining unused points have a convex hull that is uncrossed by the current path and is visible from the endpoint of the current path.

► **Definition 13.** Define a visible vertex of a finite point set S , from a point p that is not in the convex hull of S , to be a vertex q of the convex hull of S such that the open line segment pq is disjoint from the convex hull of S (Figure 1, left). We do not consider points interior to edges of the convex hull to be vertices, and in particular they are not visible vertices, even when segment pq is disjoint from the hull.

► **Observation 14.** *Every nonempty finite point set S and point p not in the convex hull of S has at least one visible vertex of S from p . If $S \cup \{p\}$ is not collinear, there are at least two visible vertices.*

Define a *visible-vertex path* to be a polygonal chain formed by a greedy algorithm that builds a sequence of vertices, beginning with any vertex of the convex hull, by repeatedly appending to the sequence a visible vertex of the remaining points not yet in the sequence, as viewed from the last point of the sequence (Figure 1, right). A *maximal visible-vertex path* is a visible-vertex path that uses all of the points in a given point set S ; this name is justified by the following lemma.

► **Lemma 15.** *Every visible-vertex path is a non-crossing path. Every maximal visible-vertex path is a non-crossing Hamiltonian path. Every non-maximal visible-vertex path can be extended to a longer visible-vertex path.*

Proof. Maximal visible-vertex paths are Hamiltonian, by definition: they are defined to be paths that use all the points. Because each step in a visible-vertex path is defined locally, without respect to the earlier parts of the path, the ability to extend every non-maximal path follows immediately from Observation 14.

It remains to prove that the resulting paths are non-crossing. For every segment pq of the path, the next segment is incident to q and therefore cannot cross pq . All segments after the next segment lie within the convex hull of the remaining points after q . Since p and q are vertices of their respective convex hulls, the convex hull of the points after them in the path does not contain them. Moreover, segment pq does not cross this hull, because if it did then q would not be visible from p . Thus, these later segments are disjoint (as closed line segments) from the closed line segment pq and so cannot cross pq nor even pass through q . Therefore, pq cannot intersect any later segment. For each pair of segments in the path, a crossing is ruled out by applying this argument to the earlier of the two segments in the path, so no crossing can exist. ◀

► **Lemma 16.** *In every finite point set S , every two vertices p and q of the convex hull of S form the endpoints of a non-crossing Hamiltonian path.*

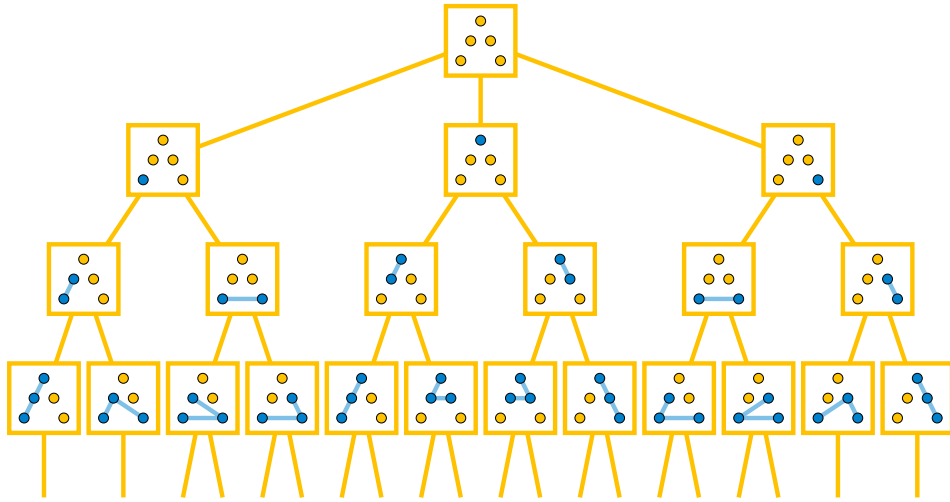
Proof. Form a maximal visible-vertex path beginning at p , at each step choosing a visible vertex that is not q when this is possible. By Observation 14, the path will have two points to choose between (one of which is not q) until the remaining points become collinear. Once they do, all remaining points that are not q will lie between q and the current end of the path, so q cannot be included in the path until it is the last point remaining. ◀

3.3 Lower bound for far-from-collinear sets

To lower-bound the number of non-crossing Hamiltonian paths in a point set S , as a function of $\text{OFFLINE}(S)$, we divide into two cases: the case where $\text{OFFLINE}(S)$ is large (at least proportional to a constant fraction of $|S|$) and the case where it is small. The following lemma is valid for all non-collinear S , but is only tight (within a constant factor of the logarithm) in the large case.

► **Lemma 17.** *For every point set S that does not lie on a single line, the number of non-crossing Hamiltonian paths is at least $\frac{3}{2} \cdot 2^{\text{OFFLINE}(S)}$.*

Proof. We form a tree T of visible-vertex paths, in which the root is the empty sequence of vertices and the parent of any nonempty path is obtained by removing its last vertex (Figure 2). By Lemma 15, the leaves of T are non-crossing Hamiltonian paths. The root



■ **Figure 2** A tree of the visible-vertex paths in a point set, where the parent of each path is obtained by removing its last point. This point set has $\text{OFFLINE}(S) = 2$. The root and the next two levels of nodes each have multiple children, but some nodes on the last level shown in the figure have only one child, because the last point on the path is collinear with all remaining points.

node of T has at least three children (one for each convex hull vertex). In the nodes of T at distance at most $\text{OFFLINE}(S)$ from the root, the last point in the path represented by each node is not collinear with the remaining points, by the definition of OFFLINE . It follows by Observation 14 that these nodes have at least two choices of visible vertices and therefore that they have at least two children.

As a tree in which the root node has at least three children and the nodes in the next $\text{OFFLINE}(S)$ levels have at least two children, T has at least $3 \cdot 2^{\text{OFFLINE}(S)}$ leaves. Each leaf is a non-crossing Hamiltonian path, and each non-crossing Hamiltonian path can come from at most two leaves (one for each endpoint, if both endpoints are convex hull vertices). Therefore, there are at least $\frac{3}{2} \cdot 2^{\text{OFFLINE}(S)}$ non-crossing Hamiltonian paths. ◀

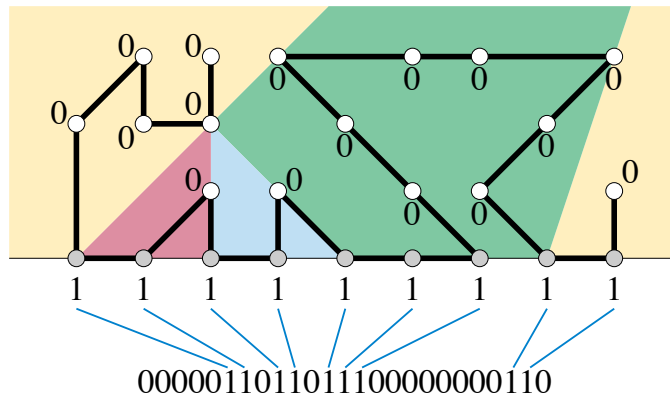
3.4 Lower bound for near-collinear sets

Although the next lemma covers only a special class of point sets, it is the key to our lower bounds for the case where $\text{OFFLINE}(S)$ is small.

▶ **Lemma 18.** *Let S be a set of points with $|S| = n$, such that ℓ points of S lie on a line L and the remaining points all lie on the same side of the line. Then*

$$\#_{\text{HAM}}(S) \geq \binom{n - \lceil \ell/2 \rceil}{\lfloor \ell/2 \rfloor}.$$

Proof. We may assume without loss of generality that $\ell \geq 2$, for otherwise the lemma states only that there exists a single non-crossing Hamiltonian path, known to be true. For convenience consider an orientation of the plane in which L is horizontal and S lies in the closed halfspace above L . We will prove the lemma by constructing many non-crossing Hamiltonian paths in which the points of $L \cap S$ appear in left-to-right order, and no point in L has two neighbors in $S \setminus L$. For any such path, define its *signature* to be the binary sequence with a 1-bit for points in L and a 0-bit for points in $S \setminus L$ (Figure 3). It has length



■ **Figure 3** Partition of the halfspace above L into convex subsets, a non-crossing Hamiltonian path respecting the partition, and its signature, a 010-avoiding binary sequence.

n , with ℓ 1-bits, and does not have any three consecutive bits in the pattern 010. The number of 010-avoiding sequences with n bits and ℓ 1-bits is [25]

$$\sum_{j=0}^{n-\ell} (-1)^j \binom{n-\ell-1}{j} \binom{|n-2j|}{\ell-j} \geq \binom{n-\lceil \ell/2 \rceil}{\lfloor \ell/2 \rfloor},$$

where for the simpler formula on the right hand side of the inequality the 1-bits are grouped into $\lfloor \ell/2 \rfloor$ pairs (with one group of three if ℓ is odd) and we count only the sequences in which each pair or triple appears consecutively. (By the assumption that $\ell \geq 2$, at least one such grouping is possible.) As we argue in the remainder of the proof, every 010-avoiding sequence is the signature of at least one non-crossing Hamiltonian path, so this lower bound on the number of 010-avoiding sequences also provides a lower bound on the number of non-crossing Hamiltonian paths.

For a given 010-avoiding sequence σ , let n_i denote the length of the i th non-empty block of consecutive 0-bits in σ . We will partition the halfplane above L into convex sets C_i , each containing n_i points of $S \setminus L$, by a greedy process that maintains a convex subset of the halfplane containing the remaining points to be partitioned. Initially the convex subset is the entire halfplane and the remaining points are $S \setminus L$. On the i th step (for any i other than the last one), let p_i be the point of $S \cap L$ that corresponds to the 1-bit of σ following the i th block of 0-bits. Sort the remaining points of $S \setminus L$ radially around p_i (in left to right order with respect to L) breaking ties in favor of closer points to p_i , and let q_i be the point in position n_i of this sorted order. Draw line $p_i q_i$, separating C_i on its left from a remaining convex subset on its right. Assign the n_i points up to q_i in the radial sorted order to set C_i , and leave the remaining points (possibly including farther points on line $p_i q_i$) unassigned. In the final step of this construction, assign all remaining points to the final remaining convex region. For instance, in Figure 3, the leftmost set C_1 (yellow) is separated from the rest of the halfplane by line $p_1 q_1$. Here p_1 is the leftmost point of L , and q_1 is the fifth point in the radial ordering around p_1 . Point q_1 lies on the boundary of four convex regions but is assigned to the first, C_1 . Line $p_1 q_1$ also contains another point of S , farther from p_1 , which is assigned to C_4 (green).

Once this partition into convex sets has been determined, use Lemma 16 to find a non-crossing Hamiltonian path within each convex set C_i that starts and ends at its (one or two) points on L , and connect these paths in sequence by segments of L to form a non-crossing Hamiltonian path for all of S , with the given 010-avoiding sequence σ as its signature. ◀

3.5 Putting the bounds together

Combining the two different lower bounds into a single formula, we have:

► **Lemma 19.** *Let S be a set of n points with $\text{OFFLINE}(S) = k$. Then*

$$\log \#_{\text{HAM}}(S) = \Omega \left(k \left(\log \frac{n}{k+1} \right) \right).$$

Proof. For any k , the logarithm of the number of non-crossing Hamiltonian paths is $\Omega(k)$ by Lemma 17. For $k \geq n/3$, $\log(n/k) = O(1)$, so for this range of k , this $\Omega(k)$ bound is equivalent to the bound stated in the theorem.

For smaller values of k , let K be any set of k points whose removal from S leaves a collinear set of size $\ell = n - k = \Omega(n)$, belonging to a line L . Partition K into the two subsets K_1 and K_2 on the two sides of L , with $|K_1| \geq |K_2|$; let $|K_1| = k' \geq k/2$. Let p be the first point of $S \cap L$, in the sorted sequence of the points along this line, let $\ell' = \ell - 1$ be the number of remaining points in $S \cap L$, and let $n' = \ell' + k'$. By Lemma 18, the number of non-crossing Hamiltonian paths of $S \setminus K_2$ that start at p is at least

$$\binom{n' - \lceil \ell'/2 \rceil}{\lfloor \ell'/2 \rfloor} = \binom{\lfloor \ell'/2 \rfloor + k'}{k'}.$$

Each such path can be extended to a non-crossing Hamiltonian path of all of S by concatenating any non-crossing path through p and the points of K_2 . By the assumption that $k < n/3$, the bottom term of the right binomial coefficient is at most half the top term, allowing us to apply Lemma 11. By this lemma, and the facts that $k' = \Theta(k)$ and $\ell' = \Theta(n)$, the logarithm of this binomial coefficient is $\Omega(k \log(n/k))$ as stated. ◀

Although the upper bound of Lemma 12 and the lower bound of Lemma 19 are not quite the same, we can combine them to achieve a constant factor approximation to the logarithm of the number of non-crossing paths, or Hamiltonian paths.

► **Theorem 20.** *For a given point set S ,*

$$\#_{\text{PATH}}(S) = (|S| \cdot \#_{\text{HAM}}(S))^{O(1)}.$$

Proof. Taking logs of both sides, it is equivalent to write that

$$\log \#_{\text{PATH}}(S) = O(\log |S| + \log \#_{\text{HAM}}(S)).$$

This follows immediately from Lemma 19 and Lemma 12, according to which $\log \#_{\text{HAM}}(S)$ is lower-bounded and $\log \#_{\text{PATH}}(S)$ upper-bounded (respectively) to within constant factors by formulas that differ from each other only in an additive $\log |S|$ term. ◀

4 Counting cycles

For counting both surrounding cycles and polygonalizations of general-position point sets, in place of $\text{OFFLINE}(S)$ (which the general-position assumption makes trivial) we use the following parameter:

► **Definition 21.** *Let $\text{INHULL}(S)$ denote the number of points of S that are interior to the convex hull of S .*

For counting cycles and polygonalizations of point sets that are not assumed to be in general position, we will use a combined analysis in terms of both OFFLINE and INHULL .

4.1 Omitted lemmas

Our bounds for surrounding cycles and polygonalizations follow similar arguments to our bounds for paths. We defer many details and all proofs to the full version of this paper because of space limitations.

► **Lemma 22.** *Let S be a set of n points, with $\text{INHULL}(S) = h$. Then*

$$\log \#_{\text{SURROUND}}(S) = O\left(h \left(\log \frac{n}{h} + 1\right)\right).$$

► **Corollary 23.** *Let S be a set of n points with $\min(\text{OFFLINE}(S), \text{INHULL}(S)) = m$. Then*

$$\log \#_{\text{SURROUND}}(S) = O\left(m \left(\log \frac{n}{m} + 1\right)\right).$$

► **Lemma 24.** *Let S be a set of n points for which at most $|S|/7$ points lie on any line, and at most $|S|/7$ points lie on the convex hull. Then the number of polygonalizations of S is at least singly exponential in S .*

► **Lemma 25.** *Let S be a set of points with $|S| = n$, such that h points of S lie on its convex hull (either as vertices or within its edges). Then*

$$\#_{\text{HAM}}(S) \geq \binom{\lfloor h/4 \rfloor + \lceil (n-h)/2 \rceil - 1}{\lceil (n-h)/2 \rceil}$$

4.2 Putting the bounds together

Combining our lower bounds into a single formula, we have:

► **Lemma 26.** *Let S be a set of n points with $\min(\text{OFFLINE}(S), \text{INHULL}(S)) = m$. Then*

$$\log \#_{\text{POLY}}(S) = \Omega\left(m \left(\log \frac{n}{m} + 1\right)\right).$$

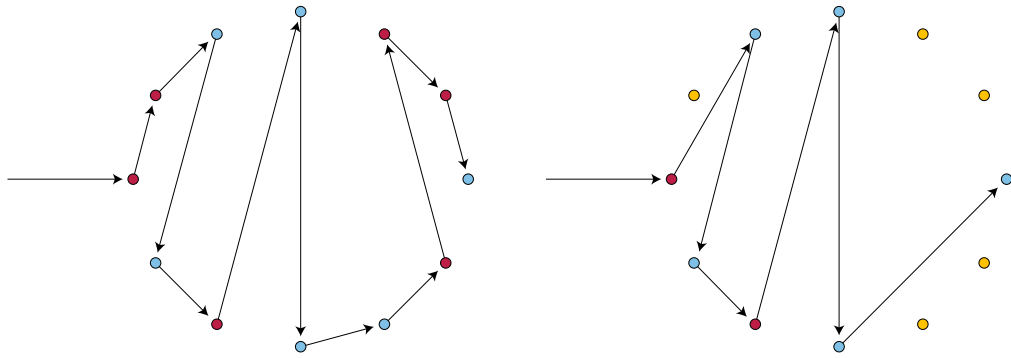
Proof. We consider the following cases:

- If $\text{INHULL}(S) \leq 6n/7$, the result follows from Lemma 25. In particular this applies when the largest subset of collinear points in S has size $\geq n/7$ and is part of the convex hull.
- If the largest subset of collinear points in S has size $\geq n/7$ but is not part of the convex hull, then let L be the line through this subset. Because L does not lie on the convex hull, $S \setminus L$ includes points on both sides of L , with at least $m/2$ points in one of these two halfplanes. By Lemma 18 the number of Hamiltonian paths through the points in L and the points in this halfplane, starting and ending at the two extreme points of L , meets or exceeds the lower bound in the statement of this lemma. Each of these Hamiltonian paths can be completed to a polygonalization through the points in the other halfplane bounded by L , by Lemma 16.
- In the remaining case, the largest subset of collinear points in S has size $< n/7$ and $\text{INHULL}(S) \geq 6n/7$. In this case, $m = \Omega(n)$ and the bound of the lemma reduces to $\Omega(n)$. The result follows from Lemma 24.

Since all cases have at least the number of polygonalizations stated, the bound holds.

For a bound of $\Omega(i)$, apply Lemma 24, and for $\Omega(i \log n/i)$ when $i \leq n/2$, apply Lemma 25, in both cases using Lemma 11 to estimate the logarithm of the binomial coefficient. ◀

From the fact that the upper bound of Corollary 23 and the lower bound of Lemma 26 have exactly the same form, we obtain a constant factor approximation to the logarithm of the number of surrounding cycles and of polygonalizations. For our bound on the complexity of the algorithm for listing polygonalizations we need it in the following form:



■ **Figure 4** Coloring-based arguments for the number of non-crossing paths and Hamiltonian paths of a point set in convex position. Each 2-coloring of the points and choice of starting point determines a non-crossing Hamiltonian path in which the colors determine the direction of the next step; each 3-coloring and choice of starting point determines a path, skipping vertices of one of the colors.

► **Theorem 27.** For a given point set S ,

$$\#SURROUND(S) = \#POLY(S)^{O(1)}.$$

5 Nonlinearity

In this section we investigate the exponent of the polynomial bounds on non-crossing paths as a function of non-crossing Hamiltonian paths, and on surrounding cycles as a function of polygonalizations. In both cases we show that the exponent is bounded away from one, for inputs in which the number of non-crossing configurations is exponential. This implies, in particular, that the backtracking algorithms that we investigate for Hamiltonian paths and polygonalizations can in some instances be forced to take an amount of time per output that is exponential in the input size, despite being polynomial in the output size.

The construction for paths and Hamiltonian paths is very simple:

► **Theorem 28.** There exist sets S of points for which

$$\#PATH(S) \geq \#HAM(S)^{\log_2 3 - o(1)} \approx \#HAM(S)^{1.585}$$

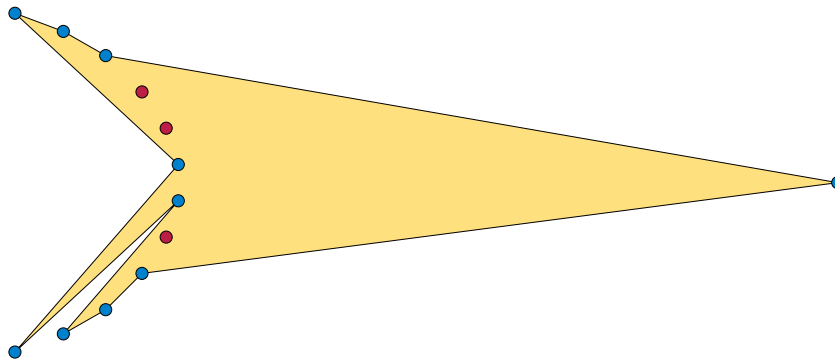
and moreover for which both the number of non-crossing paths and the number of non-crossing Hamiltonian paths is exponential in $|S|$.

Proof. We may take S to be in convex position. If a set S in convex position has n points, it has $n2^{n-3}$ non-crossing Hamiltonian paths [27], but

$$\frac{n}{4}(3^{n-1} + 3)$$

non-crossing paths in total, considering a single vertex to count as a path of length zero.

Both bounds can be proven by a simple coloring argument (Figure 4). For non-crossing Hamiltonian paths, consider the 2^n ways of coloring the points red and blue, and n choices of where to start. For each choice, follow a path that, at each red point, steps clockwise to the next available vertex, and at each blue point steps counterclockwise. Each non-crossing Hamiltonian path is found for exactly eight choices: the path can start at either end, and the final two vertex colors are irrelevant. Thus, the number of paths is $n2^n/8$.



■ **Figure 5** Points formed by replacing an edge of a triangle by a convex chain of points within the triangle, and a surrounding polygon (yellow) for these points. Points can be omitted as polygon vertices (red) only when they lie between the two neighbors of the apex (the rightmost point of the figure).

For the bound on non-crossing paths, consider instead the 3^n ways of coloring the points red, blue, and yellow, and skip the yellow points both in choosing a starting point and in considering which vertices are available in subsequent steps of the path. Cyclically permuting the colors in a coloring groups the 3^n colorings into 3^{n-1} orbits, each of which has a total of $2n$ red or blue starting points, so there are $2n \cdot 3^{n-1}$ choices. Again, each path is found for exactly eight choices, except for the single-vertex paths, which are found for only two choices. The total number of paths is obtained by dividing $2n \cdot 3^{n-1}$ by eight, and adjusting for the number of single-vertex paths. ◀

For an analogous separation between surrounding polygons and polygonalizations, we replace one edge of a triangle by a convex chain of $n - 2$ edges, within the triangle (Figure 5). The polygonalizations of this point set are obtained from non-crossing Hamiltonian paths through the points of the convex chain, with both ends of the path connected to the one point that does not belong to the convex chain (which we call the *apex*). The edges to the apex cannot cross any other edge, so all Hamiltonian paths lead to polygonalizations in this way. Therefore, by the same formula already used above, the number of polygonalizations is exactly $(n - 1)2^{n-4}$.

A surrounding polygon for these points must have the apex as a vertex (because it lies on the convex hull). It must also include as vertices all of the points of the convex chain that lie outside the two neighbors of the apex. The points of the convex chain that lie between the two neighbors of the apex may either be vertices of the surrounding polygon, or omitted; if omitted, they will automatically be surrounded. We may parameterize a surrounding polygon by three non-negative numbers: a , the number of *outer points* of the convex chain that lie outside the two neighbors of the apex, b , the number of *inner points* that lie between these two neighbors and are vertices of the polygon, and c , the number of *omitted points* that are not vertices of the surrounding polygon. Necessarily, $a + b + c = n - 3$, because the points that are counted by these three numbers include all points except the apex and its two neighbors.

Once a , b , and c have been chosen, the surrounding polygon itself may be determined by three more choices: how to partition the outer a points into left and right subsets, with $a + 1$ possibilities, how to alternate between outer and inner points along the polygon, with $\binom{a+b}{a}$ possibilities, and how to partition the points between the two neighbors of the apex

into inner points and omitted points, with $\binom{b+c}{b}$ possibilities. Therefore, the total number of surrounding polygons of this point set is

$$\sum_{a+b+c=n-3} (a+1) \binom{a+b}{a} \binom{b+c}{b}.$$

These numbers, for $n = 3, 4, 5, \dots$, are

$$1, 4, 13, 40, 120, 354, 1031, 2972, 8495, 24110, \dots,$$

a sequence having the generating function $(1-2x)/(1-3x+x^2)^2$ and growing proportionally to $n(\varphi+1)^n$, where $\varphi = (1+\sqrt{5})/2$ is the golden ratio [26]. This example proves:

► **Theorem 29.** *There exist sets S of points for which*

$$\#SURROUND(S) \geq \#POLY(S)^{\log_2(\varphi+1)-o(1)} \approx \#POLY(S)^{1.388}$$

and moreover for which both the number of surrounding polygons and the number of polygonalizations is exponential in $|S|$.

6 Conclusions

We have developed simple output-sensitive algorithms for listing all non-crossing Hamiltonian paths and all polygonalizations for a point set. However, their dependence on the output size is polynomial, not linear. It would be of interest to find alternative algorithms with a better dependence on the output size, as well as more accurate approximations for the numbers of non-crossing structures.

References

- 1 Oswin Aichholzer, Franz Aurenhammer, Clemens Huemer, and Birgit Vogtenhuber. Gray code enumeration of plane straight-line graphs. *Graphs Combin.*, 23(5):467–479, 2007. doi:10.1007/s00373-007-0750-z.
- 2 Miklós Ajtai, Vašek Chvátal, Monroe M. Newborn, and Endre Szemerédi. Crossing-free subgraphs. In Alexander Rosa, Gert Sabidussi, and Jean Turgeon, editors, *Theory and Practice of Combinatorics: A collection of articles honoring Anton Kotzig on the occasion of his sixtieth birthday*, volume 12 of *Annals of Discrete Mathematics*, pages 9–12. North-Holland, 1982. doi:10.1016/S0304-0208(08)73484-4.
- 3 Noga Alon, Sridhar Rajagopalan, and Subhash Suri. Long non-crossing configurations in the plane. *Fund. Inform.*, 22(4):385–394, 1995. doi:10.3233/FI-1995-2245.
- 4 Victor Alvarez, Karl Bringmann, Radu Curticapean, and Saurabh Ray. Counting triangulations and other crossing-free structures via onion layers. *Discrete Comput. Geom.*, 53(4):675–690, 2015. doi:10.1007/s00454-015-9672-3.
- 5 David Avis and Komei Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65(1-3):21–46, 1996. doi:10.1016/0166-218X(95)00026-N.
- 6 Sayan Bandyapadhyay, Aritra Banik, Sujoy Bhore, and Martin Nöllenburg. Geometric planar networks on bichromatic collinear points. *Theoret. Comput. Sci.*, 895:124–136, 2021. doi:10.1016/j.tcs.2021.09.035.
- 7 Sergei Bespamyatnikh. An efficient algorithm for enumeration of triangulations. *Comput. Geom. Theory & Appl.*, 23(3):271–279, 2002. doi:10.1016/S0925-7721(02)00111-6.
- 8 Gi-Sang Cheon, Hong Joon Choi, Guillermo Esteban, and Minh Song. Enumeration of bipartite non-crossing geometric graphs. *Discrete Appl. Math.*, 317:86–100, 2022. doi:10.1016/j.dam.2022.04.008.

- 9 Mirela Damian, Robin Flatland, Joseph O'Rourke, and Suneeta Ramaswami. Connecting polygonizations via stretches and twangs. *Theory Comput. Syst.*, 47(3):674–695, 2010. doi:10.1007/s00224-009-9192-8.
- 10 Erik D. Demaine, Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Joseph S. B. Mitchell. Area-optimal simple polygonalizations: the CG challenge 2019. *ACM J. Exp. Algorithmics*, 27:Article 2.4, 2022. doi:10.1145/3504000.
- 11 Adrian Dumitrescu and Csaba D. Tóth. Long non-crossing configurations in the plane. *Discrete Comput. Geom.*, 44(4):727–752, 2010. doi:10.1007/s00454-010-9277-9.
- 12 David Eppstein. *Forbidden Configurations in Discrete Geometry*. Cambridge University Press, 2018. doi:10.1017/9781108539180.
- 13 David Eppstein. Counting polygon triangulations is hard. *Discrete Comput. Geom.*, 64(4):1210–1234, 2020. doi:10.1007/s00454-020-00251-7.
- 14 Sándor P. Fekete. On simple polygonalizations with optimal area. *Discrete Comput. Geom.*, 23(1):73–110, 2000. doi:10.1007/PL00009492.
- 15 Philippe Flajolet and Marc Noy. Analytic combinatorics of non-crossing configurations. *Discrete Math.*, 204(1-3):203–229, 1999. doi:10.1016/S0012-365X(98)00372-0.
- 16 Alfredo García, Marc Noy, and Javier Tejel. Lower bounds on the number of crossing-free subgraphs of K_N . *Comput. Geom. Theory & Appl.*, 16(4):211–221, 2000. doi:10.1016/S0925-7721(00)00010-9.
- 17 H. Guggenheimer. The Jordan curve theorem and an unpublished manuscript by Max Dehn. *Archive for History of Exact Sciences*, 17(2):193–200, 1977. doi:10.1007/BF02464980.
- 18 Carmen Hernando, Michael E. Houle, and Ferran Hurtado. On local transformation of polygons with visibility properties. *Theoret. Comput. Sci.*, 289(2):919–937, 2002. doi:10.1016/S0304-3975(01)00409-1.
- 19 Michael Hoffmann, André Schulz, Micha Sharir, Adam Sheffer, Csaba D. Tóth, and Emo Welzl. Counting plane graphs: flippability and its applications. In János Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 303–325. Springer, 2013. doi:10.1007/978-1-4614-0110-0_16.
- 20 Naoki Katoh and Shin-Ichi Tanigawa. Fast enumeration algorithms for non-crossing geometric graphs. *Discrete Comput. Geom.*, 42(3):443–468, 2009. doi:10.1007/s00454-009-9164-4.
- 21 D. T. Lee. Visibility of a simple polygon. *CVGIP*, 22(2):207–221, 1983. doi:10.1016/0734-189x(83)90065-8.
- 22 Dániel Marx and Tillmann Miltzow. Peeling and nibbling the cactus: subexponential-time algorithms for counting triangulations and related problems. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG 2016, June 14-18, 2016, Boston, MA, USA*, volume 51 of *LIPICs*, pages 52:1–52:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.SoCG.2016.52.
- 23 G. H. Meisters. Polygons have ears. *Amer. Math. Monthly*, 82(6):648–651, 1975. doi:10.2307/2319703.
- 24 Joseph S. B. Mitchell and Joseph O'Rourke. Computational geometry column 42. *Internat. J. Comput. Geom. Appl.*, 11(5):573–582, 2001. doi:10.1142/S0218195901000651.
- 25 OEIS contributors. Sequence A180562. The On-Line Encyclopedia of Integer Sequences, March 2014. URL: <https://oeis.org/A180562>.
- 26 OEIS contributors. Sequence A238846. The On-Line Encyclopedia of Integer Sequences, November 2021. URL: <https://oeis.org/A238846>.
- 27 OEIS contributors. Sequence A001792. The On-Line Encyclopedia of Integer Sequences, April 2022. URL: <https://oeis.org/A001792>.
- 28 Louis V. Quintas and Fred Supnick. On some properties of shortest Hamiltonian circuits. *Amer. Math. Monthly*, 72(9):977–980, 1965. doi:10.2307/2313333.
- 29 Andreas Razen and Emo Welzl. Counting plane graphs with exponential speed-up. In Cristian S. Calude, Grzegorz Rozenberg, and Arto Salomaa, editors, *Rainbow of Computer Science*:

29:16 Non-Crossing Hamiltonian Paths and Cycles in Output-Polynomial Time

- Dedicated to Hermann Maurer on the Occasion of His 70th Birthday*, volume 6570 of *Lecture Notes in Computer Science*, pages 36–46. Springer, 2011. doi:10.1007/978-3-642-19391-0_3.
- 30 Micha Sharir and Adam Sheffer. Counting plane graphs: cross-graph charging schemes. *Combin. Probab. Comput.*, 22(6):935–954, 2013. doi:10.1017/S096354831300031X.
 - 31 Micha Sharir, Adam Sheffer, and Emo Welzl. Counting plane graphs: perfect matchings, spanning cycles, and Kasteleyn’s technique. *J. Combin. Theory Ser. A*, 120(4):777–794, 2013. doi:10.1016/j.jcta.2013.01.002.
 - 32 Hugo Steinhaus. *One Hundred Problems in Elementary Mathematics*. Basic Books, 1964. See pp. 17, 85–86.
 - 33 Shin-Ichi Tanigawa. Enumeration of non-crossing geometric graphs. In *Encyclopedia of Algorithms*, pages 638–640. Springer, 2016. doi:10.1007/978-1-4939-2864-4_729.
 - 34 Emo Welzl. Counting simple polygonizations of planar point sets. In *Proceedings of the 23rd Annual Canadian Conference on Computational Geometry, Toronto, Ontario, Canada, August 10-12, 2011*, 2011. URL: <https://www.cccg.ca/proceedings/2011/papers/invited3.pdf>.
 - 35 Manuel Wettstein. Counting and enumerating crossing-free geometric graphs. *J. Comput. Geom.*, 8(1):47–77, 2017. doi:10.20382/jocg.v8i1a4.
 - 36 Ro-Yu Wu, Jou-Ming Chang, Kung-Jui Pai, and Yue-Li Wang. Amortized efficiency of generating planar paths in convex position. *Theoret. Comput. Sci.*, 412(35):4504–4512, 2011. doi:10.1016/j.tcs.2011.04.017.
 - 37 Katsuhisa Yamanaka, David Avis, Takashi Horiyama, Yoshio Okamoto, Ryuhei Uehara, and Tanami Yamauchi. Algorithmic enumeration of surrounding polygons. *Discrete Appl. Math.*, 303:305–313, 2021. doi:10.1016/j.dam.2020.03.034.

Finding a Maximum Clique in a Disk Graph

Jared Espenant ✉

Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada

J. Mark Keil ✉

Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada

Debajyoti Mondal¹ ✉ 

Department of Computer Science, University of Saskatchewan, Saskatoon, Saskatchewan, Canada

Abstract

A disk graph is an intersection graph of disks in the Euclidean plane, where the disks correspond to the vertices of the graph and a pair of vertices are adjacent if and only if their corresponding disks intersect. The problem of determining the time complexity of computing a maximum clique in a disk graph is a long-standing open question that has been very well studied in the literature. The problem is known to be open even when the radii of all the disks are in the interval $[1, (1 + \varepsilon)]$, where $\varepsilon > 0$. If all the disks are unit disks then there exists an $O(n^3 \log n)$ -time algorithm to compute a maximum clique, which is the best-known running time for over a decade. Although the problem of computing a maximum clique in a disk graph remains open, it is known to be APX-hard for the intersection graphs of many other convex objects such as intersection graphs of ellipses, triangles, and a combination of unit disks and axis-parallel rectangles. Here we obtain the following results.

- We give an algorithm to compute a maximum clique in a unit disk graph in $O(n^{2.5} \log n)$ -time, which improves the previously best known running time of $O(n^3 \log n)$ [Eppstein '09].
- We extend a widely used “co-2-subdivision approach” to prove that computing a maximum clique in a combination of unit disks and axis-parallel rectangles is NP-hard to approximate within $4448/4449 \approx 0.9997$. The use of a “co-2-subdivision approach” was previously thought to be unlikely in this setting [Bonnet et al. '20]. Our result improves the previously known inapproximability factor of $7633010347/7633010348 \approx 0.9999$.
- We show that the parameter minimum lens width of the disk arrangement may be used to make progress in the case when disk radii are in $[1, (1 + \varepsilon)]$. For example, if the minimum lens width is at least 0.265 and $\varepsilon \leq 0.0001$, which still allows for non-Helly triples in the arrangement, then one can find a maximum clique in polynomial time.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Maximum clique, Disk graph, Time complexity, APX-hardness

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.30

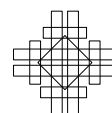
Related Version *Full Version*: <https://arxiv.org/abs/2303.07645> [15]

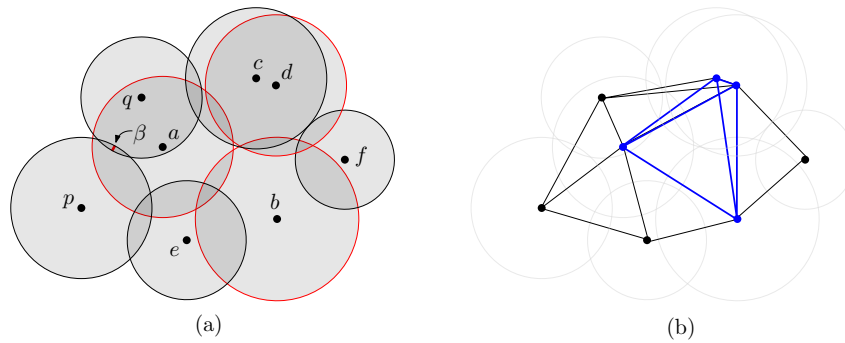
Funding The work is supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC).

1 Introduction

An *intersection graph* of a set S of geometric objects is a graph where each object in S corresponds to a vertex in G and two vertices in G are adjacent if and only if the corresponding objects intersect. A set of vertices $C \subseteq V$ is called a *clique* if they are mutually adjacent. In this paper, we are interested in the problem of finding a *maximum clique*, i.e., a largest set of mutually adjacent vertices. We mainly focus on *disk graphs*, i.e., the intersection graphs of disks in \mathbb{R}^2 (Figure 1). Disk graphs are often used to model ad-hoc wireless networks [22].

¹ Corresponding author





■ **Figure 1** (a) An arrangement of disks \mathcal{A} . The minimum width β over all the lenses is determined by the disks centered at p and q . A non-Helly triple is shown in red. (b) A disk graph corresponding to \mathcal{A} , where a maximum clique is shown in blue.

The time complexity question for finding a maximum clique in a disk graph is known to be open for over two decades [3, 4, 17]. The question is open even in severely restricted settings such as when the radii of the disks are of two types [9] or when the disk radii are in the interval $[1, 1 + \varepsilon]$ for a fixed $\varepsilon > 0$ [5]. However, there exists randomized EPTAS, deterministic PTAS, and subexponential-time algorithms for computing a maximum clique in arbitrary disk graphs [5, 6]. For unit disk graphs, i.e., when all the radii are the same, Clark et al. [13] showed that a maximum clique can be found in $O(n^{4.5})$ -time. Their algorithm searches for a maximum clique over all the lenses of pairwise intersecting disks. Later, Eppstein [14] showed how the algorithm could be implemented in $O(n^3 \log n)$ -time by searching through a careful ordering of the lenses and using a data structure of [2] to maintain a maximum clique throughout the search. Faster algorithms are known in constrained settings where the centers of the disks lie within a narrow horizontal strip [8]. Polynomial-time algorithms exist for many other intersection graph classes such as for circle graphs [29], trapezoid graphs [16], circle trapezoid graphs [16], intersection graphs of axis-parallel rectangles [23], and so on.

Although the maximum clique problem is open for disk graphs, a number of APX-hardness results are known in the literature for intersection graphs of other geometric objects. A common approach to prove the NP-hardness result for computing a maximum clique in an intersection graph class \mathcal{I} is to take a *co-2k-subdivision approach*, as follows. A *2k-subdivision*, where k is a positive integer, of a graph G is obtained by replacing each edge (u, v) of G with a path $(u, d_1, \dots, d_{2k}, v)$ of $2k$ division vertices. A *co-2k-subdivision approach* takes a graph class for which finding a maximum independent set is NP-hard and shows that the complement graph of its $2k$ -subdivision has an intersection representation in class \mathcal{I} . Since the NP-hardness of computing a maximum independent set is preserved by the even subdivision [12] and since a maximum independent set in a graph corresponds to a maximum clique in the complement graph, this establishes the NP-hardness result for computing a maximum clique in class \mathcal{I} . Some of the intersection graph classes for which the maximum clique problem has been proved to be APX-hard using the co-2k-subdivision approach are intersection graphs of ellipses [3], triangles [3], string graphs [27], grounded string graphs [25], and so on. Cabello [10] used the co-2k-subdivision approach to prove the NP-hardness of computing a maximum clique in the intersection graph of rays, which settled a 21-year-old open problem posed by Kratochvíl and Nešetřil [26]. To the best of our knowledge, no hardness of approximation result is known for this graph class.

Since there is strong evidence that a co-2k-subdivision approach may not be sufficient to prove the NP-hardness of computing a maximum clique in a disk graph [6], Bonnet et al. [7] attempted to explore alternative approaches. While they were not able to prove the

NP-hardness for disk graphs, they showed that the problem of computing a maximum clique in an intersection graph that contains both unit disks and axis-parallel rectangles is not approximable within a factor of $7633010347/7633010348$ in polynomial time, unless $P=NP$. This result is interesting since the maximum clique problem is polynomial-time solvable when all objects are either unit disks [13] or axis-parallel rectangles [23]. To obtain this result, Bonnet et al. [7] introduced a new problem called “Max Interval Permutation Avoidance”, proved it to be APX-hard, and reduced it to the problem of computing a maximum clique in a combination of unit disks and axis-parallel rectangles. Furthermore, they stated that the intersection graph of unit disks and axis-parallel rectangles is “a class for which the co-2-subdivision approach does not seem to work”.

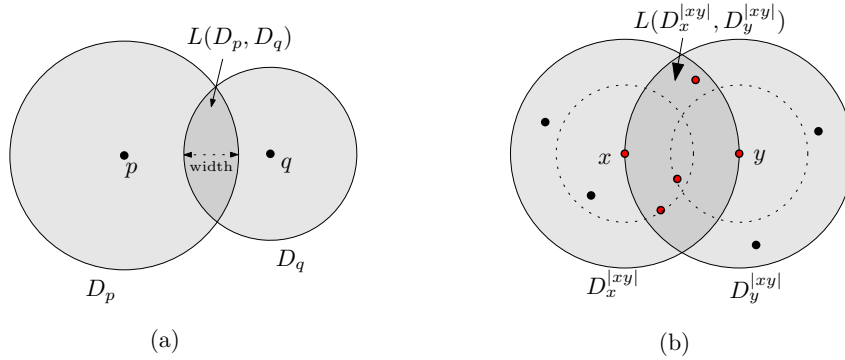
Our Contribution

In this paper we make significant progress on the maximum clique problem for unit disk graphs, disk graphs with disk radii lying in the interval $[1, 1 + \varepsilon]$, and intersection graphs of unit disks and axis-parallel rectangles.

Unit disk graph. We give an algorithm to compute a maximum clique in a unit disk graph in $O(n^{2.5} \log n)$ -time, which improves the previously best known running time of $O(n^3 \log n)$ [14]. Our algorithm is based on a divide-and-conquer approach that, unlike the previous algorithms that search a clique over all the lenses, shows how to efficiently merge solutions to the subproblems to achieve a faster time complexity. Such techniques have previously been used to accelerate computation for other computational geometry problems, e.g., when finding a closest pair in a point set [28], but appeared to be highly non-trivial while adapting it for the unit disk graph setting.

Intersection graph of unit disks and axis-parallel rectangles. We extend the co-2-subdivision approach to prove a $(4448/4449 \approx 0.9997)$ -inapproximability result for computing a maximum clique in an intersection graph that contains both unit disks and axis-parallel rectangles, and thus improve the previously known inapproximability factor of $7633010347/7633010348 \approx 0.9999$ [7]. Note that the use of a co-2-subdivision approach was previously thought to be unlikely in this setting by Bonnet et al. [7]. The key idea behind our NP-hardness reduction is to show that every Hamiltonian cubic graph admits a well-behaved edge orientation and vertex labeling, i.e., its vertices can be labeled and the edges can be oriented such that every vertex has two outgoing or two incoming edges where the labels of these corresponding neighbors are consecutive. While such orientation and labeling are of independent interest, they allow us to represent the complement of the 2-subdivision of a Hamiltonian cubic graph using a combination of unit disks and axis-parallel rectangles.

(ε, β) -disk graph. In an attempt to make progress on the case when the disk radii are in the interval $[1, 1 + \varepsilon]$, we introduce (ε, β) -disk graphs. A (ε, β) -disk graph, where ε and β are positive constants, is a disk graph where the radii of the disks are in the interval $[1, 1 + \varepsilon]$ and every lens is of width at least β . The parameter β can be thought of as the minimum width over all the lenses in the disk arrangement, where a *lens* is the convex intersection region of a pair of disks (Figure 1(a)). We show that the parameter β , i.e., the minimum lens width of the disk arrangement, may be used to make progress in the case when disk radii are in $[1, (1 + \varepsilon)]$. For example, if the minimum lens width is at least 0.265, then one can find a maximum clique for $\varepsilon \leq 0.0001$ in polynomial time.



■ **Figure 2** (a) Illustration for a lens. (b) Computation of a maximum clique, where the centers of the disks are shown in dots. The unit disks centered at x and y are shown in dotted circles. The centers inside $L(D_x^{|xy|}, D_y^{|xy|})$ are shown in red.

The existence of *non-Helly triple* in a disk arrangement, i.e., three pairwise intersecting disks without any common point of intersection (Figure 1(a)), typically makes the problem of finding a clique challenging (see Sec. 5). Since β is a lower bound on the width of every lens, a natural question is whether our choice for $\beta \geq 0.265$ already forbids the existence of non-Helly triples. We note that our choice for β still allows for non-Helly triples, and thus the result is non-trivial. We show that the lower bound on β could be leveraged to find for each non-Helly triple, a maximum clique that includes this triple. This extends the prior approach of finding a maximum clique in a unit disk graph that searches over all the pairwise intersecting disks [13] to a more general setting where the disk radii are in $[1, 1.0001]$. We believe that our proposed approach is interesting from the perspective of finding a way to make progress beyond unit disks even though the lower bound on β is large and the gain on ε is small.

2 Preliminaries

By D_q^r we denote a disk with radius r and center q . For the simplicity of the presentation, sometimes we omit the radius and simply use D_q to denote a disk with center q . Let D_p and D_q be a pair of disks. By $L(D_p, D_q)$ we denote the lens (i.e. the intersection region) of these disks (Figure 2(a)). For a line segment ab , we denote by $|ab|$ the length of the segment or the Euclidean distance between the points a and b . The *width* of a lens $L(D_p, D_q)$ is the length of the line segment determined by the intersection of pq and $L(D_p, D_q)$.

Let G be a graph. The *complement graph* \overline{G} of G is a graph on the same set of vertices where \overline{G} contains an edge if and only if it does not appear in G . A set S of vertices in G is called *independent* if no two vertices in S are adjacent in G . A *maximum independent set* $\alpha(G)$ is an independent set of largest cardinality. G is called *bipartite* if its vertices can be partitioned into two independent sets. G is called a *cobipartite graph* if the complement graph of G is a bipartite graph. G is called *cubic* if every vertex of G is of degree three. G is *Hamiltonian* if it has a cycle that contains each vertex of G exactly once.

3 Unit Disk Graph (UDG)

In this section we provide an $O(n^{2.5} \log n)$ -time algorithm to compute a maximum clique in a unit disk graph, where a geometric representation of the graph is given as an input.

Clark et al. [13] gave an $O(n^{4.5})$ -time algorithm to compute a maximum clique in a unit disk graph. The idea of the algorithm is as follows. For each edge (x, y) of the graph, consider two disks D_x and D_y such that their boundaries pass through y and x , respectively. Let S be the set of unit disks with centers in $L(D_x^{|xy|}, D_y^{|xy|})$. Clark et al. showed that the subgraph of G induced by the vertices corresponding to S is a cobipartite graph $G(S)$. One can thus find a maximum clique in $G(S)$ by computing a maximum independent set in the bipartite graph $\overline{G(S)}$. If (x, y) is the longest edge of a maximum clique M in G , then S must include all the centers of the disks in M and $G(S)$ will contain the largest clique in G . Therefore, one can try the above strategy over all edges and find a maximum clique in G . Since G contains $O(n^2)$ edges and since a maximum independent set in a bipartite graph can be computed in $O(n^{2.5})$ time by leveraging a maximum matching [21], the running time becomes $O(n^{4.5})$.

Breu [8] observed that Clark et al.'s approach [13] to find a maximum clique can be implemented in $O(n^{3.5} \log n)$ time using a result of Aggarwal et al. [2]. Specifically, Aggarwal et al. [2] showed how to compute a maximum independent set in $\overline{G(S)}$ in $O(n^{1.5} \log n)$ time using a data structure of [20, 24], and hence over $O(n^2)$ lenses the running time becomes $O(n^{3.5} \log n)$.

Eppstein [14] observed that while searching through the lenses, instead of computing the maximum independent set from scratch, one can exploit geometric properties to efficiently update and maintain a maximum independent set as follows. For a unit disk center p , let q be a point on the plane such that $|pq| = 2$. Consider two disks D_p and D_q such that their boundaries pass through q and p , respectively. One can now rotate the lens $L(D_p^{|pq|}, D_q^{|pq|})$ around p and update the maximum independent set in the graph corresponding to $L(D_p^{|pq|}, D_q^{|pq|})$ each time a point (i.e., a center of a unit disk) enters or exists from the lens. An update can be processed by an alternating path search in $O(n \log n)$ time [2]. Since the number of changes to $L(D_p^{|pq|}, D_q^{|pq|})$ is bounded by $O(n)$, the time spent for p is $O(n^2 \log n)$. Hence the overall running time is $O(n^3 \log n)$.

3.1 Idea of Our Algorithm

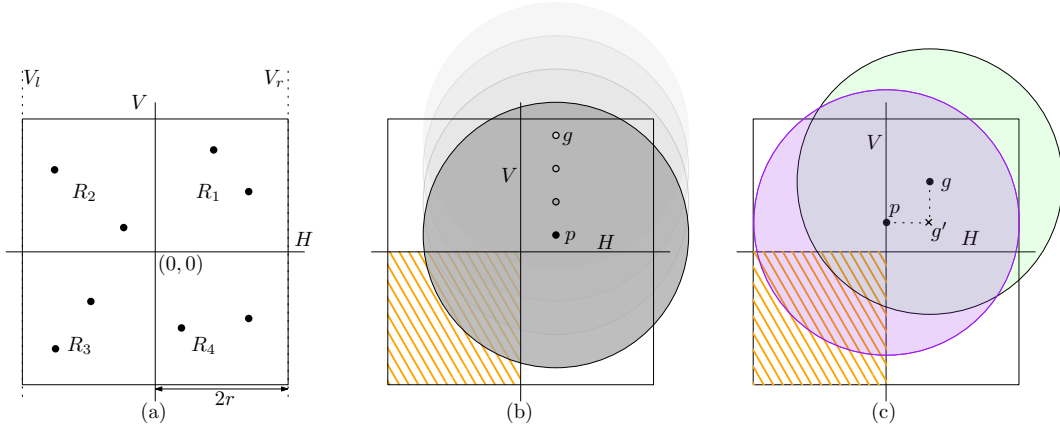
Let G be a disk graph with n vertices, where each disk is of radius r . Let P be the set of centers of the disks corresponding to the vertices of G . To find a maximum clique we take a divide-and-conquer approach as follows.

We rotate the plane so that no two points are in the same vertical or horizontal line. It is straightforward to perform such a rotation in $O(n^2)$ time. We sort the points in P with respect to their x-coordinates and find a vertical line V through a median x-coordinate such that at most $\lceil n/2 \rceil$ points of P are on each half-plane of V . Let P_l and P_r be the points on the closed left halfplane and closed right halfplane of V , respectively. We will find a maximum clique in P_l and P_r recursively.

Let M be a maximum clique in G . If the set of disk centers corresponding to M is a subset of either P_l or P_r , then such a clique must be returned as a solution to one of these two subproblems. Otherwise, each of P_l and P_r contains some points of M . To tackle such a case, it suffices to find a maximum clique in the vertical slab between the vertical lines V_l and V_r , where V_l and V_r are $2r$ units apart from V on the left halfplane and right halfplane, respectively. Let $Q \subseteq P$ be the set of points in the vertical slab. Then the maximum clique of G is the maximum clique found over the disks corresponding to the sets P_l , P_r and Q .

Let $T(n)$ be the time to compute a maximum clique in G . Let $F(n)$ be the time to compute a maximum clique in the vertical slab. Then $T(n)$ is defined as follows.

$$T(n) = 2T\left(\frac{n}{2}\right) + F(n). \tag{1}$$



■ **Figure 3** (a) The square S with the disk centers in black dots. (b)–(c) Illustration for Remark 2.

We now sort the points of Q with respect to their y -coordinates and find a horizontal line H through the median y -coordinate such that at most $\lceil |Q|/2 \rceil$ points of Q are on each half-plane of H . Let Q_t and Q_b be the points on the closed top halfplane and closed bottom halfplane of H , respectively. We now find a maximum clique in Q_t and Q_b recursively. If the set of disk centers corresponding to M is a subset of either Q_t or Q_b , then such a clique must be returned as a solution to one of these two subproblems. Otherwise, each of Q_t and Q_b contains some points of M . It now suffices to find a maximum clique in the square S of side length $4r$ with its center located at the intersection point of V and H (Figure 3(a)). Let $B(n)$ be the time to compute the maximum clique in S . Then $F(n)$ is defined as follows.

$$F(n) = 2F\left(\frac{n}{2}\right) + B(n). \quad (2)$$

In the following, we will show that a maximum clique in S can be computed in $O(n^{2.5} \log n)$ time. Consequently, $B(n) \in O(n^{2.5} \log n)$ and by Equation 2 and master theorem, $F(n) \in O(n^{2.5} \log n)$. Consequently, the time complexity determined by Equation 1 is $O(n^{2.5} \log n)$. Note that computing a maximum clique in the square S of side length $4r$ appears to be the bottleneck of our algorithm.

3.2 Computing a Maximum Clique in the Square S

Let M be a maximum clique in G and let C be the centers of the disks in M . Assume that $C \not\subseteq P_l$, $C \not\subseteq P_r$, $C \not\subseteq Q_t$ and $C \not\subseteq Q_b$, i.e., C is a subset of the points in S . We now show how to find M . Let o be the center of S , i.e., the intersection point of H and V . Without loss of generality assume that o is at $(0, 0)$. Let R_i , where $1 \leq i \leq 4$, be the region determined by the intersection of the i th quadrant and S (Figure 3(a)). We now give two remarks. Remark 1 follows directly from our assumption that $C \not\subseteq P_l$, $C \not\subseteq P_r$, $C \not\subseteq Q_t$ and $C \not\subseteq Q_b$.

► **Remark 1.** C must satisfy at least one of the following two conditions. (a) R_1 and R_3 each contains a point from C . (b) R_2 and R_4 each contains a point from C .

► **Remark 2.** Let p and g be two points inside R_1 where the x - and y -coordinates of g are at least as large as that of p . Let D_p^{2r} and D_g^{2r} be two disks of the same radius $2r$ centered at p and g , respectively. Then $(R_3 \cap D_g^{2r}) \subseteq (R_3 \cap D_p^{2r})$.

Proof. Consider first the case when g and p lie on the same vertical line (Figure 3(b)). Note that the interval $(H \cap D_g^{2r})$ increases as we move the center g vertically downward and the interval reaches the maximum when g hits H . Therefore, $(H \cap D_g^{2r}) \subseteq (H \cap D_p^{2r})$. Since g is vertically above p and both have the same radius, $(R_3 \cap D_g^{2r}) \subseteq (R_3 \cap D_p^{2r})$. The argument when g and p lie on the same horizontal line is symmetric.

Consider now the case when x- and y- coordinates of g are larger than that of p (Figure 3(c)). We can move D_g^{2r} vertically down to reach a point g' that has the same y-coordinate as that of p . Consequently, $(R_3 \cap D_g^{2r}) \subseteq (R_3 \cap D_{g'}^{2r})$. Finally, we move $D_{g'}^{2r}$ towards p . Hence we obtain $(R_3 \cap D_{g'}^{2r}) \subseteq (R_3 \cap D_p^{2r})$. ◀

We are now ready to describe the algorithm. The algorithm considers two cases depending on whether every disk center in C is within a distance of $2r$ from o . It processes each case in $O(n^{2.5} \log n)$ time, and then returns the maximum clique found over the whole process.

The high-level idea for finding a maximum clique is as follows. For the first case, we assume every disk center in C to be within a distance of $2r$ from o . The algorithm makes a guess for the farthest disk center q in C from o and then finds the other disks in the maximum clique by defining a lens that would contain all the disk centers of C . For the second case, we assume that at least one disk center in C has a distance of more than $2r$ from o . The algorithm makes a guess for the first point $p \in C$ in some particular point ordering and then finds the other disks in the maximum clique by defining a lens that would contain all the disk centers of C . We now describe the details.

Case 1 (Every disk center in C is within a distance of $2r$ from o)

Let q be a point of C that has the largest distance from o . Without loss of generality assume that q lies in R_2 . We now order the points of R_2 that are within distance $2r$ from o in decreasing order of their distances from o (breaking ties arbitrarily). Figure 4(a) illustrates this order in orange concentric circles. Let σ be the resulting point ordering. We iteratively consider each point in σ to be q and then find a maximum clique as follows.

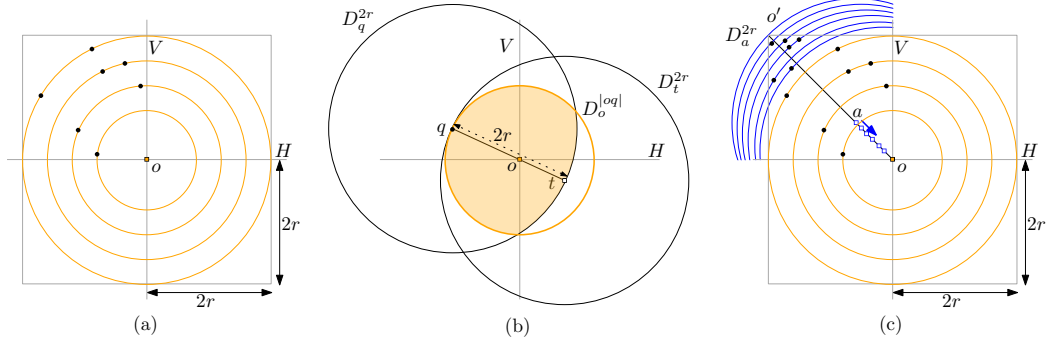
Let $D_o^{|oq|}$ be a disk centered at $o = (0,0)$ such that its boundary passes through q (Figure 4(b)). Since q is the furthest point of C from o , every point of C is contained in $D_o^{|oq|}$. Let t be a point in R_4 that lies on the line through o and q at a distance of $2r$ from q .

We now show that every point of C is in $L(D_q^{2r}, D_t^{2r})$. Suppose for a contradiction that there exists a point $e \in C$ which is not in $L(D_q^{2r}, D_t^{2r})$. If e belongs to $S \setminus D_q^{2r}$, then D_e cannot intersect D_q . Therefore, e must lie in the region $D_o^{|oq|} \cap (D_q^{2r} \setminus D_t^{2r})$. Note that q, o and t lie on the same line. Since the boundaries of both $D_o^{|oq|}$ and D_t^{2r} pass through q and $|qt| \geq |qo|$, we have $D_o^{|oq|} \subseteq D_t^{2r}$, and hence the point e cannot exist.

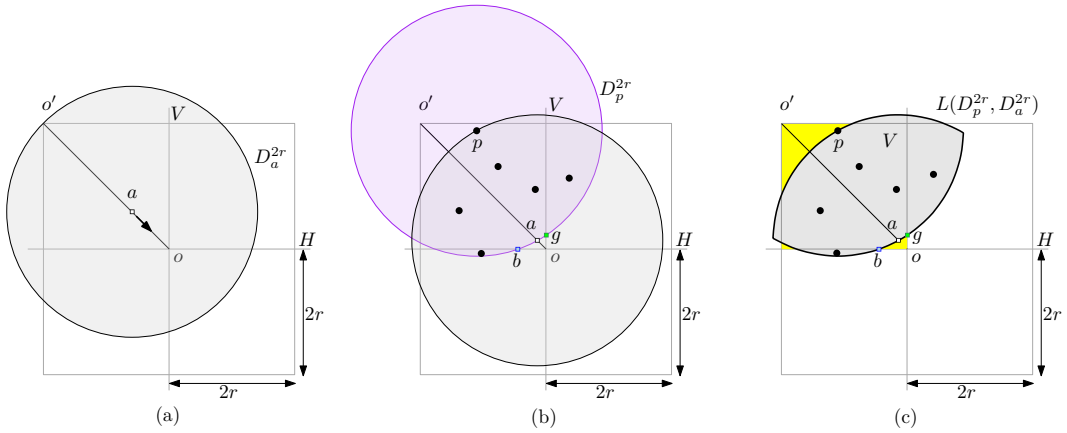
Since the intersection graph induced by the disks with centers in $L(D_q^{2r}, D_t^{2r})$ is a cobipartite graph [13], a maximum clique in this graph can be computed in $O(n^{1.5} \log n)$ time [2]. Over all choices for q in σ , the running time becomes $O(n^{2.5} \log n)$.

Case 2 (There exists a disk center in C with distance more than $2r$ to o)

Without loss of generality assume that R_2 contains a disk center that belongs to C and has a distance of more than $2r$ from o . We now consider the points of R_2 that have a distance of more than $2r$ from o and order them by sweeping a disk as described below. Let o' be the top left corner of R_2 . Let D_a^{2r} be a disk of radius $2r$ such that its boundary passes through o' and its center a lies on the line oo' (Figure 5(a)). We now move D_a^{2r} along the line $o'o$



■ **Figure 4** (a) Illustration for point ordering in R_2 in Case 1. (b) Illustration for D_q^{2r} and D_t^{2r} . (c) Point ordering in Case 2.



■ **Figure 5** (a)–(b) Illustration for sweeping D_a^{2r} . (c) Illustration for Case A of Lemma 3.

by moving the center a towards o . We order the points in the order D_a^{2r} hits them at its boundary as we move a towards o (breaking ties arbitrarily). Figure 4(c) illustrates this order in blue circular arcs. Let σ' be the resulting point ordering.

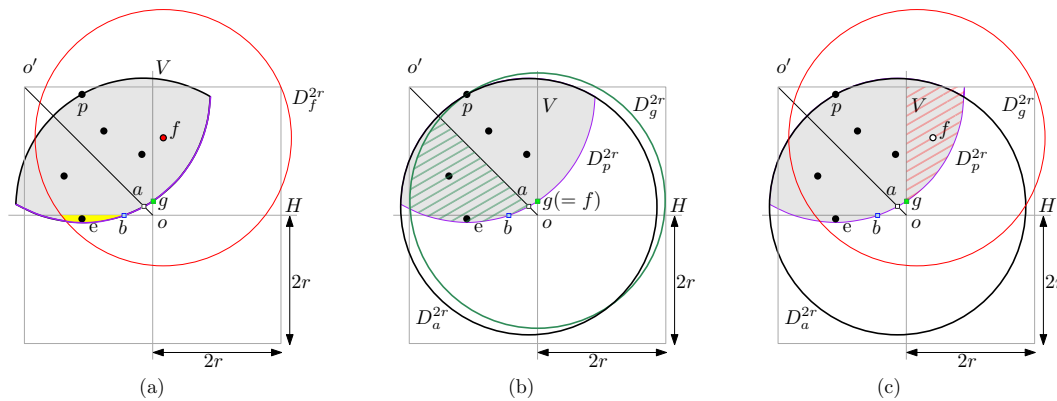
Let p be the first point of C in R_2 that is hit by D_a^{2r} at its boundary. Then the boundary of D_p^{2r} passes through a (Figure 5(b)). Let b be the point of intersection between D_p^{2r} and H that is closer to o . Let g be the point of intersection between D_p^{2r} and V that is closer to o .

In the following lemma (Lemma 3), we show that every point of C belongs to the lens $L(D_p^{2r}, D_a^{2r})$. Since the corresponding intersection graph is a cobipartite graph [13], a maximum clique in this graph can be computed in $O(n^{1.5} \log n)$ time [2]. Over all choices for p in σ' , the running time becomes $O(n^{2.5} \log n)$.

► **Lemma 3.** *Let p be the first point of C in R_2 that is hit by D_a^{2r} at its boundary. Then every point of C belongs to the lens $L(D_p^{2r}, D_a^{2r})$.*

Proof. Suppose for a contradiction that there exists a point $e \in C$ that does not belong to $L(D_p^{2r}, D_a^{2r})$. We now consider the following three subcases and in each case we show that such a point e cannot exist.

Case A ($e \in R_2$): Figure 5(c) highlights the potential locations for e in yellow. If $e \in (D_a^{2r} \setminus D_p^{2r})$, then e fails to intersect p . If $e \in (D_p^{2r} \setminus D_a^{2r})$, then $e \in C$ must be the first point (instead of p) in R_2 that is hit by D_a^{2r} , which leads to a contradiction.



■ **Figure 6** Illustration for Case C of Lemma 3. (a) The boundaries of D_f^{2r} and D_p^{2r} are shown in red and purple, respectively. (b) The scenario when f coincides with g . (c) The scenario when $f \neq g$.

Case B ($e \in R_4$): Since D_p^r and D_e^r intersect, e must lie inside D_p^{2r} . Note that $|po| > 2r$. Since p belongs to C , we have $C \subseteq D_p^{2r}$, and hence, R_4 cannot contain any point of C .

Case C ($e \in R_1$ or $e \in R_3$): Without loss of generality assume that $e \in R_3$. The argument when $e \in R_1$ is symmetric. We have explained in Case B that R_4 cannot contain any point of C . Therefore, by Remark 1, R_1 and R_3 each contains a point from C . Let $f \in C$ be a point in R_1 . Then D_e^r must intersect D_f^r . In other words, e must belong to D_f^{2r} (Figure 6(a)). It now suffices to show that the region $R_3 \cap L(D_f^{2r} \cap D_p^{2r})$, which is the potential location for e (shown in yellow), is a subset of $L(D_p^{2r}, D_a^{2r})$.

Consider first the case when f coincides with g . Since g is on the boundary of D_p^{2r} , the boundary of D_g^{2r} passes through p . Note that if we walk along the boundary of D_p^{2r} starting at g clockwise, then we first hit g and then a . If we keep walking then we must hit the boundary of D_g^{2r} before the boundary of D_a^{2r} . Therefore, the region of $L(D_p^{2r}, D_g^{2r})$ on the left halfplane of line oo' (as shown in rising pattern in Figure 6(b)) is a subset of $L(D_p^{2r}, D_a^{2r})$. Consequently, $R_3 \cap L(D_p^{2r}, D_g^{2r})$ is a subset of $L(D_p^{2r}, D_a^{2r})$.

Consider now the case when $f \neq g$ and $f \in (R_1 \cap D_p^{2r})$ (Figure 6(c)). Since the x - and y -coordinates of f are at least as large as that of g , by Remark 2, we obtain $(R_3 \cap D_f^{2r}) \subseteq (R_3 \cap D_g^{2r})$. Together with the argument that $R_3 \cap L(D_p^{2r}, D_g^{2r})$ is a subset of $L(D_p^{2r}, D_a^{2r})$, one can observe that $R_3 \cap L(D_p^{2r}, D_f^{2r})$ is a subset of $L(D_p^{2r}, D_a^{2r})$. ◀

Since a maximum clique in S can be computed in $O(n^{2.5} \log n)$ time, the strategy of Section 3.1 yields a running time of $O(n^{2.5} \log n)$.

► **Theorem 4.** *Given a set of n unit disks in the Euclidean plane, a maximum clique in the corresponding disk graph can be computed in $O(n^{2.5} \log n)$ time.*

4 Combination of Unit Disks and Axis-Parallel Rectangles

In this section we show that the maximum clique problem for an intersection graph of unit disks and axis-parallel rectangles is NP-hard to approximate within a factor of $\frac{4448}{4449} \approx 0.9997$. We first show an inapproximability result for computing a maximum independent set and then use this result to prove the APX-hardness for computing a maximum clique.

4.1 Inapproximability of Computing a Maximum Independent Set

The proof of the following theorem is obtained by leveraging an inapproximability result of [11] and a graph transformation technique of [18] (see the full version [15]).

► **Theorem 5.** *The problem of computing a maximum independent set in a 2-subdivision of a Hamiltonian cubic graph is NP-hard to approximate within $\frac{4448}{4449} \approx 0.9997$, even when a Hamiltonian cycle is given as an input.*

Let G be a Hamiltonian cubic graph with n vertices and let H be a 2-subdivision of G . In Section 4.2 we show that given a Hamiltonian cycle of G , \overline{H} can be represented as an intersection graph of unit disks and axis-parallel rectangles in polynomial time. Since a maximum independent set in H corresponds to a maximum clique in \overline{H} and vice versa, the inapproximability result follows from Theorem 5.

4.2 Representing \overline{H} with Unit Disks and Axis-parallel Rectangles

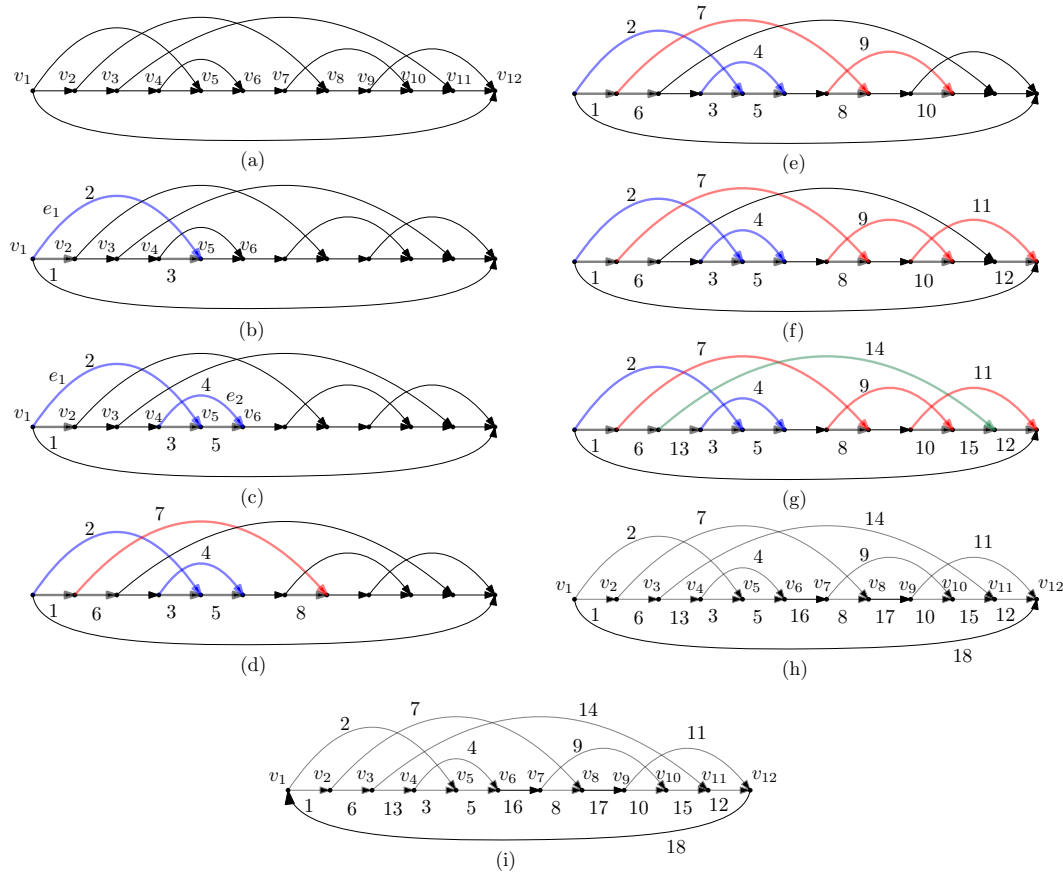
The number of edges in G is $m = 3n/2$. We first show that the edges of G can be oriented and labeled with distinct positive integers from 1 to $3n/2$ such that each vertex has exactly two of its incident edges with the same orientation and they are labeled with consecutive numbers. We will refer to such labeling as a *pair-oriented labeling*. Figure 7(i) illustrates a pair-oriented labeling of a cubic graph, e.g., v_5 has two incoming edges which are labeled with 2 and 3, and v_7 has two outgoing edges which are labeled with 8 and 9. We will use this labeling to construct the required intersection representation for \overline{H} .

► **Lemma 6.** *Let G be a Hamiltonian cubic graph with n vertices. Then G admits a pair-oriented labeling. Furthermore, given a Hamiltonian cycle C in G , a pair-oriented labeling for G can be computed in polynomial time.*

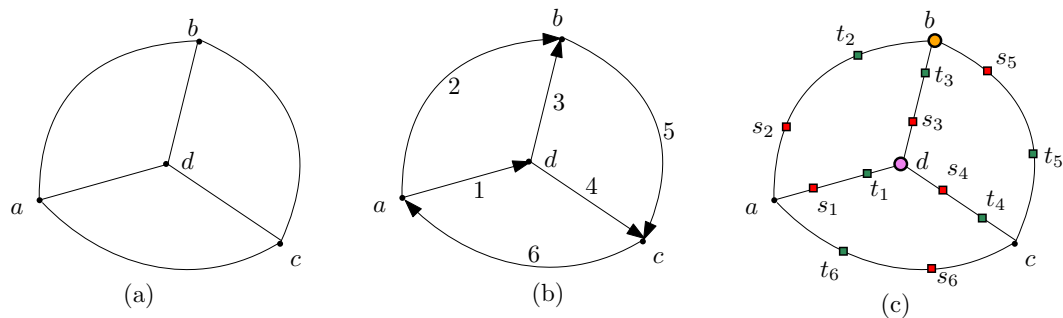
Proof (Outline). We first orient the edges of G , as follows. Let (v_1, v_2, \dots, v_n) be the ordering of the vertices of G on C . For each edge (v_i, v_j) , where $i < j$, we orient the edge from v_i to v_j , as illustrated in Figure 7(a).

We now give an incremental construction for the edge labeling. We first find the smallest index k such that v_k has a pair of outgoing edges that are not yet labeled. We now find a *maximal edge sequence* S_k of non-Hamiltonian edges e_1, e_2, \dots, e_q such that for each i from 1 to $q - 1$, there is a Hamiltonian edge that connects the source vertex of e_{i+1} to the target vertex of e_i . Figure 7(c) illustrates such a maximal edge sequence e_1, e_2 , where $e_1 = (v_1, v_5)$ and $e_2 = (v_4, v_6)$, and the edge (v_4, v_5) is a Hamiltonian edge. Let ℓ be the largest number that has been used for edge labeling so far. We then label the edges e_1, e_2, \dots, e_q with $\ell + 2, \ell + 4, \dots, \ell + 2q$ and the Hamiltonian edges that they nest with $\ell + 1, \ell + 3, \dots, \ell + (2q + 1)$. Let V_k be the set of vertices that appear on the edges of S_k . It is now straightforward to verify that every vertex of V_k has two edges with the same orientation and these edges are labeled with consecutive numbers. We repeatedly find such maximal edge sequences starting at the vertex with the smallest index that has two outgoing edges that are not yet labeled (Figure 7(a)–(g)). The remaining unlabeled edges are then labeled arbitrarily using remaining integers (Figure 7(h)) and finally, the orientation of the edge (v_1, v_n) is reversed to obtain the required pair-oriented labeling (Figure 7(i)). ◀

Let γ be a pair-oriented labeling of G (Figure 8(a)–(b)). By $\gamma(e)$ we denote the label of an edge e . Let e be an edge of G with source s and target t . We now label the two division vertices corresponding to e in the 2-subdivision H . The division vertex adjacent to s receives

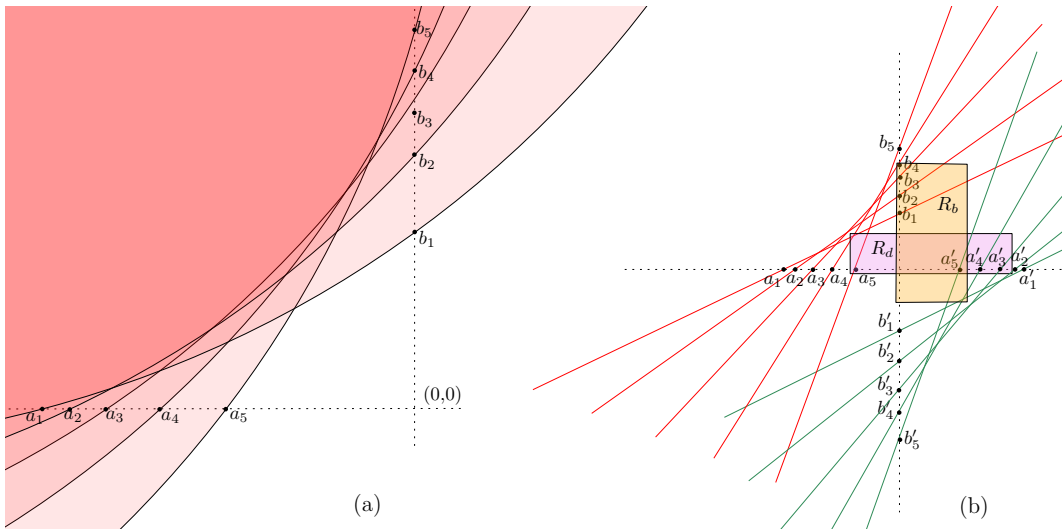


■ **Figure 7** Construction of a pair-oriented labeling of a Hamiltonian cubic graph. The first maximal edge sequence S_1 is shown in blue. S_1 consists of the non-Hamiltonian edges e_1, e_2 , and the set V_1 consists of their end vertices, i.e., $\{v_1, v_4, v_5, v_6\}$. The second and third maximal edge sequences are shown in red and green, respectively.



■ **Figure 8** (a)–(b) G , and its pair-oriented labeling. (c) Labeling of the division vertices of H .

the label $s_{\gamma(e)}$ and the division vertex adjacent to t receives the label $t_{\gamma(e)}$. We refer to $s_{\gamma(e)}$ and $t_{\gamma(e)}$ as a *type-s* and *type-t* label, respectively. By the property of γ (Lemma 6), each original vertex v in H is now adjacent to exactly two division vertices of the same type with their indices numbered with consecutive numbers. For example in Figure 8(c), the vertices d and b are adjacent to s_3, s_4 and t_2, t_3 , respectively.



■ **Figure 9** (a) Arrangement of the unit disks corresponding to type- s division vertices. (b) Illustration for the intersection representation of \overline{H} . Only a subset of unit disks and the rectangles corresponding to b and d are shown for better readability.

The intersection representation of \overline{H} now follows from the construction of [7]. We briefly describe the construction at a high level for completeness. Consider a set of unit disks C_s for the type- s vertices with centers in the second quadrant such that they intersect the negative x -axis and positive y -axis, but not the positive x -axis or negative y -axis. Furthermore, the ordering of the disks obtained by walking from $(0, 0)$ to $(-\infty, 0)$ is reversed when walking from $(0, 0)$ to $(0, \infty)$ (Figure 9(a)). Let \mathcal{I}_s be the convex region determined by the intersection of all type- s disks. The set of unit disks C_t for type- t vertices is placed on the 4th quadrant symmetrically. Let \mathcal{I}_t be the convex region determined by the intersection of all the type- t disks. For a sufficiently large radius, the disk boundaries appear similar to a set of halfplanes (Figure 9(b)) and each disk in C_s intersects every disk in C_t except for the one with the same label. Therefore, all the intersections between division vertices of \overline{H} are realized. Note that the original vertices of H in \overline{H} form a clique in \overline{H} and each original vertex is adjacent to all but three division vertices in \overline{H} . We now represent the original vertices of \overline{H} with rectangles such that all of them enclose the point $(0, 0)$. Let b be an original vertex of \overline{H} . Without loss of generality assume that b has two type- t neighbors and one type s neighbor (Figure 8(c)). By the property of the pair-oriented labeling, the type- t neighbors are labeled consecutively. Let t_i, t_{i+1}, s_j be the neighbors of b . We now create a rectangle R_b to represent b . We place the top-left corner of R_b near the intersection point of the boundaries of the disks for t_i, t_{i+1} such that R_b does not intersect these disks but intersects all other type- t disks. We place the bottom-right corner of R_b near the circular segment determined by the disk for s_j on \mathcal{I}_t such that R_b does not intersect the disk for s_j but intersects all other type- s disks. We refer to [7] for a formal reduction.

► **Theorem 7.** *The problem of computing a maximum clique in an intersection graph of unit disks and axis-parallel rectangles is NP-hard to approximate within a factor of $\frac{4448}{4449} \approx 0.9997$.*

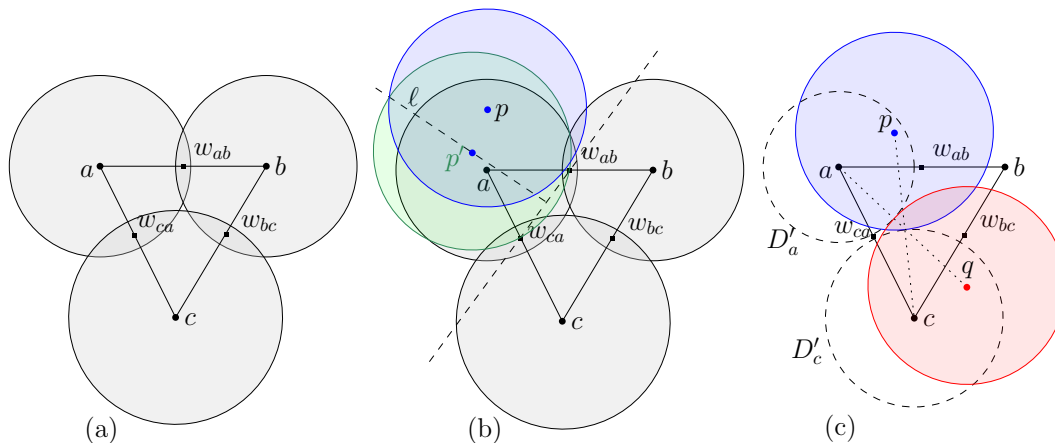


Figure 10 (a) A non-Helly triple $\{D_a, D_b, D_c\}$. Illustration for (b) Case 1 and (c) Case 2.

5 Finding a Maximum Clique in an (ϵ, β) -disk graph

In this section, we give a polynomial-time algorithm to compute a maximum clique in an (ϵ, β) -disk graph. By definition, the radii of the disks are in $[1, 1 + \epsilon]$ and every lens is of width at least β . We give an $O(n^4)$ -time algorithm when $\beta \geq 0.265$ and $\epsilon \leq 0.0001$. Although β could be expressed as a function of ϵ , for simplicity of the presentation, we set specific values to β and ϵ and often use crude bounds to simplify the arguments. Therefore, we believe one can choose slightly better parameters by using a tedious case analysis.

Let G be an (ϵ, β) -disk graph. Let M be a set of disks determining a maximum clique in G . In the following, we will use Helly’s theorem [19], i.e., for a collection of convex sets in \mathbb{R}^d , if the intersection of every $(d + 1)$ of these sets is non-empty then the collection must have a non-empty intersection. Recall that for three disks $\{D_a, D_b, D_c\}$, if $(D_a \cap D_b \cap D_c) = \emptyset$, then we call them a *non-Helly triple*. Otherwise, we refer to them as a *Helly triple*.

Consider three unit disks with centers at the corners of an equilateral triangle. If these disks intersect exactly at one point, then the width of each lens is $(2 - 2 \cos 30^\circ) \approx 0.2679$, which is larger than β . Therefore, in an (ϵ, β) -disk graph, we may have non-Helly triples. We now consider two cases depending on whether M contains a non-Helly triple or not.

5.1 M does not contain any non-Helly triple

If M does not contain any non-Helly triple, then by Helly’s theorem [19], the disks in M have a non-empty intersection. Let \mathcal{R} be the set of connected regions or cells determined by the arrangement of the disk boundaries. We examine for each cell r in \mathcal{R} , the number of disks that contains r , and find a maximum set of mutually intersecting disks. It is straightforward to compute the arrangement of n disks in $O(n^3)$ -time (even faster algorithms exist [1]) and each cell can be checked in $O(n)$ time. Hence finding a maximum clique takes $O(n^4)$ time.

5.2 M contains a non-Helly triple

If M contains a non-Helly triple, then let $\{D_a, D_b, D_c\}$ be such a non-Helly triple in M (Figure 10(a)). Let w_{ab}, w_{bc}, w_{ca} be the midpoint of the lenses $L(D_a, D_b), L(D_b, D_c)$ and $L(D_c, D_a)$, respectively. The following two lemmas give some properties corresponding to the non-Helly triple and their proof is included in the full version [15].

► **Lemma 8.** *If $\{D_a, D_b, D_c\}$ is a non-Helly triple, $\beta \geq 0.265$, and $\varepsilon \leq 0.0001$, then the lenses $L(D_a, D_b)$, $L(D_b, D_c)$ and $L(D_c, D_a)$ are of width less than 0.275. Furthermore, the interior angles of Δabc are in the interval $[58.024^\circ, 60.988^\circ]$.*

► **Lemma 9.** *If $\{D_a, D_b, D_c\}$ is a non-Helly triple, $\beta \geq 0.265$, and $\varepsilon \leq 0.0001$, then Δabc and $\Delta w_{ab}w_{bc}w_{ca}$ satisfy the following properties. (a) $1.725 \leq |ab|, |bc|, |ca| \leq (1.735 + 2\varepsilon)$. (b) For each point q in $\{w_{ab}, w_{bc}, w_{ca}\}$, the distances from q to the center of the two disks containing q are in the interval $[0.8625, (0.8675 + \varepsilon)]$. (c) The length of each side of $\Delta w_{ab}w_{bc}w_{ca}$ is in the interval $[0.883, 0.887]$.*

Let O be the disks in the disk graph representation and let O' be $O \setminus \{D_a, D_b, D_c\}$. We refer to a disk in O' as *type- k* , where $0 \leq k \leq 3$, if it contains exactly k points from $\{w_{ab}, w_{bc}, w_{ca}\}$. In the following we show that for a pair of disks D_p, D_q , if each of them intersects all the disks in $\{D_a, D_b, D_c\}$, then they must mutually intersect. As a consequence, we can find a maximum clique including $\{D_a, D_b, D_c\}$ in $O(n)$ time and a maximum clique over all possible $O(n^3)$ choices of non-Helly triples in $O(n^4)$ time.

Case 1 (At least one of D_p and D_q is of Type-0): We show that this case is trivial because a type-0 disk that intersects all the disks in the non-Helly triple but avoids $\{w_{ab}, w_{bc}, w_{ca}\}$ cannot exist.

Consider the disk D_p . We first show that if p lies inside $\Delta w_{ab}w_{bc}w_{ca}$, then D_p must contain a corner of $\Delta w_{ab}w_{bc}w_{ca}$. By Lemma 9, the maximum side length of $\Delta w_{ab}w_{bc}w_{ca}$ is at most 0.887. Therefore, the circumradius for $\Delta w_{ab}w_{bc}w_{ca}$ is bounded by $\frac{0.887}{\sqrt{3}} \leq 1$. Hence D_p must contain a corner of $\Delta w_{ab}w_{bc}w_{ca}$.

We now show that if p lies outside of $\Delta w_{ab}w_{bc}w_{ca}$, then D_p cannot create a lens of width β with D_a, D_b, D_c . Without loss of generality assume that the left-halfplane of the line through $w_{ab}w_{ca}$ contains p and the right-halfplane contains the centers b, c (Figure 10(b)).

Consider a disk $D_{p'}$ with the same radius as that of D_p such that its center p' lies outside of Δabc and its boundary passes through w_{ab} and w_{ca} . The following lemma gives an upper bound on $|ap'|$ and $\angle acp'$ and its proof is included in the full version [15].

► **Lemma 10.** *Let $D_{p'}$ be a disk such that the boundary of $D_{p'}$ passes through w_{ab} and w_{ca} and the center p' lies outside of Δabc . Then $|ap'| < (0.25 + \varepsilon)$ and $\angle acp' \leq 17.5^\circ$.*

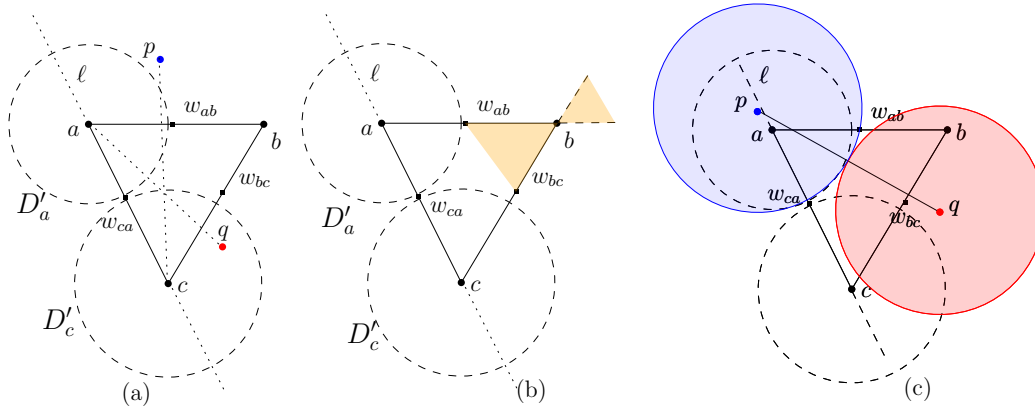
We now show that $D_{p'}$ cannot create a lens of width β with D_c . Since $|p'w_{ca}|$ is fixed, the distance $|p'c|$ decreases with the increase in $\angle p'cw_{ca}$ and decrease in $|cw_{ca}|$. Since $\angle p'cw_{ca} < 17.5^\circ$ and $|cw_{ca}| \geq 0.8625$, by using basic trigonometry on $\Delta p'cw_{ca}$ one can observe that $|p'c| \geq 1.78 > (2 - \beta)$. Therefore, $D_{p'}$ cannot create a lens of width β with D_c .

Since D_p does not contain w_{ab} and w_{bc} , p lies above or below the bisector ℓ of $w_{ab}w_{ca}$. Consider moving p' to p . Since moving p' above or below ℓ decreases the width of either $L(D_{p'}, D_b)$ or $L(D_{p'}, D_c)$, D_p cannot have a lens of width β with D_b and D_c simultaneously.

Case 2 (D_p and D_q are of Type-1): Without loss of generality assume that D_p and D_q contains w_{ab} and w_{bc} , respectively (Figure 10(c)). Let a_r, c_r, p_r, q_r be the radii of D_a, D_c, D_p, D_q , respectively. It now suffices to show that $|pq| \leq p_r + q_r$, i.e., D_p and D_q must intersect. Note that by the property of (ε, β) -graph, an intersection would imply a lens of width at least β , and hence we only show that $|pq| \leq p_r + q_r$.

Let D'_a and D'_c be the disks obtained by shrinking the radii of D_a and D_c by β . Since the width of the lenses created by the non-Helly triple is less than 0.275, the points w_{ab}, w_{bc}, w_{ca} lie outside of D'_a and D'_c . Since the width of each lens is at least β , D_p must intersect D'_c . Consider a line ℓ through ac with b on its right half-plane.

Consider first the scenario when p and q are on the right half-plane of ℓ . If p is above the line through bc and q is below the line through ab , then aq and pc intersect (Figure 11(a)). Therefore, $|pq| \leq |aq| + |pc| - |ac| \leq (a_r + q_r - 0.265) + (p_r + c_r - 0.265) - (a_r + c_r - 0.275) < (p_r + q_r)$. Otherwise, p, q lie on the right halfplane of the line through $w_{ab}w_{bc}$ in the wedge determined by $\angle abc$ and its opposite angle, as shaded in orange in Figure 11(b). Since $\max\{|w_{ab}w_{bc}|, |bw_{ab}|, |bw_{bc}|\} \leq 1 + \varepsilon$, it is straightforward to observe that D_p and D_q intersect.



■ **Figure 11** Illustration for the locations of p and q . (a)–(b) Case 1. (c) Case 2.

If p and q are on different sides of ℓ (Figure 11(c)), then without loss of generality assume that p lies on the left half-plane and q lies on the right half-plane. Here we show that $|ap| \leq (0.25 - \varepsilon)$ (see the full version [15] for details). Consequently, $|pq| \leq |ap| + |aq| \leq (0.25 - \varepsilon) + (a_r + q_r - 0.265) = (p_r + q_r) + (a_r - p_r) - \varepsilon - 0.015 < (p_r + q_r)$.

Case 3 (D_p and D_q are of Type-2 or Type-3): Since D_p and D_q each contains at least two points from $\{w_{ab}, w_{bc}, w_{ca}\}$, they must intersect.

Case 4 (One of D_p and D_q is of type-1 and the other is of type-2 or type-3): The case when D_p and D_q contains a common point from $\{w_{ab}, w_{bc}, w_{ca}\}$ is trivial. Therefore, without loss of generality assume that D_p is of type-1 and contains w_{ab} , and D_q is of type-2 and contains w_{bc} and w_{ca} . We use the same setting as in Case 2, i.e., ℓ is the line through ac and b lies on the right half-plane. We now move D_q counter-clockwise without changing the distance of $|pq|$ and stop as soon as w_{ca} hits the boundary of D_q . By an analysis similar to Case 2, we now can observe that $|pq| \leq p_r + q_r$, and hence D_p and D_q must intersect.

► **Theorem 11.** *Given a set of n disks in the Euclidean plane such that the width of every lens is at least 0.265 and the radii are in the interval $[1, 1.0001]$, a maximum clique in the corresponding disk graph can be computed in $O(n^4)$ time.*

6 Conclusion and Directions for Future Work

We gave an $O(n^{2.5} \log n)$ -time algorithm to compute a maximum clique in a unit disk graph. A natural avenue for future research would be to improve the time complexity of the algorithm. We proved that for the combination of unit disks and axis-parallel rectangles, a maximum clique is NP-hard to approximate within a factor of 4448/4449. We obtained the result using a co-2-subdivision approach, and along the way, we showed that every Hamiltonian cubic graph admits a pair-oriented labeling. It would be interesting to improve the inapproximability

factor, and one way to achieve this would be to examine whether pair-oriented labelings exist also for non-Hamiltonian cubic graphs. We showed that if the width of every lens is at least 0.265, then one can find a maximum clique in polynomial time in a more general setting where the disk radii are in $[1, 1.0001]$. We believe that with tedious case analysis, these numbers may be improved slightly, however, it would be challenging to lower β down to 0.2 using the current technique.

References

- 1 Pankaj K. Agarwal and Micha Sharir. Arrangements and their applications. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. Elsevier, 2000. doi:10.1016/b978-0-444-82537-7.x5000-1.
- 2 Alok Aggarwal, Hiroshi Imai, Naoki Katoh, and Subhash Suri. Finding k points with minimum diameter and related problems. *J. Algorithms*, 12(1):38–56, 1991. doi:10.1016/0196-6774(91)90022-Q.
- 3 Christoph Ambühl and Uli Wagner. The clique problem in intersection graphs of ellipses and triangles. *Theory of Computing Systems*, 38(3):279–292, 2005. doi:10.1007/s00224-005-1141-6.
- 4 J. Bang-Jensen, B. Reed, M. Schacht, R. Šámal, B. Toft, and U. Wagner. *Topics in Discrete Mathematics, Dedicated to Jarik Nešetřil on the Occasion of his 60th birthday*, volume 26 of *Algorithms and Combinatorics*, pages 613–627. Springer, 2006.
- 5 Marthe Bonamy, Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Panos Giannopoulos, Eun Jung Kim, Pawel Rzazewski, Florian Sikora, and Stéphan Thomassé. EPTAS and subexponential algorithm for maximum clique on disk and unit ball graphs. *J. ACM*, 68(2):9:1–9:38, 2021. doi:10.1145/3433160.
- 6 Édouard Bonnet, Panos Giannopoulos, Eun Jung Kim, Pawel Rzazewski, and Florian Sikora. QPTAS and subexponential algorithm for maximum clique on disk graphs. In Bettina Speckmann and Csaba D. Tóth, editors, *Proceedings of the 34th International Symposium on Computational Geometry (SoCG)*, volume 99 of *LIPICs*, pages 12:1–12:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICs.SoCG.2018.12.
- 7 Édouard Bonnet, Nicolas Grelier, and Tillmann Miltzow. Maximum Clique in Disk-Like Intersection Graphs. In Nitin Saxena and Sunil Simon, editors, *Proceedings of the 40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, volume 182 of *LIPICs*, pages 17:1–17:18. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.FSTTCS.2020.17.
- 8 Heinz Breu. *Algorithmic aspects of constrained unit disk graphs*. PhD thesis, University of British Columbia, 1996.
- 9 Sergio Cabello. Maximum clique for disks of two sizes. Open problems from Geometric Intersection Graphs: Problems and Directions, CG Week Workshop, 2015.
- 10 Sergio Cabello, Jean Cardinal, and Stefan Langerman. The clique problem in ray intersection graphs. *Discret. Comput. Geom.*, 50(3):771–783, 2013. doi:10.1007/s00454-013-9538-5.
- 11 Miroslav Chlebík and Janka Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theor. Comput. Sci.*, 354(3):320–338, 2006. doi:10.1016/j.tcs.2005.11.029.
- 12 Miroslav Chlebík and Janka Chlebíková. The complexity of combinatorial optimization problems on d -dimensional boxes. *SIAM J. Discret. Math.*, 21(1):158–169, 2007. doi:10.1137/050629276.
- 13 Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discret. Math.*, 86(1-3):165–177, 1990. doi:10.1016/0012-365X(90)90358-0.
- 14 David Eppstein. Graph-theoretic solutions to computational geometry problems. In Christophe Paul and Michel Habib, editors, *Proceedings of the 35th International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, pages 1–16, 2009. doi:10.1007/978-3-642-11409-0_1.

- 15 Jared Espenant, J. Mark Keil, and Debajyoti Mondal. Finding a maximum clique in a disk graph, 2023. doi:10.48550/ARXIV.2303.07645.
- 16 Stefan Felsner, Rudolf Müller, and Lorenz Wernisch. Trapezoid graphs and generalizations, geometry and algorithms. *Discret. Appl. Math.*, 74(1):13–32, 1997. doi:10.1016/S0166-218X(96)00013-3.
- 17 Aleksei V. Fishkin. Disk graphs: A short survey. In Klaus Jansen and Roberto Solis-Oba, editors, *Approximation and Online Algorithms, First International Workshop, WAOA 2003, Budapest, Hungary, September 16-18, 2003, Revised Papers*, volume 2909 of *Lecture Notes in Computer Science*, pages 260–264. Springer, 2003. doi:10.1007/978-3-540-24592-6_23.
- 18 Herbert Fleischner, Gert Sabidussi, and Vladimir I. Sarvanov. Maximum independent sets in 3- and 4-regular hamiltonian graphs. *Discret. Math.*, 310(20):2742–2749, 2010. doi:10.1016/j.disc.2010.05.028.
- 19 E. Helly. über mengen konvexer körper mit gemeinschaftlichen punkten. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 32:175–176, 1923.
- 20 John Hershberger and Subhash Suri. Finding tailored partitions. *J. Algorithms*, 12(3):431–463, 1991. doi:10.1016/0196-6774(91)90013-0.
- 21 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. doi:10.1137/0202019.
- 22 Mark L. Huson and Arunabha Sen. Broadcast scheduling algorithms for radio networks. In *Proceedings of MILCOM'95*, volume 2, pages 647–651. IEEE, 1995.
- 23 Hiroshi Imai and Takao Asano. Finding the connected components and a maximum clique of an intersection graph of rectangles in the plane. *J. Algorithms*, 4(4):310–323, 1983. doi:10.1016/0196-6774(83)90012-3.
- 24 Hiroshi Imai and Takao Asano. Efficient algorithms for geometric graph search problems. *SIAM J. Comput.*, 15(2):478–494, 1986. doi:10.1137/0215033.
- 25 J. Mark Keil, Debajyoti Mondal, Ehsan Moradi, and Yakov Nekrich. Finding a maximum clique in a grounded 1-bend string graph. *Journal of Graph Algorithms and Applications*, 26(4), 2022. doi:10.7155/jgaa.00608.
- 26 Jan Kratochvíl and Jaroslav Nešetřil. Independent set and clique problems in intersection-defined classes of graphs. *Commentationes Mathematicae Universitatis Carolinae*, 31(1):85–93, 1990.
- 27 Matthias Middendorf and Frank Pfeiffer. The max clique problem in classes of string-graphs. *Discret. Math.*, 108(1-3):365–372, 1992. doi:10.1016/0012-365X(92)90688-C.
- 28 Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 151–162, 1975. doi:10.1109/SFCS.1975.8.
- 29 Alexander Tiskin. Fast distance multiplication of unit-monge matrices. *Algorithmica*, 71(4):859–888, 2015. doi:10.1007/s00453-013-9830-z.




Linear Size Universal Point Sets for Classes of Planar Graphs

Stefan Felsner   

Institute of Mathematics, Technische Universität Berlin, Germany

Hendrik Schrezenmaier   

Institute of Mathematics, Technische Universität Berlin, Germany

Felix Schröder   

Institute of Mathematics, Technische Universität Berlin, Germany

Raphael Steiner   

Institute of Theoretical Computer Science, Department of Computer Science, ETH Zürich, Switzerland

Abstract

A finite set P of points in the plane is n -universal with respect to a class \mathcal{C} of planar graphs if every n -vertex graph in \mathcal{C} admits a crossing-free straight-line drawing with vertices at points of P .

For the class of all planar graphs the best known upper bound on the size of a universal point set is quadratic and the best known lower bound is linear in n .

Some classes of planar graphs are known to admit universal point sets of near linear size, however, there are no truly linear bounds for interesting classes beyond outerplanar graphs.

In this paper, we show that there is a universal point set of size $2n - 2$ for the class of bipartite planar graphs with n vertices. The same point set is also universal for the class of n -vertex planar graphs of maximum degree 3. The point set used for the results is what we call an exploding double chain, and we prove that this point set allows planar straight-line embeddings of many more planar graphs, namely of all subgraphs of planar graphs admitting a one-sided Hamiltonian cycle.

The result for bipartite graphs also implies that every n -vertex plane graph has a 1-bend drawing all whose bends and vertices are contained in a specific point set of size $4n - 6$, this improves a bound of $6n - 10$ for the same problem by Löffler and Tóth.

2012 ACM Subject Classification Mathematics of computing → Graph theory; Mathematics of computing → Graphs and surfaces; Human-centered computing → Graph drawings; Theory of computation → Computational geometry; Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Graph drawing, Universal point set, One-sided Hamiltonian, 2-page book embedding, Separating decomposition, Quadrangulation, 2-tree, Subcubic planar graph

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.31

Related Version *Full Version:* <https://arxiv.org/abs/2303.00109>

Funding *Stefan Felsner:* DFG Project FE 340/12-1.

Hendrik Schrezenmaier: DFG Project FE 340/12-1.

Raphael Steiner: DFG-GRK 2434 Facets of Complexity and an ETH Zürich Postdoctoral Fellowship.

Acknowledgements We are highly indebted to Henry Förster, Linda Kleist, Joachim Orthaber and Marco Ricci due to discussions during GG-Week 2022 resulting in a solution to the problem of separating 2-cycles in our proof for subcubic graphs.



© Stefan Felsner, Hendrik Schrezenmaier, Felix Schröder, and Raphael Steiner; licensed under Creative Commons License CC-BY 4.0

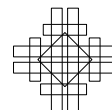
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 31; pp. 31:1–31:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Given a family \mathcal{C} of planar graphs and a positive integer n , a point set $P \subseteq \mathbb{R}^2$ is called an *n-universal point set* for the class \mathcal{C} or simply *n-universal* for \mathcal{C} if for every graph $G \in \mathcal{C}$ on n vertices there exists a straight-line crossing-free drawing of G such that every vertex of G is placed at a point of P .

To determine the minimum size of universal sets for classes of planar graphs is a fundamental problem in geometric graph theory, see e.g. Problem [17] in the Open Problem Garden. More specifically, the quest is for good bounds on the minimum size $f_{\mathcal{C}}(n)$ of an n -universal point set for a class \mathcal{C} .

Schnyder [21] showed that for $n \geq 3$ the $[n-1] \times [n-1]$ -grid forms an n -universal point set for planar graphs, even if the combinatorial embedding of the planar graph is prescribed. This shows that $f(n) := f_{\mathcal{P}}(n) \leq n^2 \in O(n^2)$, where \mathcal{P} is the class of all planar graphs. Asymptotically, the quadratic upper bound on $f(n)$ remains the state of the art. Only the multiplicative constant in this bound has seen some improvement, the current upper bound is $f(n) \leq \frac{1}{4}n^2 + O(n)$ by Bannister et al. [5]. For several subclasses \mathcal{C} of planar graphs, better upper bounds are known: A classical result by Gritzmann et al. [13] is that every outerplanar n -vertex graph embeds straight-line on *any* set of n points in general position, and hence $f_{\text{out-pl}}(n) = n$. Near-linear upper bounds of $f_{\mathcal{C}}(n) = O(n \text{ polylog}(n))$ are known for 2-outerplanar graphs, simply nested graphs, and for the classes of bounded pathwidth [4, 5]. Finally, for the class \mathcal{C} of planar 3-trees (also known as Apollonian networks or stacked triangulations), $f_{\mathcal{C}}(n) = O(n^{3/2} \log n)$ has been proved by Fulek and Tóth [12].

As for lower bounds, the trivial bounds $n \leq f_{\mathcal{C}}(n) \leq f(n)$ hold for all $n \in \mathbb{N}$ and all planar graph classes \mathcal{C} . The current lower bound $f(n) \geq 1.293n - o(n)$ from [20] has been shown using planar 3-trees, we refer to [6, 8, 9, 15] for earlier work on lower bounds.

Choi, Chrobak and Costello [7] recently proved that point sets chosen uniformly at random from the unit square must have size $\Omega(n^2)$ to be universal for n -vertex planar graphs with high probability. This suggests that universal point sets of size $o(n^2)$ -if they exist- will not look nice, e.g., they will have a large ratio between shortest and largest distances.

In this paper we study a specific ordered point set H (the exploding double chain) and denote the initial piece of size $2n-2$ in H as H_n . Let \mathcal{C} be the class of all planar graphs G which have a plane straight-line drawing on the point set H_n where $n = |V(G)|$. That is, H_n forms an n -universal point set for \mathcal{C} .

A graph is POSH (partial one-sided Hamiltonian) if it is a spanning subgraph of a graph admitting a plane embedding with a one-sided Hamiltonian cycle (for definitions see Section 2). Triangulations with a one-sided Hamiltonian cycle have been studied before by Alam et al. [2] in the context of cartograms. They conjectured that every plane 4-connected triangulation has a one-sided Hamiltonian cycle. Later Alam and Kobourov [3] found a plane 4-connected triangulation on 113 vertices which has no one-sided Hamiltonian cycle.

Our main result (Theorem 3) is that every POSH graph is in \mathcal{C} . We let

$$\mathcal{C}' := \{\mathcal{G} : \mathcal{G} \text{ is POSH}\}.$$

Theorem 3 motivates further study of \mathcal{C}' . On the positive side we show that every bipartite plane graph is POSH (proof in Section 4). We proceed to use the construction for bipartite graphs to show that subcubic planar graphs have a POSH embedding in Section 5. On the negative side, we also show that not all 2-trees are POSH. We conclude with some conjectures and open problems in Section 7.

An exploding double chain was previously used by Löffler and Tóth [16]. They show that every planar graph with n vertices has a 1-bend drawing on a subset S_n of H with $|S_n| = 6n - 10$. Our result about bipartite graphs implies a better bound:

► **Corollary 1.** *There is a point set $P = H_{2n-2}$ of size $4n - 6$ such that every n -vertex planar graph admits a 1-bend drawing with bends and vertices on P .*

Proof. The dual of a plane triangulation is a bridgeless 3-regular graph of $2n - 4$ vertices; it has a perfect matching by Petersen's Theorem [19]. Hence, subdividing at most $n - 2$ edges can make any planar graph on n vertices bipartite. Thus H_{n+n-2} of size $2(n+n-2) - 2 = 4n - 6$ is sufficient to accommodate 1-bend drawings of all n -vertex planar graphs. ◀

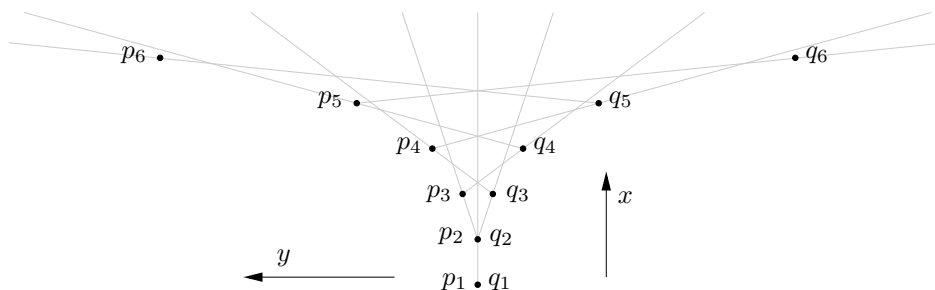
Universality for 1-bend and 2-bend drawings with no restriction on the placement of bends has been studied by Kaufmann and Wiese [14], they show that every n -element point set is universal for 2-bend drawings of planar graphs.

2 The point set and the class of POSH graphs

In this section we define the exploding double chain H and the class \mathcal{C}' of POSH graphs and show that for every $n \geq 2$ the initial part H_n of size $2n - 2$ of H is n -universal for \mathcal{C}' .

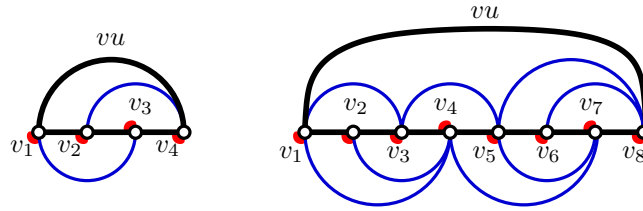
A sequence $(y_i)_{i \in \mathbb{N}}$ of real numbers satisfying $y_1 = 0$, $y_2 = 0$ is *exploding* and the corresponding point set $H = \{p_i, q_i | i \in \mathbb{N}\}$, where $p_i = (i, y_i)$, $q_i = (i, -y_i)$, is an *exploding double chain*, if for all $n \in \mathbb{N}$, y_{n+1} is large enough that all intersections of lines going through two points of $H_n = \{p_i, q_i | i \in [n]\}$ with the line $x = n + 1$ lie strictly between y_{n+1} and $-y_{n+1}$. It is $p_1 = q_1$ and $p_2 = q_2$, thus $|H_n| = 2n - 2$. Figure 1 shows H_6 . This fully describes the order type of the exploding double chain. Note that the coordinates given here can be made integers, but the largest coordinate of H_n is exponential in n , which is unavoidable for the order type. However, the ratio of largest to smallest distance does not have to be: We can alter the construction setting $y_i = i$, but letting the x -coordinates grow slowly enough as to achieve the same order type, but with a linear ratio.

An explicit construction of a point set H in this order type is given in the full version.



■ **Figure 1** An example of a point set H_6 in a rotated coordinate system.

A plane graph G has a *one-sided Hamiltonian cycle with special edge vu* if it has a Hamiltonian cycle $(v = v_1, v_2, \dots, v_n = u)$ such that vu is incident to the outer face and for every $j = 2, \dots, n$, the two edges incident to v_j in the Hamiltonian cycle, i.e., edges $v_{j-1}v_j$ and $v_{j+1}v_j$, are consecutive in the rotation of v_j in the subgraph induced by v_1, \dots, v_j, v_{j+1} in G . In particular, the one-sided condition depends on the Hamiltonian cycle, its direction and its special edge. A more visual reformulation of the second condition is obtained using the closed bounded region D whose boundary is the Hamiltonian cycle. It is that in the embedding of G for every j either all the back-edges $v_i v_j$ with $i < j$ are drawn inside D or in the open exterior of D . We let V_I be the set of vertices v_j which have a back-edge $v_i v_j$ with $i < j - 1$ drawn inside D and $V_O = V \setminus V_I$. The set V_I is the set of vertices having back-edges only inside D while vertices in V_O have back-edges only outside D .



■ **Figure 2** K_4 and a slightly larger graph both with a one-sided Hamiltonian cycle. Red angles indicate a side with no back-edge.

Recall that \mathcal{C}' is the class of planar graphs which are *spanning* subgraphs of plane graphs admitting a one-sided Hamiltonian cycle. It is worth noting all subgraphs are POSH.

► **Proposition 2.** *Any subgraph of a POSH graph is POSH.*

Proof. As edge deletions preserve the POSH property by definition, it suffices to show that deleting a vertex preserves it as well. Let G be a POSH graph and let G' be its supergraph with a one-sided Hamiltonian cycle. Now after deleting v from G' , adding an edge between its neighbours on the Hamiltonian cycle (if it does not exist) can be done along the two edges of v along the cycle. This is a supergraph of $G \setminus v$ with a one-sided Hamiltonian cycle. ◀

3 The embedding strategy

Our interest in POSH graphs is motivated by the following theorem.

► **Theorem 3.** *Let G' be POSH and let v_1, \dots, v_n be a one-sided Hamiltonian cycle of a plane supergraph G of G' on the same vertex set. Then there is a crossing-free embedding of G' on H_n with the property that v_i is placed on either p_i or q_i .*

Proof. It is sufficient to describe the embedding of the supergraph G on H_n . For the proof we assume that in the plane drawing of G the sequence v_1, \dots, v_n traverses the boundary of D in counter-clockwise direction. For each i vertex v_i is embedded at $\bar{v}_i = p_i$ if $v_i \in V_I$ and at $\bar{v}_i = q_i$ if $v_i \in V_O$.

Let $G_i = G[v_1, \dots, v_i]$ be the subgraph of G induced by $\{v_1, \dots, v_i\}$. The path $\Lambda_i = v_1, \dots, v_i$ separates G_i . The *left part* GL_i consists of the intersection of G_i with D , the *right part* GR_i is G_i minus all edges which are interior to D . The intersection of GL_i and GR_i is Λ_i and their union is G_i . The counter-clockwise boundary walk of G_i consists of a path ∂R_i from v_1 to v_i which is contained in GR_i and a path from v_i to v_1 which is contained in GL_i , let ∂L_i be the reverse of this path.

Let \bar{G}_i be the straight-line drawing of the plane graph G_i obtained by placing each vertex v_j at the corresponding \bar{v}_j . A vertex \bar{v} of \bar{G}_i is said to *see a point* p if there is no crossing between the segment $\bar{v}p$ and an edge of \bar{G}_i . By induction on i we show:

1. The drawing \bar{G}_i is plane, i.e., non-crossing.
2. \bar{G}_i and G_i have the same outer boundary walks.
3. Every vertex of ∂L_i in \bar{G}_i sees all the points p_j with $j > i$ and every vertex of ∂R_i in \bar{G}_i sees all the points q_j with $j > i$.

For $i = 2$ the graph G_i is just an edge and the three claims are immediate, for Property 3 just recall that the line spanned by p_1 and p_2 separates the p -side and the q -side of H_n .

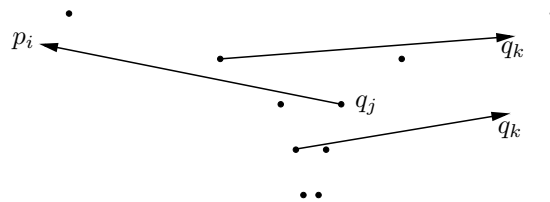
Now assume that $i \in \{3, \dots, n\}$, the properties are true for \bar{G}_{i-1} and suppose that $v_i \in V_I$ (the argument in the case $v_i \in V_O$ works symmetrically). This implies that all the back-edges

of v_i are in the interior of D whence all the neighbors of v_i belong to ∂L_{i-1} . Since $v_i \in V_I$ we have $\bar{v}_i = p_i$ and Property 3 of \bar{G}_{i-1} implies that the edges connecting to \bar{v}_i can be added to \bar{G}_{i-1} without introducing a crossing. This is Property 1 of \bar{G}_i .

Since G_{i-1} and \bar{G}_{i-1} have the same boundary walks and v_i (respectively \bar{v}_i) belong to the outer faces of G_i (respectively \bar{G}_i) and since v_i has the same incident edges in G_i as \bar{v}_i in \bar{G}_i , the outer walks of G_i and \bar{G}_i again equal each other, i.e., Property 2.

Let j be minimal such that $v_j v_i$ is an edge and note that ∂L_i is obtained by taking the prefix of ∂L_{i-1} whose last vertex is v_j and append v_i . The line spanned by \bar{v}_j and $\bar{v}_i = p_i$ separates all the edges incident to \bar{v}_i in \bar{G}_i from all the segments $\bar{v}_\ell p_k$ with $\ell < j$ and $\bar{v}_\ell \in \partial L_i$ and $k > i$. This shows that every vertex of ∂L_i in \bar{G}_i sees all the points p_k with $k > i$. For the proof of the second part of Property 3 assume some edge $\bar{v}_i \bar{v}_j$ crosses the line of sight from \bar{v}_l to $q_k, k > i$, we refer to Figure 3. First note that this is only possible if $l \leq j$, since otherwise $\bar{v}_j \bar{v}_l$ separates $\bar{v}_i = p_i$ and q_k , because p_i is on the left as can be seen at $x = i$ and q_k is on the right as can be seen at $x = k$ by definition. Since $j = l$ is impossible by construction, we are left with the case $l < j$. Then one of \bar{v}_i and \bar{v}_l , say \bar{v} , lies to the right of the oriented line $\bar{v}_j q_k$. However that implies that $\bar{v}_j \bar{v}$ has q_k on its left, which is a contradiction to the definition of q_k at $x = k$. This completes the proof of Property 3 and thus the inductive step.

Finally, Property 1 for \bar{G}_n implies the theorem. ◀



■ **Figure 3** Vertices from ∂R_i see q_k

4 Plane bipartite graphs

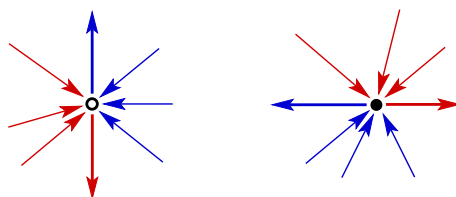
In this section we consider bipartite plane graphs and show that they are POSH.

► **Theorem 4.** *Every bipartite plane graph $G = (V, E)$ is a subgraph of a plane graph G' on the same vertex set V which has a one-sided Hamiltonian cycle, i.e., G is POSH.*

Proof. Quadrangulations are the plane graphs with all faces of degree four. Equivalently they are the maximal plane bipartite graphs, i.e., any bipartite plane graph except stars is a subgraph of a quadrangulation. Thus since POSH graphs are closed under taking subgraphs, it suffices to prove the theorem for quadrangulations.

Let Q be a quadrangulation and let V_B and V_W be the *black* and *white* vertices of a 2-coloring. Label the two black vertices of the outer face as s and t . Henceforth, when talking about a quadrangulation we think of an embedded quadrangulation endowed with s and t . A *separating decomposition* is a pair $D = (Q, Y)$ where Q is a quadrangulation and Y is an orientation and coloring of the edges of Q with colors red and blue such that:

1. The edges incident to s and t are incoming in color red and blue, respectively.
2. Every vertex $v \notin \{s, t\}$ is incident to a non-empty interval of red edges and a non-empty interval of blue edges. If v is white, then, in clockwise order, the first edge in the interval of a color is outgoing and all the other edges of the interval are incoming. If v is black, the outgoing edge is the clockwise last in its color (see Figure 4).

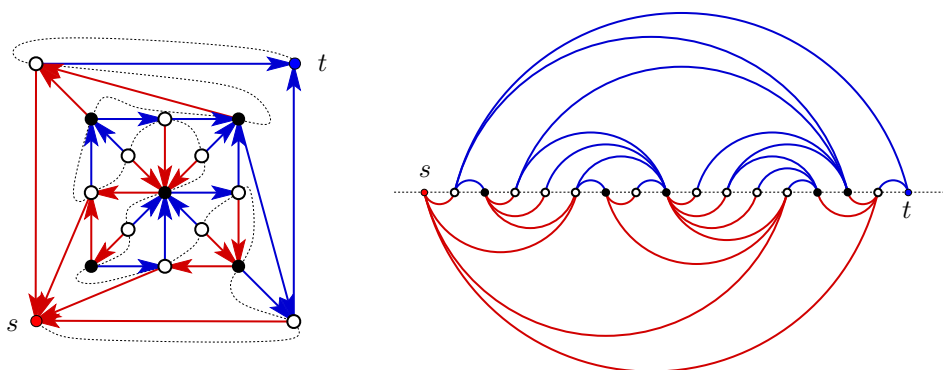


■ **Figure 4** Edge orientations and colors at white and black vertices.[10]

Separating decompositions of a quadrangulation Q have been defined by de Fraysseix and Ossona de Mendez [18]. They show a bijection between separating decompositions and 2-orientations (orientations of the edges of Q such that every vertex $v \notin \{s, t\}$ has out-degree 2) and show the existence of a 2-orientation of Q with an argument related to flows and matchings. An inductive proof for the existence of separating decompositions was given by Felsner et al. [11], this proof is based on identifying pairs of opposite vertices on faces.

In a separating decomposition the red edges form a tree directed towards s , and the blue edges form a tree directed towards t . Each of the trees connects all the vertices $v \notin \{s, t\}$ to the respective root. Felsner et al. ([10, 11]) show that the edges of the two trees can be separated by a curve which starts in s , ends in t , and traverses every vertex and every inner face of Q . This curve is called the *equatorial line*.

If Q is redrawn such that the equatorial line is mapped to the x -axis with s being the left end and t being the right end of the line, then the red tree and the blue tree become *alternating trees* ([11], defined below) drawn in the upper respectively lower half-plane defined by the x -axis. Note that such a drawing of Q is a 2-page book embedding, we call it an *alternating 2-page book embedding* to emphasize that the graphs drawn on the two pages of the book are alternating trees.



■ **Figure 5** A quadrangulation Q with a separating decomposition S , and the alternating 2-page book embedding induced by the equatorial line of S [10].

An *alternating tree* is a plane tree T with a plane drawing such that the vertices of T are placed at different points of the x -axis and all edges are embedded in the half-plane above the x -axis (or all below). Moreover, for every vertex v it holds that all its neighbors are on one side, either they are all left of v or all right of v . In these cases we call the vertex v respectively a *right* or a *left vertex* of the alternating layout. Note that every vertex is a left vertex in one of the two trees and a right vertex in the other.

Let Q be a plane quadrangulation on n vertices and let S be a separating decomposition of Q . Let $s = v_1, v_2, \dots, v_n = t$ be the spine of the alternating 2-page book embedding of Q based on S . Let Q^+ be obtained from Q by adding $v_n v_1$ and all the edges $v_i v_{i+1}$ which do

not yet belong to the edge set of Q . By construction v_1, v_2, \dots, v_n is a Hamiltonian cycle of Q^+ and since the trees are alternating, black vertices have only blue edges to the left and white vertices have only red edges to the left. Thus this Hamiltonian cycle is one-sided with reverse edge $v_n v_1 = ts$. Hence Q is POSH. ◀

It is worth noting that the Hamiltonian cycle read in the reverse direction, i.e., as v_n, v_{n-1}, \dots, v_1 , is again one-sided, now the reverse edge is $v_1 v_n = st$.

5 Planar subcubic graphs

In this section we identify another large subclass of the \mathcal{C}' . Recall that 3-regular graphs are also known as cubic graphs and in subcubic graphs all vertices have degree at most 3.

► **Theorem 5.** *Every planar subcubic graph G is a spanning subgraph of a planar graph G' which has an embedding with a one-sided Hamiltonian cycle, i.e., G has a POSH embedding.*

► **Remark 6.** Note that we do *not* claim the theorem for all *plane* subcubic graphs. However, we are not aware of any connected subcubic plane graph, which is not POSH.

To prove this, we use Theorem 4 and the following lemmas:

► **Lemma 7.** *Let G be a subcubic graph. Then G admits a matching M such that contracting all the edges of M results in a bipartite multi-graph.*

Proof. Let (X, Y) be a partition the vertex-set of G such that the size of the cut, i.e., the number of edges in G with one endpoint in X and one endpoint in Y , is maximized. We claim that the induced subgraphs $G[X]$ and $G[Y]$ of G are matchings. Suppose that a vertex $v \in X$ has at least two neighbors in $G[X]$. Then v has at most one neighbor in Y , and hence moving v from X to Y increases the size of the cut by at least one, a contradiction. The same argument works for $G[Y]$.

Let M be the matching in G consisting of all the edges in $G[X]$ and $G[Y]$. Contracting the edges in M transforms $G[X]$ and $G[Y]$ into independent sets, and hence results in a bipartite multi-graph G/M . ◀

A *separating k -cycle* of a plane graph D is a simple cycle of length k , i.e., k edges, such that there are vertices of D inside the cycle.

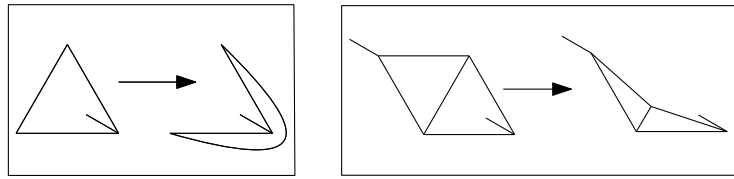
► **Lemma 8.** *Let G be a subcubic planar graph. Then G admits a plane embedding D_G and a matching M such that contracting all the edges of M in D_G results in a bipartite multi-graph without separating 2-cycles.*

Proof. Let G be a subcubic planar graph. Without loss of generality G is connected, otherwise we just deal with the components first, then embed G in a way that all components are incident to the outer face.

Note that a 2-cycle can only arise by contracting one matching edge of a triangle or two matching edges of a quadrilateral. Consider an embedding D of G which minimizes the number of separating 3-cycles and among those minimizes the number of separating 4-cycles.

▷ **Claim 9.** D has no separating 3-cycle.

Proof. For illustration, see Figure 6. We will first show D has no *separating diamond*, that is, two triangles sharing an edge $e = uv$, at least one of which is a separating 3-cycle. Otherwise place u very closely to v . Now e is short and we reroute the other two edges of u such that



■ **Figure 6** Procedure to eliminate triangles with an inner vertex. The procedure on the left eliminates isolated separating triangles, while the one on the right deals with separating diamonds.

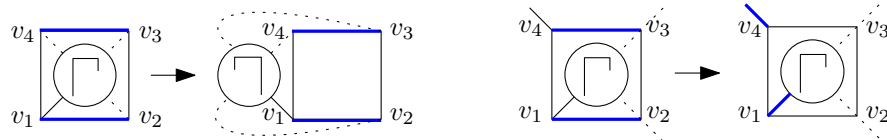
they stay close to the corresponding edge of v . Since one of the triangles containing e was assumed to be separating the new drawing has fewer separating 3-cycles, a contradiction.

We are ready to show D has no separating 3-cycle. If T is a separating 3-cycle some edge has to go from a vertex v of T into its interior. Since v has degree at most 3 it has no edge to the outside of T . We can then redraw the edge e of T not incident to v outside of T closely to its two other edges. Again the new drawing has fewer separating 3-cycles: indeed, if the redrawn edge would be part of another 3-cycle, T is part of a separating diamond. \triangleleft

Now choose an edge set M of minimum cardinality, such that contracting it yields a bipartite multi-graph. The proof of Lemma 7 implies that M is a matching. Among those matchings, we choose M such that the number of separating 4-cycles which have 2 edges in M is minimized. Such separating 4-cycles are said to be *covered* by M .

▷ **Claim 10.** M covers no separating 4-cycle.

Proof. Suppose $Q = v_1v_2v_3v_4$ is a separating 4-cycle such that v_1v_2 and $v_3v_4 \in M$ and v_1 has an edge e_I to the inside, thus no edge to the outside.



■ **Figure 7** Procedure to eliminate quadrilaterals with an inner vertex. The redrawing (left) cannot be applied in the right case, where we are changing the blue matching to avoid a separating 2-cycle.

If v_4 has no edge to the outside either, we change D to a drawing D' by redrawing the part Γ of D inside Q outside of it reflected across v_1v_4 , see Figure 7. In D' the original separating 4-cycle is no longer separating. We claim that no new separating 3-cycle or 4-cycle that is covered by M was created. The claim contradicts the choice of D or M .

To prove the claim note that $S = \{v_2, v_3\}$ is a 2-separator, unless Q is the outer face of D , so let's assume first that it is not. Thus a separating 3- or 4-cycle has to live on one side of S , since the shortest path between them in $Q \cup \Gamma$ except their edge is of length 3 except if both v_2 and v_3 are adjacent to the same vertex of Γ , in which case Q is the outer face, a contradiction. Let X be the component of $G \setminus S$ containing Γ . Then the number of vertices inside 3- or 4-cycles that are not part of X is unchanged in D' , since the face X is located in is still the same. The only 3- or 4-cycles in $X \cup S$ that were not reflected in their entirety are the ones containing the edge v_2v_3 . Since Q is assumed not to be the outer face, at least one of v_2 and v_3 is not connected to Γ . Thus such a cycle C is a 4-cycle consisting of v_2, v_3 , one of v_1 or v_4 as well as a common neighbour of v_2 and v_4 or v_1 and v_3 in Γ . However v_1v_2 or v_3v_4 respectively would be the only edge in $M \cap C$. This is a contradiction to the fact that contracting M yields a bipartite graph.

Now if Q is the outer face of D , it is still true that the only cycles not reflected in their entirety contain v_2v_3 . However v_2 and v_3 could both be adjacent to a vertex in Γ , either a common neighbour for a 3-cycle or two adjacent neighbours for a 4-cycle. Since v_2 and v_3 are already covered by M , this 3-cycle would contain no edge in M , whereas the 4-cycle would contain at most one. Therefore both of these contradict the definition of M .

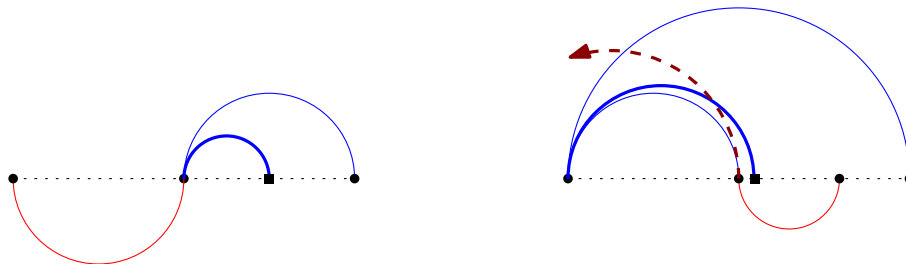
Therefore, we know that v_4 has an edge e_O to the outside. This edge does not go to any vertex of the quadrilateral, because the only candidate left would be v_2 , but this would yield that one of the triangles $v_2v_3v_4$ and $v_1v_2v_4$ is separating.

Change the matching M to an edge set M' by removing v_1v_2 and v_3v_4 from it and adding e_O and e_I . Contracting M' still results in a bipartite graph, because the same four facial cycles that contained our previous edges contain exactly one new edge each as well, so their size after contraction does not change. Thus M' is a matching, because it has the same cardinality as M and is therefore minimal as well. We conclude M' does not cover v_2 or v_3 , because M did not contain any other edge than v_1v_2 and v_3v_4 at them either. Since M' does not contain two edges from quadrilateral v_1, \dots, v_4 but M is minimal, there has to be a separating quadrilateral, of which M' contains two edges, but M doesn't. If such a separating quadrilateral Q contains e_I , then it has to contain another edge incident to v_1 . It cannot contain v_1v_2 , because we know v_2 is not covered by M' . Therefore it contains v_1v_4 and consequently e_O . The same argumentation works to show that if it contains e_O , then it also contains e_I . This is a contradiction to the existence of M' because the endpoints of e_O and e_I are on the outside and the inside of the quadrilateral respectively and therefore non-adjacent. ◁

So we proved that our choice of M makes sure that no separating 2-cycles will be present in the contracted plane bipartite multi-graph. ◀

► **Remark 11.** The embedding D and the matching M can be constructed starting from an arbitrary embedding and matching by iterative application of the operations used in the proof.

Proof of Theorem 5. Now let B be the plane bipartite multi-graph obtained from G by contracting the edges in M without changing the embedding any further. Let B' be the underlying simple graph of B and let Q be a quadrangulation or a star which has B' as a spanning subgraph. The proof of Theorem 4 shows that there is a left to right placement v_1, \dots, v_s of the vertices of Q on the x -axis such that for each $i \in [s]$ all the edges v_jv_i with $j < i - 1$ are in one half-plane and all edges v_iv_j with $j > i + 1$ are in the other half-plane. Delete all the edges from Q which do not belong to B' , and duplicate the multi-edges of B in the drawing. This yields a 2-page book embedding Γ of B .



■ **Figure 8** How to add leaves: The leaf is plotted as a square, its new adjacent edge fat.

Let v be a contracted vertex of B . Vertex v was obtained by contracting an edge $uw \in M$. If u and/or w did not have degree 3, we add edges at the appropriate places into the embedding that end in leaves, see Figure 8. To add an edge to u for instance, choose a face f incident to u that is not contracted into a 2-cycle. Let e and e' be the two edges incident to both v and f . If the angle between e and e' contains part of the spine (the x -axis), we put the leaf on the spine close to v connected to v with a short edge below or above the spine, in a way to accommodate the local vertex condition of v . If it doesn't, assume without loss of generality it is in the upper half-plane and that edge e is the edge closer to the spine. This edge is unique because both edges at v delimiting f go upwards and therefore both to the same side, say right of v . Route the new edge closely along e then put the leaf just next to the other endpoint x of e . Edges that would cross this new edge cannot cross e , thus the only possibility are edges incident to x that emanate into the upper halfspace. However those edges have to go to the left of x by its local vertex condition. These edges do not exist, as any such edge would have to cross e' , see the dashed line in Figure Figure 8. Thus the new edge is uncrossed. This procedure will be done to every vertex first. Note that the resulting graph stays bipartite and the local vertex conditions are still fulfilled, but now every contracted vertex has degree 4. This makes the case distinction of splitting the vertices easier.

We now show how to undo the contractions, i.e., *split* vertices, in the drawing Γ in such a way that at the end we arrive at a one-sided 2-page book drawing Γ^* of G , that is, a 2-book embedding of G with vertex-sequence v_1, \dots, v_n such that for every $j \in \{1, \dots, n\}$ the incident back-edges $v_i v_j$ with $1 \leq i < j$ are all drawn either on the spine or on the same page of the book embedding (all above or all below the spine). Once we have obtained such a book embedding, we can delete the artificial added leaves, then add the spine edges (including the back edge from the rightmost to the leftmost vertex) to G to obtain a supergraph G^+ of G which has a one-sided Hamiltonian cycle, showing that G is POSH.

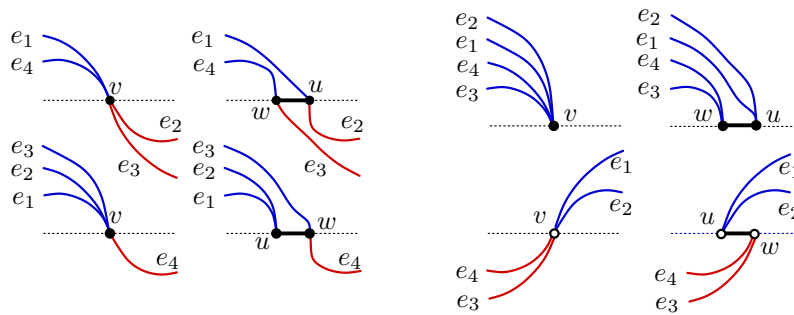
Before we advance to show how we split a single vertex v of degree four into an edge $uw \in M$, we first want to give an overview of the order in which the different splits, the *far splits* and *local splits* are applied. We will then describe what these different splits actually mean. To split all the degree four vertices we proceed as follows:

First we split all vertices which are subject to a far split, from the outside inwards. More precisely, define a partially ordered set on the edges incident¹ to vertices subject to a far split in the following way: Every edge e defines a region R_e which is enclosed by e and the spine. Now order the edges by the containment order of regions R_e . From this poset, choose a maximum edge and then a vertex that needs a far split incident to that edge. When no further far split is possible we do all the local splits. These splits are purely local, so they cannot conflict with each other. Therefore their order can be chosen arbitrarily.

We label the edges of v in clockwise order as e_1, e_2, e_3, e_4 such that in G the edges e_1, e_2 are incident to u and e_3, e_4 are incident to w . If the two angles $\angle e_2 e_3$ and $\angle e_4 e_1$ together take part of both half-planes defined by the spine, then it is possible to select two points left and right of the point representing v in Γ and to slightly detour the edges e_i such that no crossings are introduced and one of the two points is incident to e_1, e_2 and the other to e_3, e_4 . The addition of an edge connecting the two points completes the split of v into the edge $uw \in M$. Figure 9 shows a few instances of this *local* split.

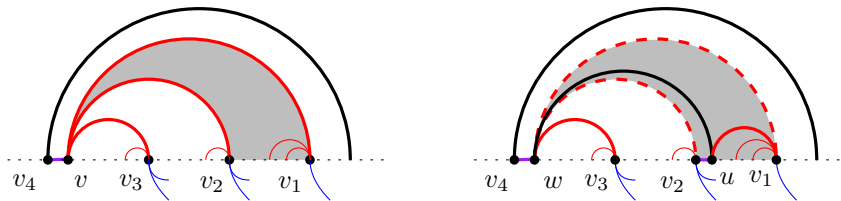
The above condition about the two angles is not fulfilled if and only if all four edges of v emanate into the same halfspace, say the upper one, and the clockwise numbering starting at the x -axis is either e_4, e_1, e_2, e_3 or e_2, e_3, e_4, e_1 . The two cases are the same up to exchanging

¹ There will be a clarification later as to what this means exactly.



■ **Figure 9** Four cases for the local split of a vertex v .

the names of u and w , therefore we can assume the first one. A more important distinction is whether most e_i end to the left or right of v . Note that in the ordering given by Γ , all e_i go to the same side, since they are all in the same halfplane. However, if v is not the first vertex we are splitting, it may happen, that a single edge on the spine to the other side exists, see Figure 10. For all $i \in [4]$ let v_i be the other endpoint of e_i than v . While it can happen that some of the v_i coincide due to multi-edges, we will first discuss the case that they don't. In the left case we put u slightly left of v_1 while in the right case u is put slightly right of v_2 , connecting u to this close vertex by a spine edge. In both cases we leave w at the former position of v . Figure 10 shows the right case and Figure 11 the left.



■ **Figure 10** Far split with v_i to the right except for the spine edge neighbor.

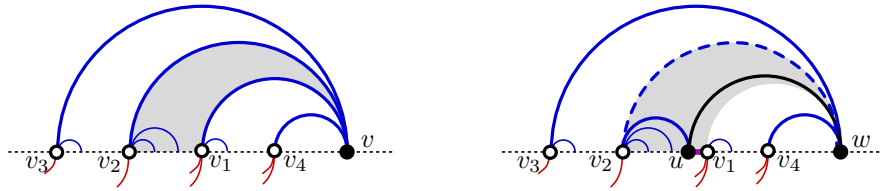
To see that in the left case edges uv_2 and uw are completely free of crossings, observe that we can route them close to the path v_2vv_1 and the edge v_1v respectively in the original drawing (dashed in Figure 11). It is important to note here, that due to the order in which we chose to do the splits, v_1 and v_2 are still original vertices of B , that is, they have not been split in the upper half-plane and thus still don't have two edges emanating to the upper half-plane to both sides. Therefore, similarly to the argumentation for adding leaves, no edge incident to v_1 crosses uw or uv_2 . The right case is analogous, just exchange the roles of v_1 and v_2 .

This kind of split is a *far* split. For the purposes of incidence in the poset structure mentioned above, vertices are not only considered incident to any edge they are an endpoint of, but the spine neighbour of u (v_1 or v_2) is also considered to be incident to the edge uw . For illustration, consider the outermost black edge in Figure 10 (left), it is considered incident to v .

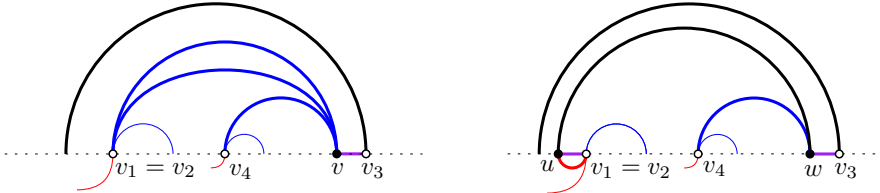
In the following we describe how the different kinds of splits are affected by the presence of multi-edges. The first thing to note is that local splits can be done in the same way, since we did not mention the end vertices at all.

Concerning the far splits, firstly we talk about the case that exactly two edges go from one vertex to another: As depicted in Figures 10 and 11 the case $v_2 = v_3$ and/or $v_4 = v_1$ is unproblematic, in this case we keep the dashed line(s) in the drawing. Double-edges are

31:12 Linear Size Universal Point Sets for Classes of Planar Graphs

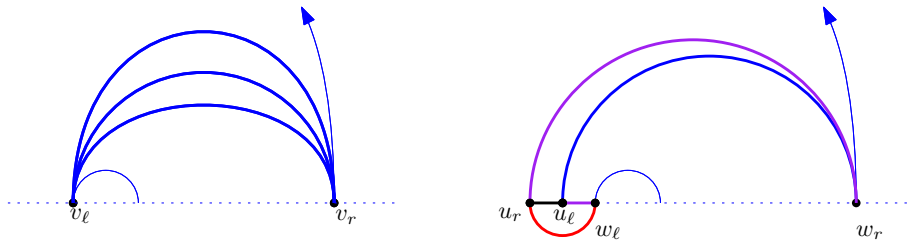


■ **Figure 11** Far split within the gray region with v_i to the left in the upper half-plane.



■ **Figure 12** If $v_1 = v_2$, a double spine edge is created. Here $e_3 = vv_3$ is a spine edge.

consecutive because non-consecutive double-edges are separating 2-cycles, which we avoided in the construction. Thus the last case of a double-edge to consider is $v_1 = v_2$. In this case, we follow the same strategy of placement of u and w , but this results in a double-edge on the spine between u and $v_1 = v_2$, see Figure 12. As in later local splits, we might be interested what half-space the angle between the two spine edges is part of, we interpret one of these edges as a spine edge and the other as an edge which is above or below the spine depending on the right vertex of the two. This might be u or v_1 , depending on whether we are in the left or right case. It is important for the one-sidedness condition to choose this direction so that all left neighbours of the right vertex of the two are reached by edges emanating into the same halfspace and/or spine edges.



■ **Figure 13** Doing a double split means splitting two vertices simultaneously.

Secondly, if there are three edges between a left vertex v_ℓ and a right vertex v_r , say in the upper half-plane, we will split both simultaneously, for illustration, see Figure 13. Since three edges go between these two vertices, there is just one more edge e left for v_ℓ . Therefore we can find a place on the spine just to the right or to the left of v_ℓ which is free, because the edge e is on the other side. Now we split v_ℓ into u_ℓ and w_ℓ and v_r into u_r and w_r simultaneously where w_ℓ and w_r are the vertices with the edge that goes somewhere else on both sides. From left to right we put u_r then u_ℓ just left of the position of v_l , which is the new position of w_ℓ . The three of them are connected by spine edges, just u_r and w_ℓ have an edge in the lower half-plane. These edges are not crossed, because the vertices are close enough together. Finally we put w_r at the position of v_r and add edges to w_r and w_ℓ in the upper half-plane. These edges are not crossed, because any edge crossing them would have crossed the triple edge in the original drawing.

This kind of split is a *double* split. These splits are purely local, so they can be performed together with the local splits in the end.

The last case is that all four edges of a given vertex go to the same vertex, this is a full connected component of the bipartite graph, because it has maximum degree 4. This component goes back to a K_4 component in the cubic graph that had two independent edges contracted. A one-sided Hamiltonian cycle of K_4 is illustrated in Figure 2. We apply another local double split which consists of replacing the 4 parallel edges by this drawing, embedded close to the place of one of the original vertices.

This completes the proof of Theorem 5. ◀

6 2-Trees

From the positive results in Sections 4 and 5 one might expect that “sufficiently sparse” planar graphs are POSH. This section shows that 2-trees are not.

A *2-tree* is a graph which can be obtained, starting from a K_3 , by repeatedly selecting an edge of the current graph and adding a new vertex which is made adjacent to the endpoints of that edge. We refer to this operation as *stacking* a vertex over an edge.

From the recursive construction it follows that a 2-tree on n vertices is a planar graph with $2n - 3$ edges. We also mention that 2-trees are series-parallel planar graphs. Another well studied class which contains 2-trees as a subclass is the class of (planar) Laman graphs.

Fulek and Tóth have shown that planar 3-trees admit n -universal point sets of size $O(n^{3/2} \log n)$. Since every 2-tree is an induced subgraph of a planar 3-tree the bound carries over to this class.

► **Theorem 12.** *There is a 2-tree G on 499 vertices that is not POSH.*

Proof. Throughout the proof we assume that a 2-tree G is given together with a left to right placement v_1, \dots, v_n of the vertices on the x -axis such that adding the spine edges and the reverse edge $v_n v_1$ to G yields a plane graph with a one-sided Hamiltonian cycle.

For an edge e of G we let $X(e)$ be the set of vertices which are stacked over e and $S(e)$ the set of edges which have been created by stacking over e , i.e., each edge in $S(e)$ has one vertex of e and one vertex in $X(e)$. We partition the set $X(e)$ of an edge $e = v_i v_j$ with $i < j$ into a left part $XL(e) = \{v_k \in X(e) : k < i\}$, a middle part $XM(e) = \{v_k \in X(e) : i < k < j\}$, and a right part: $XR(e) = \{v_k \in X(e) : j < k\}$.

▷ **Claim 13.** For every edge $|XR(e)| \leq 2$.

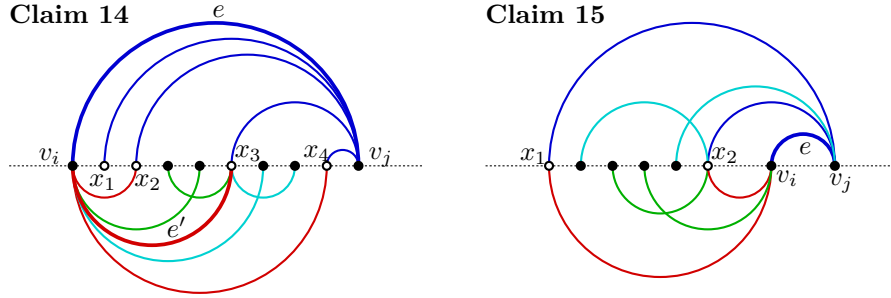
Suppose that $|XR(e)| \geq 3$. Each vertex in this set has all its back-edges on the same side. Two of them use the same side for the back edges to the vertices of e . This implies a crossing pair of edges, a contradiction.

▷ **Claim 14.** If for all $e' \in S(e)$ we have $|X(e')| \geq 3$, then $|XM(e)| \leq 3$.

Suppose that $e = v_i v_j$ with $i < j$ is in the upper half-plane and there are four vertices x_1, x_2, x_3, x_4 in $XM(e)$. One-sidedness implies that the four edges $x_k v_j$ are in the upper half-plane. Thus if x_1, x_2, x_3, x_4 is the left to right order, then the edges $v_i x_2, v_i x_3$, and $v_i x_4$ have to be in the lower half-plane. Now let $e' = v_i x_3$ and consider the three vertices in $X(e')$. Two of them, say y_1, y_2 , are on the same side of x_3 . First suppose $y_1, y_2 \in X(e')$ are left of x_3 . The edges of $v_i x_2$ and $x_2 v_j$ enforce that y_1, y_2 are between x_2 and x_3 . Due to edge $x_2 v_j$ the edges $v_i y_1, v_i y_2$ are in the lower half-plane. One-sidedness at x_3 requires that $y_1 x_3$ and $y_2 x_3$ are also in the lower half-plane. This makes a crossing unavoidable.

31:14 Linear Size Universal Point Sets for Classes of Planar Graphs

Now suppose that $y_1, y_2 \in X(e')$ are right of x_3 . The edges $v_i x_4$ and $x_4 v_j$ enforce that y_1, y_2 are between x_3 and x_4 . Due to the edge $x_3 v_j$ the edges $v_i y_1$ and $v_i y_2$ are in the lower half-plane. Now let y_1 be left of y_2 . One-sidedness at y_2 requires that $x_3 y_2$ is also in the lower half-plane, whence, there is a crossing between $v_i y_1$ and $x_3 y_2$. This completes the proof of the claim.



■ **Figure 14** Illustrating the proofs of the claims.

▷ **Claim 15.** If $XL(e) \geq 2$ and x is the rightmost element of $XL(e)$, then $XL(e') \leq 1$ for some $e' \in S(e)$ incident with x and $XR(e') = \emptyset$ for both.

Suppose that $e = v_i v_j$ with $i < j$ is in the upper half-plane and there are two vertices x_1, x_2 in $XL(e)$. We assume that x_2 is the rightmost element of $XL(e)$. From one-sidedness at v_j we know that $x_1 v_j$ and $x_2 v_j$ are in the upper half-plane. Now $x_1 v_i$ and hence also $x_2 v_i$ are in the lower half-plane. All the vertices of $X(x_2 v_i)$ and $X(x_2 v_j)$ are in the region bounded by $x_1 v_j, v_j v_i, v_i x_1$, in particular $XR(e') = \emptyset$ for both. Suppose for contradiction that we have $y_1, y_2 \in XL(x_2 v_i)$ and $z_1, z_2 \in XL(x_2 v_j)$. By one-sidedness the edges from x_2 to the four vertices y_1, y_2, z_1, z_2 are in the same half-plane. If they are in the lower half-plane and y_1 is left of y_2 there is a crossing between $y_1 x_2$ and $y_2 v_i$. If they are in the upper half-plane and z_1 is left of z_2 there is a crossing between $z_1 x_2$ and $z_2 v_j$. The contradiction shows that $XL(x_2 v_i) \leq 1$ or $XL(x_2 v_j) \leq 1$, since $x = x_2$ this completes the proof of the claim.

We are ready to define the graph G and then use the claims to prove that G is not POSH. The graph G contains a *base edge* e and seven vertices stacked on e , i.e., $|X(e)| = 7$. For each edge $e' \in S(e)$ there are five vertices stacked on e' . Finally, for each edge e'' introduced like that three vertices are stacked on e'' . Note that there are $7 \cdot 2 = 14$ edges e' , $14 \cdot 5 \cdot 2 = 140$ edges e'' and $140 \cdot 3 \cdot 2 = 840$ edges introduced by stacking on an edge e'' . In total the number of edges is $995 = 2n - 3$, hence, the graph has 499 vertices.

Now suppose that G is POSH and let v_1, \dots, v_n be the order of vertices on the spine of a certifying 2-page book embedding. Let $e = v_i v_j$ with $i < j$ be the base edge. Assume by symmetry that e is in the upper half-plane. From Claim 13 we get $|XR(e)| \leq 2$ and from Claim 14 we get $|XM(e)| \leq 3$, it follows that $|XL(e)| \geq 2$. Let x_1 and x_2 be elements of $XL(e)$ such that x_2 is the rightmost element of $XL(e)$. Let $e' = x_2 v_i$ and $e'' = x_2 v_j$ then $XR(e') = \emptyset = XR(e'')$ by Claim 15. From Claim 14 applied to e' and e'' we deduce that $|XM(e')| \leq 3$ and $|XM(e'')| \leq 3$. Hence $|XL(e')| \geq 2$ and $|XL(e'')| \geq 2$. This is in contradiction with Claim 13. Thus there is no spine ordering for G which leads to a one-sided crossing-free 2-page book embedding. ◀

7 Concluding remarks

We have examined the exploding double chain as a special point set (order type) and shown that the initial part H_n of size $2n - 2$ is n -universal for graphs on n vertices that are POSH. We believe that the class of POSH graphs is quite rich. On the sparse side, the result on bipartite graphs might be generalized, while for triangulations, the sheer number of Hamiltonian cycles in 5-connected graphs [1] makes it likely one of them is one-sided.

► **Conjecture 16.** *Every triangle-free planar graph is POSH.*

► **Conjecture 17.** *Every 5-connected planar triangulation is POSH.*

We have shown that 2-trees and their superclasses series-parallel and planar Laman graphs are not contained in the class \mathcal{C}' of POSH graphs. The question whether these classes admit universal point sets of linear size remains intriguing.

References

- 1 A. Alahmadi, R.E.L. Aldred, and C. Thomassen. Cycles in 5-connected triangulations. *Journal of Combinatorial Theory, Series B*, 140:27–44, 2020. doi:10.1016/j.jctb.2019.04.005.
- 2 Md. Jawaherul Alam, Therese C. Biedl, Stefan Felsner, Michael Kaufmann, Stephen G. Kobourov, and Torsten Ueckerdt. Computing cartograms with optimal complexity. *Discret. Comput. Geom.*, 50:784–810, 2013.
- 3 Md. Jawaherul Alam and Stephen G. Kobourov. Proportional contact representations of 4-connected planar graphs. In *Graph Drawing*, volume 7704 of *LNCS*, pages 211–223. Springer, 2012.
- 4 Patrizio Angelini, Till Bruckdorfer, Giuseppe Di Battista, Michael Kaufmann, Tamara Mchedlidze, Vincenzo Roselli, and Claudio Squarcella. Small universal point sets for k -outerplanar graphs. *Discrete & Computational Geometry*, pages 1–41, 2018. doi:10.1007/s00454-018-0009-x.
- 5 Michael J. Bannister, Zhanpeng Cheng, William E. Devanny, and David Eppstein. Superpatterns and Universal Point Sets. *Journal of Graph Algorithms and Applications*, 18(2):177–209, 2014. doi:10.7155/jgaa.00318.
- 6 Jean Cardinal, Michael Hoffmann, and Vincent Kusters. On Universal Point Sets for Planar Graphs. *Journal of Graph Algorithms and Applications*, 19(1):529–547, 2015. doi:10.7155/jgaa.00374.
- 7 Alexander Choi, Marek Chrobak, and Kevin Costello. An $\Omega(n^2)$ lower bound for random universal sets for planar graphs. *arXiv preprint, arXiv1908.07097*, 2019. URL: <https://arxiv.org/abs/1908.07097>.
- 8 Marek Chrobak and Howard J. Karloff. A Lower Bound on the Size of Universal Sets for Planar Graphs. *ACM SIGACT News*, 20(4):83–86, 1989. doi:10.1145/74074.74088.
- 9 Hubert De Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. doi:10.1007/BF02122694.
- 10 Stefan Felsner, Éric Fusy, Marc Noy, and David Orden. Bijections for Baxter families and related objects. *Journal of Combinatorial Theory, Series A*, 118(3):993–1020, 2011. doi:10.1016/j.jcta.2010.03.017.
- 11 Stefan Felsner, Clemens Huemer, Sarah Kappes, and David Orden. Binary labelings for plane quadrangulations and their relatives. *Discrete Mathematics and Theoretical Computer Science*, 12:3:115–138, 2010.
- 12 Radoslav Fulek and Csaba D. Tóth. Universal point sets for planar three-trees. *Journal of Discrete Algorithms*, 30:101–112, 2015. doi:10.1016/j.jda.2014.12.005.


31:16 Linear Size Universal Point Sets for Classes of Planar Graphs

- 13 Peter Gritzmann, Bojan Mohar, János Pach, and Richard Pollack. Embedding a planar triangulation with vertices at specified points. *American Mathematical Monthly*, 98:165–166, 1991. doi:10.2307/2323956.
- 14 Michael Kaufmann and Roland Wiese. Embedding vertices at points: Few bends suffice for planar graphs. *Journal of Graph Algorithms and Applications*, 6(1):115–129, 2002. doi:10.7155/jgaa.00046.
- 15 Maciej Kurowski. A $1.235n$ lower bound on the number of points needed to draw all n -vertex planar graphs. *Information Processing Letters*, 92(2):95–98, 2004. doi:10.1016/j.ipl.2004.06.009.
- 16 Maarten Löffler and Csaba D. Tóth. Linear-size universal point sets for one-bend drawings. In *Graph Drawing*, volume 9411 of *LNCS*, pages 423–429. Springer, 2015. doi:10.1007/978-3-319-27261-0_35.
- 17 Bojan Mohar. *Universal point sets for planar graphs*. Open Problem Garden, 2007. URL: http://www.openproblemgarden.org/op/small_universal_point_sets_for_planar_graphs.
- 18 Patrice Ossona de Mendez and Hubert de Fraysseix. On topological aspects of orientations. *Discrete Mathematics*, 229(1–3):57–72, 2001. doi:doi.org/10.1016/S0012-365X(00)00201-6.
- 19 Julius Petersen. Die Theorie der regulären graphs. *Acta Mathematica*, 15(1):193–221, 1891. doi:10.1007/BF02392606.
- 20 Manfred Scheucher, Hendrik Schrezenmaier, and Raphael Steiner. A note on universal point sets for planar graphs. *Journal of Graph Algorithms and Applications*, 24(3):247–267, 2020. doi:10.7155/jgaa.00529.
- 21 Walter Schnyder. Embedding Planar Graphs on the Grid. In *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138–148. Society for Industrial and Applied Mathematics, 1990.

When Ternary Triangulated Disc Packings Are Densest: Examples, Counter-Examples and Techniques

Thomas Fernique ✉

CNRS & LIPN, Univ. Paris Nord, 93430 Villetaneuse, France

Daria Pchelina ✉ 

LIPN, Univ. Paris Nord, 93430 Villetaneuse, France

Abstract

We consider *ternary* disc packings of the plane, i.e. the packings using discs of three different radii. Packings in which each “hole” is bounded by three pairwise tangent discs are called *triangulated*. Connelly conjectured that when such packings exist, one of them maximizes the proportion of the covered surface: this holds for unary and binary disc packings. For ternary packings, there are 164 pairs (r, s) , $1 > r > s$, allowing triangulated packings by discs of radii 1, r and s . In this paper, we enhance existing methods of dealing with maximal-density packings in order to study ternary triangulated packings. We prove that the conjecture holds for 31 triplets of disc radii and disprove it for 40 other triplets. Finally, we classify the remaining cases where our methods are not applicable. Our approach is based on the ideas present in the Hales’ proof of the Kepler conjecture. Notably, our proof features local density redistribution based on computer search and interval arithmetic.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Disc packing, density, interval arithmetic

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.32

Related Version *Full version:* <https://arxiv.org/abs/2211.02905> [17]

Supplementary Material *Software:* https://github.com/tooticki/ternary_triangulated_disc_packings, archived at `swh:1:dir:9630535719e287d4e42724550cdaf81df94181c5`

1 Introduction

Given a finite set S of discs, a *packing* of the plane by S is a collection of translated copies of discs from S with disjoint interiors.

Given a packing P , its *density* $\delta(P)$ is the proportion of the plane covered by the discs. More formally,

$$\delta(P) := \limsup_{n \rightarrow \infty} \frac{\text{area}([-n, n]^2 \cap P)}{\text{area}([-n, n]^2)}.$$

Nowadays, the density of disc packings is widely studied in different contexts. The worst-case optimal density of packings in triangular and circular containers is found in [11, 12]. In computer science, there are various connections between sphere packings and error-correcting codes [4]. Researchers in chemical physics used Monte Carlo simulations on 2-disc packings and, among others, obtained lower bounds on the maximal density of packings with particular disc sizes [7]. Two other groups of physicists found lower bounds on maximal densities of packings in \mathbb{R}^3 with 2 sizes of spheres [26, 31]. Upper bounds on the density are usually much harder to obtain.

The main problem we are interested in is the following: given a finite set of ball sizes in \mathbb{R}^2 (or \mathbb{R}^3), find a packing of the plane (or of the space) maximizing the density.



© Thomas Fernique and Daria Pchelina;

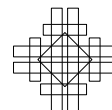
licensed under Creative Commons License CC-BY 4.0

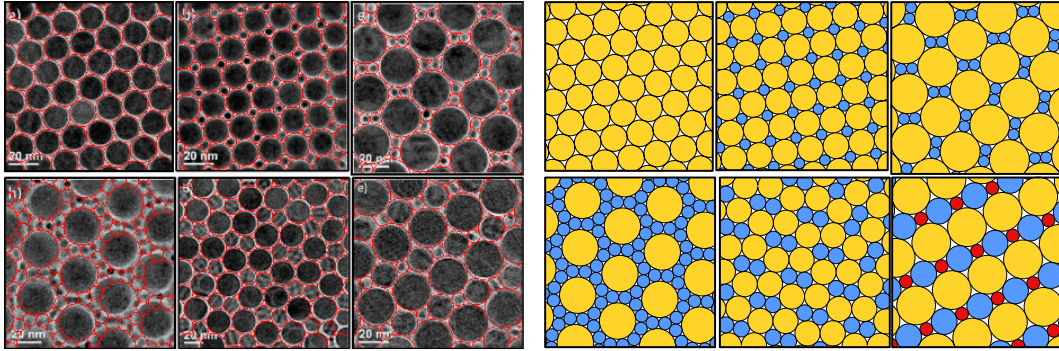
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 32; pp. 32:1–32:17

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Disc packings self-assembled from colloidal nanodiscs and nanorods in [32] (on the left) which very accurately correspond to triangulated packings (on the right).

Answering this question has a few practical applications. Chemists, for example, are interested in the disc and sphere sizes maximizing the density in order to eventually design compact materials using spherical nanoparticles of given sizes [7, 26, 32]. Figure 1 gives an illustration of experimental results from [32].

The first known studies of the densest packings go back to Kepler. Many advances in this area have been made since then.

1.1 1-sphere packings

In a Kepler manuscript dated by 1611, we find a description of the “cannonball” packing followed by an assertion that it is a densest *1-sphere packing* (i.e. packing by equally sized spheres) of the three-dimensional Euclidean space. This assertion is widely known by name of the Kepler conjecture. The “cannonball” packing, also called face-centered-cubic (FCC) packing, belongs to a family of packings formed by stacking layers of spheres centered in the vertices of a triangular lattice, like it is shown in Figure 2. After placing the first two layers, at each step, there are two choices of how to place the next layer. This gives us an uncountable set of packings having the same density. These packings are called *close-packings*.

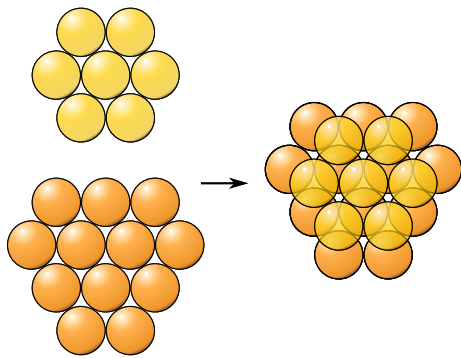
► **Conjecture 1** (Kepler 1611). *The density $\delta(P)$ of packing P of \mathbb{R}^3 by unit spheres never exceeds the density of a close-packing:*

$$\delta(P) \leq \frac{\pi}{3\sqrt{2}}. \quad (1)$$

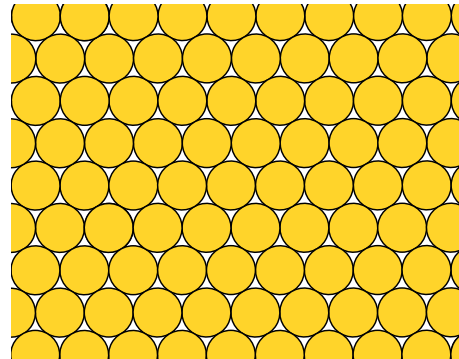
The first advancement in a proof of the Kepler conjecture was made by Gauss who, in 1831, showed that close sphere packings maximize the density among all possible *lattice* packings, i.e. those where the disc centers form a lattice [19]. However, the proof of the whole conjecture took four centuries to be found. Hilbert included this conjecture, also named “the sphere packing problem”, in his famous list of 23 problems published in 1900.

The Kepler conjecture was finally proved in a series of 6 papers submitted by Hales and Ferguson in 1998 [20, 22]. Their computer-assisted proof took 8 years to be fully reviewed. In 2003, Hales founded a project called Flyspeck in order to fully verify his proof by an automated theorem prover. Flyspeck was completed in 2014 including the proof of the Kepler conjecture in the list of computer verified proofs [21].

The rough idea of the proof consists of locally redistributing the density function and showing inequality (1) for this redistributed density. Lagarias calls this approach “localization” [29]. In our work, we use the same general ideas discussed in detail in Section 2.



■ **Figure 2** First step of construction of a 3D close-packing.



■ **Figure 3** 2D hexagonal packing.

1.2 Disc packings

The two-dimensional variant of the Kepler conjecture claims the 2D hexagonal packing on the plane (see Fig. 3) to have the highest density among all planar packings by identical discs.

In 1772, Lagrange proved it to be a densest among lattice packings. The general result was first shown by Thue in 1910 [33]. His proof was however considered incomplete, a reliable proof was given by Fejes-Tóth in 1942 [8].

The proof of the two-dimensional Kepler conjecture contains the basics of the strategy used to prove similar results for packings with several disc sizes, like binary packings (discussed below) and ternary packings which are studied in this paper.

Packings of the plane where, as in the hexagonal one, each “hole” is bounded by three pairwise tangent discs are called *triangulated*. More formally,

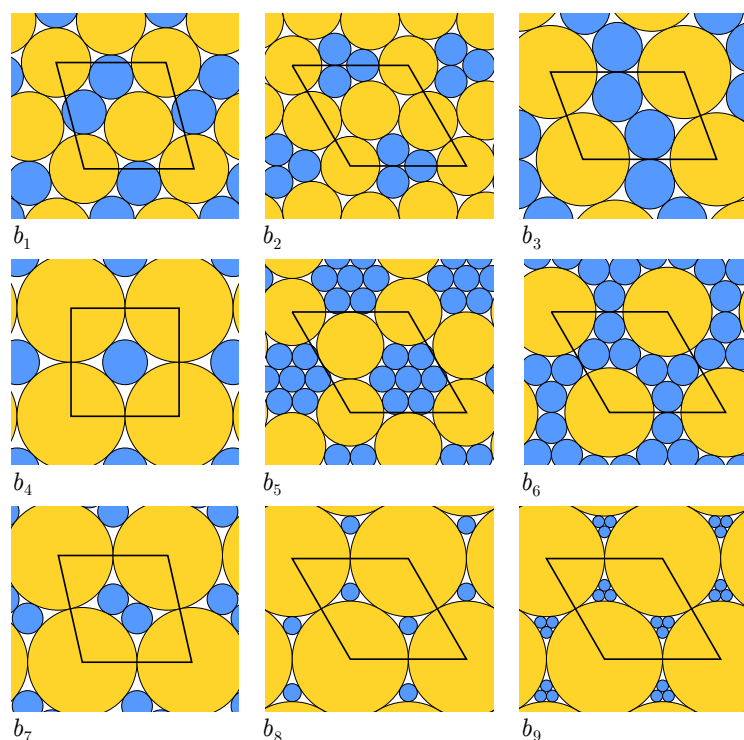
► **Definition 2.** *A packing is called **triangulated** if the graph formed by connecting the centers of every pair of tangent discs is a triangulation.*

Fejes Tóth in [9] called such packings “compact”: since triangulated packings have no “huge holes”, they intuitively look the most compact. Moreover, around each disc, its neighbors form a corona of tangent discs which looks like a locally “optimal” way to pack. For these reasons, triangulated packings appear to be the best candidates to maximize the density on the whole plane.

Notice that, for a fixed n , there exists only a finite number of n -tuples of disc radii (r_1, \dots, r_n) s.t. $1 = r_1 > \dots > r_n > 0$ allowing a triangulated packing where all n disc sizes are present [30].

Let us consider binary packings of the plane. Given two discs of radii 1 and $r < 1$, what is the maximal density of a packing by copies of these discs? We can always obtain $\frac{\pi}{2\sqrt{3}}$, the density of the hexagonal packing, by using only one of the discs which gives as a lower bound on the maximal density. Florian in [18] derived an upper bound on the density which is equal to the density in the triangle formed by 2 small and one big pairwise tangent discs. [14] gives tighter lower and upper bounds of maximal density of binary packings of the plane, for all values of $r \in (0, 1)$.

There are 9 values of r allowing triangulated binary packings where both disc sizes are present [28]. Such packings are shown in Fig. 4. Each of the depicted packings is *periodic*, i.e. if P is a packing in question, there are two non-collinear vectors u and v , called periods, such that $P + u = P + v = P$. Notice that in this paper, we always consider packings of the



■ **Figure 4** 9 triangulated periodic binary packings maximizing the density among packings with the respective disc sizes.

whole plane, and since the triangulated packings we show here and below are all periodic, it is enough to represent their fundamental domain (a parallelogram formed by the period vectors, marked in black in Fig. 4) to see how the whole plane is packed.

Notice that for each of these values of r , there is actually an infinite number of packings having the same density as the one depicted in Fig. 4. First, changing a finite portion of a packings does not affect its density. Moreover, for b_1 , b_3 , and b_7 , there exist non-periodic triangulated packings with a different global structure, having the same density as the ones from Fig. 4 [28]. For the sake of simplicity, we choose to depict the periodic ones.

It turns out that for each of these 9 radii, the density is maximized by a triangulated binary packing – namely, the ones shown in Figure 4 [1, 24, 25, 27]. This result suggests the following conjecture [2].

► **Conjecture 3** (Connelly, 2018). *If a finite set of discs allows a triangulated **saturated** packing, then the density of packings by these discs is maximized on a triangulated packing.*

A packing by a set of discs is called *saturated* if no more discs from this set can be added to the packing without intersecting already placed discs. In our setup, we always assume packings to be saturated since we are interested in the upper bounds on the density.

The Connelly conjecture holds for 1-disc (*unary*) packings and 2-disc (*binary*) packings. To study this conjecture, the next step is to verify it for 3-disc (*ternary*) packings which was the main motivation of our work.

1.3 Our results

Let us turn to the ternary packings. To begin with, we need to find the sizes of discs allowing triangulated ternary packings. This problem was solved in [15]: there are 164 pairs (r, s) featuring triangulated packings with discs of radii $1, r, s$, $1 > r > s$. In this paper, a triplet of disc radii associated to each of such pairs is called a *case*.

The ternary cases are indexed by positive integers from 1 to 164, like in [15]. To avoid confusion, the binary cases (pairs of disc radii allowing binary triangulated packings) are denoted by b_1, \dots, b_9 which respectively correspond to the cases 1–9 in [1].

The Connelly conjecture is applicable only to the cases having triangulated *saturated* packings. This eliminates 15 cases where no triangulated packing is saturated and leaves us with 149 cases.

Our main contribution is a classification of 71 cases formulated in the following theorem:

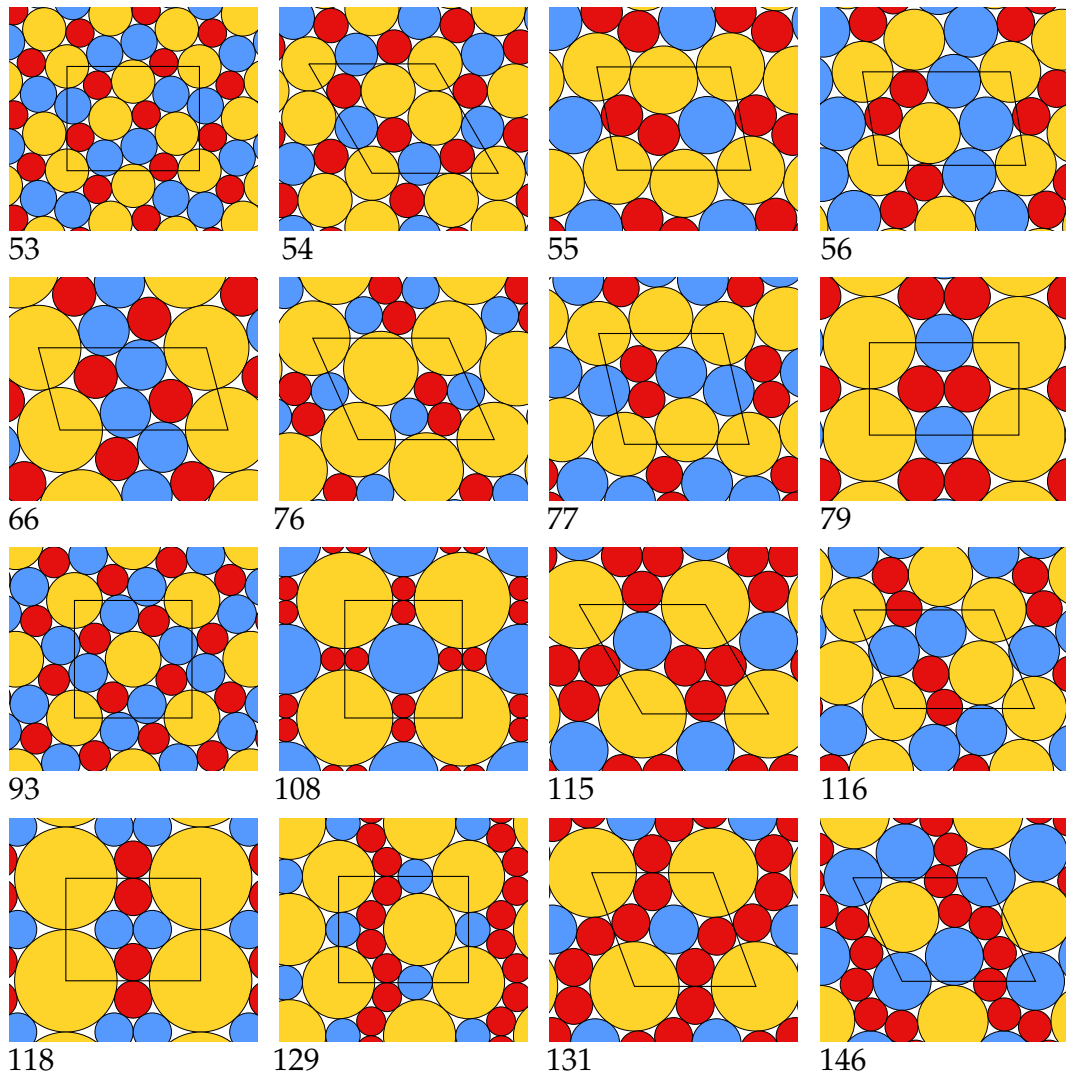
► **Theorem 4.**

- (a) For the 16 following cases: 53, 54, 55, 56, 66, 76, 77, 79, 93, 108, 115, 116, 118, 129, 131, 146, the density is maximized by a triangulated ternary packing.
- (b) For the cases 1–15, the density is maximized by triangulated binary packings. For cases 1–5, it is the triangulated packing of b_8 ; for case 6 – b_4 ; for cases 7–9 – b_7 ; for cases 10–16 – b_9 .
- (c) For the 40 following cases: 19, 20, 25, 47, 51, 60, 63, 64, 70, 73, 80, 92, 95, 97, 98, 99, 100, 104, 110, 111, 117, 119, 126, 132, 133, 135, 136, 137, 138, 139, 141, 142, 151, 152, 154, 159, 161, 162, 163, 164, there exists a non-triangulated packing denser than any triangulated one.

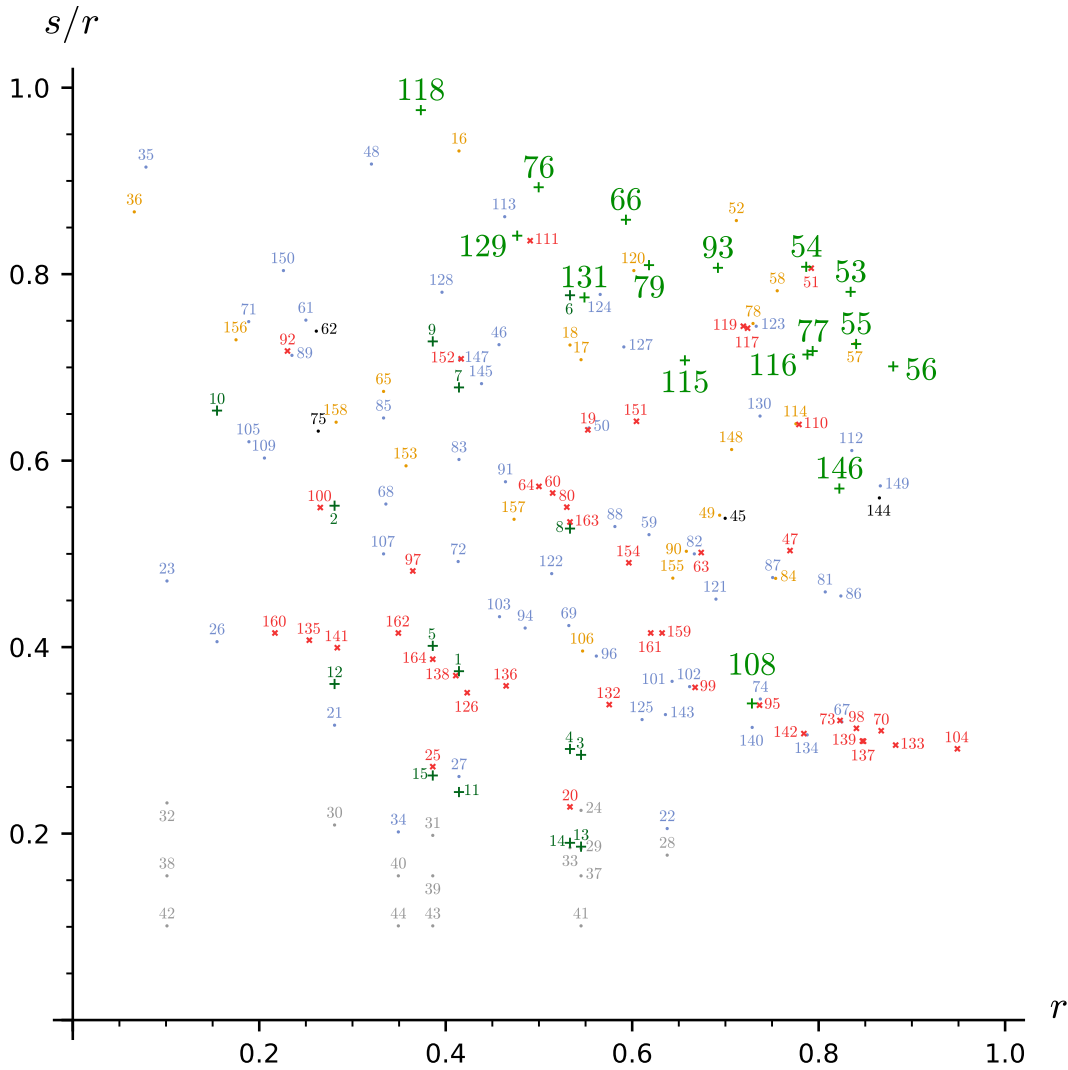
The values of radii corresponding to the cases from Theorem 4 are given in [15]. The triangulated packings maximizing the density for the cases from Th. 4.(a) are depicted in Fig. 5. For Th. 4.(b), the binary triangulated packings which maximize the density are present in Fig. 4 while the ternary triangulated packings are given in Fig. 7. An instance of a triangulated ternary packing and a non-triangulated binary denser packing for Th. 4.(c) are given in Fig. 8 while the complete list can be found in the appendix of the extended version of the paper [17].

All in all, we proved the Connelly conjecture to be false and classified the 149 cases where it was applicable in several groups: 16 cases for which the conjecture holds (Th. 4.(a)), 15 cases where the density is maximized on a triangulated packing using only two discs out of three (Th. 4.(b)), 40 (periodic) counter examples to the initial conjecture (Th. 4.(c)), and the other cases where our proof strategy does not work. Figure 6 represents each case (i.e. a triplet of disc radii $1, r, s$, $1 > r > s$) as a point with coordinates $(r, \frac{s}{r})$ and its number from [15]. The color of the point and the number corresponds to the class we assigned to the case.

Section 2 is dedicated to the cases where a ternary triangulated packing is proved to maximize the density. We explain the approach used in the similar proof for binary packings from [1] and how we enhance it to make it work in our context. The first improvement was the generalization of the code universal to all the cases (instead of treating them one by one as in [1]). The second necessary generalization was leaving a bunch of parameters as free variables instead of fixing them arbitrary. The theoretical background of the proof strategy is given in Section 2.1. Section 2.2 provides the main ideas of the computational part of the proof of Th. 4.(a) (the detailed version of this section is given in Section 3 of the full version [17]). We prove Th. 4.(b) in Section 2.3 by adjusting the proof of Th. 4.(a).



■ **Figure 5** The 16 triangulated ternary packings proved to maximize the density (the numbers correspond to the numbering in [15]).



■ **Figure 6** The “map” of the 164 cases with triangulated ternary packings. Each case (i.e. a triplet of disc radii $1, r, s, 1 > r > s$) corresponds to a point with coordinates $(r, \frac{s}{r})$ and its number from [15]. The cases where no triangulated packing is saturated are marked in grey. The cases with a ternary triangulated packing proved to maximize the density are marked by green + with larger case numbers. The cases where we proved a triangulated binary packing to maximize the density are marked by dark green +. The cases with counterexamples are red (*). The cases featuring two coronas (find the details in Section 5.1) are orange. The cases with empty polyhedra (see Section 5.2) are blue. The remaining cases are marked in black (Section 5.3).

Our proof, as quite a few recent results in the domain, like [11,14,20], is based on computer calculations. The main details of the implementation are provided in Section 3. The complete code is given at the url: https://github.com/tooticki/ternary_triangulated_disc_packings.

Cases from Th. 4.(c) are treated in Section 4. We obtain a counter-example for each of these cases by applying the flip-and-flow method [3] on the triangulated binary packings with disc radii ratio close to the radii ratios of pairs of discs of this case.

Section 5 is dedicated to the remaining cases. Section 5.1 presents the 22 cases where one of the discs appears with at least two different neighborhoods. Our proof technique is not sufficient to treat such cases, handling them requires a less local approach.

Section 5.2 treats the 52 cases where we did not find a set of constants satisfying all required inequalities needed in our proof. Even though after several attempts with higher and higher precision, we concluded that the existence of valid constants is quite unlikely, it cannot be rigorously proved for the moment. We thus leave this as an open problem.

Finally, Section 5.3 is dedicated to the 4 cases where the existence of such set of constants is more probable since we could find the parameters satisfying the majority of constraints, but a few of them were still not satisfied. Whether the density is maximized in these cases is also an open problem.

2 Proof of Th. 4 (a) and (b)

In this section, we give the proof of the first two parts of Theorem 4. We follow almost the same steps of the proof as in [1] where the same result is proven for binary triangulated packings and in [13] which treats computationally the “simplest” case among the ternary triangulated packings (case 53).

From the theoretical point of view, the transition from binary packings to ternary ones seems to be straightforward. In practice, however, we have much more cases to treat (149 instead of 9) and for each of them, the problem is much more complex due to the high number of local combinatorial configurations in possible packings. This requires a more refined and sensitive choice of parameters than in [1].

2.1 Proof strategy

This section is strongly based on [1]: we use the idea of the proof and quite a few intermediate results. Thus, for the sake of simplicity, we preserve the same notations.

Let us describe the theoretical background of the proof which is common for all cases, the only difference being the choice of the parameters described in Section 3.

We are given 3 discs of radii $1, r$ and s , $1 > r > s$ and a ternary triangulated packing of the plane by copies of these discs conjectured to maximize the density, let us denote it by P^* . Our aim is to prove that for any other packing P using the same discs, its density $\delta(P)$ does not exceed the density δ^* of P^* .

The main idea common to all the results about the maximal density of triangulated packings was called “cell balancing” by Heppes [25] and it perfectly matches this title. It consists of two steps: first we locally “redistribute” the density among some well-defined cells (triangles of the triangulation in [1,25,27] and a mixture of Delaunay simplices and Voronoi cells, both encoded in so-called decomposition stars, in [23]) preserving the global density value. Then we prove that the redistributed density of any cell of P never exceeds δ^* .

First, let us define triangulations for packings by several sizes of discs. The *FM-triangulation* of a packing was introduced in [10] (it is a particular case of weighted Delaunay triangulations [5]). Some of its useful properties are given in [1] (Section 4). The vertices of

the FM-triangulation are the disc centers. There is an edge between two disc centers if and only if there is a point $p \in \mathbb{R}^2$ and a distance $d > 0$ such that p is at distance d from the both discs and at least d from any other disc.

Let \mathcal{T} and \mathcal{T}^* respectively denote the FM-triangulations of P and P^* . The cells we are interested in are the triangles of these triangulations. Instead of working with densities, we introduce an additive function E , called *emptiness*, which, for a triangle T in \mathcal{T} , is defined by

$$E(T) := \text{area}(T) \times \delta^* - \text{area}(T \cap P).$$

This function was used in [27] by the name of “excess”. It was inspired by “surplus area” introduced in [25] defined as $\text{area}(T) - \frac{\text{area}(T \cap P)}{\delta^*}$, identical to emptiness up to multiplication by δ^* . A similar but more complex function called “score” is used in the proof of the Kepler conjecture [29].

The emptiness function reflects how “empty” the triangle is compared to δ^* . Indeed, $E(T)$ is positive if the density of T is less than δ^* , negative if T is denser, and equals zero if $\delta(T) = \delta^*$. We use it rather than the density because of its additivity: the emptiness of a union of two triangles equals the sum of their emptiness values. This property does not hold for the density.

To prove that $\delta \leq \delta^*$, it is enough to show that $\sum_{T \in \mathcal{T}} E(T) \geq 0$ [1]. This intuitively means that P is globally more empty and less dense than P^* .

Instead of working directly with the emptiness, we define a so-called potential which plays the role of density redistribution mentioned above. We do it since this function, constructed explicitly, is easier to manipulate. We will construct a potential U such that for any triangle $T \in \mathcal{T}$, its potential does not exceed its emptiness:

$$E(T) \geq U(T) \tag{2}$$

and the sum of potentials of all triangles in \mathcal{T} is non-negative:

$$\sum_{T \in \mathcal{T}} U(T) \geq 0 \tag{3}$$

If, for P^* , there exists U satisfying (2) and (3) for any packing P , then P^* maximizes the density among packings using the same disc radii:

$$(2),(3) \implies \sum_{T \in \mathcal{T}} E(T) \geq 0 \implies \delta^* \geq \delta.$$

The rest of the proof consists in construction of potential U satisfying both (2) and (3) for any packing P .

2.2 Sketch of our proof of Th. 4 (a)

This section provides the short version of our proof, please find the detailed version in [17].

We follow the method of “localizing potentials” introduced by Kennedy in [27]. The idea is to distribute the potential U of each triangle among its vertices in a way that the sum of vertex potentials of the triangles around each vertex of any packing is non-negative. This local constraint of non-negativity implies inequality (3).

We choose the potential function as simple as possible to facilitate further calculations. As in [27] and [1], the potential in a vertex of a triangle depends only on the three disc radii of the triangle and the angle in the vertex.

32:10 When Ternary Triangulated Disc Packings Are Densest

Let us first introduce *tight triangles*: the triangles formed by three tangent discs. For ternary packings, there are always 10 of them. Let V_{xyz} denote the vertex potential of the tight triangle formed by discs of radii x, y, z in the vertex corresponding to the center of the y -disc, we call these constants *tight vertex potentials* and we fix them below.

Given a triangle T with discs of radii x, y, z and with vertex v in the center of the y -disc, we denote by \hat{v} the angle of v in T and by \widehat{xyz} the angle in the center of the y -disc of the tight triangle formed by the discs of radii x, y, z . We define the vertex potential of T in v as

$$\dot{U}_v(T) := V_{xyz} + m_y |\hat{v} - \widehat{xyz}|,$$

where m_y is a constant fixed below.

As in [1], we choose constants V_{xyz} and m_y in a way to satisfy the non-negativity around each vertex $v \in \mathcal{T}$:

$$\sum_{T \in \mathcal{T} | v \in T} \dot{U}_v(T) \geq 0. \quad (\bullet)$$

Besides that, tight vertex potentials should satisfy 5 equations in order to guarantee (2),(3) in P^* (you can find them in Section 3.2.1 of the full version of the paper [17]). In [1], the remaining tight vertex potentials are all set to 0 for the sake of simplicity. This strategy does not work in our case; we thus leave 6 tight vertex potentials $V_{1r1}, V_{1s1}, V_{r1r}, V_{rsr}, V_{s1s}, V_{srs}$ as free variables at this point.

The solutions of inequalities (\bullet) form a polyhedron in \mathbb{R}^9 containing all valid values of the tight vertex potentials and constants m_1, m_r, m_s . You can find more details on the vertex potentials in [17], the implementation details of the polyhedra are discussed in Section 3.2.

Choosing values of V_{xyz} and m_y among all the solutions found above, we seek to satisfy (2): we pick the solution where V_{xyz} and m_y are “small” (find details in [17]).

This is however not enough because of two “limit” cases of triangles. The first are so-called “stretched” triangles: those where one of the angles is very large which causes high vertex potential and low emptiness. However, the triangle sharing the longest edge of a stretched triangle always features low vertex potentials and high emptiness. We thus introduce an *edge* potential \bar{U} aiming to make stretched triangles share their potential with their empty neighbors. You can find the exact formulas of the edge potentials in Section 3.2.2 of the full version of the paper [17].

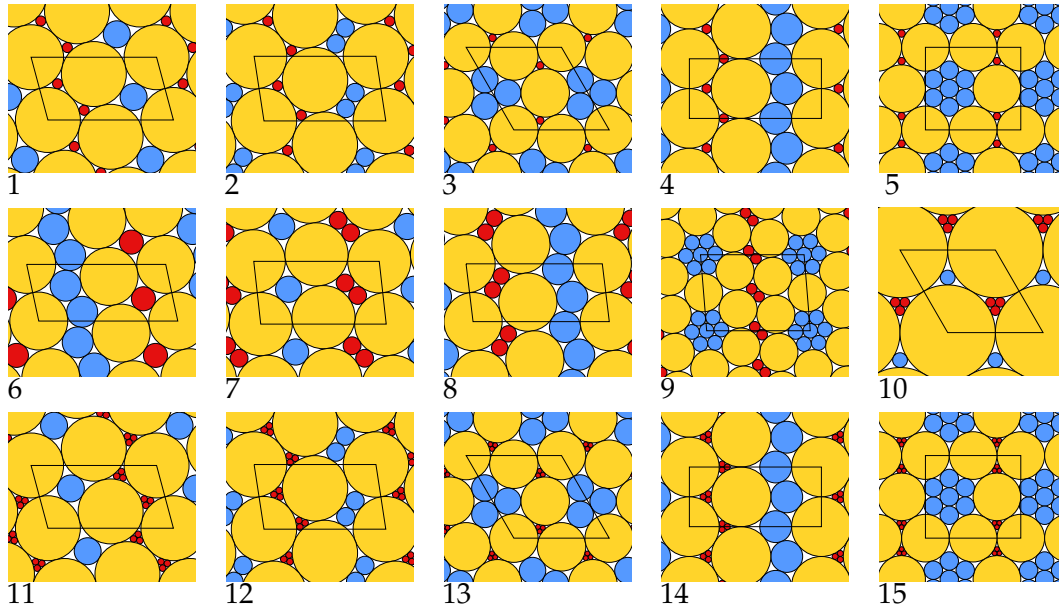
The second problematic case are so-called ϵ -*tight* triangles: those which are close to tight ones. In tight triangles, the emptiness is equal to the potential by definition which means that these values are close in ϵ -tight triangles. To verify (2) in this case, we have to compare the derivatives of the emptiness and the potential. This part is explained in Section 3.3.1 of [17].

As these cases are treated, we verify (2) for the remaining triangles using interval arithmetic which is discussed in Section 3.1. The details of this verification process are given in Section 3.3.2 of [17].

2.3 Proof of Th. 4 (b)

Cases 1-18 are special: they are called *large separated* in [15] since they do not contain pairs of adjacent medium and small discs (see Fig. 7 for the first 15). For each of these cases, in addition to ternary triangulated packings, there are other triangulated packings using only two discs out of three. It happens because the radii of small and medium discs coincide with the radii of small discs of two cases among b_1 – b_9 . It is thus possible to assemble

packings having the same density as the binary packings of mentioned cases using only two of three discs. It turns out that in all these cases, the density of one of the mentioned binary packings exceeds the density of the ternary one. That means, for each of cases 1-18, the densest packing among the triangulated ones is a binary packing corresponding to a case from b_1 - b_9 (Fig. 4).



■ **Figure 7** Triangulated ternary packings for cases 1-15, where a triangulated binary packing maximizes the density. For cases 1-5, it is the triangulated packing of b_8 ; for case 6 - b_4 ; for cases 7-9 - b_7 ; for cases 10-16 - b_9 .

Indeed, each of these ternary packings is formed as a “combination” of two binary packings one of which is denser than the other. Thus, the densest of the binary packings will also be denser than its combination with a less dense packing.

We were able to show that the denser triangulated binary packing maximizes the density among all packings (not only triangulated ones) for the cases from 1 to 15 (Fig. 7). The proof is almost the same as in Section 2.

Let i be the case number and P_3 denote its triangulated ternary packing. Let P_2^* denote the densest triangulated binary packing using two discs of case i and let P_2 denote the less dense triangulated binary packing using two discs of case i . We already know that P_2^* is denser than the two others, $\delta(P_2^*) > \delta(P_3) > \delta(P_2)$. Our aim is to show that P_2^* maximizes the density among all packings by the discs of case i .

The only difference with the strategy used for other cases concerns vertex potentials. Since P_2^* uses only two discs out of three, it features only 2 coronas instead of 3. Thus, these 2 coronas together with the 10 equations for tight triangles, give us at most 11 independent equations instead of 12.

We now need to chose 7 free variables instead of 6. We can pick 6 tight potentials of isosceles triangles as before. There remains to choose the last free variable. Vertex potentials of equilateral tight triangles cannot be picked because of the equations of type $V_{xxx} = E_{xxx}$: they are already fixed. The remaining vertex potentials of isosceles triangles (V_{xxy} , $x \neq y$) cannot be used since they are dependent of the first 6 free variables and the equations $2V_{xxy} + V_{xyx} = E_{xyx}$. The only candidates thus are $V_{1rs}, V_{1sr}, V_{r1s}$; we add one of them.

32:12 When Ternary Triangulated Disc Packings Are Densest

For cases 16, 17, and 18, the densest binary packing is b_5 which features two different coronas around the small disc, so our method is not applicable to them as discussed in Section 5.1.

To summarize, for cases 1-18, among triangulated packings, the density is maximized by a binary packing, not a ternary one as in the Connelly conjecture. However, whether this packing maximizes the density among all packings is still an open question for cases 16–18.

3 Computer implementation

As many proofs of the domain, notably the proof of the Kepler Conjecture [20], the proofs of the maximal density for triangulated packings, like ours and those from [1, 13, 27], essentially rely on computer calculations. In this section, we discuss the details of computer implementation. You can find the complete code at https://github.com/tooticki/ternary_triangulated_disc_packings.

The treatment of each case consists of two steps summarized in Section 2.2. We first choose all the values necessary to define the potential: tight vertex potentials V_{xqy} , constants m_q and capping values Z_q (Section 3.2.1, [17]), the value of ϵ (Section 3.3.1, [17]), and the constants l_{xy}, q_{xy} of the edge potentials (Section 3.2.2, [17]). We choose them in a way to satisfy the “global” inequality (3). The second step is to verify the “local” inequality (2) on all possible triangles.

3.1 Interval arithmetic

We use interval arithmetic in two completely different contexts: to work with real numbers non representable in computer memory and to verify inequalities on uncountable but compact sets of values. More precisely, we use intervals to store the values of radii of discs which are algebraic numbers obtained as roots of polynomials in [15] as well as the value of π . The other situation where we use intervals is to verify the local inequalities on a compact continuum set of triangles.

In interval arithmetic, each value is represented by an interval which contains it and whose endpoints are exact values finitely representable in computer memory (floating-point numbers). Performing functions in interval arithmetic preserves both properties. More precisely, if x_1, \dots, x_n are intervals, and f is an n -ary function, the interval $f(x_1, \dots, x_n)$ must contain $f(y_1, \dots, y_n)$ for all $(y_1, \dots, y_n) \in x_1 \times \dots \times x_n$ and its endpoints are floating-point numbers.

To verify an inequality on two intervals $x_1 < x_2$, it is enough to compare the right endpoint of x_1 and the left endpoint of x_2 . The returned value is `True` only if each pair of values from these intervals satisfy the inequality. However, if the result is `False`, that does not mean that the inequality is false on the numbers represented by x_1 and x_2 , it might also mean that these intervals overlap.

We worked with interval arithmetic implemented in SageMath [6], called Arbitrary Precision Real Intervals¹. The intervals endpoints are floating-point numbers, the precision we use in the majority of cases is the default precision of the library where the mantissa encoding has 53 bits.

¹ https://doc.sagemath.org/html/en/reference/rings_numerical/sage/rings/real_mphi.html

3.2 Polyhedra

As mentioned in Section 2.2, we choose the values of vertex potentials in tight triangles and constants m_1, m_r, m_s in a way to satisfy all the necessary constraints (more details are given in [17]). These constraints together define a subset of \mathbb{R}^9 (where the variables are 6 tight vertex potentials $V_{1r1}, V_{1s1}, V_{r1r}, V_{rsr}, V_{s1s}, V_{sr s}$ and 3 constants m_1, m_r, m_s). We use the Polyhedra module² of SageMath to work with them (it allows us to store the solutions of a system of linear inequalities as a convex polyhedron).

Even more constraints are added by ϵ -tight triangles, since there should exist a positive value of ϵ satisfying the inequalities on derivatives of emptiness and potential given in [17]. To guarantee that, we verify if the inequality hold for $\epsilon = 0$, in other words, we make sure that it holds for some non-negative ϵ . We do it in SageMath: to compute both parts of the inequality, we use interval arithmetic and calculations of derivatives. The obtained inequalities are intersected with the polyhedron calculated above. For all the cases considered in this section, this intersection is not empty (the cases where it was empty are discussed in Section 5.2). Then we find the maximal value of $\epsilon > 0$ allowing the intersection not to be empty and this permits us to fix ϵ .

For all the cases treated in this section, these constraints together define a compact polyhedron in \mathbb{R}^9 (where the variables are the 6 tight vertex potentials and m_1, m_r, m_s).

After we get a polyhedron of valid values, we are free to choose a point inside to fix them. Our aim at that step is to minimize potentials of all triangles in order to satisfy (2). We thus find the three vertices of the polyhedron minimizing m_1, m_r and m_s respectively, compute a linear combination of them (the weights that worked well in practice were respectively 1,1 and 4), and take a point between this one and the center of the polyhedron in order to avoid the approximations problems on the border which are discussed in the next paragraph. Our method to choose the point described above is a heuristic.

Implementing construction of polyhedra, we encounter the following problem: the Polyhedra class does not allow coefficients of constraints to be intervals, while some of the coefficients of our inequalities are stored as such due to their dependency of π and disc radii. Polyhedra do not support intervals as a base ring for a good reason: solutions of a system of linear inequalities with interval coefficients might not form a convex polyhedron. We choose to replace the intervals with their centers and work with an approximation of the actual set of valid values for tight potentials and m_1, m_r, m_s . Our polyhedron is stored in a field of rational values, since this field is computationally quite efficient.

That means, after choosing a point inside this approximated polyhedra, we cannot know if this point actually satisfies all the constraints. To make sure it does, we then rigorously verify that all the inequalities with interval coefficients hold in this point.

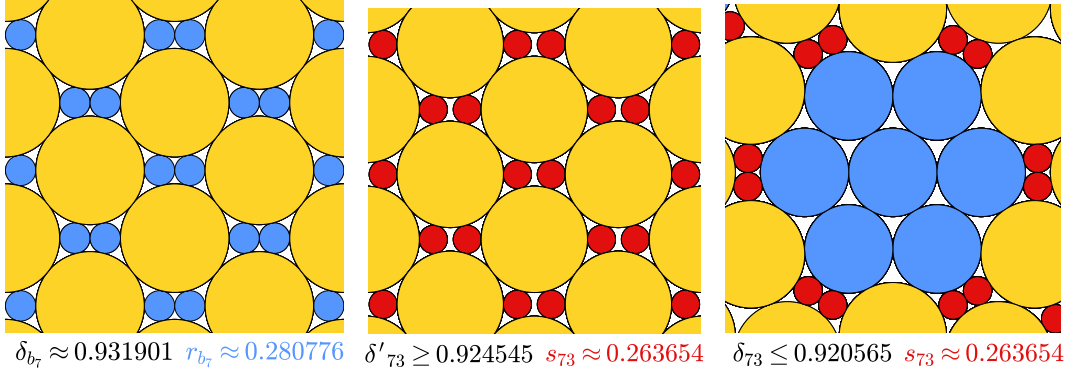
4 Counter-examples: proof of Th. 4 (c)

Starting to work on the density of ternary saturated triangulated packings, we believed the Connelly conjecture to hold, i.e. that for all of the 149 cases, a triangulated packing would maximize the density. Realization that our proof strategy failed for many of them made us suspect the conjecture to be false. Knowing that the density of binary triangulated packings (all of them are given in Figure 4) often exceeds the density of ternary triangulated packings in question gave us an idea to use them in order to find counter examples.

² https://doc.sagemath.org/html/en/reference/discrete_geometry/sage/geometry/polyhedron/constructor.html

32:14 When Ternary Triangulated Disc Packings Are Densest

The first result we obtained was for case 110 [16]. After generalization, we ended up with 40 counter examples (19, 20, 25, 47, 51, 60, 63, 64, 70, 73, 80, 92, 95, 97, 98, 99, 100, 104, 110, 111, 117, 119, 126, 132, 133, 135, 136, 137, 138, 139, 141, 142, 151, 152, 154, 159, 161, 162, 163, 164). They are all non triangulated packings using only two discs out of three which have greater densities than triangulated packings using all three discs. We obtained each of them deforming a triangulated binary packing with discs whose size ratio is close to the one of a pair of discs in the triplet associated to the case. Tiny deformations do not dramatically lower the density and these packings are dense enough to beat the ternary triangulated ones.



■ **Figure 8** Left: a triangulated binary packing of case b_7 . Middle: a deformation where the small discs are replaced with the small discs of case 73. Right: a triangulated periodic packing of case 73, its fundamental domain and description are given in [15].

Let us explain our method on an example. Recall that the pairs of discs allowing binary triangulated packings are denoted by b_1, \dots, b_9 while the triplets with ternary triangulated packings are indexed by positive integers from 1 to 164. Let us consider case 73, its triangulated ternary packing is given in Figure 8, on the right. Notice that the radius of the small disc ($s_{73} \approx 0.263$) of case 73 is close to the radius of the small disc ($r_{b_7} \approx 0.281$) of case b_7 . Let us deform the triangulated binary packing of b_7 (Figure 8, on the left) replacing the small disc of b_7 by the small disc from 73. We choose a deformation which breaks as few contacts between discs as possible (Figure 8, in the middle). Observe that the only broken contact is between the two small discs: they are not tangent anymore. The density of this new non-triangulated packing $\delta' \approx 0.9245$ is higher than the density of the triangulated packing 73 $\delta_{73} \approx 0.9206$ (Figure 8, on the right).

This method is called flip-and-flow [3]. The 40 counter examples were found by computer search. First, for each case b_i , we find the set of pairs of radii from the cases 1-164 with radii ratio “close enough” (we choose the distance heuristically) to the ratio of the discs of b_i . Then we deform the triangulated packing of b_i to obtain packings with the found disc ratios. Our way to deform packings was chosen in order to minimize the number of broken contacts between discs since intuitively it is the best way to keep the density high. Finally, the densities of 40 packings obtained by our method were higher than the densities of the respective ternary triangulated packings which leaves us with the counter examples illustrated in the appendix of the full version of the paper [17].

Our method is not universal: there might be other deformations for certain cases to obtain even higher density and even more counter examples. Besides that, there might be other cases with ternary counter examples (notably, among the cases discussed in Sections 5.2).

5 Other cases

5.1 2 coronas

Among the necessary conditions on vertex potentials in tight triangles given in [17], we saw that the sum of potentials in the corona around any vertex of triangulated packing T^* must be equal to zero. In all the proved cases, each disc has only one possible corona in T^* . It is not always the case, more precisely, among the cases where T^* is saturated, and for which we did not find counter examples, there are 22 cases where one of the discs appears with at least two different coronas in T^* : 16, 17, 18, 36, 49, 52, 57, 58, 65, 78, 84, 90, 106, 114, 120, 148, 153, 155, 156, 157, 158, 160. Each of these cases features a supplementary corona consisting of 6 discs of the same size as the central one. We thus have to add a supplementary condition $6V_{xxx} = 0$, where x is the radius of the disc with two coronas. This however contradicts the condition $3V_{xxx} = E_{xxx}$ in all of these cases. Our density redistribution would need to be less local to solve this problem. In the context of binary triangulated packings, such a case (b_5 , see Figure 4) is treated in detail in Section 5.3 of [1].

5.2 Empty polyhedra

In Section 3.2, we construct a polyhedron in \mathbb{R}^9 aiming to contain all valid values of tight vertex potentials and m_1, m_r, m_s . In this section, we talk about the 52 cases where the polyhedron obtained by our computations is empty: 21, 22, 23, 26, 27, 34, 35, 46, 48, 50, 59, 61, 67, 68, 69, 71, 72, 74, 81, 82, 83, 85, 86, 87, 88, 89, 91, 94, 96, 101, 102, 103, 105, 107, 109, 112, 113, 121, 122, 123, 124, 125, 127, 128, 130, 134, 140, 143, 145, 147, 149, 150.

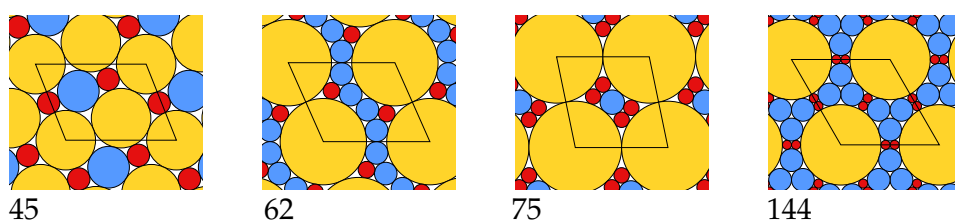
The polyhedron formed by the inequalities on vertex potentials and the inequality for ϵ -triangles (which are given in [17]), represents the values satisfying (\bullet) featuring a non-negative valid ϵ . These constraints are necessary for our proof to be correct. If this polyhedron is empty there are no valid values of tight potentials and m_1, m_r, m_s and thus our strategy of proof is not applicable.

Nevertheless, our computations are limited by computer memory which can represent only certain values. Normally, we avoid this problem by using interval arithmetic (Section 3.1). However, we can not apply this solution with polyhedra. First, as mentioned in Section 3.2, in SageMath, the polyhedra module does not support the interval field as a base ring. Implementing another way to represent “interval polyhedra” would be unreasonable due to memory and time constraints of calculations: the polyhedra are constructed from thousands of inequalities, and performing computations in interval field significantly increases time and memory costs. Instead, we use the ring of rationals to store the inequalities coefficients. Therefore, the polyhedron we work with is an approximation of the actual polyhedron and may not contain all the valid sets of values.

Yet, we believe that the polyhedra in question are probably actually empty in these cases, so the precision issues are not the principal obstacle. All in all, some of the cases from this section might actually maximize the density but we would need an essentially different approach to be able to prove it. Looking forward, further attempts to treat these cases would likely need to use a less local density distribution.

5.3 The 4 mysterious cases

In the four remaining cases (45, 62, 75, and 144) the polyhedron from Section 3.2 is not empty, like for the cases from the previous section. Nevertheless, we could not find a point in it to guarantee the local inequality (2) in all triangles: the problematic triangles are always



■ **Figure 9** Triangulated ternary packings of the four *mysterious* cases.

those close to one of the tight ones. Minimizing m_q and the tight potentials is an obvious strategy to minimize the potentials and eventually satisfy (2) but the capping constants Z_q also dramatically affect potentials.

Trying to find appropriate values of V_{xyz} , m_q and Z_q , we represented all the constraints coming from (3) as a linear optimization problem. This allowed us to encode problematic triangles violating (2) as constraints and add them to the system, one by one, each time one appears during local verification, in hope to finally “converge” to a solution which would satisfy (2) on all triangles. However, this method failed: no solutions were found.




The fact that we could not choose a set of appropriate constants in these cases does not prove that they do not exist (due to the approximation issues already discussed in the previous section as well as the new ones coming from encoding our constraints into a rational linear problem). We, however, believe that these cases, as well as those from the previous section, just cannot be treated by our proof methods. They probably require a less local emptiness redistribution than the one we use.

References

- 1 N. Bedaride and T. Fernique. Density of Binary Disc Packings: The Nine Compact Packings. *Discrete and Computational Geometry*, 67:1–24, 2022. doi:10.1007/s00454-021-00348-7.
- 2 R. Connelly, S. Gortler, E. Solomonides, and M. Yampolskaya. Circle packings, triangulations, and rigidity. *Oral presentation at the conference for the 60th birthday of Thomas C. Hales*, 2018.
- 3 R. Connelly and S. J. Gortler. Packing Disks by Flipping and Flowing. *Discret. Comput. Geom.*, 66:1262–1285, 2021.
- 4 J. Conway and N.J.A. Sloane. *Sphere Packings, Lattices and Groups*. Grundlehren der mathematischen Wissenschaften. Springer New York, 1998.
- 5 S. L. Devados and J. O’Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011.
- 6 The Sage Developers. Sage mathematics software (version 9.0). <http://www.sagemath.org>, 2020.
- 7 E. Fayen, A. Jagannathan, G. Foffi, and F. Smallenburg. Infinite-pressure phase diagram of binary mixtures of (non)additive hard disks. *The Journal of Chemical Physics*, 152(20):204901, 2020.
- 8 L. Fejes Tóth. Über die dichteste Kugellagerung. *Math. Z.*, 48:676–684, 1943. doi:10.1007/BF01180035.
- 9 L. Fejes Tóth. Compact Packing of Circles. *Studia Sci. Math. Hungar.*, 19:103–107, 1984.
- 10 L. Fejes Tóth and J. Molnár. Unterdeckung und Überdeckung der Ebene durch Kreise. *Mathematische Nachrichten*, 18:235–243, 1958.
- 11 S. P. Fekete, P Keldenich, and C. Scheffer. Packing disks into disks with optimal worst-case density. *Discrete and Computational Geometry*, 2022. doi:<https://link.springer.com/article/10.1007/s00454-022-00422-8>.

- 12 S. P. Fekete, S. Morr, and C. Scheffer. Split packing: Packing circles into triangles with optimal worst-case density. In *Algorithms and Data Structures*, pages 373–384. Springer International Publishing, 2017.
- 13 T. Fernique. A densest ternary circle packing in the plane. <https://arxiv.org/abs/1912.02297>, 2019.
- 14 T. Fernique. Density of binary disc packings: Lower and upper bounds. *Experimental Mathematics*, pages 1–12, 2022.
- 15 T. Fernique, A. Hashemi, and O. Sizova. Compact packings of the plane with three sizes of discs. *Discret. Comput. Geom.*, 66(2):613–635, 2021.
- 16 T. Fernique and D. Pchelina. Compact packings are not always the densest. <https://arxiv.org/abs/2104.12458>, 2021.
- 17 T. Fernique and D. Pchelina. Density of triangulated ternary disc packings. <https://arxiv.org/abs/2211.02905>, 2022.
- 18 A. Florian. Ausfüllung der Ebene durch Kreise. *Rendiconti del Circolo Matematico di Palermo*, 9:300–312, 1960.
- 19 C. F. Gauss. Untersuchungen über die Eigenschaften der positiven ternären quadratischen Formen von Ludwig August Seber. *Göttingische gelehrte Anzeigen*, 1831.
- 20 T. C. Hales. A proof of the Kepler conjecture. *Annals of Mathematics*, 162(3):1065–1185, 2005.
- 21 T. C. Hales, M. Adams, G. Bauer, D. T. Dang, J. Harrison, T. L. Hoang, C. Kaliszyk, V. Magron, S. McLaughlin, T. T. Nguyen, T. Q. Nguyen, T. Nipkow, S. Obua, J. Pleso, J. Rute, A. Solovyev, A. H. T. Ta, T. N. Tran, D. T. Trieu, J. Urban, K. K. Vu, and R. Zumkeller. A formal proof of the Kepler conjecture. *Forum of Mathematics, Pi*, 5:e2, 2017. doi:10.1017/fmp.2017.1.
- 22 T. C. Hales and S. P. Ferguson. The Kepler conjecture. *Discrete Comput. Geom.*, 36(1):1–269, 2006.
- 23 T. C. Hales and S. P. Ferguson. *A Formulation of the Kepler Conjecture*, pages 83–133. Springer New York, New York, NY, 2011.
- 24 A. Heppes. On the densest packing of discs of radius 1 and $\sqrt{2} - 1$. *Studia Scientiarum Mathematicarum Hungarica*, 36:433–454, 2000.
- 25 A. Heppes. Some densest two-size disc packings in the plane. *Discrete and Computational Geometry*, 30:241–262, 2003.
- 26 A. B. Hopkins, F. H. Stillinger, and S. Torquato. Densest binary sphere packings. *Phys. Rev. E*, 85:021130, 2012.
- 27 T. Kennedy. A densest compact planar packing with two sizes of discs. <https://arxiv.org/abs/math/0412418>, 2005.
- 28 T. Kennedy. Compact packings of the plane with two sizes of discs. *Discret. Comput. Geom.*, 35(2):255–267, 2006. doi:10.1007/s00454-005-1172-4.
- 29 J. Lagarias. Bounds for local density of sphere packings and the Kepler conjecture. *Discrete and Computational Geometry*, 27:165–193, 2002. doi:10.1007/s00454-001-0060-9.
- 30 M. Messerschmidt. The number of configurations of radii that can occur in compact packings of the plane with discs of n sizes is finite. <https://arxiv.org/abs/2110.15831>, 2021. doi:10.48550/ARXIV.2110.15831.
- 31 P. I. O’Toole and T. S. Hudson. New high-density packings of similarly sized binary spheres. *The Journal of Physical Chemistry C*, 115(39):19037–19040, 2011.
- 32 T. Paik, B. T. Diroll, C. R. Kagan, and C. B. Murray. Binary and Ternary Superlattices Self-Assembled from Colloidal Nanodisks and Nanorods. *Journal of the American Chemical Society*, 137(20):6662–6669, 2015.
- 33 A. Thue. Über die dichteste Zusammenstellung von kongruenten Kreisen in einer Ebene. *Christiania Videnskabs-Selskabets Skrifter*, I. Math.-Naturv. Klasse, 1:1–9, 1910.

Labeled Nearest Neighbor Search and Metric Spanners via Locality Sensitive Orderings

Arnold Filtser   

Department of Computer Science, Bar-Ilan University, Ramat-Gan, Israel

Abstract

Chan, Har-Peled, and Jones [SICOMP 2020] developed locality-sensitive orderings (LSO) for Euclidean space. A (τ, ρ) -LSO is a collection Σ of orderings such that for every $x, y \in \mathbb{R}^d$ there is an ordering $\sigma \in \Sigma$, where all the points between x and y w.r.t. σ are in the ρ -neighborhood of either x or y . In essence, LSO allow one to reduce problems to the 1-dimensional line. Later, Filtser and Le [STOC 2022] developed LSO's for doubling metrics, general metric spaces, and minor free graphs.

For Euclidean and doubling spaces, the number of orderings in the LSO is exponential in the dimension, which made them mainly useful for the low dimensional regime. In this paper, we develop new LSO's for Euclidean, ℓ_p , and doubling spaces that allow us to trade larger stretch for a much smaller number of orderings. We then use our new LSO's (as well as the previous ones) to construct path reporting low hop spanners, fault tolerant spanners, reliable spanners, and light spanners for different metric spaces.

While many nearest neighbor search (NNS) data structures were constructed for metric spaces with implicit distance representations (where the distance between two metric points can be computed using their names, e.g. Euclidean space), for other spaces almost nothing is known. In this paper we initiate the study of the labeled NNS problem, where one is allowed to artificially assign labels (short names) to metric points. We use LSO's to construct efficient labeled NNS data structures in this model.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry; Theory of computation \rightarrow Sparsification and spanners

Keywords and phrases Locality sensitive ordering, nearest neighbor search, high dimensional Euclidean space, doubling dimension, planar and minor free graphs, path reporting low hop spanner, fault tolerant spanner, reliable spanner, light spanner

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.33

Related Version Full Version: <https://arxiv.org/abs/2211.11846> [46]

Funding This research was supported by the Israel Science Foundation (grant No. 1042/22).

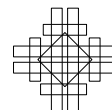
1 Introduction

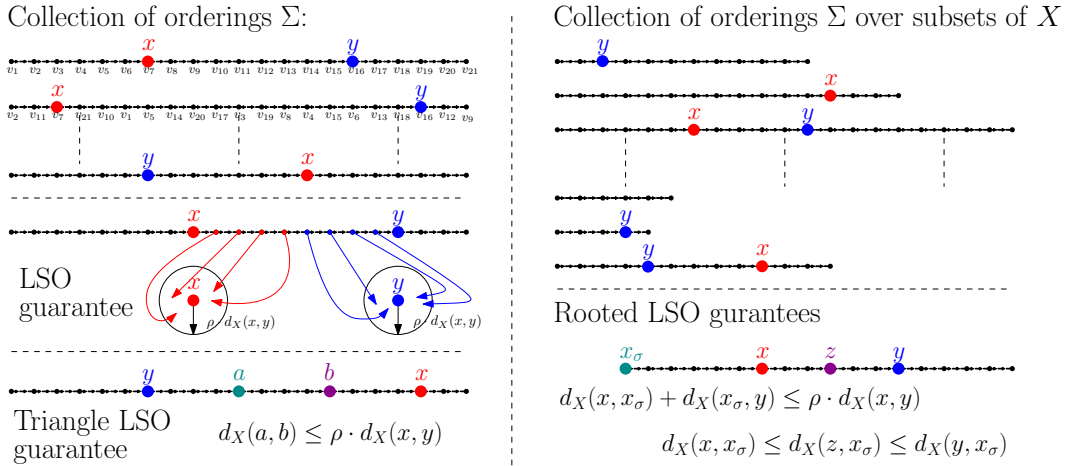
1.1 Locality Sensitive Ordering

Chan, Har-Peled, and Jones [30] recently introduce a new and powerful tool into the algorithmist's toolkit, called *locality sensitive ordering* (abbreviated LSO). LSO provides an order over the points of a metric space (X, d_X) , this order being very useful, as it helps to store, sort, and search the data (among other manipulations).

► **Definition 1** ((τ, ρ) -LSO). *Given a metric space (X, d_X) , we say that a collection Σ of orderings is a (τ, ρ) -LSO if $|\Sigma| \leq \tau$, and for every $x, y \in X$, there is a linear ordering $\sigma \in \Sigma$ such that (w.l.o.g.¹) $x \preceq_\sigma y$ and the points between x and y w.r.t. σ could be partitioned into two consecutive intervals I_x, I_y where $I_x \subseteq B_X(x, \rho \cdot d_X(x, y))$ and $I_y \subseteq B_X(y, \rho \cdot d_X(x, y))$. ρ is called the stretch parameter.*

¹ That is either $x \preceq_\sigma y$ or $y \preceq_\sigma x$, and the guarantee holds w.r.t. all the points between x and y in the order σ .





■ **Figure 1** Illustration of different types of LSO.

Morally speaking, given a problem, LSO can reduce it from a general and complicated space to a much simpler space: 1-dimensional line. Chan et al. [30] constructed $(O_d(\epsilon^{-d}) \cdot \log \frac{1}{\epsilon}, \epsilon)$ -LSO for the d -dimensional Euclidean space. They used their LSO to design simple dynamic algorithms for approximate nearest neighbor search, bichromatic closest pair, MST, spanners, and fault-tolerant spanners. Later, Buchin, Har-Peled, and Oláh [27, 28] constructed reliable spanners using LSO, obtaining considerably superior results compared with previous techniques.

Filtser and Le [49] generalized Chan et al. [30] result to doubling spaces,² showing that every metric space with doubling dimension d admits a $(\epsilon^{-O(d)}, \epsilon)$ -LSO. Furthermore, they generalized the concept of LSO to other metric spaces, defining the two related notions of *triangle-LSO* (which turn to be useful for general metric spaces), and *left-sided LSO* (which turn to be useful for topologically restricted graphs). Here, instead of presenting the left-sided LSO's of [49], we introduce the closely related notion of *rooted-LSO*, which has some additional structure. All the results and constructions for left-sided LSO in [49] hold for rooted LSO as well. We refer to [49] for a comparison between the different notions, and to Figure 1 for an illustration.

► **Definition 2** $((\tau, \rho)$ -Triangle-LSO). *Given a metric space (X, d_X) , we say that a collection Σ of orderings is a (τ, ρ) -triangle-LSO if $|\Sigma| \leq \tau$, and for every $x, y \in X$, there is an ordering $\sigma \in \Sigma$ such that (w.l.o.g.¹) $x \prec_\sigma y$, and for every $a, b \in X$ such that $x \preceq_\sigma a \preceq_\sigma b \preceq_\sigma y$ it holds that $d_X(a, b) \leq \rho \cdot d_X(x, y)$.*

► **Definition 3** $((\tau, \rho)$ -rooted-LSO). *Given a metric space (X, d_X) , we say that a collection Σ of orderings over subsets of X is a (τ, ρ) -rooted-LSO if the following hold:*

- *Each point $x \in X$ belongs to at most τ orderings in Σ .*
- *Each ordering $\sigma \in \Sigma$ is associated with a point $x_\sigma \in X$, which is the first in the order, and such that the ordering is w.r.t. distances from x_σ (i.e. $y \prec_\sigma z \Rightarrow d_X(x_\sigma, y) \leq d_X(x_\sigma, z)$).*
- *For every pair of points u, v , there is some $\sigma \in \Sigma$ containing both x, y , and such that $d_G(u, x_\sigma) + d_G(x_\sigma, v) \leq \rho \cdot d_G(u, v)$.*

² A metric (X, d) has doubling dimension d if any ball of radius $2r$ can be covered by 2^d balls of radius r .

Filtser and Le [49] constructed triangle LSO for general metrics, and rooted LSO for the shortest path metrics of trees, treewidth graphs, planar graphs, and graph excluding a fixed minor. They used their LSO’s to construct oblivious reliable spanners for the respective metric spaces, considerably improving previous constructions (that used different techniques). All the known results on LSO’s are summarized in Table 1.

■ **Table 1** Summary of all known results, on all the different types of locality sensitive orderings (LSO). $k \in \mathbb{N}$, $t > 1$, $\epsilon \in (0, 1)$ is an arbitrarily small parameter. ^(*) O_d hides an arbitrary function of d , the number of orderings in [30] LSO is $O_d(\epsilon^{-d}) \cdot \log \frac{1}{\epsilon} = 2^{O(d)} \cdot d^{\frac{3}{2}d} \cdot \epsilon^{-d} \cdot \log \frac{1}{\epsilon}$.

LSO type	Metric Space	# of orderings (τ)	Stretch (ρ)	Ref
(Classic) LSO	Euclidean space \mathbb{R}^d	$O_d(\epsilon^{-d}) \cdot \log \frac{1}{\epsilon}$ ^(*)	ϵ	[30]
	Doubling dimension d	$\epsilon^{-O(d)}$	ϵ	[49]
Triangle-LSO	General metric	$O(n^{\frac{1}{k}} \cdot \log n \cdot \frac{k^2}{\epsilon} \cdot \log \frac{k}{\epsilon})$	$2k + \epsilon$	[49]
	Euclidean space \mathbb{R}^d	$e^{\frac{d}{2t^2 \cdot (1 + \frac{2}{t^2})}} \cdot \tilde{O}(\frac{d^{1.5}}{\epsilon \cdot t})$	$(1 + \epsilon)t$	Thm. 4
	ℓ_p^d for $p \in [1, 2]$	$e^{O(d/t^p)} \cdot \tilde{O}(d)$	t	Thm. 5
	ℓ_p^d for $p \in [2, \infty]$	$\tilde{O}(d)$	$d^{1 - \frac{1}{p}}$	FullV[46]
	Doubling dimension d	$2^{O(d/t)} \cdot d \cdot \log^2 t$	t	Thm. 6
Rooted LSO	Tree	$\log n$	1	[49]
	Treewidth k	$k \cdot \log n$	1	[49]
	Planar / fixed minor free	$O(\frac{1}{\epsilon} \cdot \log^2 n)$	$1 + \epsilon$	[49]

Previously constructed LSO for the Euclidean space [30], as well as for metric spaces with doubling dimension d [49], have exponential dependency on the dimension in their cardinality, a phenomena often referred to as “the curse of dimensionality”. When the dimension is high, it can be a major obstacle. Indeed, the distances induced by n point in an $O(\log n)$ -dimensional Euclidean space induce a metric space which is much more structured than a general metric space. Therefore one might expect them to admit better LSO. However, using [30] one can only construct (n, ϵ) -LSO (note that every metric admits $(\lceil \frac{n}{2} \rceil, 0)$ -LSO ³).

Every n point metric space has doubling dimension at most $\log n$. Consider the case where the doubling dimension is somewhat large (e.g. $\sqrt{\log n}$) but not maximal. It is much more structured than general metric, however the only construction we have [49] gives us $\epsilon^{-O(d)}$ orderings, which might be too large. In the small number of orderings regime, could we take advantage of the doubling structure to construct better LSO then for general metrics?

Our Contribution. In this paper we construct new triangle-LSO for high dimensional spaces. We then present many applications for the newly constructed LSO’s, as well as for the previously constructed LSO’s. Old and new LSO construction are summarized in Table 1.

► **Theorem 4.** For every $t \in [4, 2\sqrt{d}]$, $\delta \in (0, 1]$, and $d \geq 1$, the d -dimensional Euclidean space \mathbb{R}^d admits $\left(O\left(\frac{d^{1.5}}{\delta \cdot t} \cdot \log\left(\frac{2\sqrt{d}}{t}\right) \cdot \log \frac{d}{\delta} \cdot e^{\frac{d}{2t^2 \cdot (1 + \frac{2}{t^2})}} \right), (1 + \delta)t \right)$ -triangle LSO.

For $t = \frac{2}{3}\sqrt{d}$ and $\delta = \frac{1}{2}$, we obtain $(O(d \log d), \sqrt{d})$ -triangle LSO. In particular, for every set of n points in ℓ_2 , using the Johnson Lindenstrauss dimension reduction [61], for every fixed $t > 1$, we can construct $(n^{\frac{1}{t^2}} \cdot \tilde{O}(\frac{\log^{1.5} n}{t}), O(t))$ -triangle LSO, or $(\tilde{O}(\log n), O(\sqrt{\log n}))$ -triangle LSO, a quadratic improvement compared with general n -point metric spaces!

³ This follows from a theorem by Walecki [7] who showed that the edges of the K_n clique graph can be partitioned into $\lceil \frac{n}{2} \rceil$ Hamiltonian paths.

Interestingly, we show that the $(O(d \log d), \sqrt{d})$ -triangle LSO Σ for ℓ_2 , is in the same time also a $(O(d \log d), d^{\frac{1}{p}})$ -triangle LSO for ℓ_p where $p \in [1, 2]$, and $(O(d \log d), d^{1-\frac{1}{p}})$ -triangle LSO for ℓ_p where $p \in [2, \infty]$. For $p \in [1, 2]$, we generalize Theorem 4 to ℓ_p spaces to get the entire #ordering-stretch trade-off. Finally, we generalize Theorem 4 to general metric spaces with doubling dimension d .

► **Theorem 5.** For every $p \in [1, 2]$, $t \in [5, d^{\frac{1}{p}}]$ and $d \geq 1$, the d -dimensional ℓ_p space admits $(e^{O(\frac{d}{t^p})} \cdot \tilde{O}(d), t)$ -triangle LSO.

► **Theorem 6.** Given a metric space (X, d_X) with doubling dimension d , and parameter $t \in [\Omega(1), d]$, X admits $(2^{O(d/t)} \cdot d \cdot \log^2 t, t)$ -triangle LSO.

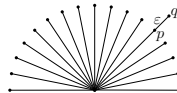
For $t = d$, we get $(\tilde{O}(d), d)$ -triangle LSO, again much better than general metric spaces!

1.2 Labeled Nearest Neighbor Search

Nearest neighbor search (abbreviated NNS) is a classical and fundamental task used in numerous domains including machine learning, clustering, document retrieval, databases, statistics, data compression, database queries, computational biology, data mining, pattern recognition, and many others. In the NNS problem we are given a set P of points in a metric space (X, d_X) . The goal is to construct a succinct data structure that given a query point $q \in X$, quickly returns a point $p \in P$ closest to q (i.e. $\arg \min_{p \in P} d_X(p, q)$). In order to keep the size of the data structure, and the query time small, usually approximation is allowed. In the t -approximate nearest neighbor problem (abbreviated t -NNS) the goal is to return a point p at distance at most $t \cdot \min_{p \in P} d_X(p, q)$ from q . The problem was extensively studied in ℓ_p spaces (see the survey [11]), and also in various norm spaces over \mathbb{R}^d (see e.g. [12, 13]). NNS data structures were also constructed beyond normed spaces. Some examples are Earth-Mover distance [60], Edit Distance [79, 11], and Fréchet distance [59, 41, 43, 47]. We observe that a crucial property shared by these examples, is that they have an “implicit distance representation”. That is, it is possible to compute the distance between two points using only their names (e.g. the coordinates values in \mathbb{R}^d used as names: $d_{\mathbb{R}^d}((x_1, \dots, x_d), (y_1, \dots, y_d)) = \|(x_1, \dots, x_d) - (y_1, \dots, y_d)\|_2$).

For general metric spaces, Krauthgamer and Lee [66] introduced the *black box model*. Here one is given access to an exact distance oracle ⁴ DO that answer distance queries in t_{DO} time. They showed that one can construct an efficient $(1 + \epsilon)$ -NNS (that is with polynomial space, and polylogarithmic query time), if and only if the doubling dimension of X is at most $O(\log \log n)$.

Indeed, for metric spaces with large doubling dimension, distance queries provide very limited information. Consider for example the case where the input metric is the star graph (inducing uniform metric on the leaves, see illustration below), and the query point attached to one of the leaves with an edge of infinitesimal weight, one must query all the points before finding any finite approximation to the nearest neighbor.



⁴ An exact distance oracle D is a data structure that given two points x, y , returns $\text{est}(x, y) = d_X(x, y)$. A distance oracle of stretch t returns a value $\text{est}(x, y)$ in $[d_X(x, y), t \cdot d_X(x, y)]$.

An interesting case studied by Abraham, Chechik, Krauthgamer, and Wieder [3] is that of planar graphs. Here we are given a huge weighted planar graph $G = (V, E, w)$ with N vertices, and a subset of n vertices $X \subseteq V$. The goal is to solve the $(1 + \epsilon)$ -NNS problem w.r.t. the shortest path metric d_G , input set X and queries from V . Assuming access to an exact distance oracle⁴ DO that answer distance queries in t_{DO} time, and given a planar graph G of maximum degree Δ , Abraham et al. [3] constructed a $(1 + \epsilon)$ -NNS data structure for planar graph of size $n \cdot O(\epsilon^{-1} \cdot \log \log N + \Delta \cdot \log^2 n)$ and query time $O((\epsilon^{-1} \cdot \log \log n + t_{\text{DO}}) \cdot \log \log N + \log n \cdot \Delta \cdot t_{\text{DO}})$.

Linear dependence on the degree is a very limiting requirement, as planar graphs have a priori unbounded degree. Moreover, exact distance computations (even in planar graphs) are time consuming, and if the graph is big enough could be infeasible. Exact distance oracle is a highly non-trivial assumption, it is an expensive data structure,⁵ better to be avoided. One might hope to relax either the max degree assumption, or to use the much more reasonable and efficient data structure of approximate distance oracle [84, 64, 70]. Unfortunately, Abraham et al. [3] showed both assumptions to be necessary. Specifically, the dependence on the degree is necessary, as every NNS data structure with space at most $O(\frac{N}{\Delta \log_{\Delta} n})$ must probe the distance oracle at least $\Omega(\Delta \log_{\Delta} n)$ times. Furthermore, they show that if one is only given access to a $(1 + \epsilon)$ -distance oracle, then there is a planar graph (in fact a tree) with maximum degree $O(\log n)$, aspect ratio $O(\frac{\log n}{\epsilon})$, $N \leq n^2$, and the NNS data structure is forced to make $\Omega(n)$ queries to the distance oracle.

To conclude this discussion, exact distance oracle (assumed both by the black box model [66] and [3]) is an expensive data structure, which enables us to construct efficient NNS only under very limiting assumptions (small doubling dimension / constant maximum degree in planar graphs). On the other hand in many metric spaces with “implicit distance representation” efficient NNS were constructed. The crux is that the information stored in the name (e.g. coordinate values) used to perform various manipulations on the data, in addition to distance computation. What if in planar graphs, or even in completely general metric spaces, we could choose the names of the metric points, or alternatively assigning each point a short label, would it be possible to construct efficient NNS data structures?

To answer this question, we introduce the *labeled t -NNS* problem.

► **Definition 7 (Labeled t -NNS).** Consider an N -point metric space (X, d_X) , where one can assign to each point $x \in X$ an arbitrary short label l_x . Given a subset $P \subseteq X$ of size n (unknown in advance) together with their labels $\{l_x\}_{x \in P}$ (but without access to (X, d_X) or any additional information) the goal is to construct a NNS search data structure as follows: given a query $q \in X$ together with its assigned label l_q , the data structure will return a t -approximate nearest neighbor $p \in P$: $d_X(p, q) \leq t \cdot \min_{x \in P} d_X(x, q)$. The parameters of study are: label size, data structure size, query time, and approximation factor t .

We also consider the scenario where the set P is changing dynamically: points are added and removed from P . Here we are required to maintain a data structure for P , while minimizing the update time (as well as all the other parameters).⁶

In the labeled NNS model we get to assign a short label (alternatively choose a name) for each point in a big metric space (X, d_X) . These labels try to imitate the natural hint provided by the name of the points themselves in metric spaces with implicit distance

⁵ After a long line of work, the state of the art (by Long and Pettie [73]) requires either super-linear space $N^{1+o(1)}$, or very large query time $N^{o(1)}$, both quite undesirable.

⁶ For example, consider a NNS data structure for a set P . Dynamic NNS, should be able to efficiently update the data structure to work w.r.t. a slightly updated set $P' = P \cup \{x\} \setminus \{y\}$ instead of P .

representation. The main object of study here is the trade-off between label size, and the approximation of the resulting NNS. A trivial choice of label for each point x will be simply to store distances to all other points. However the label size $\Omega(N)$ is infeasible. A more sophisticated solution is the following: fix constants $k, t \in \mathbb{N}$, and embed all the points in (X, d_X) into $d = \tilde{O}(N^{\frac{1}{k}})$ -dimensional ℓ_∞ [75, 1]. That is we assign each point x a vector $v_x \in \mathbb{R}^d$ such that $\forall x, y \in X, d_X(x, y) \leq \|v_x - v_y\|_\infty \leq (2k - 1) \cdot d_X(x, y)$, and use the vectors as labels. Given an n point subset $P \subseteq X$ with its respective labels (vectors), use Indyk's NNS [58] over $\{v_x\}_{x \in P}$ to construct a NNS data structure \mathbb{D}_{Ind} with approximation factor $O(\log_{1+\frac{1}{t}} \log d) = O(t \cdot \log \log N)$ w.r.t. the ℓ_∞ vectors, space $\tilde{O}(d \cdot n^{1+\frac{1}{t}}) = \tilde{O}(N^{\frac{1}{k}} \cdot n^{1+\frac{1}{t}})$ and query time $\tilde{O}(n^{1+\frac{1}{k}})$. Given a query q , we will simply query \mathbb{D}_{Ind} on the vector v_q , and on answer v_p will return p . Note that the query time and space are the same as above, while the approximation factor will be $O(k \cdot t \cdot \log \log N)$.

Our Contribution. Our results for the labeled t -NNS are summarized in Table 2. We begin by proving meta theorem showing that (τ, ρ) -rooted LSO implies a labeled ρ -NNS with label size $O(\tau)$, space $O(n \cdot \tau)$, query time $O(\tau)$, and update time $O(\tau \cdot \log \log N)$. As a result we conclude efficient labeled $(1 + \epsilon)$ -NNS data structures for fixed minor free graphs (and planar), and exact labeled NNS for treewidth graphs. Another interesting corollary is an efficient labeled NNS for metrics with small *correlation dimension* (a generalization of doubling, see [29]).

■ **Table 2** Labeled NNS data structures for different families. The $*$ sign is replacing $O(\log \log N)$. The second to last line is a lower bound. Space is measured in machine words. The label size and query time always equal. The space in all the cases above equals n times the label size.

Family	stretch	label	query time	update time	Ref
Minor free	$1 + \epsilon$	$O(\frac{1}{\epsilon} \log^2 N)$	$O(\frac{1}{\epsilon} \log^2 N)$	$\frac{1}{\epsilon} \cdot \tilde{O}(\log^2 N)$	FullV[46]
Treewidth k	1	$O(k \log N)$	$O(k \log N)$	$k \cdot \tilde{O}(\log N)$	FullV[46]
Correlation k	$1 + \epsilon$	$\tilde{O}_{k,\epsilon}(\sqrt{N})$	$\tilde{O}_{k,\epsilon}(\sqrt{N})$	$\tilde{O}_{k,\epsilon}(\sqrt{N})$	FullV[46]
Ultrametric	1	$O(\log N)$	*	*	FullV[46]
General Metric	$8(1 + \epsilon)k$	$O(\frac{k}{\epsilon} N^{\frac{1}{k}} \cdot \log N)$	$O(\frac{1}{\epsilon} \cdot *)$	$O(\frac{k}{\epsilon} N^{\frac{1}{k}} \cdot *)$	FullV[46]
	$t < 2k + 1$	$\tilde{\Omega}(N^{\frac{1}{k}})$	arbitrary	arbitrary	FullV[46]
Doubling d	t	$2^{O(d/t)} \cdot \tilde{O}(d) \cdot \log N$	$2^{O(d/t)} \cdot \tilde{O}(d) \cdot *$	$2^{O(d/t)} \cdot \tilde{O}(d) \cdot *$	FullV[46]

Next, we prove a meta theorem, showing that (τ, ρ) -triangle LSO implies a labeled 2ρ -NNS with label size $O(\tau \cdot \log N)$, space $O(n \cdot \tau \cdot \log N)$, and query and update time $O(\tau \cdot \log \log N)$. We conclude an efficient labeled NNS for graphs with large doubling dimension. For the high-dimensional Euclidean space, approximate nearest neighbor search was extensively studied (see the survey [11], and additional discussion in the full version [46]). However, for the case of doubling metrics, NNS never went beyond $1 + \epsilon$ approximation. In particular, in all existing solutions the query time and space have exponential dependence on the dimension (see references in the full version [46]). Thus ours are the first results in this regime, removing “the curse of dimensionality”.

As an additional corollary of the triangle LSO to labeled NNS meta theorem one can derive a NNS of for general metric spaces which considerably improved upon the labeled NNS based on [75]+[58] discussed above. However, the query time turns out to be somewhat large. We provide direct constructions for labeled NNS for general metrics, getting label size $\tilde{O}(\epsilon^{-1} \cdot N^{\frac{1}{k}})$, stretch $8(1 + \epsilon)k$ and very small query time: $O(\epsilon^{-1} \cdot \log \log N)$. We show that the standard information theoretic bound applies for the labeled NNS as well, specifically, for

stretch $t < 2k + 1$, the label size must be $\tilde{\Omega}(n^{\frac{1}{k}})$ (regardless of query time). Finally, we put special focus on the regime where the stretch is $O(\log N)$. We obtain labeled NNS scheme with very short label and small query time. Most notably, assuming polynomial aspect ratio, and allowing the bound on the label to be only in expectation, we can obtain $O(1)$ label size, and $O(\log \log N)$ query time.

1.3 Spanners

Given a metric space (X, d_X) , a metric *spanner* is a graph H over X points, such that that the shortest path metric d_H in H , closely resembles the metric d_X . Formally, a t -spanner for X is a weighted graph $H(X, E, w)$ that has $w(u, v) = d_X(u, v)$ for every edge $(u, v) \in E$ and $d_H(x, y) \leq t \cdot d_X(x, y)$ for every pair of points $x, y \in X$.⁷ The classic parameter of study is the trade-off between stretch and sparsity (number of edges). Althöfer et al. [8] showed that every n point metric space admits a $2k - 1$ spanner with $O(n^{1+\frac{1}{k}})$ edges, while every set of n points in \mathbb{R}^d , or more generally metric space of doubling dimension d , admits a $(1 + \epsilon)$ -spanner with $n \cdot \epsilon^{-O(d)}$ edges [38, 52]. We refer to the book [77], and the survey [4] for an overview.

Path Reporting Low Hop Spanners. Recently, Kahalon, Le, Milenkovic, and Solomon [62] studied *path reporting low-hop spanners*. While a t -spanner guarantees that a “short” path exists between every two points, such a path might be very long, and finding it is a time consuming operation. A path reporting t -spanner, is a spanner accompanied with a data structure that given a query pair $\{x, y\}$, efficiently retrieves a path between x and y (of total weight $\leq t \cdot d_X(x, y)$). A path P with h edges is called an h -hop path. H is an h -hop t -spanner of X if for every $x, y \in X$, there is an h -hop path P from x to y in H , such that $w(P) \leq t \cdot d_X(x, y)$. Clearly, the time required to report a path is at least as large as the number of edges along the path, thus we wish to minimize the number of hops.

Low number of hops is a highly desirable property in network design, as each transmission causes delays, which are non-negligible when the number of transmissions is large [5, 23]. Low hop networks are also known to be more reliable [23, 87, 82], and used in electricity and telecommunications [23], and many other (practical) network design problems [71, 16, 55, 54, 81]. Hop-constrained network approximation is often used in parallel computing [36, 14], as the number of hops governs the number of required parallel rounds (e.g. in Dijkstra).

Kahalon et al. [62] constructed path reporting low-hop spanners for many spaces, such as path reporting 2-hop $O(k)$ -spanners with $O(n^{1+\frac{1}{k}} \cdot k \cdot \log n)$ edges, and $O(1)$ query time for general metrics, and path reporting 2-hop $(1 + \epsilon)$ -spanners with $O(\frac{n}{\epsilon^2} \cdot \log^3 n)$ edges and $O(\epsilon^{-2} \cdot \log^2 n)$ query time for planar graphs. They showed a plethora of applications for their spanners: compact routing schemes, fault tolerant routing, spanner sparsification, approximate shortest path trees, minimum weight trees (MST), and online MST verification.

Our Contribution. Kahalon et al. [62] first constructed path reporting low hop spanners for trees, and then reduced each type of metric to the case of trees. We observe that it is actually enough to reduce to the even simpler case of paths, and obtain a host of such spanners using LSO’s. We then manually improve some of the resulting spanners, most notably we create

⁷ Frequently the literature is concerned with graph spanners, where given a graph $G = (V, E, w)$ the goal is to find a subgraph H preserving distances. Here we study metric spanners, where there is no underlying graph.

■ **Table 3** Summary of old and new results on path reporting low hop spanners. The spanners are for n point metrics, and all report paths with hop bound 2. Here $\epsilon \in (0, 1)$, $k, d \geq 1$ are integers. The space required for the path reporting data structure is asymptotically equal to the sparsity of the spanner in all the cases other than Euclidean space, where there is an additional additive factor of $O_d(\epsilon^{-2d}) \log \frac{1}{\epsilon}$.

Metric family	stretch	sparsity	query time	Ref
General Metric	$O(k)$	$O\left(n^{1+\frac{1}{k}} \cdot k \cdot \log n\right)$	$O(1)$	[62]
	$2k - 1$	$O(n^{1+\frac{1}{k}} \cdot k)$	$O(k)$	FullV[46], [85]
	$(1 + \epsilon)(4k - 2)$	$O(n^{1+\frac{1}{k}} \cdot \epsilon^{-1} \cdot k \cdot \log \Phi)$	$O(\epsilon^{-1} \cdot \log 2k)$	FullV[46]
Doubling Dimension	$1 + \epsilon$	$\epsilon^{-O(d)} \cdot n \cdot \log n$	$\epsilon^{-O(d)}$	[62]
	t	$2^{-O(d/t)} \cdot \tilde{O}(n)$	$2^{-O(d/t)} \cdot d \cdot \log^2 t$	FullV[46]
Euclidean \mathbb{R}^d	$1 + \epsilon$	$O_d(\epsilon^{-d}) \cdot \log \frac{1}{\epsilon} \cdot n \cdot \log n$	$O_d(1)$	FullV[46]
	$(1 + \epsilon)t$	$\tilde{O}\left(\frac{d^{1.5}}{\epsilon \cdot t}\right) \cdot e^{\frac{2d}{t^2} \cdot (1 + \frac{8}{t^2})} \cdot n \log n$	$\tilde{O}\left(\frac{d^{1.5}}{\epsilon \cdot t}\right) \cdot e^{\frac{2d}{t^2} \cdot (1 + \frac{8}{t^2})}$	FullV[46]
$\ell_p^d, p \in [1, 2]$	t	$\tilde{O}(d) \cdot e^{O(\frac{d}{t^p})} \cdot n \log n$	$\tilde{O}(d) \cdot e^{O(\frac{d}{t^p})}$	FullV[46]
$\ell_p^d, p \in [2, \infty]$	$2 \cdot d^{1-\frac{1}{p}}$	$\tilde{O}(d) \cdot n \log n$	$\tilde{O}(d)$	FullV[46]
Tree	1	$O(n \cdot \log n)$	$O(1)$	[62]
Fixed Minor Free	$1 + \epsilon$	$O(n \cdot \epsilon^{-2} \cdot \log^3 n)$	$O(\epsilon^{-2} \cdot \log^2 n)$	[62]
	$1 + \epsilon$	$O(n \cdot \epsilon^{-1} \cdot \log^2 n)$	$O(\epsilon^{-1} \cdot \log n)$	FullV[46]
Planar	$1 + \epsilon$	$O(n \cdot \epsilon^{-1} \cdot \log^2 n)$	$O(\epsilon^{-1})$	FullV[46]
Treewidth k	1	$O(n \cdot k \cdot \log n)$	$O(k)$	FullV[46]

path reporting 2-hop $(1 + \epsilon)$ -spanner for planar graph with $O(\frac{n}{\epsilon} \log^2 n)$ edges and $O(\frac{1}{\epsilon})$ -query time, and a path reporting 2-hop $(1 + \epsilon)$ -spanner for points in d -dimensional Euclidean space with $O_d(\epsilon^{-d}) \cdot \log \frac{1}{\epsilon} \cdot n \log n$ edges and $O_d(1)$ -query time. See Table 3 for a summary of old and new results.

Fault tolerant spanners. Levcopoulos, Narasimhan, and Smid [72] introduced the notion of a fault-tolerant spanner. A graph $H = (X, E_H, w)$ is an f -vertex-fault-tolerant t -spanner of a metric space (X, d_X) , if for every set $F \subset X$ of at most f vertices, it holds that $\forall u, v \notin F$, $d_{H \setminus F}(u, v) \leq t \cdot d_X(u, v)$. For general metrics, after a long line of work [34, 39, 20, 22, 40, 21, 80], it was shown that every n -vertex graph admits an efficiently constructible f -vertex-fault-tolerant $(2k - 1)$ -spanner with $O(f^{1-1/k} \cdot n^{1+1/k})$ edges, which is optimal assuming the Erdős' Girth Conjecture [44]. For n -points in d dimensional Euclidean space, or more generally in a space of doubling dimension d , f -vertex fault tolerant $(1 + \epsilon)$ -spanner were constructed with $\epsilon^{-O(d)} \cdot f \cdot n$ edges [72, 74, 83].

Kahalon et al. [62] initiated the study of low-hop fault tolerant spanners (previous constructions had $\Omega(\log n)$ hops). An h -hop f -fault tolerant t -spanner H of a metric (X, d_x) is a graph over X such that for every set $F \subseteq X$ of at most f vertices, for every $x, y \notin F$, the spanner without F : $H[X \setminus F]$ contains an h -hop path between x to y of weight at most $t \cdot d_X(x, y)$. The advantages of such a spanner are straightforward, we refer to [62] for a discussion. Kahalon et al. constructed a 2-hop f -fault tolerant spanner for doubling spaces with $n \cdot f^2 \cdot \epsilon^{-O(d)} \cdot \log n$ edges. Note that a linear dependence on f is necessary (as if a point has degree $\leq f$ in H , we can delete all it's neighbors and get distortion ∞). It is natural to ask whether it is possible to construct such a spanner with only a linear dependence, and not quadratic as in [62].

Our Contribution. One can easily construct f -fault tolerant 1-spanner for the path graph with $O(nf)$ edges. We observe that using $O(nf \log n)$ edges, it is possible to obtain f -fault tolerant 2-hop 1-spanner for the path graph (note that $O(n \log n)$ edges are necessary for

■ **Table 4** Summary of old and new results on 2-hop f -fault tolerant spanners. The spanners are for n point metrics, and all report paths with hop bound 2. Here $\epsilon \in (0, 1)$, $k, d \geq 1$ are integers.

Family	Stretch	Edges	Ref
Doubling dimension d	$1 + \epsilon$	$\epsilon^{-O(d)} \cdot f^2 \cdot n \cdot \log n$	[62]
	$1 + \epsilon$	$\epsilon^{-O(d)} \cdot f \cdot n \cdot \log n$	FullV[46]
	t	$2^{-O(d/t)} \cdot f \cdot \tilde{O}(n)$	FullV[46]
General Metric	$4k + \epsilon$	$\tilde{O}(n^{1+\frac{1}{k}} \cdot f \cdot \epsilon^{-1})$	FullV[46]
Euclidean \mathbb{R}^d	$1 + \epsilon$	$O_d(\epsilon^{-d}) \log \frac{1}{\epsilon} \cdot f \cdot n \cdot \log n$	FullV[46]
	$(1 + \epsilon)k$	$e^{\frac{2d}{k^2} \cdot (1 + \frac{8}{k^2})} \cdot \tilde{O}(\frac{d^{1.5}}{\epsilon \cdot k}) \cdot f \cdot n \cdot \log n$	FullV[46]
$\ell_p^d, p \in [1, 2]$	k	$e^{O(\frac{d}{k^p})} \cdot \tilde{O}(d) \cdot f \cdot n \cdot \log n$	FullV[46]
$\ell_p^d, p \in [2, \infty]$	$2 \cdot d^{1-\frac{1}{p}}$	$\tilde{O}(d) \cdot f \cdot n \cdot \log n$	FullV[46]
Treewidth k	2	$O(n \cdot k \cdot f \cdot \log n)$	FullV[46]
Fixed Minor Free	$2 + \epsilon$	$O(\frac{n}{\epsilon} \cdot f \cdot \log^2 n)$	FullV[46]

every 2-hop spanner [6, 68]). Using the various old and new LSO's, we obtain a host of f -fault tolerant 2-hop spanners for various metric spaces. Most notably, for metrics with doubling dimension d , we obtain an f -fault tolerant 2-hop $(1 + \epsilon)$ -spanner with $\epsilon^{-O(d)} \cdot f \cdot n \cdot \log n$ edges, getting the desired linear dependence on f . See Table 4 for a summary of results.

Reliable spanners. A major limitation of fault tolerant spanners is that the number of failures must be determined in advance. In particular, such spanners cannot withstand a massive failure. One can imagine a scenario where a significant portion (even 90%) of a network fails and ceases to function (due to, e.g., close-down during a pandemic), it is important that the remaining parts of the network (or at least most of it) will remain highly connected and functioning. To this end, Bose et al. [26] introduced the notion of a *reliable spanner*. A ν -reliable spanner is a graph such that for every failure set $B \subseteq X$, the residual spanner $H \setminus B$ is a t -spanner for $X \setminus B^+$, where $B^+ \supseteq B$ is a superset of cardinality at most $(1 + \nu) \cdot |B|$. An oblivious ν -reliable t -spanner is a distribution \mathcal{D} over spanners, such that for every failure set B , $H \setminus B$ is a t -spanner for $X \setminus B_H^+$, where the superset B_H^+ depends on both B and the sampled spanner H . The guarantee is that the cardinality of B_H^+ is bounded by $(1 + \nu) \cdot |B|$ in expectation.

ν -Reliable spanners were constructed for d dimensional Euclidean and doubling spaces with $n \cdot \epsilon^{-O(d)} \cdot \tilde{O}(\log n)$ edges [27, 28, 49] by a reduction from (classic) LSO's. Oblivious reliable spanners were constructed also for planar, minor free, treewidth graphs, and general metrics [49] by reductions from triangle, and rooted LSO's (as well as from sparse covers [57]).

Our Contribution. Our newly constructed triangle LSO's for high dimensional Euclidean, ℓ_p spaces, and doubling spaces, directly imply reliable spanners for these spaces, obtaining the first results without exponential dependence on the dimension. See Table 5 for a summary.

Light spanners. An extensively studied parameter is the *lightness* of a spanner, defined as the ratio $w(H)/w(MST(X))$, where $w(H)$ resp. $w(MST(X))$ is the total weight of edges in H resp. a minimum spanning tree (MST) of X . Obtaining spanners with small lightness (and thus total weight) is motivated by applications where edge weights denote e.g. establishing cost. The best possible total weight that can be achieved in order to ensure finite stretch is the weight of an MST, thus making the definition of lightness very natural.

■ **Table 5** Summary of previous and new constructions of ν -reliable spanners.

Family	stretch	guarantee	size	ref
Euclidean ($\mathbb{R}^d, \ \cdot\ _2$)	$1 + \epsilon$	Deterministic	$n \cdot \tilde{O}_d(\epsilon^{-7d}) \nu^{-6} \cdot \tilde{O}(\log n)$	[27]
	$1 + \epsilon$	Oblivious	$n \cdot \tilde{O}_d(\epsilon^{-2d}) \cdot \tilde{O}(\nu^{-1}(\log \log n)^2)$	[28]
	$(1 + \epsilon)t$	Oblivious	$\nu^{-1} \cdot e^{\frac{4d}{t^2} \cdot (1 + \frac{8}{t^2})} \cdot \tilde{O}(n \cdot \frac{d^3}{\epsilon^2 \cdot t^2})$	FullV[46]
ℓ_p^d for $p \in [1, 2]$	t	Oblivious	$\nu^{-1} \cdot e^{O(\frac{d}{t^p})} \cdot \tilde{O}(n \cdot d^2)$	FullV[46]
ℓ_p^d for $p \in [2, \infty]$	$2 \cdot d^{1 - \frac{1}{p}}$	Oblivious	$\nu^{-1} \cdot \tilde{O}(n \cdot d^2)$	FullV[46]
Doubling dimension d	$1 + \epsilon$	Deterministic	$n \cdot \epsilon^{-O(d)} \nu^{-6} \cdot \tilde{O}(\log n)$	[49]
	$1 + \epsilon$	Oblivious	$n \cdot \epsilon^{-O(d)} \nu^{-1} \log \nu^{-1} \cdot \tilde{O}(\log \log n)^2$	[49]
	t	Oblivious	$\tilde{O}(n) \cdot \nu^{-1} \cdot 2^{O(d/t)}$	FullV[46]
General metric	$8t + \epsilon$	Oblivious	$\tilde{O}(n^{1+1/t} \cdot \epsilon^{-2}) \cdot \nu^{-1}$	[49]
Tree	2	Oblivious	$n \cdot O(\nu^{-1} \log^3 n)$	[49]
Treewidth k	2	Oblivious	$n \cdot O(\nu^{-1} k^2 \log^3 n)$	[49]
Planar/Minor-free	$2 + \epsilon$	Oblivious	$n \cdot O(\nu^{-1} \epsilon^{-2} \log^5 n)$	[49]

■ **Table 6** Summary of previous and new results of light spanners for high dimensional metric spaces. Interestingly, for $p \in [1, 2]$ [49] obtain lightness $O(\frac{t^{1+p}}{\log^2 t} \cdot n^{O(\frac{\log^2 t}{t^p})} \cdot \log n)$ regardless of dimension, which is superior to ours for $d \gg \log n$.

Metric space	Stretch	Lightness	Ref
Euclidean space	$O(t)$	$O(n^{\frac{1}{t^2}} \cdot \log n \cdot t)$	[69]
	$O(t)$	$O(e^{\frac{d}{t^2}} \cdot \log^2 n \cdot t)$	[50]
	$(1 + \epsilon)2t$	$e^{\frac{d}{2t^2} \cdot (1 + \frac{2}{t^2})} \cdot \tilde{O}(\frac{d^{1.5}}{\epsilon^2}) \cdot \log n$	FullV[46]
	$(1 + \epsilon)4t$	$e^{\frac{d}{2t^2} \cdot (1 + \frac{2}{t^2})} \cdot \tilde{O}(\frac{d^{1.5}}{\epsilon^2}) \cdot \log^* n$	FullV[46]
Doubling dimension d	$O(t)$	$O(2^{\frac{d}{t}} \cdot t \cdot \log^2 n)$	[50]
	$O(t)$	$2^{O(d/t)} \cdot d \cdot \log^2 t \cdot \log^* n$	FullV[46]
	d	$O(d \cdot \log^2 n)$	[50]
	d	$O(d \cdot \log^2 d \cdot \log^* n)$	FullV[46]
ℓ_p^d for $p \in [1, 2]$	t	$O(\frac{t^{1+p}}{\log^2 t} \cdot n^{O(\frac{\log^2 t}{t^p})} \cdot \log n)$	[50]
	t	$e^{O(\frac{d}{t^p})} \cdot \tilde{O}(d \cdot t) \cdot \log^* n$	FullV[46]
ℓ_p^d for $p \in [2, \infty]$	$4 \cdot d^{1 - \frac{1}{p}}$	$\tilde{O}(d^{2 - \frac{1}{p}}) \cdot \log^* n$	FullV[46]

Obtaining light spanners for general graphs has been the subject of an active line of work [8, 31, 42, 18, 35, 51], where the state of the art is by Le and Solomon [69] who obtained $(1 + \epsilon)(2k - 1)$ spanner with lightness $O(\epsilon^{-1} \cdot n^{\frac{1}{k}})$. Light spanners were also studied extensively in Euclidean spaces (see the book [77]), doubling spaces [53, 51, 25], planar and minor free graphs [63, 65, 24, 67, 37], and high dimensional Euclidean and doubling spaces [56, 50, 69].

Our Contribution. Recently Le and Solomon [69] obtain a general framework for constructing light spanners from spanner oracles. We construct new spanner oracles using LSO's. As a result we derive new light spanners, that improve the state of the art for high dimensional spaces (and match the state of the art for low dimensional doubling spaces). See Table 6 for a summary of results.

1.4 Technical ideas

Triangle LSO for high dimensional Euclidean space. Our construction is very natural: partition the space randomly in every distance scale ξ_i (for some large ξ) into clusters of diameter ξ_i , such that close-by points are likely to be clustered together. In the created ordering σ , points in each cluster will be ordered consecutively and recursively. In particular, the ordering σ will correspond to a laminar partition obtained by the clustering in all possible scales. For a pair of points $x, y \in \mathbb{R}^d$ to be satisfied in the resulting ordering σ , they have to be clustered together in all the distance scales $\xi^i \geq t \cdot \|x - y\|_2$.

Our space partition in each scale is done using ball carving (ala [10]): pick a uniformly random series of centers z_1, z_2, \dots . Each points is assigned to the cluster of the first center at distance at most $R = \frac{1}{2} \cdot \xi^i$. We show that a finite random seed of size $d^{O(d)}$ is enough to sample such a clustering (in all possible distance scales, simultaneously). The probability that two points x, y are clustered together is then equal to the ratio between the volumes of intersection and union of balls: $\Pr[x, y \text{ clustered together}] = \frac{\text{Vol}_d(B(x, R) \cap B(y, R))}{\text{Vol}_d(B(x, R) \cup B(y, R))} \geq \Omega\left(\frac{1}{\sqrt{d}}\right) \cdot \left(1 - \left(\frac{\|x-y\|_2}{R}\right)^2\right)^{d/2}$. We bound this ratio for the case $\|x - y\|_2 \leq \frac{R}{\sqrt{d}}$ using a lemma from [33]. For the general case, we prove that the ratio between these volumes is at least $\Omega\left(\frac{R}{\sqrt{d} \cdot \|p-q\|_2}\right) \cdot \left(1 - \left(\frac{\|p-q\|_2}{R}\right)^2\right)^{\frac{d}{2}}$, slightly improving a similar fact from [9], by a $\frac{R}{\|p-q\|_2}$ factor. This ratio eventually governs our success probability (when replacing $R/\|p-q\|_2$ by twice the stretch $2t$). The improved analysis of the volumes ratio is significant for the $O(\sqrt{d})$ -stretch regime, improving the number of orderings to $\tilde{O}(d)$ (compared with $\tilde{O}(d^{1.5})$ orderings if we were using [9]).

To generalize this construction to ℓ_p spaces, we use the exact same construction, replacing ℓ_2 balls with ℓ_p balls. The volume ratio lemma from [32] for close-by points is replaced by a crude observation without any significant consequences to the resulting number of orderings. For the general case, we directly analyze the ratio of volumes for ℓ_p -balls (our computation is similar to [78]). The rest of the analysis is the same.

Triangle LSO for doubling spaces. Ultrametrics are trees with additional structure, where each ultrametric admits a $(1, 1)$ -triangle LSO. (τ, ρ) -ultrametric cover of a metric space (X, d_X) is a collection \mathcal{U} of τ ultrametrics such that every pair $x, y \in X$ is well approximated by the ultrametrics: $d_X(x, y) \leq \min_{U \in \mathcal{U}} d_U(x, y) \leq \rho \cdot d_X(x, y)$. Filtser and Le [49] showed that (τ, ρ) -ultrametric cover implies (τ, ρ) -triangle LSO. We construct $(2^{O(d/t)} \cdot d \cdot \log^2 t, t)$ -ultrametric cover for spaces with doubling dimension d , implying Theorem 6.

Our starting point for constructing the ultrametric cover is Filtser's [45] padded partition cover, which is a collection of $\approx 2^{O(d/t)}$ space partitions where all clusters are of diameter at most Δ , and every ball of radius $\frac{\Delta}{t}$ is fully contained in a single cluster in one of the partitions. We take a single partition from each distance scales, where the gap between the distance scales is somewhat large: $O\left(\frac{\Delta}{\epsilon}\right)$. Initially these partitions are unrelated, and we "force" them to be laminar, while keeping the padding property. Each such laminar partition induces an ultrametric, and their union is the desired ultrametric cover.

Labeled NNS. Morally, given a (τ, ρ) LSO (or triangle LSO), the NNS label of every point is simply its position in each ordering. Given a query q , we simply find its successor and predecessor in each one of the orderings, one of them is guaranteed to be an approximate nearest neighbor (abbreviated ANN). We can find the successor and predecessor in each ordering in $O(\log \log N)$ time using Y-fast trie [86], it only remains to choose one of the 2τ candidates to be the ANN. To solve this problem we again deploy the LSO structure, and

construct a 2-hop 1-spanner for the implicit path graph induced by each ordering. Specifically, each point will be associated with $O(\log N)$ edges (the name and weight of which will be added to the NNS label), where given two points $x \prec_\sigma y$, in $O(1)$ time we will be able to find a point z such that $x \preceq_\sigma z \preceq_\sigma y$ and x and y stored $\{x, z\}, \{y, z\}$ respectively. Then $d_X(x, z) + d_X(z, y)$ will provide us the desired estimate of $d_X(x, y)$, which will be used to choose the ANN.

The case of rooted LSO is simpler- the label of each point z will consist of its position in all the orderings σ it belongs to, and the distance to the first point x_σ (w.r.t. d_X). Given a query q , for each ordering σ containing q , the leftmost point $y_\sigma \in P$ in the ordering will be a candidate ANN. We will estimate the distance from q to y_σ by $d_X(q, x_\sigma) + d_X(x_\sigma, y_\sigma)$, and return the point with minimum estimation.

For general metrics, the number of orderings is polynomial, $N^{\frac{1}{k}}$ which results in similar NNS label size, and query time (following the approach above). While the NNS label essentially cannot be improved, we can significantly reduce the query time. Our solution is to use Ramsey trees [19, 76, 17, 2], which are a collection of embeddings into ultrametrics \mathcal{U} such that each point x has a single home ultrametric $U_x \in \mathcal{U}$ which well approximate all the distances to x . We thus reduce the labeled NNS problem to ultrametrics, where it can be efficiently solved. For the case of approximation factor $O(\log N)$ the required number of ultrametrics is $O(\log N)$, which leads us to label size $O(\log^2 N)$. To reduce it even farther, we use the novel clan embedding [48], where instead of embedding the space X into a collection of ultrametrics, we embed it into a single ultrametric (but where each point might have several copies). This allows us to reduce the label size to $O(\log N)$ (in expectation), and with one additional easing assumption (either polynomial aspect ratio or small failure probability) to even $O(1)$ label size.

Path reporting low hop spanners. A (τ, ρ) -tree cover is similar to ultrametric cover discussed above, where the ultrametrics are replaced by trees. Kahalon et al. [62] first constructed path reporting low hop spanner for a tree metric, and then for each metric space of interest, they considered it's tree cover, and constructed a path reporting low hop spanner for each tree in the cover. The spanner for the global metric is obtained by taking the union of all these spanners constructed for the trees in the cover. To report a queried distance, they simply computed the paths in all the trees, and returned the shortest observed path.

Thus Kahalon et al. idea is to reduce the problem to the fairly simple case of tree metrics. We reduce each metric space into the even simpler case of paths using LSO. Given an LSO (or triangle LSO) we simply construct a path reporting 2-hop path for each path associated with an ordering, and similarly to [62], check all the path spanners and return the shortest observed path. The resulting query time has linear dependence on the number of orderings. The case of rooted LSO is simpler, where it is enough to add a single edge per ordering, to the leftmost point in the ordering.

Next we present some improvements to the query. First, for the case of Euclidean space (low dimensional), we observe that given two points x, y , the ordering satisfying them could be computed in $O_d(1)$ time, implying that we don't need to check all the orderings, and return a 2 hop path in $O_d(1)$ time. Next, for the case of planar graphs, using the structure of cycle separators (which are used to construct the rooted LSO), in $O(1)$ time one can narrow the number of potential orderings to $O(\epsilon^{-1})$, implying $O(\epsilon^{-1})$ query time. For general graphs we observe that the celebrated Thorup Zwick distance oracle [85] can be used to produce a path reporting 2-hop $(2k - 1)$ -spanner with $O(n^{1+\frac{1}{k}} \cdot k)$ edges and $O(k)$ query time. Finally, we use sparse covers [15] to obtain an exponential improvement in the query time, while incurring a factor 2 increase in the stretch.

Fault tolerant spanners. The 2-hop f -fault tolerant spanner for doubling metrics by Kahalon et al. [62] is based on a quite sophisticated tool of robust tree cover. We have a superior, and an extremely simple construction. First we observe that the path graph has a 2-hop f -fault tolerant 1-spanner with $O(nf \log n)$ edges. Indeed, add edges from all the vertices to the middle $f + 1$ vertices, delete the middle vertices and recurse on each side. We then apply this construction on each of the path graphs induced by the LSO (or triangle LSO) to obtain our results. The case of rooted LSO is even simpler: for every path it is enough to add all the edges to the first $f + 1$ points.

References

- 1 Ittai Abraham, Yair Bartal, and Ofer Neiman. Advances in metric embedding theory. *Advances in Mathematics*, 228(6):3026–3126, 2011. doi:10.1016/j.aim.2011.08.003.
- 2 Ittai Abraham, Shiri Chechik, Michael Elkin, Arnold Filtser, and Ofer Neiman. Ramsey spanning trees and their applications. *ACM Trans. Algorithms*, 16(2):19:1–19:21, 2020. preliminary version published in SODA 2018. doi:10.1145/3371039.
- 3 Ittai Abraham, Shiri Chechik, Robert Krauthgamer, and Udi Wieder. Approximate nearest neighbor search in metrics of planar graphs. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24–26, 2015, Princeton, NJ, USA*, volume 40 of *LIPICs*, pages 20–42. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPICs.APPROX-RANDOM.2015.20.
- 4 Abu Reyan Ahmed, Greg Bodwin, Faryad Darabi Sahneh, Keaton Hamm, Mohammad Javad Latifi Jebelli, Stephen G. Kobourov, and Richard Spence. Graph spanners: A tutorial review. *Comput. Sci. Rev.*, 37:100253, 2020. doi:10.1016/j.cosrev.2020.100253.
- 5 Ibrahim Akgün and Barbaros Ç. Tansel. New formulations of the hop-constrained minimum spanning tree problem via miller-tucker-zemlin constraints. *Eur. J. Oper. Res.*, 212(2):263–276, 2011. doi:10.1016/j.ejor.2011.01.051.
- 6 N. Alon and B. Schieber. Optimal preprocessing for answering on-line product queries. Technical report, Tel-Aviv University, 1987.
- 7 B. Alspach. The wonderful Walecki construction. *Bull. Inst. Combin. Appl.*, 52:7–20, 2008. see here.
- 8 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discret. Comput. Geom.*, 9:81–100, 1993. doi:10.1007/BF02189308.
- 9 Alexandr Andoni. *Nearest neighbor search: the old, the new, and the impossible*. PhD thesis, Massachusetts Institute of Technology, 2009. see here.
- 10 Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008. Preliminary version published in FOCS 2006. doi:10.1145/1327452.1327494.
- 11 Alexandr Andoni, Piotr Indyk, and Ilya P. Razenshteyn. Approximate nearest neighbor search in high dimensions. *CoRR*, abs/1806.09823, 2018. arXiv:1806.09823.
- 12 Alexandr Andoni, Huy L. Nguyen, Aleksandar Nikolov, Ilya P. Razenshteyn, and Erik Waingarten. Approximate near neighbors for general symmetric norms. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19–23, 2017*, pages 902–913. ACM, 2017. doi:10.1145/3055399.3055418.
- 13 Alexandr Andoni, Aleksandar Nikolov, Ilya P. Razenshteyn, and Erik Waingarten. Approximate nearest neighbors beyond space partitions. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 1171–1190. SIAM, 2021. doi:10.1137/1.9781611976465.72.
- 14 Alexandr Andoni, Clifford Stein, and Peilin Zhong. Parallel approximate undirected shortest paths via low hop emulators. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani,

- Gautam Kamath, and Julia Chuzhoy, editors, *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 322–335. ACM, 2020. doi:10.1145/3357713.3384321.
- 15 Baruch Awerbuch and David Peleg. Sparse partitions. In *Proceedings of the 31st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 503–513, 1990. doi:10.1109/FSCS.1990.89571.
 - 16 Anantaram Balakrishnan and Kemal Altinkemer. Using a hop-constrained model to generate alternative communication network design. *INFORMS J. Comput.*, 4(2):192–205, 1992. doi:10.1287/ijoc.4.2.192.
 - 17 Yair Bartal. Advances in metric ramsey theory and its applications. *CoRR*, abs/2104.03484, 2021. arXiv:2104.03484.
 - 18 Yair Bartal, Arnold Filtser, and Ofer Neiman. On notions of distortion and an almost minimum spanning tree with constant average distortion. *J. Comput. Syst. Sci.*, 105:116–129, 2019. preliminary version published in SODA 2016. doi:10.1016/j.jcss.2019.04.006.
 - 19 Yair Bartal, Nathan Linial, Manor Mendel, and Assaf Naor. Some low distortion metric ramsey problems. *Discret. Comput. Geom.*, 33(1):27–41, 2005. doi:10.1007/s00454-004-1100-z.
 - 20 Greg Bodwin, Michael Dinitz, Merav Parter, and Virginia Vassilevska Williams. Optimal vertex fault tolerant spanners (for fixed stretch). In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 1884–1900, 2018. doi:10.1137/1.9781611975031.123.
 - 21 Greg Bodwin, Michael Dinitz, and Caleb Robelle. Optimal vertex fault-tolerant spanners in polynomial time. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2924–2938. SIAM, 2021. doi:10.1137/1.9781611976465.174.
 - 22 Greg Bodwin and Shyamal Patel. A trivial yet optimal solution to vertex fault tolerant spanners. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*, pages 541–543, 2019. doi:10.1145/3293611.3331588.
 - 23 Jérôme De Boeck and Bernard Fortz. Extended formulation for hop constrained distribution network configuration problems. *Eur. J. Oper. Res.*, 265(2):488–502, 2018. doi:10.1016/j.ejor.2017.08.017.
 - 24 G. Borradaile, H. Le, and C. Wulff-Nilsen. Minor-free graphs have light spanners. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science, FOCS '17*, pages 767–778, 2017. doi:10.1109/FOCS.2017.76.
 - 25 G. Borradaile, H. Le, and C. Wulff-Nilsen. Greedy spanners are optimal in doubling metrics. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '19*, pages 2371–2379, 2019. doi:10.1137/1.9781611975482.145.
 - 26 Prosenjit Bose, Vida Dujmovic, Pat Morin, and Michiel H. M. Smid. Robust geometric spanners. *SIAM J. Comput.*, 42(4):1720–1736, 2013. preliminary version published in SOCG 2013. doi:10.1137/120874473.
 - 27 Kevin Buchin, Sariel Har-Peled, and Dániel Oláh. A spanner for the day after. In *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, pages 19:1–19:15, 2019. doi:10.4230/LIPIcs.SoCG.2019.19.
 - 28 Kevin Buchin, Sariel Har-Peled, and Dániel Oláh. Sometimes reliable spanners of almost linear size. In *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, pages 27:1–27:15, 2020. doi:10.4230/LIPIcs.ESA.2020.27.
 - 29 T.-H. Hubert Chan and Anupam Gupta. Approximating TSP on metrics with bounded global growth. *SIAM J. Comput.*, 41(3):587–617, 2012. preliminary version published in SODA 2008. doi:10.1137/090749396.
 - 30 Timothy M. Chan, Sariel Har-Peled, and Mitchell Jones. On locality-sensitive orderings and their applications. *SIAM J. Comput.*, 49(3):583–600, 2020. preliminary version published in ITCS 2019. doi:10.1137/19M1246493.

- 31 Barun Chandra, Gautam Das, Giri Narasimhan, and José Soares. New sparseness results on graph spanners. *Int. J. Comput. Geom. Appl.*, 5:125–144, 1995. preliminary version published in SOCG 1992. doi:10.1142/S0218195995000088.
- 32 Moses Charikar, Chandra Chekuri, Ashish Goel, and Sudipto Guha. Rounding via trees: deterministic approximation algorithms for group steiner trees and k-median. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 114–123, New York, NY, USA, 1998. ACM Press. doi:10.1145/276698.276719.
- 33 Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha, and Serge A. Plotkin. Approximating a finite metric by a small number of tree metrics. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*, pages 379–388. IEEE Computer Society, 1998. doi:10.1109/SFCS.1998.743488.
- 34 Shiri Chechik, Michael Langberg, David Peleg, and Liam Roditty. Fault tolerant spanners for general graphs. *SIAM J. Comput.*, 39(7):3403–3423, 2010. preliminary version published in STOC 2009. doi:10.1137/090758039.
- 35 Shiri Chechik and Christian Wulff-Nilsen. Near-optimal light spanners. *ACM Trans. Algorithms*, 14(3):33:1–33:15, 2018. preliminary version published in SODA 2016. doi:10.1145/3199607.
- 36 Edith Cohen. Polylog-time and near-linear work approximation scheme for undirected shortest paths. *J. ACM*, 47(1):132–166, 2000. preliminary version published in STOC 1994. doi:10.1145/331605.331610.
- 37 Vincent Cohen-Addad, Arnold Filtser, Philip N. Klein, and Hung Le. On light spanners, low-treewidth embeddings and efficient traversing in minor-free graphs. *CoRR*, abs/2009.05039, 2020. To appear in FOCS 2020, <https://arxiv.org/abs/2009.05039>. arXiv:2009.05039.
- 38 Gautam Das, Paul J. Heffernan, and Giri Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In Chee Yap, editor, *Proceedings of the Ninth Annual Symposium on Computational Geometry San Diego, CA, USA, May 19-21, 1993*, pages 53–62. ACM, 1993. doi:10.1145/160985.160998.
- 39 Michael Dinitz and Robert Krauthgamer. Fault-tolerant spanners: better and simpler. In *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, June 6-8, 2011*, pages 169–178, 2011. doi:10.1145/1993806.1993830.
- 40 Michael Dinitz and Caleb Robelle. Efficient and simple algorithms for fault-tolerant spanners. In *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*, pages 493–500, 2020. doi:10.1145/3382734.3405735.
- 41 Anne Driemel and Francesco Silvestri. Locality-Sensitive Hashing of Curves. In *Proceedings of the 33rd International Symposium on Computational Geometry*, volume 77, pages 37:1–37:16, Brisbane, Australia, July 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2017.37.
- 42 Michael Elkin, Ofer Neiman, and Shay Solomon. Light spanners. *SIAM J. Discret. Math.*, 29(3):1312–1321, 2015. doi:10.1137/140979538.
- 43 Ioannis Z. Emiris and Ioannis Psarros. Products of euclidean metrics and applications to proximity questions among curves. In Bettina Speckmann and Csaba D. Tóth, editors, *34th International Symposium on Computational Geometry, SoCG 2018, June 11-14, 2018, Budapest, Hungary*, volume 99 of *LIPIcs*, pages 37:1–37:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.SoCG.2018.37.
- 44 P. Erdős. Extremal problems in graph theory. *Theory of Graphs and Its Applications (Proc. Sympos. Smolenice)*, pages 29–36, 1964. see here.
- 45 Arnold Filtser. On strong diameter padded decompositions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA*, pages 6:1–6:21, 2019. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.6.
- 46 Arnold Filtser. Labeled nearest neighbor search and metric spanners via locality sensitive orderings. *CoRR*, abs/2211.11846, 2022. doi:10.48550/arXiv.2211.11846.

- 47 Arnold Filtser, Omrit Filtser, and Matthew J. Katz. Approximate nearest neighbor for curves - simple, efficient, and deterministic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPICs*, pages 48:1–48:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICs.ICALP.2020.48.
- 48 Arnold Filtser and Hung Le. Clan embeddings into trees, and low treewidth graphs. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 342–355. ACM, 2021. doi:10.1145/3406325.3451043.
- 49 Arnold Filtser and Hung Le. Locality-sensitive orderings and applications to reliable spanners. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1066–1079. ACM, 2022. doi:10.1145/3519935.3520042.
- 50 Arnold Filtser and Ofer Neiman. Light spanners for high dimensional norms via stochastic decompositions. *Algorithmica*, 2022. doi:10.1007/s00453-022-00994-0.
- 51 Arnold Filtser and Shay Solomon. The greedy spanner is existentially optimal. *SIAM J. Comput.*, 49(2):429–447, 2020. preliminary version published in PODC 2016. doi:10.1137/18M1210678.
- 52 J. Gao, L. J. Guibas, and A. Nguyen. Deformable spanners and applications. *Computational Geometry*, 35(1):2–19, 2006. doi:10.1016/j.comgeo.2005.10.001.
- 53 Lee-Ad Gottlieb. A light metric spanner. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 759–772, 2015. doi:10.1109/FOCS.2015.52.
- 54 Luis Eduardo Neves Gouveia and Thomas L. Magnanti. Network flow models for designing diameter-constrained minimum-spanning and steiner trees. *Networks*, 41(3):159–173, 2003. doi:10.1002/net.10069.
- 55 Luis Eduardo Neves Gouveia, Pedro Patrício, Amaro de Sousa, and Rui Valadas. MPLS over WDM network design with packet level qos constraints based on ILP models. In *Proceedings IEEE INFOCOM 2003, The 22nd Annual Joint Conference of the IEEE Computer and Communications Societies, San Francisco, CA, USA, March 30 - April 3, 2003*, pages 576–586, 2003. doi:10.1109/INFCOM.2003.1208708.
- 56 Sariel Har-Peled, Piotr Indyk, and Anastasios Sidiropoulos. Euclidean spanners in high dimensions. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 804–809, 2013. doi:10.1137/1.9781611973105.57.
- 57 Sariel Har-Peled, Manor Mendel, and Dániel Oláh. Reliable Spanners for Metric Spaces. In *37th International Symposium on Computational Geometry, SoCG'21*, pages 43:1–43:13, 2021. Full version at <https://arxiv.org/abs/2007.08738>. doi:10.4230/LIPICs.SoCG.2021.43.
- 58 Piotr Indyk. On approximate nearest neighbors under ℓ_∞ norm. *J. Comput. Syst. Sci.*, 63(4):627–638, 2001. doi:10.1006/jcss.2001.1781.
- 59 Piotr Indyk. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. In *Proceedings of the 8th Symposium on Computational Geometry*, pages 102–106, Barcelona, Spain, June 2002. ACM Press. doi:10.1145/513400.513414.
- 60 Piotr Indyk and Nitin Thaper. Fast image retrieval via embeddings. In *3rd international workshop on statistical and computational theories of vision*, volume 2(3), page 5. Nice, France, 2003. see here.
- 61 William Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984. see here.
- 62 Omri Kahalon, Hung Le, Lazar Milenkovic, and Shay Solomon. Can't see the forest for the trees: Navigating metric spaces by bounded hop-diameter spanners. In Alessia Milani and

- Philipp Woelfel, editors, *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*, pages 151–162. ACM, 2022. doi:10.1145/3519270.3538414.
- 63 P. N. Klein. Subset spanner for planar graphs, with application to subset TSP. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, STOC '06*, pages 749–756, 2006. doi:10.1145/1132516.1132620.
- 64 Philip N. Klein. Preprocessing an undirected planar network to enable fast approximate distance queries. In *Proceedings of the Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 6-8, 2002, San Francisco, CA, USA*, pages 820–827, 2002. see here. URL: <http://dl.acm.org/citation.cfm?id=545381.545488>.
- 65 Philip N. Klein. A linear-time approximation scheme for TSP in undirected planar graphs with edge-weights. *SIAM J. Comput.*, 37(6):1926–1952, 2008. doi:10.1137/060649562.
- 66 Robert Krauthgamer and James R. Lee. The black-box complexity of nearest-neighbor search. *Theor. Comput. Sci.*, 348(2-3):262–276, 2005. doi:10.1016/j.tcs.2005.09.017.
- 67 Hung Le. A PTAS for subset TSP in minor-free graphs. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2279–2298, 2020. doi:10.1137/1.9781611975994.140.
- 68 Hung Le, Lazar Milenkovic, and Shay Solomon. Sparse euclidean spanners with tiny diameter: A tight lower bound. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPICs*, pages 54:1–54:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.SoCG.2022.54.
- 69 Hung Le and Shay Solomon. A unified framework of light spanners II: fine-grained optimality. *CoRR*, abs/2111.13748, 2021. arXiv:2111.13748.
- 70 Hung Le and Christian Wulff-Nilsen. Optimal approximate distance oracle for planar graphs. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 363–374. IEEE, 2021. doi:10.1109/FOCS52979.2021.00044.
- 71 Larry J. LeBlanc, Jerome Chifflet, and Philippe Mahey. Packet routing in telecommunication networks with path and flow restrictions. *INFORMS J. Comput.*, 11(2):188–197, 1999. doi:10.1287/ijoc.11.2.188.
- 72 Christos Levcopoulos, Giri Narasimhan, and Michiel H. M. Smid. Improved algorithms for constructing fault-tolerant spanners. *Algorithmica*, 32(1):144–156, 2002. preliminary version published in STOC 1998. doi:10.1007/s00453-001-0075-x.
- 73 Yaowei Long and Seth Pettie. Planar distance oracles with better time-space tradeoffs. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2517–2537. SIAM, 2021. doi:10.1137/1.9781611976465.149.
- 74 Tamás Lukovszki. New results on fault tolerant geometric spanners. In Frank Dehne, Jörg-Rüdiger Sack, Arvind Gupta, and Roberto Tamassia, editors, *Algorithms and Data Structures*, pages 193–204, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg. doi:10.1007/3-540-48447-7_20.
- 75 Jiří Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel Journal of Mathematics*, 93(1):333–344, 1996. doi:10.1007/BF02761110.
- 76 Manor Mendel and Assaf Naor. Ramsey partitions and proximity data structures. *Journal of the European Mathematical Society*, 9(2):253–275, 2007.
- 77 Giri Narasimhan and Michiel H. M. Smid. *Geometric spanner networks*. Cambridge University Press, 2007. doi:10.1017/CB09780511546884.
- 78 Huy L. Nguyen. Approximate nearest neighbor search in ℓ_p . *CoRR*, abs/1306.3601, 2013. arXiv:1306.3601.
- 79 Rafail Ostrovsky and Yuval Rabani. Low distortion embeddings for edit distance. *J. ACM*, 54(5):23, 2007. doi:10.1145/1284320.1284322.
- 80 Merav Parter. Nearly optimal vertex fault-tolerant spanners in optimal time: sequential, distributed, and parallel. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th*

Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022, pages 1080–1092. ACM, 2022. doi:10.1145/3519935.3520047.

- 81 Hasan Pirkul and Samit Soni. New formulations and solution procedures for the hop constrained network design problem. *Eur. J. Oper. Res.*, 148(1):126–140, 2003. doi:10.1016/S0377-2217(02)00366-1.
- 82 André Rossi, Alexis Aubry, and Mireille Jacomino. Connectivity-and-hop-constrained design of electricity distribution networks. *Eur. J. Oper. Res.*, 218(1):48–57, 2012. doi:10.1016/j.ejor.2011.10.006.
- 83 Shay Solomon. From hierarchical partitions to hierarchical covers: optimal fault-tolerant spanners for doubling metrics. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 363–372, 2014. doi:10.1145/2591796.2591864.
- 84 M. Thorup. Compact oracles for reachability and approximate distances in planar digraphs. *Journal of the ACM*, 51(6):993–1024, 2004. doi:10.1145/1039488.1039493.
- 85 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *J. ACM*, 52(1):1–24, 2005. preliminary version published in STOC 2001. doi:10.1145/1044731.1044732.
- 86 Dan E. Willard. Log-logarithmic worst-case range queries are possible in space $\theta(n)$. *Inf. Process. Lett.*, 17(2):81–84, 1983. doi:10.1016/0020-0190(83)90075-3.
- 87 Kathleen A. Woolston and Susan L. Albin. The design of centralized networks with reliability and availability constraints. *Comput. Oper. Res.*, 15(3):207–217, 1988. doi:10.1016/0305-0548(88)90033-0.

Polynomial-Time Approximation Schemes for Independent Packing Problems on Fractionally Tree-Independence-Number-Fragile Graphs

Esther Galby ✉

Institute for Algorithms and Complexity, Technische Universität Hamburg, Germany

Andrea Munaro ✉ 

Department of Mathematical, Physical and Computer Sciences, University of Parma, Italy

Shizhou Yang ✉

School of Mathematics and Physics, Queen's University Belfast, UK

Abstract

We investigate a relaxation of the notion of treewidth-fragility, namely tree-independence-number-fragility. In particular, we obtain polynomial-time approximation schemes for independent packing problems on fractionally tree-independence-number-fragile graph classes. Our approach unifies and extends several known polynomial-time approximation schemes on seemingly unrelated graph classes, such as classes of intersection graphs of fat objects in a fixed dimension or proper minor-closed classes. We also study the related notion of layered tree-independence number, a relaxation of layered treewidth.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases Independent packings, intersection graphs, polynomial-time approximation schemes, tree-independence number

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.34

Related Version *Full Version*: <https://arxiv.org/abs/2303.07444>

Acknowledgements We thank the anonymous reviewers for valuable comments.

1 Introduction

Many optimization problems involving collections of geometric objects in the d -dimensional space are known to admit a polynomial-time approximation scheme (PTAS). Arguably the earliest example of such behavior is the problem of finding the maximum number of pairwise non-intersecting disks or squares in a collection of unit disks or unit squares, respectively [38]. Such subcollection is usually called an *independent packing*. This result was later extended to collections of arbitrary disks and squares and, more generally, fat objects [11, 30]. The reason for the abundance of approximation schemes for geometric problems is that shifting and layering techniques can be used to reduce the problem to small subproblems that can be solved by dynamic programming. In fact, the same phenomenon occurs for graph problems, as evidenced by the seminal work of Baker [4] on approximation schemes for local problems, such as INDEPENDENT SET, on planar graphs and its generalizations first to apex-minor-free graphs [29] and further to graphs embeddable on a surface of bounded genus with a bounded number of crossings per edge [37]. The notion of intersection graph allows to jump from the geometric world to the graph-theoretic one. Given a collection \mathcal{O} of geometric objects in \mathbb{R}^d , we can consider its *intersection graph*, the graph whose vertices are the objects in \mathcal{O} and where two vertices $O_i, O_j \in \mathcal{O}$ are adjacent if and only if $O_i \cap O_j \neq \emptyset$. An independent packing in \mathcal{O} is then nothing but an independent set in the corresponding intersection graph. Notice that intersection graphs of unit disks or squares are not minor-closed, as they contain arbitrarily large cliques. Our motivating question is the following:



© Esther Galby, Andrea Munaro, and Shizhou Yang;

licensed under Creative Commons License CC-BY 4.0

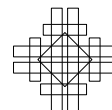
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 34; pp. 34:1–34:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Is there any underlying graph-theoretical reason for the existence of the seemingly unrelated PTASes for INDEPENDENT SET mentioned above?

We provide a positive answer to this question that also allows us to further generalize to a family of independent packing problems. The similar question of whether there is a general notion under which PTASes using Baker's technique can be obtained was asked in [37].

Baker's layering technique relies on a form of decomposition theorem for planar graphs that can be roughly summarized as follows. Given a planar graph G and $k \in \mathbb{N}$, the vertex set of G can be partitioned into $k + 1$ possibly empty sets in such a way that deleting any part induces a graph of treewidth at most $O(k)$ in G . Moreover, such a partition together with tree decompositions of width at most $O(k)$ of the respective graphs can be found in polynomial time. A statement of this form is typically referred to as a *Vertex Decomposition Theorem* (VDT) [48]. VDTs are known to exist in planar graphs [4], graphs of bounded-genus and apex-minor-free graphs [29], and H -minor-free graphs [17, 19]. However, their existence is in general something too strong to ask for, as is the case of intersection graphs of unit disks or squares and hence fat objects in general. There are then two natural ways in which one can try to relax the notion of VDT. First, we can consider an approximate partition of the vertex set, where a vertex can belong to some constant number of sets. Second, we can look for a width parameter less restrictive than treewidth.

Dvořák [24] pursued the first direction and introduced the notion of *efficient fractional treewidth-fragility*. We state here an equivalent formulation from [28]. A class of graphs \mathcal{G} is *efficiently fractionally treewidth-fragile* if there exists a function $f: \mathbb{N} \rightarrow \mathbb{N}$ and an algorithm that, for every $k \in \mathbb{N}$ and $G \in \mathcal{G}$, returns in time $\text{poly}(|V(G)|)$ a collection of subsets $X_1, X_2, \dots, X_m \subseteq V(G)$ such that each vertex of G belongs to at most m/k of the subsets and moreover, for $i = 1, \dots, m$, the algorithm also returns a tree decomposition of $G - X_i$ of width at most $f(k)$. Several graph classes are known to be efficiently fractionally treewidth-fragile. In fact, a hereditary class \mathcal{G} is efficiently fractionally treewidth-fragile in each of the following cases (see, e.g., [28]): \mathcal{G} has sublinear separators and bounded maximum degree, \mathcal{G} is proper minor-closed, or \mathcal{G} consists of intersection graphs of convex objects with bounded aspect ratio in \mathbb{R}^d (for fixed d) and the graphs in \mathcal{G} have bounded clique number. Dvořák [24] showed that INDEPENDENT SET admits a PTAS on every efficiently fractionally treewidth-fragile graph class. This result was later extended [26, 28] to a framework of maximization problems including, for example, MAX WEIGHT DISTANCE- d INDEPENDENT SET, MAX WEIGHT INDUCED FOREST and MAX WEIGHT INDUCED MATCHING. However, the notion of fractional treewidth-fragility falls short of capturing classes such as unit disk graphs, as it implies the existence of sublinear separators [24].

One can then try to pursue the second direction mentioned above and further relax the notion of efficient fractional fragility by considering width parameters *more powerful* than treewidth (i.e., bounded on a larger class of graphs) and algorithmically useful. A natural candidate is the recently introduced *tree-independence number* [15], a width parameter defined in terms of tree decompositions which is more powerful than treewidth (see Section 3). Several algorithmic applications of boundedness of tree-independence number have been provided, most notably polynomial-time solvability of MAX WEIGHT INDEPENDENT PACKING [15] (see Section 5 for the definition), a common generalization of MAX WEIGHT INDEPENDENT SET and MAX WEIGHT INDUCED MATCHING, and of its distance- d version, for d even [45]. Investigating the notion of efficient fractional tree-independence-number-fragility (tree- α -fragility for short) was recently suggested in a talk by Dvořák [25], where it was stated that, using an argument from [27], it is possible to show that intersection graphs of balls and cubes in \mathbb{R}^d are fractionally tree- α -fragile.

A successful notion related to fractional treewidth-fragility is the layered treewidth of a graph [21]. Despite currently lacking any direct algorithmic application, it proved useful especially in the context of coloring problems (we refer to [22] for additional references). We just mention that classes of bounded layered treewidth include planar graphs and, more generally, apex-minor-free graphs and graphs embeddable on a surface of bounded genus with a bounded number of crossings per edge, amongst others [20]. It can be shown that bounded layered treewidth implies fractional treewidth-fragility (see Section 4). Layered treewidth is also related to local treewidth, a notion first introduced by Eppstein [29], and in fact, on proper minor-closed classes, having bounded layered treewidth coincides with having bounded local treewidth (see, e.g., [20]).

1.1 Our results

In this paper, we investigate the notion of efficient fractional tree- α -fragility and show that it answers our motivating question in the positive, thus allowing to unify and extend several known results. Our main result can be summarized as follows and will be proved in Section 4 and Section 5.

► **Theorem 1.** *MAX WEIGHT INDEPENDENT PACKING admits a PTAS on every efficiently fractionally tree- α -fragile class. Moreover, the class of intersection graphs of fat objects in \mathbb{R}^d , for fixed d , is efficiently fractionally tree- α -fragile¹.*

The message of Theorem 1 is that a doubly-relaxed version of a VDT suffices for algorithmic applications and is general enough to hold for several interesting graph classes. Theorem 1 cannot be improved to guarantee an EPTAS, unless $\text{FPT} = \text{W}[1]$. Indeed, Marx [42] showed that INDEPENDENT SET remains $\text{W}[1]$ -complete on intersection graphs of unit disks and unit squares. The natural trade-off in extending the tractable families with respect to approximation is that fewer problems will admit a PTAS. In our case this is exemplified by the minimization problem FEEDBACK VERTEX SET, which admits no PTAS, unless $\text{P} = \text{NP}$, on unit ball graphs in \mathbb{R}^3 [32] but admits an EPTAS on disk graphs in \mathbb{R}^2 [41].

In Section 4, we also show that fractionally tree- α -fragile classes have bounded biclique number, where the *biclique number* of a graph G is the maximum $n \in \mathbb{N}$ such that the complete bipartite graph $K_{n,n}$ is an induced subgraph of G . This shows in particular that, unsurprisingly, the notion of fractional tree- α -fragility falls short of capturing intersection graphs of rectangles in the plane. Whether INDEPENDENT SET admits a PTAS on these graphs remains one of the major open problems in the area (see, e.g., [34]). We also show that the absence of large bicliques is not sufficient for guaranteeing fractional tree- α -fragility: n -dimensional grids of width n are $K_{2,3}$ -free but not fractionally tree- α -fragile.

We begin our study of fractional tree- α -fragility by introducing, in Section 3, a subclass of fractionally tree- α -fragile graphs, namely the class of graphs with bounded layered tree-independence number. We obtain the notion of *layered tree-independence number* by relaxing the successful notion of layered treewidth and show that, besides graphs of bounded layered treewidth, classes of intersection graphs of unit disks in \mathbb{R}^2 and of paths with bounded horizontal part² on a grid have bounded layered tree-independence number. Moreover, we observe that, for minor-closed classes, having bounded layered tree-independence number is equivalent to having bounded layered treewidth, thus extending a characterization of bounded layered treewidth from [20].

¹ Here we use a definition of fatness slightly generalizing that of Chan [11] (see Section 4.1).

² The *horizontal part* of a path is the interval corresponding to the projection of the path onto the x -axis.

We then consider the behavior of layered tree-independence number with respect to graph powers. We show that odd powers of graphs of bounded layered tree-independence number have bounded layered tree-independence number and that this does not extend to even powers. Combined with Theorem 1, this gives the following result which applies, for example, to unit disk graphs and cannot be extended to odd $d \in \mathbb{N}$ (see Section 5.2).

► **Theorem 2.** *For a fixed positive even integer d , the distance- d version of MAX WEIGHT INDEPENDENT PACKING admits a PTAS on every class of bounded layered tree-independence number, provided that a tree decomposition and a layering witnessing small layered tree-independence number can be computed efficiently.*

Finally, we show that the approach to PTASes through tree-independence number is competitive in terms of running time for some classes of intersection graphs. Specifically, in Section 5.3, we obtain PTASes for MAX WEIGHT INDEPENDENT SET for intersection graphs of families of unit disks, unit-height rectangles, and paths with bounded horizontal part on a grid, which improve or generalize results from [6, 12, 43] mentioned in the next section.

We believe that the notion of fractional tree- α -fragility can find further applications in the design of PTASes. In fact, it would be interesting to obtain an algorithmic meta-theorem similar to those for fractionally treewidth-fragile classes [28, 26] and classes of bounded tree-independence number [45]. Although our interest is in approximation schemes, we notice en passant that the observations from Section 3 lead to a subexponential-time algorithm for the distance- d version of MAX WEIGHT INDEPENDENT PACKING, for d even, on unit disk graphs. We finally remark that all our PTASes for intersection graphs of geometric objects are not robust i.e., they require a geometric realization to be part of the input.

1.2 Other related work

Disk graphs. Very recently, Lokshtanov et al. [41] established a framework for designing EPTASes for a broad class of minimization problems (specifically, vertex-deletion problems) on disk graphs including, among others, FEEDBACK VERTEX SET and d -BOUNDED DEGREE VERTEX DELETION. Previous sporadic PTASes on this class were known only for VERTEX COVER [30, 50], DOMINATING SET [35], INDEPENDENT SET [11, 30] and MAX CLIQUE [8]. Theorem 1 adds several maximization problems to this list (see Section 5).

Unit disk graphs. Unit disk graphs are arguably one of the most well-studied graph classes in computational geometry, as they naturally model several real-world problems. Great attention has been devoted to approximation algorithms for MAX WEIGHT INDEPENDENT SET on this class (see, e.g., [39, 46, 49]). To the best of our knowledge, the fastest known PTAS is a $(1 - 1/k)$ -approximation algorithm with running time $O(kn^{4\lceil \frac{2(k-1)}{\sqrt{3}} \rceil})$ [43]. We also remark that a special type of Decomposition Theorem was recently shown to hold for the class of unit disk graphs. A Contraction Decomposition Theorem (CDT) is a statement of the following form: given a graph G , for any $p \in \mathbb{N}$, one can partition the edge set of G into E_1, \dots, E_p such that contracting the edges in each E_i in G yields a graph of treewidth at most $f(p)$, for some function $f: \mathbb{N} \rightarrow \mathbb{N}$. CDTs are useful in designing efficient approximation and parameterized algorithms and are known to hold for classes such as graphs of bounded genus [18] and unit disk graphs [5]. Since these classes are efficiently fractionally tree- α -fragile, our results can be seen as providing a different type of relaxed decomposition theorems for them.

Intersection graphs of unit-height rectangles. As observed by Agarwal et al. [1], this class of graphs arises naturally as a model for the problem of labeling maps with labels of the same font size. Improving on [38], they obtained a $(1 - 1/k)$ -approximation algorithm for MAX WEIGHT INDEPENDENT SET on this class with running time $O(n^{2k-1})$. Chan [12] provided a $(1 - 1/k)$ -approximation algorithm with running time $O(n^k)$.

Intersection graphs of paths on a grid. Asinowski et al. [3] introduced the class of *Vertex intersection graphs of Paths on a Grid* (*VPG graphs* for short). A graph G is a *VPG graph* if there exists a collection \mathcal{P} of paths on a grid \mathcal{G} such that \mathcal{P} is in one-to-one correspondence with $V(G)$ and two vertices are adjacent in G if and only if the corresponding paths intersect. It is not difficult to see that this class coincides with the well-known class of string graphs. If every path in \mathcal{P} has at most k bends i.e., 90 degrees turns at a grid-point, the graph is a B_k -*VPG graph*. Golumbic et al. [36] introduced the class of *Edge intersection graphs of Paths on a Grid* (*EPG graphs* for short) which is defined similarly to VPG, except that two vertices are adjacent if and only if the corresponding paths share a grid-edge. It turns out that every graph is EPG [36] and B_k -EPG graphs have been defined similarly to B_k -VPG graphs. Approximation algorithms for INDEPENDENT SET on VPG and EPG graphs have been deeply investigated, especially when the number of bends is a small constant (see, e.g., [7, 33, 40, 44]). It is an open problem whether INDEPENDENT SET admits a PTAS on B_1 -VPG graphs [44]. Concerning EPG graphs, Bessy et al. [6] showed that the problem admits no PTAS on B_1 -EPG graphs, unless $P = NP$, even if each path has its vertical segment or its horizontal segment of length at most 1. On the other hand, they provided a PTAS for INDEPENDENT SET on B_1 -EPG graphs where the length of the horizontal part of each path is at most a constant c with running time $O^*(n^{\frac{3c}{\epsilon}})$.

2 Preliminaries

We consider only finite simple graphs. If G' is a subgraph of G and G' contains all the edges of G with both endpoints in $V(G')$, then G' is an *induced subgraph* of G and we write $G' = G[V(G')]$. For a vertex $v \in V(G)$ and $r \in \mathbb{N}$, the *r -closed neighborhood* $N_G^r[v]$ is the set of vertices at distance at most r from v in G . The *degree* $d_G(v)$ of a vertex $v \in V(G)$ is the number of edges incident to v in G . The *maximum degree* $\Delta(G)$ of G is the quantity $\max\{d_G(v) : v \in V\}$. Given a graph $G = (V, E)$ and $V' \subseteq V$, the operation of *deleting the set of vertices* V' from G results in the graph $G - V' = G[V \setminus V']$. A graph is *Z -free* if it does not contain induced subgraphs isomorphic to graphs in a set Z . The complete bipartite graph with parts of sizes r and s is denoted by $K_{r,s}$. An *independent set* of a graph is a set of pairwise non-adjacent vertices. The maximum size of an independent set of G is denoted by $\alpha(G)$. A *clique* of a graph is a set of pairwise adjacent vertices. A *matching* of a graph is a set of pairwise non-incident edges. An *induced matching* in a graph is a matching M such that no two vertices belonging to different edges in M are adjacent in the graph.

Intersection graphs of unit disks and rectangles. We now explain how the geometric realizations of these intersection graphs are encoded. A collection of unit disks with a common radius $c \in \mathbb{R}$ is encoded by a collection of points in \mathbb{R}^2 representing the centers of the disks. Unless otherwise stated, when we refer to a rectangle we mean an axis-aligned closed rectangle in \mathbb{R}^2 . As is typically done for intersection graphs of rectangles, we assume that the vertices of the rectangles are on an integer grid \mathcal{G} and each rectangle is encoded by the coordinates of its vertices. Given an intersection graph G of a family \mathcal{R} of rectangles, a *grid representation* of G is a pair $(\mathcal{G}, \mathcal{R})$ as above.

VPG and EPG graphs. Given a rectangular grid \mathcal{G} , its horizontal lines are referred to as *rows* and its vertical lines as *columns*. For a VPG (EPG) graph G , the pair $\mathcal{R} = (\mathcal{G}, \mathcal{P})$ is a *VPG representation* (*EPG representation*) of G . More generally, a *grid representation* of a graph G is a triple $\mathcal{R} = (\mathcal{G}, \mathcal{P}, x)$ where $x \in \{e, v\}$, such that $(\mathcal{G}, \mathcal{P})$ is an EPG representation of G if $x = e$, and $(\mathcal{G}, \mathcal{P})$ is a VPG representation of G if $x = v$. Note that, irrespective of whether $x = e$ (that is, G is an EPG graph) or $x = v$ (that is, G is a VPG graph), if two vertices $u, v \in V(G)$ are adjacent in G then P_u and P_v share at least one grid-point. A *bend-point* of a path $P \in \mathcal{P}$ is a grid-point corresponding to a bend of P and a *segment* of P is either a vertical or horizontal line segment in the polygonal curve constituting P . Paths in \mathcal{P} are encoded as follows. For each $P \in \mathcal{P}$, we have one sequence $s(P)$ of points in \mathbb{R}^2 : $s(P) = (x_1, y_1), (x_2, y_2), \dots, (x_{\ell_P}, y_{\ell_P})$ consists of the endpoints (x_1, y_1) and (x_{ℓ_P}, y_{ℓ_P}) of P and all the bend-points of P in their order of appearance when traversing P from (x_1, y_1) to (x_{ℓ_P}, y_{ℓ_P}) . If each path in \mathcal{P} has a number of bends polynomial in $|V(G)|$, then the size of this data structure is polynomial in $|V(G)|$. Given $s(P)$, we can easily determine the horizontal part $h(P)$ of the path P . Note that our results for VPG and EPG graphs (Theorems 11 and 26), although stated for constant number of bends, still hold for polynomial (in $|V(G)|$) number of bends, with a worse polynomial running time.

PTAS. A PTAS for a maximization problem is an algorithm which takes an instance I of the problem and a parameter $\varepsilon > 0$ and produces a solution within a factor $1 - \varepsilon$ of the optimal in time $n^{O(f(1/\varepsilon))}$. A PTAS with running time $f(1/\varepsilon) \cdot n^{O(1)}$ is called an efficient PTAS (EPTAS for short).

3 Layered and local tree-independence number

The key definitions of this section are those of tree-independence number and layering, which we now recall. A *tree decomposition* of a graph G is a pair $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$, where T is a tree whose every node t is assigned a vertex subset $X_t \subseteq V(G)$, called a *bag*, such that the following conditions are satisfied:

- (T1) Every vertex of G belongs to at least one bag;
 - (T2) For every $uv \in E(G)$, there exists a bag containing both u and v ;
 - (T3) For every $u \in V(G)$, the subgraph T_u of T induced by $\{t \in V(T) : u \in X_t\}$ is connected.
- The *width* of $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ is the maximum value of $|X_t| - 1$ over all $t \in V(T)$. The *treewidth* of a graph G , denoted $\text{tw}(G)$, is the minimum width of a tree decomposition of G . The *independence number* of \mathcal{T} , denoted $\alpha(\mathcal{T})$, is the quantity $\max_{t \in V(T)} \alpha(G[X_t])$. The *tree-independence number* of a graph G , denoted $\text{tree-}\alpha(G)$, is the minimum independence number of a tree decomposition of G . Clearly, $\text{tree-}\alpha(G) \leq \text{tw}(G) + 1$, for any G . On the other hand, tree-independence number is more powerful than treewidth, as there exist classes with bounded tree-independence number and unbounded treewidth (for example, chordal graphs have tree-independence number 1 [15]).

A *layering* of a graph G is a partition $(V_0, V_1, \dots, V_\ell)$ of $V(G)$ such that, for every edge $vw \in E(G)$, if $v \in V_i$ and $w \in V_j$, then $|i - j| \leq 1$. Each set V_i is a *layer*. The *layered width* of a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of a graph G is the minimum integer ℓ such that, for some layering (V_0, V_1, \dots) of G , and for each bag X_t and layer V_i , we have $|X_t \cap V_i| \leq \ell$. The *layered treewidth* of a graph G is the minimum layered width of a tree decomposition of G . Layerings with one layer show that the layered treewidth of G is at most $\text{tw}(G) + 1$. We now introduce the analogue of layered treewidth for the width parameter tree-independence number.

► **Definition 3.** The layered independence number of a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of a graph G is the minimum integer ℓ such that, for some layering (V_0, V_1, \dots) of G , and for each bag X_t and layer V_i , we have $\alpha(G[X_t \cap V_i]) \leq \ell$. The layered tree-independence number of a graph G is the minimum layered independence number of a tree decomposition of G .

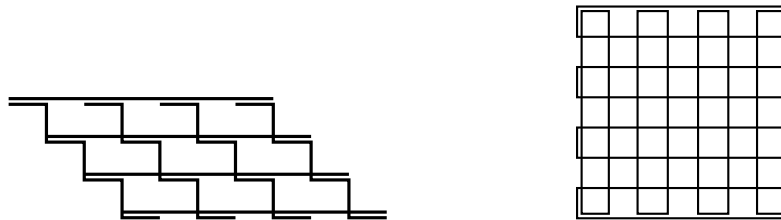
Layerings with one layer show that the layered tree-independence number of G is at most $\text{tree-}\alpha(G)$. Moreover, the layered tree-independence number of a graph is clearly at most its layered treewidth. The proof of [21, Lemma 10] shows, mutatis mutandis, that graphs of bounded layered tree-independence number have $O(\sqrt{n})$ tree-independence number:

► **Lemma 4.** Every n -vertex graph with layered tree-independence number k has tree-independence number at most $2\sqrt{kn}$.

Given a width parameter p , a graph class \mathcal{G} has bounded local p if there is a function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that for every integer $r \in \mathbb{N}$, graph $G \in \mathcal{G}$, and vertex $v \in V(G)$, the subgraph $G[N^r[v]]$ has p -width at most $f(r)$. In [21], it is shown that if every graph in a class \mathcal{G} has layered treewidth at most ℓ , then \mathcal{G} has bounded local treewidth with $f(r) = \ell(2r + 1) - 1$.

► **Lemma 5** (*). If every graph in a class \mathcal{G} has layered tree-independence number at most ℓ , then \mathcal{G} has bounded local tree-independence number with $f(r) = \ell(2r + 1)$.

► **Corollary 6** (*). The layered tree-independence number of $K_{n,n}$ is at least $n/5$.



■ **Figure 1** Examples showing that VPG/EPG graphs and intersection graphs of rectangles have unbounded layered tree-independence number: VPG/EPG representation (left) and representation by intersection of rectangles (right) of $K_{4,4}$.

► **Theorem 7** (*). The following are equivalent for a minor-closed class \mathcal{G} :

1. Some apex³ graph is not in \mathcal{G} ;
2. \mathcal{G} has bounded local tree-independence number;
3. \mathcal{G} has linear local tree-independence number (i.e., $f(r)$ is linear in r);
4. \mathcal{G} has bounded layered tree-independence number.

For $p \in \mathbb{N}$, the p -th power of a graph G is the graph G^p with vertex set $V(G^p) = V(G)$, where $uv \in E(G^p)$ if and only if u and v are at distance at most p in G . Bonomo-Braberman and Gonzalez [9] showed that fixed powers of bounded treewidth and bounded degree graphs are of bounded treewidth: For any graph G and $p \geq 2$, $\text{tw}(G^p) \leq (\text{tw}(G) + 1)(\Delta(G) + 1)^{\lceil \frac{p}{2} \rceil} - 1$. It follows from [23] that powers of graphs of bounded layered treewidth and bounded maximum degree have bounded layered treewidth. The upper bound therein was later improved by Dujmović et al. [22], who showed that if G has layered treewidth k , then G^p has layered treewidth less than $2pk\Delta(G)^{\lfloor \frac{p}{2} \rfloor}$. Using a result from [45], we show that odd powers of bounded layered tree-independence number graphs have bounded layered tree-independence number and that this does not extend to even powers.

³ An apex graph is a graph that can be made planar by deleting a single vertex.

► **Theorem 8** (\star). *Let G be a graph and let d be a positive integer. Given a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of G and a layering (V_1, \dots, V_m) of G such that, for each bag X_t and layer V_i , $\alpha(G[X_t \cap V_i]) \leq k$, it is possible to compute in $O(|V(T)| \cdot (|V(G)| + |E(G)|))$ time a tree decomposition $\mathcal{T}' = (T, \{X'_t\}_{t \in V(T)})$ of G^{1+2d} and a layering $(V'_1, \dots, V'_{\lceil \frac{m}{1+2d} \rceil})$ of G^{1+2d} such that, for each bag X'_t and layer V'_i , $\alpha(G^{1+2d}[X'_t \cap V'_i]) \leq (1 + 4d)k$. In particular, if G has layered tree-independence number k , then G^{1+2d} has layered tree-independence number at most $(1 + 4d)k$.*

► **Lemma 9** (\star). *Fix an even $k \in \mathbb{N}$. There exist graphs G with layered tree-independence number 1 and such that the layered tree-independence number of G^k is arbitrarily large.*

3.1 Intersection graphs with bounded layered tree-independence number

► **Theorem 10** (\star). *Let G be the intersection graph of a family \mathcal{D} of n unit disks. It is possible to compute, in $O(n)$ time, a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ and a layering (V_1, V_2, \dots) of G such that $|V(T)| = O(n)$ and, for each bag X_t and layer V_i , $\alpha(G[X_t \cap V_i]) \leq 8$. In particular, G has layered tree-independence number at most 8.*

► **Theorem 11** (\star). *Let G be a graph on n vertices together with a grid representation $\mathcal{R} = (\mathcal{G}, \mathcal{P}, x)$ such that each path in \mathcal{P} has horizontal part of length at most $\ell - 1$, for some fixed $\ell \geq 1$, and number of bends constant. It is possible to compute, in $O(n^2)$ time, a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ and a layering (V_1, V_2, \dots) of G such that $|V(T)| = O(n^2)$ and, for each bag X_t and layer V_i , $\alpha(G[X_t \cap V_i]) \leq 4\ell - 1$. In particular, G has layered tree-independence number at most $4\ell - 1$.*

4 Fractional tree- α -fragility

Let p be a width parameter in $\{\text{tw}, \text{tree-}\alpha\}$. Fractional tw -fragility was first defined in [24]. We provide here an equivalent definition from [26], which was explicitly extended to the case $p = \text{tree-}\alpha$ in [25].

► **Definition 12.** *For $\beta \leq 1$, a β -general cover of a graph G is a multiset \mathcal{C} of subsets of $V(G)$ such that each vertex belongs to at least $\beta|\mathcal{C}|$ elements of the cover. The p -width of the cover is $\max_{C \in \mathcal{C}} p(G[C])$.*

For a parameter p , a graph class \mathcal{G} is fractionally p -fragile if there exists a function $f: \mathbb{N} \rightarrow \mathbb{N}$ such that, for every $r \in \mathbb{N}$, every $G \in \mathcal{G}$ has a $(1 - 1/r)$ -general cover with p -width at most $f(r)$.

A fractionally p -fragile class \mathcal{G} is efficiently fractionally p -fragile if there exists an algorithm that, for every $r \in \mathbb{N}$ and $G \in \mathcal{G}$, returns in $\text{poly}(|V(G)|)$ time a $(1 - 1/r)$ -general cover \mathcal{C} of G and, for each $C \in \mathcal{C}$, a tree decomposition of $G[C]$ of width (if $p = \text{tw}$) or independence number (if $p = \text{tree-}\alpha$) at most $f(r)$, for some function $f: \mathbb{N} \rightarrow \mathbb{N}$.

Note that classes of bounded tree-independence number are efficiently fractionally tree- α -fragile thanks to [14]. Hence, the family of efficiently fractionally tree- α -fragile classes contains the two incomparable families of bounded tree-independence number classes and efficiently fractionally tw -fragile classes (to see that they are incomparable, consider chordal graphs and planar graphs). We now identify one more subfamily:

► **Lemma 13.** *Let $\ell \in \mathbb{N}$ and let G be a graph. For each $r \in \mathbb{N}$, given a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of G and a layering (V_0, V_1, \dots) of G such that, for each bag X_t and layer V_i , $\alpha(G[X_t \cap V_i]) \leq \ell$, it is possible to compute in $O(|V(G)|)$ time a $(1 - 1/r)$ -general*

cover \mathcal{C} of G and, for each $C \in \mathcal{C}$, a tree decomposition of $G[C]$ with independence number at most $\ell(r - 1)$. In particular, if every graph in a class \mathcal{G} has layered tree-independence number at most ℓ , then \mathcal{G} is fractionally tree- α -fragile with $f(r) = \ell(r - 1)$.

Proof. Fix $r \in \mathbb{N}$. Let $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ and (V_0, V_1, \dots) be the given tree decomposition and layering of G , respectively. For each $m \in \{0, \dots, r - 1\}$, let $C_m = \bigcup_{i \equiv m \pmod{r}} V_i$. We claim that $\mathcal{C} = \{C_m : 0 \leq m \leq r - 1\}$ is a $(1 - 1/r)$ -general cover of G with tree-independence number at most $\ell(r - 1)$. Observe first that each $v \in V(G)$ is not covered by exactly one element of \mathcal{C} and so it belongs to $r - 1 = (1 - 1/r)|\mathcal{C}|$ elements of \mathcal{C} . Let now $C \in \mathcal{C}$. Each component K of $G[C]$ is contained in at most $r - 1$ (consecutive) layers and so, since $\alpha(G[X_t \cap V_i]) \leq \ell$ for each bag X_t and layer V_i , restricting the bags in \mathcal{T} to $V(K)$, gives a tree decomposition of K with independence number at most $\ell(r - 1)$. We then merge the tree decompositions of the components of $G[C]$ into a tree decomposition of $G[C]$ with independence number at most $\ell(r - 1)$ in linear time. ◀

Note that the same argument of Lemma 13 shows that, if every graph in a class \mathcal{G} has bounded layered treewidth, then \mathcal{G} is fractionally tw-fragile. The following result implies that, if a class is fractionally tree- α -fragile, then it has bounded biclique number.

► **Theorem 14.** *For any function $f : \mathbb{N} \rightarrow \mathbb{N}$ and integer $r > 2$, there exists $n \in \mathbb{N}$ such that no $(1 - 1/r)$ -general cover of $K_{n,n}$ has tree-independence number less than $f(r)$. Hence, the class $\{K_{n,n} : n \in \mathbb{N}\}$ is not fractionally tree- α -fragile.*

Proof. Fix arbitrary $f : \mathbb{N} \rightarrow \mathbb{N}$ and $r > 2$. Consider a copy G of $K_{n,n}$, with $n > f(r)/(1 - 2/r)$. Let \mathcal{C} be a $(1 - 1/r)$ -general cover of G . Then, every vertex of G belongs to at least $(1 - 1/r)|\mathcal{C}|$ elements of \mathcal{C} and so there exists $C \in \mathcal{C}$ of size at least $2n(1 - 1/r)$. Let A and B be the two bipartition classes of G . Then, $|A \cap C| \geq |C| - |B| \geq 2n(1 - 1/r) - n = n(1 - 2/r) > f(r)$ and, similarly, $|B \cap C| > f(r)$. Therefore, $G[C]$ contains $K_{f(r), f(r)}$ as an induced subgraph and since $\text{tree-}\alpha(K_{f(r), f(r)}) = f(r)$ [15], $\text{tree-}\alpha(G[C]) \geq f(r)$. ◀

However, the following result shows that small biclique number does not guarantee fractional tree- α -fragility.

► **Theorem 15.** *The class of $K_{2,3}$ -free graphs is not fractionally tree- α -fragile.*

Proof. Let G_n be the n -dimensional grid graph of width n , i.e., the graph with vertex set $V(G_n) = [n]^n = \{(a_1, \dots, a_n) : 1 \leq a_1, \dots, a_n \leq n\}$, where two vertices (a_1, \dots, a_n) and (b_1, \dots, b_n) are adjacent if and only if $\sum_{1 \leq i \leq n} |a_i - b_i| = 1$. It is not difficult to see that G_n is $K_{2,3}$ -free, for each $n \in \mathbb{N}$. We show that the class $\{G_n : n \in \mathbb{N}\}$ is not fractionally tree- α -fragile.

Fix arbitrary $f : \mathbb{N} \rightarrow \mathbb{N}$ and $r > 2$. For such a choice, fix $n \in \mathbb{N}$ such that $\frac{r-4}{2r}n + 1 \geq R(3, f(r))$, where $R(3, s)$ denotes the smallest integer m for which every graph on m vertices either contains a clique of size 3 or an independent set of size s . We now show that every $(1 - 1/r)$ -general cover of G_n has tree-independence at least $f(r)$. Let \mathcal{C} be a $(1 - 1/r)$ -general cover of G_n . Then, every vertex of G_n belongs to at least $(1 - 1/r)|\mathcal{C}|$ elements of \mathcal{C} and so there exists $C \in \mathcal{C}$ containing at least $(1 - 1/r)|V(G_n)| = (1 - 1/r)n^n$ vertices of G_n . Fix such a C and let G be the subgraph of G_n induced by C . We claim that $\text{tree-}\alpha(G) \geq f(r)$.

Observe first that, for each $v \in V(G_n)$, $n \leq d_{G_n}(v) \leq 2n$. Hence, $2|E(G_n)| = \sum_{v \in V(G_n)} d_{G_n}(v) \geq n \cdot n^n$. Consider now the graph G' obtained from G_n by deleting the vertex set C . Clearly, G' has at most n^n/r vertices. Since deleting a vertex from G_n decreases the number of edges of the resulting graph by at most $2n$, we have that $|E(G)| \geq$

$|E(G_n)| - 2n|V(G')|$, from which $\sum_{v \in V(G)} d_G(v) \geq n \cdot n^n - 2 \cdot 2n \cdot n^n / r = n \cdot n^n (1 - 4/r)$. Therefore, the average degree of G is at least $n(1 - 4/r)$ and so $\text{tw}(G) \geq \frac{r-4}{2r}n$, for example by [13, Corollary 1]. This implies that every tree decomposition of G has a bag of size at least $\frac{r-4}{2r}n + 1 \geq R(3, f(r))$ and, since G is triangle-free, it follows that $\text{tree-}\alpha(G) \geq f(r)$. \blacktriangleleft

4.1 Intersection graphs of fat objects

In this section we show that the class of intersection graphs of fat objects in \mathbb{R}^d is efficiently fractionally tree- α -fragile. Let $d \geq 2$ be a fixed integer. A *box of size r* is an axis-aligned hypercube in \mathbb{R}^d of side length r . The *size* of an object O in \mathbb{R}^d , denoted $s(O)$, is the side length of its smallest enclosing axis-aligned hypercube.

Chan [11] considered the following definition of fatness: A collection of objects in \mathbb{R}^d is *fat* if, for any r and size- r box R , we can choose c points in \mathbb{R}^d such that every object that intersects R and has size at least r contains at least one of the chosen points. Chan also stated that a collection of balls or boxes with bounded aspect ratios are fat (recall that the aspect ratio of a box is the ratio of its largest side length over its smallest side length). We slightly generalize this fatness definition as follows.

► **Definition 16.** *A collection of objects in \mathbb{R}^d is c -fat if, for any r and any size- r closed box R , for every sub-collection \mathcal{P} of pairwise non-intersecting objects which intersect R and are of size at least r , we can choose c points in \mathbb{R}^d such that every object in \mathcal{P} contains at least one of the chosen points.*

► **Remark 17.** When working with a c -fat collection of objects, we assume that some reasonable operations can be done in constant time: determining the center and size of an object, deciding if two objects intersect and constructing the geometric realization of the collection.

► **Theorem 18** (\star). *Let \mathcal{O} be a c -fat collection of objects in \mathbb{R}^d and let G be its intersection graph. For each $r_0 > 1$, let $f(r_0) = 2 \left\lceil \frac{1}{1 - (1 - \frac{1}{r_0})^{\frac{1}{d}}} \right\rceil$. Then, we can compute in linear time a $(1 - 1/r_0)$ -general cover \mathcal{C} of G of size at most $(f(r_0)/2 - 1)^d$. Moreover, for each $C \in \mathcal{C}$, we can compute in linear time a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of $G[C]$, with $|V(T)| \leq |V(G)| + 1$, such that $\alpha(\mathcal{T}) \leq cf(r_0)^{2d}$.*

► **Corollary 19** (\star). *There exist fractionally tree- α -fragile classes of unbounded local tree-independence number.*

5 PTASes

Let us begin by defining MAX WEIGHT INDEPENDENT PACKING. Given a graph G and a finite family $\mathcal{H} = \{H_j\}_{j \in J}$ of connected non-null subgraphs of G , an *independent \mathcal{H} -packing* in G is a subfamily $\mathcal{H}' = \{H_i\}_{i \in I}$ of subgraphs from \mathcal{H} (that is, $I \subseteq J$) that are at pairwise distance at least 1, that is, they are vertex-disjoint and there is no edge between any two of them. If the subgraphs in \mathcal{H} are equipped with a weight function $w: J \rightarrow \mathbb{Q}_+$ assigning weight w_j to each subgraph H_j , the *weight* of an independent \mathcal{H} -packing $\mathcal{H}' = \{H_i\}_{i \in I}$ in G is $\sum_{i \in I} w_i$. Given a graph G , a finite family $\mathcal{H} = \{H_j\}_{j \in J}$ of connected non-null subgraphs of G , and a weight function $w: J \rightarrow \mathbb{Q}_+$ on the subgraphs in \mathcal{H} , the problem MAX WEIGHT INDEPENDENT PACKING asks to find an independent \mathcal{H} -packing in G of maximum weight. In the special case when \mathcal{F} is a *fixed* finite family of connected non-null graphs and \mathcal{H} is the set of all subgraphs of G isomorphic to a member of \mathcal{F} , the problem is called MAX WEIGHT INDEPENDENT \mathcal{F} -PACKING and is a common generalization of several problems, among

which: INDEPENDENT \mathcal{F} -PACKING [10], MAX WEIGHT INDEPENDENT SET ($\mathcal{F} = \{K_1\}$), MAX WEIGHT INDUCED MATCHING ($\mathcal{F} = \{K_2\}$), DISSOCIATION SET ($\mathcal{F} = \{K_1, K_2\}$) and the weight function assigns to each subgraph H_j the weight $|V(H_j)|$ [47, 51].

5.1 Packing subgraphs at distance at least 1 in efficiently fractionally tree- α -fragile classes

Our PTAS relies on the following result.

► **Theorem 20** (Dallard et al. [15]). *Let k and h be two positive integers. Given a graph G and a finite family $\mathcal{H} = \{H_j\}_{j \in J}$ of connected non-null subgraphs of G such that $|V(H_j)| \leq h$ for every $j \in J$, MAX WEIGHT INDEPENDENT PACKING can be solved in time $O(|V(G)|^{h(k+1)} \cdot |V(T)|)$ if G is given together with a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ with $\alpha(\mathcal{T}) \leq k$.*

► **Theorem 21.** *Let $h \in \mathbb{N}$ and let $f: \mathbb{N} \rightarrow \mathbb{N}$ be a function. There exists an algorithm that, given $r \in \mathbb{N}$, an n -vertex graph G equipped with a $(1 - 1/r)$ -general cover $\mathcal{C} = \{C_1, C_2, \dots\}$ and, for each i , a tree decomposition $\mathcal{T}_i = (T_i, \{X_t\}_{t \in V(T_i)})$ of $G[C_i]$ with $\alpha(\mathcal{T}_i) \leq f(r)$, a finite family $\mathcal{H} = \{H_j\}_{j \in J}$ of connected non-null subgraphs of G such that $|V(H_j)| \leq h$ for every $j \in J$, and a weight function $w: J \rightarrow \mathbb{Q}_+$ on the subgraphs in \mathcal{H} , returns in time $|\mathcal{C}| \cdot O(n^{h(f(r)+1)} \cdot t)$, where $t = \max_i |V(T_i)|$, an independent \mathcal{H} -packing in G of weight at least a factor $(1 - h/r)$ of the optimal.*

Proof. For each $i \geq 1$, we proceed as follows. Using the algorithm from Theorem 20, we simply compute a maximum-weight independent \mathcal{H} -packing \mathcal{P}_i in $G[C_i]$ in time $O(n^{h(f(r)+1)} \cdot t)$. The total running time is then $|\mathcal{C}| \cdot O(n^{h(f(r)+1)} \cdot t)$. For a collection \mathcal{A} of subgraphs of G , each isomorphic to a member of \mathcal{H} , and a subset $C \subseteq V(G)$, let $w(\mathcal{A}) = \sum_{A \in \mathcal{A}} w(A)$ and let $\mathcal{A} \cap C = \{A \in \mathcal{A} : A \subseteq C\}$. Observe that, given a subgraph H of G , each vertex $v \in V(H)$ is not contained in at most $|\mathcal{C}|/r$ elements of the $(1 - 1/r)$ -general cover \mathcal{C} . Hence, $V(H)$ is contained in at least $(1 - |V(H)|/r)|\mathcal{C}|$ elements of \mathcal{C} . Let $\mathcal{P} = \{P_1, P_2, \dots\}$ be an independent \mathcal{H} -packing in G of maximum weight. Then,

$$\begin{aligned} \sum_{C_i \in \mathcal{C}} w(\mathcal{P} \cap C_i) &= \sum_{C_i \in \mathcal{C}} \sum_{P_j \in \mathcal{P}} w(P_j) \mathbb{1}_{\{P_j \subseteq C_i\}} \\ &= \sum_{P_j \in \mathcal{P}} w(P_j) \sum_{C_i \in \mathcal{C}} \mathbb{1}_{\{P_j \subseteq C_i\}} \\ &\geq \sum_{P_j \in \mathcal{P}} w(P_j) (1 - |V(P_j)|/r) |\mathcal{C}| \\ &\geq \sum_{P_j \in \mathcal{P}} w(P_j) (1 - h/r) |\mathcal{C}| \\ &= |\mathcal{C}| (1 - h/r) w(\mathcal{P}). \end{aligned}$$

By the pigeonhole principle, there exists $C_i \in \mathcal{C}$ such that $w(\mathcal{P} \cap C_i) \geq (1 - h/r)w(\mathcal{P})$. We then return the maximum-weight independent \mathcal{H} -packing \mathcal{P}_i in $G[C_i]$ computed above. Since $\mathcal{P} \cap C_i$ is an independent \mathcal{H} -packing in $G[C_i]$, we have that $w(\mathcal{P}_i) \geq w(\mathcal{P} \cap C_i) \geq (1 - h/r)w(\mathcal{P})$. ◀

Theorem 21 immediately implies that MAX WEIGHT INDEPENDENT PACKING admits a PTAS in any efficiently fractionally tree- α -fragile class. A special case is the following.

► **Corollary 22** (\star). *There exists an algorithm that, given $r \in \mathbb{N}$, a c -fat collection \mathcal{O} of n objects in \mathbb{R}^d and its intersection graph G , and a weight function $w: V(G) \rightarrow \mathbb{Q}_+$, returns in time $(f(r)/2 - 1)^d \cdot O(n^{(cf(r)^{2d+2})})$, where $f(r) = 2 \left\lceil \frac{1}{1 - (1 - \frac{1}{r})^{\frac{1}{d}}} \right\rceil$, an independent set in G of weight at least a factor $(1 - 1/r)$ of the optimal.*

5.2 Packing subgraphs at distance at least d in graphs with bounded layered tree-independence number

MAX WEIGHT INDEPENDENT PACKING has a natural generalization. For a fixed positive integer d , given a graph G and a finite family $\mathcal{H} = \{H_j\}_{j \in J}$ of connected non-null subgraphs of G , a *distance- d \mathcal{H} -packing* in G is a subfamily $\mathcal{H}' = \{H_i\}_{i \in I}$ of subgraphs from \mathcal{H} that are at pairwise distance at least d . If we are also given a weight function $w: J \rightarrow \mathbb{Q}_+$, MAX WEIGHT DISTANCE- d PACKING is the problem of finding a distance- d \mathcal{H} -packing in G of maximum weight. The case $d = 2$ coincides with MAX WEIGHT INDEPENDENT PACKING.

► **Theorem 23** (\star). *Let $h, \ell \in \mathbb{N}$. Let d be an even positive integer. There exists an algorithm that, given $r \in \mathbb{N}$, an n -vertex graph G equipped with a tree decomposition $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ and a layering (V_1, V_2, \dots) of G such that, for each bag X_t and layer V_i , $\alpha(G[X_t \cap V_i]) \leq \ell$, a finite family $\mathcal{H} = \{H_j\}_{j \in J}$ of connected non-null subgraphs of G such that $|V(H_j)| \leq h$ for every $j \in J$, and a weight function $w: J \rightarrow \mathbb{Q}_+$, returns in time $r \cdot |V(T)| \cdot n^{O(r)}$ a distance- d \mathcal{H} -packing in G within a factor $(1 - h/r)$ of the optimal.*

Combining Theorem 23 with Theorem 10, we obtain the following:

► **Corollary 24**. *Let $d \in \mathbb{N}$ be even. MAX WEIGHT DISTANCE- d PACKING admits a PTAS for unit disk graphs.*

Observe that Theorem 23 cannot be extended to odd values of d , unless $P = NP$. Indeed, Eto et al. [31] showed that, for each $\varepsilon > 0$ and fixed odd $d \geq 3$, it is NP-hard to approximate DISTANCE- d INDEPENDENT SET to within a factor of $n^{1/2-\varepsilon}$ for chordal graphs.

Since unit disk graphs have $O(\sqrt{n})$ tree-independence number (Theorem 10 and Lemma 4) and since MAX WEIGHT DISTANCE- d PACKING is solvable in time $n^{O(k)}$, where k is the tree-independence number of the input graph [45], we immediately obtain a subexponential-time algorithm on unit disk graphs.

► **Lemma 25**. *For any fixed even $d \in \mathbb{N}$, MAX WEIGHT DISTANCE- d PACKING can be solved in $2^{O(\sqrt{n} \log n)}$ time on unit disk graphs.*

A subexponential-time algorithm for INDEPENDENT SET on unit disk graphs was first given in [2] and later extended in [16] to intersection graphs of fat objects.

5.3 Packing independent unit disks, unit-width rectangles and paths with bounded horizontal part on a grid

The following PTASes are obtained by showing that the tree-independence number of graphs whose geometric realizations are contained in an axis-aligned rectangle with bounded width is bounded.

- **Theorem 26** (\star). *MAX WEIGHT INDEPENDENT SET admits a PTAS when restricted to:*
- *Intersection graphs of a family of n unit disks of common radius $c \geq 1$. The running time is $O(c \lceil \frac{2}{\varepsilon} \rceil \cdot n^{2 \lceil \frac{2}{\varepsilon} \rceil + 3})$.*
 - *Intersection graphs of a family of n width- c rectangles together with a grid representation $(\mathcal{G}, \mathcal{R})$. The running time is $O(c \lceil \frac{1}{\varepsilon} \rceil \cdot n^{\lceil \frac{1}{\varepsilon} \rceil \cdot \frac{5}{2} + 4})$.*

- *Graphs on n vertices with a grid representation $\mathcal{R} = (\mathcal{G}, \mathcal{P}, x)$ such that each path in \mathcal{P} has number of bends constant and the horizontal part of each path in \mathcal{P} has length at most c , for some fixed $c \in \mathbb{N}$. If $x = v$, the running time is $O(c^{\lceil \frac{1}{\varepsilon} \rceil} \cdot n^{\lceil \frac{1}{\varepsilon} \rceil c + 4})$. If $x = e$, the running time is $O(c^{\lceil \frac{1}{\varepsilon} \rceil} \cdot n^{3(\lceil \frac{1}{\varepsilon} \rceil c + 1)})$.*

References

- 1 Pankaj K. Agarwal, Marc van Kreveld, and Subhash Suri. Label placement by maximum independent set in rectangles. *Computational Geometry*, 11(3):209–218, 1998.
- 2 Jochen Alber and Jiří Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *Journal of Algorithms*, 52(2):134–151, 2004.
- 3 Andrei Asinowski, Elad Cohen, Martin Charles Golumbic, Vincent Limouzy, Marina Lipshteyn, and Michal Stern. Vertex intersection graphs of paths on a grid. *Journal of Graph Algorithms and Applications*, 16(2):129–150, 2012.
- 4 Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994.
- 5 Sayan Bandyapadhyay, William Lochet, Daniel Lokshtanov, Saket Saurabh, and Jie Xue. True contraction decomposition and almost ETH-tight bipartization for unit-disk graphs. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
- 6 Stéphane Bessy, Marin Bougeret, Steven Chaplick, Daniel Gonçalves, and Christophe Paul. On independent set in B_1 -EPG graphs. *Discrete Applied Mathematics*, 278:62–72, 2020.
- 7 Therese Biedl and Martin Derka. Splitting B_2 -VPG graphs into outer-string and co-comparability graphs. In Faith Ellen, Antonina Kolokolova, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures - 15th International Symposium, (WADS 2017)*, volume 10389 of *Lecture Notes in Computer Science*, pages 157–168. Springer, 2017.
- 8 Marthe Bonamy, Édouard Bonnet, Nicolas Bousquet, Pierre Charbit, Panos Giannopoulos, Eun Jung Kim, Paweł Rzażewski, Florian Sikora, and Stéphan Thomassé. EPTAS and subexponential algorithm for maximum clique on disk and unit ball graphs. *J. ACM*, 68(2), 2021.
- 9 Flavia Bonomo-Braberman and Carolina Lucía Gonzalez. A new approach on locally checkable problems. *Discrete Applied Mathematics*, 314:53–80, June 2022.
- 10 Kathie Cameron and Pavol Hell. Independent packings in structured graphs. *Mathematical Programming*, 105(2-3):201–213, 2006.
- 11 Timothy M Chan. Polynomial-time approximation schemes for packing and piercing fat objects. *Journal of Algorithms*, 46(2):178–189, 2003.
- 12 Timothy M. Chan. A note on maximum independent sets in rectangle intersection graphs. *Information Processing Letters*, 89(1):19–23, 2004.
- 13 L.Sunil Chandran and C.R. Subramanian. Girth and treewidth. *Journal of Combinatorial Theory, Series B*, 93(1):23–32, 2005.
- 14 Clément Dallard, Fedor V. Fomin, Petr A. Golovach, Tuukka Korhonen, and Martin Milanič. Computing tree decompositions with small independence number. *CoRR*, abs/2207.09993, 2022. URL: <https://arxiv.org/abs/2207.09993>.
- 15 Clément Dallard, Martin Milanič, and Kenny Štorgel. Treewidth versus clique number. II. Tree-independence number. *CoRR*, abs/2111.04543, 2022. URL: <https://arxiv.org/abs/2111.04543>.
- 16 Mark de Berg, Hans L. Bodlaender, Sándor Kisfaludi-Bak, Dániel Marx, and Tom C. van der Zanden. A framework for exponential-time-hypothesis-tight algorithms and lower bounds in geometric intersection graphs. *SIAM Journal on Computing*, 49(6):1291–1331, 2020.
- 17 Erik D. Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005)*, pages 637–646. IEEE Computer Society, 2005.

- 18 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Bojan Mohar. Approximation algorithms via contraction decomposition. *Combinatorica*, 30(5):533–552, 2010.
- 19 Matt DeVos, Guoli Ding, Bogdan Oporowski, Daniel P. Sanders, Bruce Reed, Paul Seymour, and Dirk Vertigan. Excluding any graph as a minor allows a low tree-width 2-coloring. *Journal of Combinatorial Theory, Series B*, 91(1):25–41, 2004.
- 20 Vida Dujmović, David Eppstein, and David R. Wood. Structure of graphs with locally restricted crossings. *SIAM Journal on Discrete Mathematics*, 31(2):805–824, 2017.
- 21 Vida Dujmović, Pat Morin, and David R. Wood. Layered separators in minor-closed graph classes with applications. *Journal of Combinatorial Theory, Series B*, 127:111–147, 2017.
- 22 Vida Dujmović, Louis Esperet, Pat Morin, Bartosz Walczak, and David R. Wood. Clustered 3-colouring graphs of bounded degree. *Combinatorics, Probability and Computing*, 31(1):123–135, 2022.
- 23 Vida Dujmović, Pat Morin, and David R. Wood. Graph product structure for non-minor-closed classes. *CoRR*, abs/1907.05168, 2022. URL: <https://arxiv.org/abs/1907.05168>.
- 24 Zdeněk Dvořák. Sublinear separators, fragility and subexponential expansion. *European Journal of Combinatorics*, 52:103–119, 2016.
- 25 Zdeněk Dvořák. Talk at GWP 2022 - satellite workshop of ICALP 2022, 2022. URL: <https://homepages.ecs.vuw.ac.nz/~bretteni/GWP2022/>.
- 26 Zdeněk Dvořák. Approximation Metatheorems for Classes with Bounded Expansion. In Artur Czumaj and Qin Xin, editors, *18th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2022)*, volume 227 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
- 27 Zdeněk Dvořák, Daniel Gonçalves, Abhiruk Lahiri, Jane Tan, and Torsten Ueckerdt. On comparable box dimension. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022.
- 28 Zdeněk Dvořák and Abhiruk Lahiri. Approximation schemes for bounded distance problems on fractionally treewidth-fragile graphs. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.
- 29 David Eppstein. Diameter and treewidth in minor-closed graph families. *Algorithmica*, 27(3):275–291, 2000.
- 30 Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34(6):1302–1323, 2005.
- 31 Hiroshi Eto, Fengrui Guo, and Eiji Miyano. Distance- d independent set problems for bipartite and chordal graphs. *Journal of Combinatorial Optimization*, 27(1):88–99, 2014.
- 32 Fedor V. Fomin, Daniel Lokshtanov, and Saket Saurabh. Bidimensionality and geometric graphs. In Yuval Rabani, editor, *Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 1563–1575. SIAM, 2012.
- 33 Jacob Fox and János Pach. Computing the independence number of intersection graphs. In Dana Randall, editor, *Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2011)*, pages 1161–1165. SIAM, 2011.
- 34 Waldo Gálvez, Arindam Khan, Mathieu Mari, Tobias Mömke, Madhusudhan Reddy Pittu, and Andreas Wiese. A 3-approximation algorithm for Maximum Independent Set of rectangles. In Joseph (Seffi) Naor and Niv Buchbinder, editors, *Proceedings of the 2022 ACM-SIAM Symposium on Discrete Algorithms (SODA 2022)*, pages 894–905. SIAM, 2022.
- 35 Matt Gibson and Imran A. Pirwani. Algorithms for dominating set in disk graphs: Breaking the $\log n$ barrier. In Mark de Berg and Ulrich Meyer, editors, *Algorithms – ESA 2010*, volume 6346 of *Lecture Notes in Computer Science*, pages 243–254. Springer Berlin Heidelberg, 2010.

- 36 Martin Charles Golumbic, Marina Lipshteyn, and Michal Stern. Edge intersection graphs of single bend paths on a grid. *Networks*, 54(3):130–138, 2009.
- 37 Alexander Grigoriev and Hans L. Bodlaender. Algorithms for graphs embeddable with few crossings per edge. *Algorithmica*, 49(1):1–11, 2007.
- 38 Dorit S. Hochbaum and Wolfgang Maass. Approximation schemes for covering and packing problems in image processing and VLSI. *J. ACM*, 32(1):130–136, 1985.
- 39 Harry B Hunt, Madhav V Marathe, Venkatesh Radhakrishnan, S.S Ravi, Daniel J Rosenkrantz, and Richard E Stearns. NC-approximation schemes for NP- and PSPACE-hard problems for geometric graphs. *Journal of Algorithms*, 26(2):238–274, 1998.
- 40 Abhiruk Lahiri, Joydeep Mukherjee, and C. R. Subramanian. Maximum Independent Set on B_1 -VPG graphs. In Zaixin Lu, Donghyun Kim, Weili Wu, Wei Li, and Ding-Zhu Du, editors, *Combinatorial Optimization and Applications - 9th International Conference (COCO 2015)*, volume 9486 of *Lecture Notes in Computer Science*, pages 633–646. Springer, 2015.
- 41 Daniel Lokshantov, Fahad Panolan, Saket Saurabh, Jie Xue, and Meirav Zehavi. A framework for approximation schemes on disk graphs. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms (SODA 2023)*, pages 2228–2241. SIAM, 2023.
- 42 Dániel Marx. Efficient approximation schemes for geometric problems? In Gerth Støtting Brodal and Stefano Leonardi, editors, *Algorithms – ESA 2005*, volume 3669 of *Lecture Notes in Computer Science*, pages 448–459. Springer Berlin Heidelberg, 2005.
- 43 Tomomi Matsui. Approximation algorithms for maximum independent set problems and fractional coloring problems on unit disk graphs. In Jin Akiyama, Mikio Kano, and Masatsugu Urabe, editors, *Discrete and Computational Geometry*, pages 194–200. Springer Berlin Heidelberg, 1998.
- 44 Saeed Mehrabi. Approximating domination on intersection graphs of paths on a grid. In Roberto Solis-Oba and Rudolf Fleischer, editors, *Approximation and Online Algorithms - 15th International Workshop (WAOA 2017)*, volume 10787 of *Lecture Notes in Computer Science*, pages 76–89. Springer, 2017.
- 45 Martin Milanič and Paweł Rzżewski. Tree decompositions with bounded independence number: beyond independent sets. *CoRR*, abs/2209.12315, 2022. URL: <https://arxiv.org/abs/2209.12315>.
- 46 Tim Nieberg, Johann L. Hurink, and Walter Kern. A robust PTAS for maximum weight independent sets in unit disk graphs. In Juraj Hromkovic, Manfred Nagl, and Bernhard Westfechtel, editors, *Graph-Theoretic Concepts in Computer Science, 30th International Workshop (WG 2004)*, volume 3353 of *Lecture Notes in Computer Science*, pages 214–221. Springer, 2004.
- 47 Yury Orlovich, Alexandre Dolgui, Gerd Finke, Valery Gordon, and Frank Werner. The complexity of dissociation set problems in graphs. *Discrete Applied Mathematics*, 159(13):1352–1366, 2011.
- 48 Fahad Panolan, Saket Saurabh, and Meirav Zehavi. Contraction decomposition in unit disk graphs and algorithmic applications in parameterized complexity. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*, pages 1035–1054. SIAM, 2019.
- 49 Erik Jan van Leeuwen. Approximation algorithms for unit disk graphs. In Dieter Kratsch, editor, *Graph-Theoretic Concepts in Computer Science, 31st International Workshop (WG 2005)*, volume 3787 of *Lecture Notes in Computer Science*, pages 351–361. Springer, 2005.
- 50 Erik Jan van Leeuwen. Better approximation schemes for disk graphs. In Lars Arge and Rusins Freivalds, editors, *Algorithm Theory - SWAT 2006, 10th Scandinavian Workshop on Algorithm Theory*, volume 4059 of *Lecture Notes in Computer Science*, pages 316–327. Springer, 2006.
- 51 Mihalis Yannakakis. Node-deletion problems on bipartite graphs. *SIAM Journal on Computing*, 10(2):310–327, 1981.

Voronoi Diagrams in the Hilbert Metric

Auguste H. Gezalyan  

Department of Computer Science, University of Maryland, College Park, MD, USA

David M. Mount   

Department of Computer Science, University of Maryland, College Park, MD, USA

Abstract

The Hilbert metric is a distance function defined for points lying within a convex body. It generalizes the Cayley-Klein model of hyperbolic geometry to any convex set, and it has numerous applications in the analysis and processing of convex bodies. In this paper, we study the geometric and combinatorial properties of the Voronoi diagram of a set of point sites under the Hilbert metric. Given any m -sided convex polygon Ω in the plane, we present two randomized incremental algorithms and one deterministic algorithm. The first randomized algorithm and the deterministic algorithm compute the Voronoi diagram of a set of n point sites. The second randomized algorithm extends this to compute the Voronoi diagram of the set of n sites, each of which may be a point or a line segment. Our algorithms all run in expected time $O(mn \log n)$. The algorithms use $O(mn)$ storage, which matches the worst-case combinatorial complexity of the Voronoi diagram in the Hilbert metric.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Voronoi diagrams, Hilbert metric, convexity, randomized algorithms

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.35

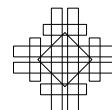
Related Version *Full Version:* <https://arxiv.org/abs/2112.03056>

1 Introduction

The Hilbert metric was introduced by David Hilbert in 1895 [12]. Given a convex body Ω in d -dimensional space, it defines a distance function between any pair of points in the interior of Ω . (Definitions are presented in Section 2.1.) The Hilbert geometry has a number of natural properties. For example, straight line segments are geodesics. It generalizes the Cayley-Klein model of hyperbolic geometry (on Euclidean balls) to any convex body. It is also invariant under projective transformations. Hilbert geometry provides new insights into classical questions from convexity theory. It also provides new insights into the study of metric and differential geometries (such as Finsler geometries). An excellent resource on the Hilbert geometries is the handbook on Hilbert geometry by Papadopoulos and Troyanov [18].

Hilbert geometry is also relevant to the topic of convex approximation. Efficient approximations of convex bodies have been applied to a wide range of applications, including approximate nearest neighbor searching both in Euclidean space [6] and more general metrics [1], optimal construction of ε -kernels [4], solving the closest vector problem approximately [10, 11, 14, 19], computing approximating polytopes with low combinatorial complexity [3, 5]. These works all share one thing in common – they approximate a convex body by covering it with elements that behave much like metric balls. These covering elements go under various names: Macbeath regions, Macbeath ellipsoids, Dikin ellipsoids, and $(2, \varepsilon)$ -covers. While these all behave like metric balls, the question is in what metric space? Vernicos and Walsh showed that these shapes are, up to a constant scaling factor, equivalent to Hilbert balls [2, 21].

In spite of its obvious appeals, there has been remarkably little work the design of algorithms in the Hilbert geometry on convex polygons and polytopes. Two notable exceptions are the work of Nielsen and Shao, which investigates properties and efficient construction of



Hilbert balls in convex polygons [15], and Nielsen and Sun, which investigates clustering in Hilbert simplex geometry [16].

In this paper, we investigate perhaps the most fundamental computational questions one might ask about a metric geometry: How to construct the Voronoi diagram of a set of sites, points and/or line segments, in the plane? Given any convex polygon Ω bounded by m sides, we present two randomized algorithms and one deterministic algorithm for computing the Voronoi diagram of an n -element point set in the Hilbert metric induced by Ω . The first randomized algorithm works for point sites, while the second works for point sites and segment sites. Both run in $O(mn \log n)$ expected time. The deterministic algorithm works for point sites in $O(mn \log n)$ time and uses a divide-and-conquer approach. Due to space limitations the deterministic algorithm will be presented in the arXiv version of the paper. All the algorithms use $O(mn)$ space. We show that the worst-case combinatorial complexity of the Voronoi diagram is $\Theta(mn)$, so all the algorithms are worst-case optimal up to logarithmic factors.

2 Preliminaries

Throughout, let Ω denote a *convex body* in \mathbb{R}^d , that is, a compact, full-dimensional convex set. Let $\partial\Omega$ and $\text{int}(\Omega)$ denote its boundary and interior, respectively. Given points $p, q \in \mathbb{R}^d$, let $\|p - q\|$ denote the Euclidean distance between these points. Given two distinct points $p, q \in \Omega$, let $\chi(p, q)$ denote the *chord* defined as the intersection of the line passing through p and q with Ω .

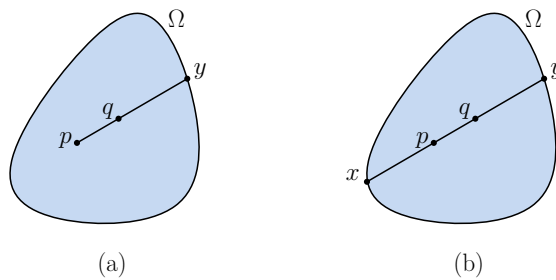
2.1 Funk and Hilbert Metrics

Before defining the Hilbert metric, it will be convenient to define a simpler (asymmetric) distance function called the *Funk weak metric*.

► **Definition 1** (Funk weak metric). *Given a convex body Ω in \mathbb{R}^d and two distinct points $p, q \in \text{int}(\Omega)$, let y denote point where a ray shot from p to q intersects $\partial\Omega$. Define the Funk weak metric to be*

$$F_{\Omega}(p, q) = \ln \frac{\|p - y\|}{\|q - y\|},$$

and define $F_{\Omega}(p, p) = 0$ (see Figure 1(a)).



■ **Figure 1** (a) The Funk weak metric and (b) the Hilbert metric.

Observe that in the limit as q approaches y along the chord $\chi(p, y)$ the Funk distance increases to $+\infty$, and thus, the boundary is infinitely far away from any interior point of Ω . The Funk weak metric is not symmetric, but it satisfies triangle inequality. Symmetrizing this yields a true metric, called the Hilbert metric [12].

► **Definition 2** (Hilbert metric). *Given a convex body Ω in \mathbb{R}^d and two distinct points $p, q \in \text{int}(\Omega)$, let x and y denote endpoints of the chord $\chi(p, q)$, so that the points are in the order $\langle x, p, q, y \rangle$. Define the Hilbert metric to be*

$$d_{\Omega}(p, q) = \frac{F_{\Omega}(p, q) + F_{\Omega}(q, p)}{2} = \frac{1}{2} \ln \frac{\|p - y\| \|q - x\|}{\|q - y\| \|p - x\|},$$

and define $d_{\Omega}(p, p) = 0$ (see Figure 1(b)).

Hilbert showed that line segments are geodesics, and if q lies on the line segment between p and r , then $d_{\Omega}(p, q) + d_{\Omega}(q, r) = d_{\Omega}(p, r)$. Note however that generally there may be multiple shortest paths between two points, and hence geodesics need not be line segments (see, e.g., [7]). As in the Funk weak metric, the boundary of Ω is infinitely far away from any interior point. Observe that the quantity in the \ln term in the definition is the cross ratio of $(p, q; y, x)$. It follows that the Hilbert metric is invariant under projective transformations. For further information, see the first chapters of the handbook on Hilbert geometry by Papadopoulos and Troyanov [18].

3 Hilbert Metric Balls

In this section we consider the properties of metric balls in the Hilbert metric. Given a convex body Ω in \mathbb{R}^d , $p \in \text{int}(\Omega)$, and $r \geq 0$, let us denote the *Hilbert ball* of radius r centered at p by

$$B_{\Omega}(p, r) = \{q \in \Omega : d_{\Omega}(q, p) \leq r\}.$$

We can extend the notion of distance to any nonempty set P in the natural way by taking the minimum distance to the set, that is, $d_{\Omega}(q, P) = \inf\{d_{\Omega}(q, p) : p \in P\}$. This allows us to talk about the Hilbert ball of other shapes, such as line segments. We will consider balls generated by both points and line segments.

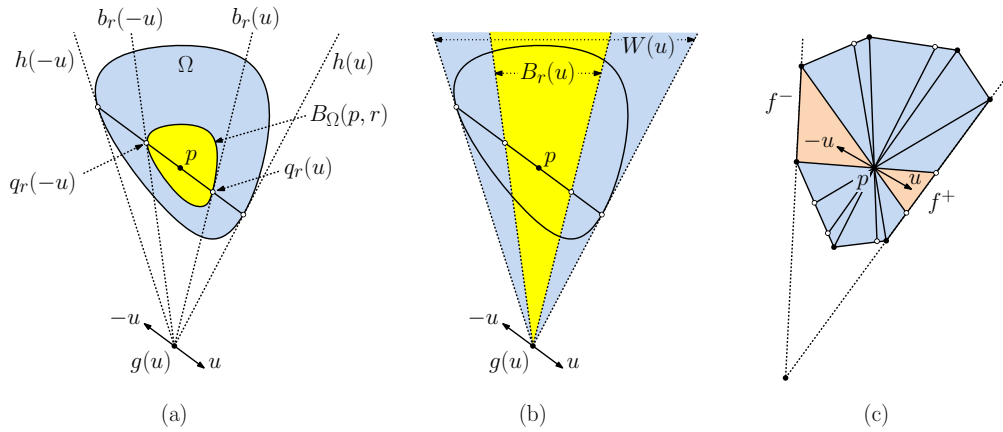
3.1 Hilbert Balls of Points

Nielsen and Shao [15] provided a characterization of Hilbert balls when Ω is an m -sided convex polygon in \mathbb{R}^2 , showing that the ball is a convex polygon bounded by at most $2m$ sides. We generalize their result to arbitrary convex polytopes in \mathbb{R}^d . This provides an alternative (and more elementary) proof that Hilbert balls are convex (also proved in [17, 20]).

► **Lemma 3.** *Given any convex polygon Ω in \mathbb{R}^2 bounded by m sides, $p \in \text{int}(\Omega)$, and $r \geq 0$, the ball $B_{\Omega}(p, r)$ is a convex polygon bounded by at most $2m$ sides. It can be constructed in $O(m)$ time.*

Proof. For now, let us take Ω to be any convex body (not necessarily a polytope) in \mathbb{R}^2 . Let \mathbb{S}^1 denote the unit sphere, that is the set of all unit vectors in \mathbb{R}^2 . For each $u \in \mathbb{S}^1$, consider the ray emanating from p in direction u , and define $q_r(u) \in \Omega$ to be the (unique) point along this ray whose Hilbert distance from p is r . Clearly, the boundary of $B_{\Omega}(p, r)$ is just the set of points $q_r(u)$ over all $u \in \mathbb{S}^1$.

For any $u \in \mathbb{S}^1$, let $h(u)$ denote any support line of Ω at the point where the ray emanating from p in the direction u intersects the boundary of Ω (see Figure 2(a)). Define $h(-u)$ analogously for the ray emanating from p in the direction $-u$. Let $g(u) (= g(-u))$ denote the point where $h(u)$ and $h(-u)$ intersect. (Through an infinitesimal perturbation of Ω , we may assume that $h(u)$ and $h(-u)$ are not parallel.) Let $b_r(u)$ denote the ray originating at $g(u)$ and passing through $q_r(u)$, and define $b_r(-u)$ analogously for $q_r(-u)$.



■ **Figure 2** The Hilbert ball of a point p .

Let $W(u)$ denote the convex wedge whose apex is at $g(u)$ and is bounded by $h(u)$ and $h(-u)$ (see Figure 2(b)). It follows from basic properties of projective geometry that every line passing through p on this plane is cut by $h(u)$, $h(-u)$, $b_r(u)$, and p into four points that share the same cross ratio. This implies that every point on $b_r(u)$ is at Hilbert distance r from p with respect to $W(u)$. The same applies symmetrically to $b_r(-u)$, and therefore the sub-wedge of $W(u)$ bounded by $b_r(u)$ and $b_r(-u)$ is just the Hilbert ball of radius r centered at p for $W(u)$. (This was observed by Nielsen and Shao in their analysis of the two-dimensional case.) Let us call this sub-wedge $B_r(u)$ (shaded in yellow in Fig. 2(b)).

Note that $p \in \Omega \subseteq B_r(u)$, and therefore Hilbert distances from p in $B_r(u)$ are at least as large as they are in Ω . Therefore, for all $u \in \mathbb{S}^1$, $B_\Omega(p, r) \subseteq B_r(u)$, and hence

$$B_\Omega(p, r) \subseteq \bigcap_{u \in \mathbb{S}^1} B_r(u).$$

On the other hand, by definition of $B_r(u)$, we know that $B_r(u)$ and $B_\Omega(p, r)$ both cover the exactly the same portion of the chord parallel to u passing through p . Since these chords cover all the boundary points of $B_\Omega(p, r)$, we have

$$B_\Omega(p, r) = \bigcap_{u \in \mathbb{S}^1} B_r(u).$$

Clearly, $B_r(u)$ is the intersection of a (possibly infinite) set of convex wedges, and so it is also convex.

Suppose now that Ω is an m -sided convex polygon. We say that two sides f^+ and f^- form a *complementary pair* if there exists $u \in \mathbb{S}^1$ such that the rays emanating from p in the directions u and $-u$ hit f^+ and f^- , respectively (see Figure 2(c)). We can partition the elements of \mathbb{S}^1 into equivalence classes according to the associated complementary pair. All the unit vectors u from any one equivalence class share the same support lines $h(u)$ and $h(-u)$, and therefore all of them contribute the same wedge $B_r(u)$ to $B_\Omega(p, r)$. By considering the chords $\chi(v, p)$ for each each of the m vertices of Ω , it follows directly that the number of complementary pairs is at most m . These chords partition Ω into at most m double wedges about p , each of which contributes two edges to the final ball, for a total of $2m$ sides.

The final Hilbert ball $B_\Omega(p, r)$ can be constructed in $O(m)$ time by observing that the sorted sequence of double wedges about p can be generated in linear time and then generating the two line segments bounding $B_r(u)$ within each double wedge (assuming a standard representation of Ω as a cyclic sequence of vertices). ◀

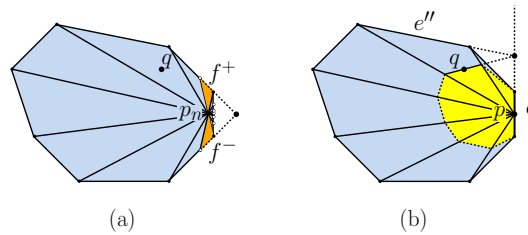
This result can be readily generalized to the higher dimensional case, but the worst-case number of bounding facets is quadratic in the number of facets. (Due to space limitations, the proof appears in the arXiv of the version of the paper.)

► **Lemma 4.** *Given any convex polytope Ω in \mathbb{R}^d bounded by m facets, $p \in \text{int}(\Omega)$, and $r \geq 0$, $B_\Omega(p, r)$ is a convex polytope bounded by $O(m^2)$ facets.*

Given the characterization on Hilbert ball in $\Omega \in \mathbb{R}^2$, a natural question arises, how can we define Hilbert balls for points at infinity, that is, on $\partial\Omega$? Let q be a point in Ω , we define the Hilbert ball at $p \in \partial\Omega$ whose boundary passes through q to be limit of any sequence of Hilbert balls passing through q whose centers approach p . We show this limit is unique and yields a convex polygon with at most m sides.

► **Lemma 5.** *Given any convex polygon Ω in \mathbb{R}^2 bounded by m sides, $p \in \partial\Omega$, $q \in \text{int}(\Omega)$, there exists a unique ball, B , centered at p containing q on its boundary which is a convex polygon with at most m sides.*

Proof. Let us begin by considering an arbitrary sequence of Hilbert balls $\{B_n\}_{n \in \mathbb{N}}$ centered at points p_n such that $\{p_n\}_{n \in \mathbb{N}}$ converges to a point p on a side e of Ω (see Fig. 3(a)). Since our sequences converges to p there must be some $n_0 \in \mathbb{N}$ such that $\forall n \geq n_0$ in the sequence the wedges that contributes the boundary sides to $B_n, n \geq n_0$ all share the side e except the one made by the spokes through the vertices of e and p . Call this other wedge W with sides f^+ and f^- . It can be see that as $\{p_n\}_{n \geq n_0}$ approaches p , the measure of the region contributed by the wedge W vanishes to 0. Likewise, for any wedge W' between some e' and e , the portion of the boundary contributed between p_n and e vanishes in the limit. Hence we need only look at the boundaries contributed by the wedges between e and other sides of Ω .



■ **Figure 3** The Hilbert ball of a point p on $\partial\Omega$ through a point q .

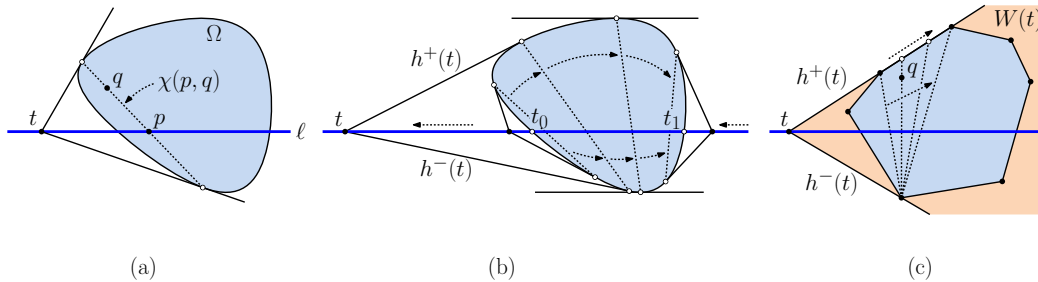
To construct this boundary let q lie on the wedge between e and e'' from the perspective of p_n . By tracing the line through q to the apex of the wedge we are able to determine the boundary of an arbitrary p_n through q and a point on the boundaries of the neighboring wedges, from here we can walk along the wedges using the same process to complete the boundary. As p_n approaches p we can see that this converges to the unique polygonal boundary made by the same procedure applied to p . Since each wedge contributes one side to the boundary and there are $m - 1$, paired with the edge e this gives the ball centered at p has at most m sides. ◀

3.2 Hilbert Balls of Lines and Line Segments

In this section we will describe the ball of radius r about an arbitrary line segment in Ω . Let us first consider the case of line ℓ that intersects Ω . Let $B_\Omega(\ell, r)$ denote the set of points of Ω that lie within Hilbert distance r of ℓ . We will consider just the 2-dimensional case, but the general case in \mathbb{R}^d can be generated by considering the union of 2-dimensional slices generated by planes passing through ℓ . Given any point q in the interior of Ω , the following lemma characterizes the point of ℓ that is closest to q .

► **Lemma 6.** *Given any convex body Ω in \mathbb{R}^2 in general position and a line ℓ that intersects the interior of Ω , for any $q \in \text{int}(\Omega)$ that is not on ℓ , its closest point on ℓ is uniquely determined to be the point $p \in \ell$ such that there exist support lines from each of the endpoints of the chord $\chi(q, p)$ that intersect at a point lying on ℓ (see Fig. 4(a)).*

Proof. We begin by showing that, assuming general position, for any point $q \in \text{int}(\Omega)$, there exists a unique point p satisfying the desired properties. We first consider the simpler case where Ω is strictly convex, and we will generalize later. Let us also assume for the sake of illustration that ℓ is horizontal. Define t_0 and t_1 to be the leftmost and rightmost points of intersection of ℓ with Ω , respectively (see Fig. 4(b)).



■ **Figure 4** The closest point to q on line ℓ .

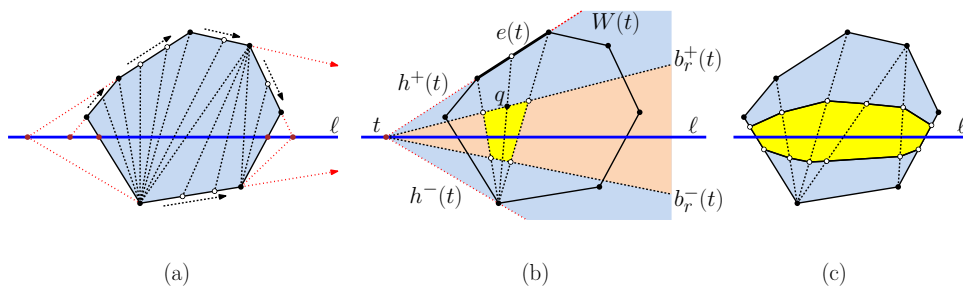
Consider a point t that moves continuously along $\ell \setminus \Omega$, which starts at t_0 , moves to the left until reaching $x = -\infty$, then wraps around to $x = +\infty$, and finally moves to t_1 . At each such point t , let $h^+(t)$ and $h^-(t)$ denote the two support lines of Ω that pass through t above and below ℓ , respectively. Since Ω is strictly convex, each pair of supporting lines defines a unique chord by joining the points where these support lines intersect $\partial\Omega$. It is a straightforward consequence of convexity that points where these support lines intersect $\partial\Omega$ move continuously and monotonically along both sides of the boundary of Ω from t_0 to t_1 . Thus, every point of Ω lies on exactly one of these chords. Given point $q \in \text{int}(\Omega)$, the chord passing through q satisfies the conditions of the lemma. (Note that if Ω has vertices, multiple points t may generate the same chord. So even though the chord is unique, the associated supporting lines need not be.)

We can generalize to the case where Ω is not strictly convex. Suppose, for example, that it is an m -sided convex polygon. We may assume through an infinitesimal perturbation of ℓ that the linear extensions of any two edges of Ω do not intersect on ℓ . If we extend each edge of Ω until it intersects ℓ , we have m distinct points on $\ell \setminus \Omega$. It is no longer true that each point t generates a unique chord, but each point along the extended edge generates a chord in conjunction with the supporting vertex on the opposite side ℓ (see Fig. 4(c)). It follows from monotonicity that for any $q \in \text{int}(\Omega)$, exactly one of these chords passes through q .

Because the endpoints of each of these chords lies on opposite sides of ℓ , each chord intersects ℓ . We assert that the point p where this chord intersects ℓ is the unique closest point to q . To see why, consider the wedge $W(t)$ bounded by $h^+(t)$ and $h^-(t)$. Because the three lines ℓ , $h^+(t)$, and $h^-(t)$ are coincident at t , every line passing through q cuts these lines so that the cutting points have the same cross ratios. It follows that every point on ℓ is equidistant to q with respect to the Hilbert distance defined by $W(t)$. But (assuming general position) Ω intersects each such line through q along a strictly smaller subsegment, and hence the Hilbert distance between q and any point on ℓ other than p is strictly larger than $d_\Omega(q, p)$. Therefore, p is the unique closest point to q on ℓ . ◀

► **Lemma 7.** *Given any convex polygon Ω in \mathbb{R}^2 bounded by m sides, any line ℓ that intersects the interior of Ω , and $r \geq 0$, the Hilbert ball $B_\Omega(\ell, r)$ is a convex polygon bounded by at most $2m$ sides. Furthermore, this ball can be constructed in $O(m)$ time.*

Proof. Our proof is constructive. As in Lemma 6, let us assume that ℓ is horizontal. Consider the m points where the linear extensions of the edges of Ω intersect ℓ . We assume by general position that these points are distinct. As mentioned in the earlier proof, each of these points t generates two support lines $h^+(t)$ and $h^-(t)$ lying above and below ℓ , respectively. It also generates a continuous family of chords, by joining each point on the corresponding side $e(t)$ of Ω to the vertex of Ω through which the other support line passes (see Fig. 5(a) and (b)).



■ **Figure 5** The Hilbert ball of a line ℓ .

Let $W(t)$ denote the double wedge bounded by $h^+(t)$ and $h^-(t)$. Given the radius r , we can draw a ray $b_r^+(t)$ emanating from t and passing above ℓ so that for every point q on $b_r^+(t)$, its Hilbert distance from ℓ with respect to $W(t)$ is r . As shown in Lemma 6, if q also lies on one of the chords generated by edge $e(t)$, then this is also true with respect to the Hilbert distance defined by Ω . Therefore, such a point q lies on the upper boundary of $B_\Omega(\ell, r)$ (see Fig. 5(b)). Analogously, we can find the ray $b_r^-(t)$, which generates the lower boundary of $B_\Omega(\ell, r)$ for the chords generated by $e(t)$.

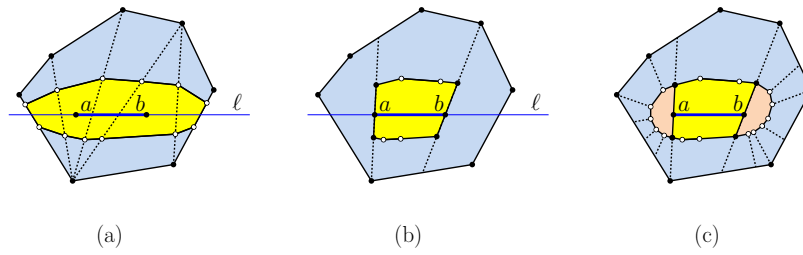
All that remains to complete the entire ball is to repeat this process for each of the m points t where the extensions of the sides of Ω intersect ℓ . The ball $B_\Omega(\ell, r)$ is clearly a simple polygon bounded by at most $2m$ sides. Convexity follows by observing that, due to the monotonicity of the points t along the line ℓ , the slopes of its sides along the upper boundary of the ball decrease monotonically and the slopes of sides along the lower boundary of the ball increase monotonically. ◀

Given two points $a, b \in \Omega$, the Hilbert ball of a line segment \overline{ab} can be formed by first constructing the Hilbert ball for the line ℓ that passes through these points (see Fig. 6(a)), then cutting this ball through the chords of the line-ball construction passing through points a and b (see Fig. 6(b)), and finally filling in the missing ends with the portions of the Hilbert balls centered at the points a and b (see Fig. 6(c)).

► **Lemma 8.** *Given any convex polygon Ω in \mathbb{R}^2 bounded by m sides, any line segment \overline{ab} for $a, b \in \Omega$, and $r \geq 0$, the Hilbert ball $B_\Omega(\overline{ab}, r)$ is a convex polygon bounded by at most $4m$ sides. Furthermore, this ball can be constructed in $O(m)$ time.*

4 Characterizing Voronoi Diagrams in the Hilbert Metric

Using our understanding of Hilbert balls we can characterize the Voronoi diagram of a set of point sites in the Hilbert metric. Throughout, let Ω denote a convex polygon in \mathbb{R}^2 , and unless otherwise stated, distances will be in the Hilbert metric induced by Ω , which we



■ **Figure 6** Three-step process for building the Hilbert ball of a line segment \overline{ab} .

denote by $d_\Omega(\cdot, \cdot)$ or simply $d(\cdot, \cdot)$, when Ω is clear. Let S denote a set of n points lying within the interior of Ω , which we call *sites*. For $p \in S$, define its *Voronoi cell* to be

$$V(p) = V_S(p) = \{q \in \Omega : d(q, p) \leq d(q, p'), \forall p' \in S \setminus \{p\}\}.$$

Although points on the boundary of Ω are infinitely far from points in the interior of Ω , we can compare the relative distances between a fixed boundary point and two interior points by considering the limit as an interior point approaches this boundary point. The *Voronoi diagram* of S in the Hilbert metric induced by Ω , denoted $\text{Vor}_\Omega(S)$, is the cell complex of Ω induced by the Voronoi cells $V(p)$ for all $p \in S$. We assume that the points of S are in general position, and in particular, the line passing through any pair of sites of S and the lines extending any two edges of Ω are not coincident at a common point (including all three being parallel). If this assumption does not hold, the bisectors separating Voronoi cells can widen into 2-dimensional regions.

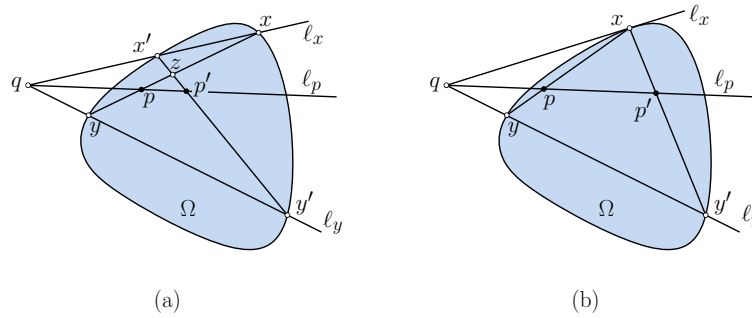
Recall that a set $R \subseteq \mathbb{R}^d$ is a *star* (or is *star-shaped*) with respect to a point $p \in R$ if for each $q \in R$, the segment pq lies within R . We next show that Hilbert Voronoi cells are *stars*.

► **Lemma 9.** *Voronoi cells in the Hilbert Metric are stars with respect to their defining sites.*

Proof. If this were not the case, there would exist a site p and points $x, y \in \Omega$ such that $x \in V(p)$, $y \notin V(p)$, and y lies on the line segment px . By collinearity, $d(p, x) = d(p, y) + d(x, y)$. Letting q be the closest site to y , we have $d(q, y) < d(p, y)$ and $d(p, x) \leq d(q, x)$. Combining these we have $d(q, x) + d(p, y) > d(q, y) + d(p, x)$, or equivalently $d(q, x) > d(q, y) + d(x, y)$. But this violates the triangle inequality, yielding a contradiction. ◀

4.1 Bisectors in the Hilbert Metric

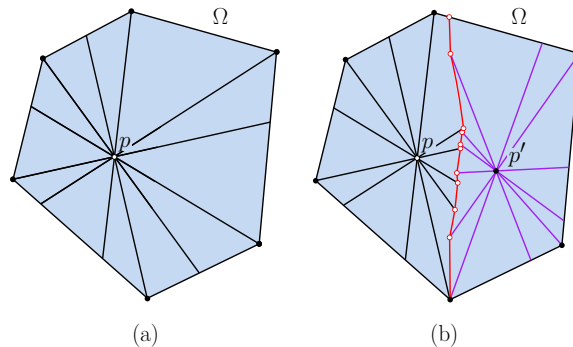
Given two sites $p, p' \in \text{int}(\Omega)$, we define their Hilbert bisector, denoted the (p, p') -bisector, to be $\{z \in \Omega : d_\Omega(z, p) = d_\Omega(z, p')\}$. We will explore the conditions for a point z to lie on the bisector. Let x and y denote the endpoints of the chord $\chi(z, p)$, and define x' and y' analogously for $\chi(z, p')$. Label these points in the order $\langle x, z, p, y \rangle$ and $\langle x', z, p', y' \rangle$ (see Figure 7(a)). Finally, let ℓ_x, ℓ_p , and ℓ_y denote the lines passing through the line segments xx', pp' and yy' , respectively. If these three lines are coincident on some point q then by basic properties of projective geometry, the cross ratios $(z, p; y, x)$ and $(z, p'; y', x')$ are equal. It follows that $d_\Omega(z, p) = d_\Omega(z, p')$, and hence z is on the bisector. If not, then the cross ratios are different and the Hilbert distances are different. Observe that as z approaches the boundary of Ω (on the same side of ℓ_p as ℓ_x), the points z, x , and x' converge on a common point on $\partial\Omega$, and ℓ_x approaches a support line for Ω at this point (see Figure 7(b)). Thus, we obtain the following characterization of bisector points.



■ **Figure 7** Conditions for a point z to lie on the Hilbert bisector between sites p and p' .

► **Lemma 10.** *Given a convex body Ω in \mathbb{R}^2 , sites $p, p' \in \text{int}(\Omega)$ and any other point $z \in \text{int}(\Omega)$, z lies on the (p, p') -bisector if and only if lines ℓ_x, ℓ_p , and ℓ_y (defined above) are coincident. Further, a point $x \in \partial\Omega$ is on the Hilbert bisector if the coincidence holds when ℓ_x is any support line at x .*

When Ω is an m -sided convex polygon in \mathbb{R}^2 , we can provide a more precise characterization of the bisectors. Given two sites $p, p' \in \text{int}(\Omega)$, we will show below that the (p, p') -bisector is a piecewise conic (see Lemma 14). Assuming this for now, we can characterize the breakpoints in this curve. Letting $\{v_1, \dots, v_m\}$ denote the vertices of Ω , the $2m$ chords $\chi(v_i, p)$ subdivide Ω into $2m$ triangular regions, which we call p 's sectors with respect to Ω (see Figure 8(a)).



■ **Figure 8** (a) The sectors of p with respect to Ω and (b) the Hilbert bisector (in red) between p and p' .

4.2 Bisector Segments and Combinatorial Complexity

For each point z on the (p, p') -bisector, let e and f denote the edges where the endpoints of chord $\chi(z, p)$ intersect $\partial\Omega$, and let e' and f' denote the corresponding edges of $\chi(z, p')$. Observe that the pair (e, f) is uniquely determined by the sector of p containing z , and the pair (e', f') is similarly determined by the sector of p' containing z . Therefore, the points of the (p, p') -bisector can be grouped into a discrete set of equivalence classes based on their sector memberships with respect to p and p' (see Figure 8(b)). We refer to these as *bisector segments*. Observe that by the star-shaped nature of Voronoi cells, as we travel along the bisector, we encounter the (up to $2m$) sectors of p in cyclic order (say clockwise as in the figure), and we visit (up to $2m$) the sectors of p' in the opposite cyclic order (counterclockwise). We will see below that, each bisector segment can be described by a

simple parametric function involving p, p' , and the four edges defining the equivalence class. Star-shapedness implies that the bisector is simply connected. Combining the total number of sectors involved for each site, we have:

► **Lemma 11.** *Given an m -sided convex polygon Ω in \mathbb{R}^2 and sites $p, p' \in \text{int}(\Omega)$, the (p, p') -bisector is a simply connected piecewise curve consisting of at most $4m$ bisector segments.*

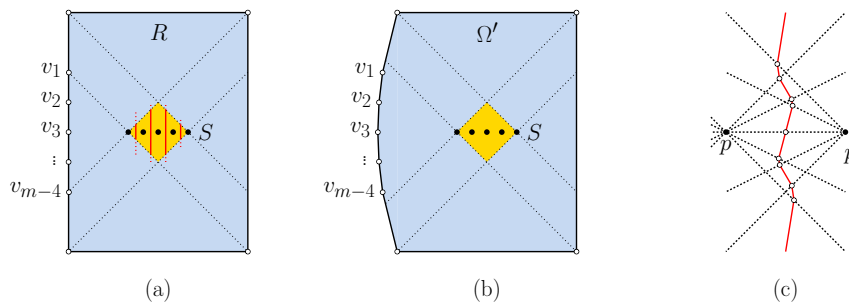
Given a set of n sites S in Ω , the Voronoi diagram consists of a collection of n Voronoi cells. The intersection of two cells $V(p)$ and $V(p')$ (if nonempty) is a portion of (p, p') -bisector, which we call a *Voronoi edge*. As shown in the above lemma, each such edge is composed of at most $4m = O(m)$ bisector segments. A cell's boundary may also contain a portion of the boundary of Ω . The intersection of two Voronoi edges (if nonempty) is a *Voronoi vertex*. Because the diagram is a planar graph, we have the following bounds by a straightforward application of Euler's formula.

► **Lemma 12.** *Given an m -sided convex polygon Ω in \mathbb{R}^2 and a set of n sites S in Ω , $\text{Vor}_\Omega(S)$ has n Voronoi cells, at most $3n$ Voronoi edges, and at most $2n$ Voronoi vertices. Each Voronoi edge consists of at most $4m$ bisector segments. Therefore, the entire diagram has total combinatorial complexity $O(mn)$. The average number of Voronoi edges per cell is $O(1)$, and the average number of bisector segments per cell is $O(m)$.*

The following lemma shows that the bound on the combinatorial complexity is tight in the worst case.

► **Lemma 13.** *For all sufficiently large m and n , there exists a convex polygon Ω' with m sides and a set S of n sites within Ω' such that $\text{Vor}_{\Omega'}(S)$ has combinatorial complexity $\Omega(mn)$. (Here we are using $\Omega(\cdot)$ in the asymptotic sense.)*

Proof. We start the construction with an axis parallel rectangle R that is slightly taller than wide, and the set S consists of n points positioned on a short horizontal line segment near the center of this rectangle (see Figure 9(a)). (This violates our general position assumptions, but the construction works if we perturb the sides of the rectangle.) Also, create a set of $m - 4$ points $V = \{v_1, \dots, v_{m-4}\}$ along the left edge of R . We can adjust the spacing of points of V and S and the side lengths of R so that there exists a diamond shape (shaded in orange in the figure) so that for any point q in this region, the chord $\chi(v_i, q)$ intersects the boundary of R along its vertical sides. Observe that within the diamond, each consecutive pair of sites contributes an edge to the Voronoi diagram, which is easily verified to be a vertical line segment within the diamond.



■ **Figure 9** Proof of Lemma 13.

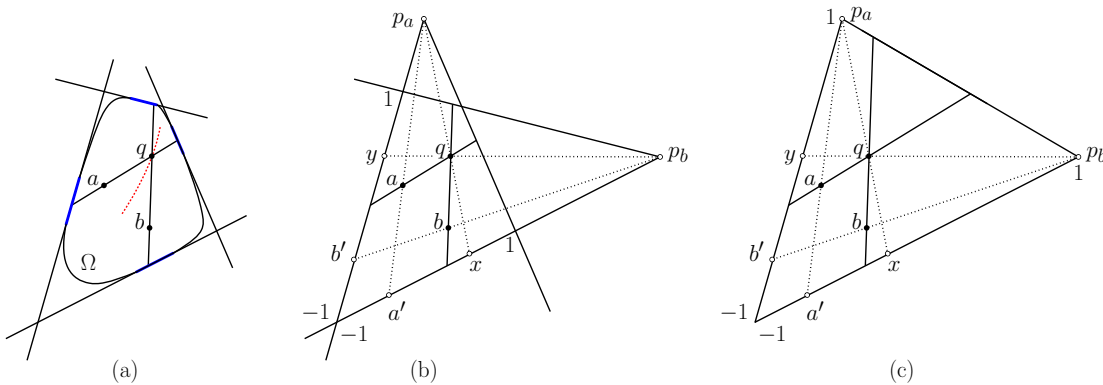
To form Ω' , we bend the left side of the rectangle out infinitesimally (see Figure 9(b)) so the points of V become vertices of a convex polygon. The bending does not alter the shape of the Voronoi edges significantly, except to break each edge up into multiple arcs.

We assert that each consecutive pair of points $p, p' \in S$ contributes at least $m - 4$ arcs to the Voronoi diagram. To see why, observe that there are $m - 4$ sectors about p and p' , and vertical line passing between p and p' intersects the boundaries between these sectors (see the red curve in Figure 9(c)). Due to the infinitesimal bending of the left side of R , each crossing produces a new segment on the bisector. Since there are $n - 1$ consecutive pairs of sites, the total complexity of $\text{Vor}_{\Omega'}(S)$ is at least $(m - 4)(n - 1) = \Omega(mn)$, as desired. ◀

4.3 Bisector Segments Structure

In this section we discuss the properties of bisectors in the Hilbert metric. We consider cases involving both points, lines, and combinations thereof. In particular, we prove that the bisectors are piece-wise conics.

First, we will characterize the local structure of a bisector between two point sites a and b with respect to Ω . Let us hypothesize that a point $q \in \text{int}(\Omega)$ lies on the bisector between these sites (see Fig. 10(a)). Draw chords \overline{aq} and \overline{bq} . Let us make the general-position assumption that these chords do not intersect vertices of Ω . Consider the two edges of Ω incident chord \overline{aq} , and let p_a be the point (possibly at infinity) where the linear extensions of these edges intersect (see Fig. 10(b)). Define p_b analogously for the chord \overline{bq} .



■ **Figure 10** Segments of a point-point bisector.

If there are only two distinct edges involved (implying that $p_a = p_b$), the problem can be reduced to a simple 1-dimensional case by projection through the point where the extensions of the two edges meet. Otherwise, there are two wedges with apexes at the points p_a and p_b such that Ω is contained within their intersection. The shape resulting from the intersection of the two wedges may be a convex quadrilateral (as shown in Fig. 10(b)), but if two edges coincide, we obtain a triangle (see Fig. 10(c)). Also, this shape may be unbounded, which we treat as wrapping around the projective plane.

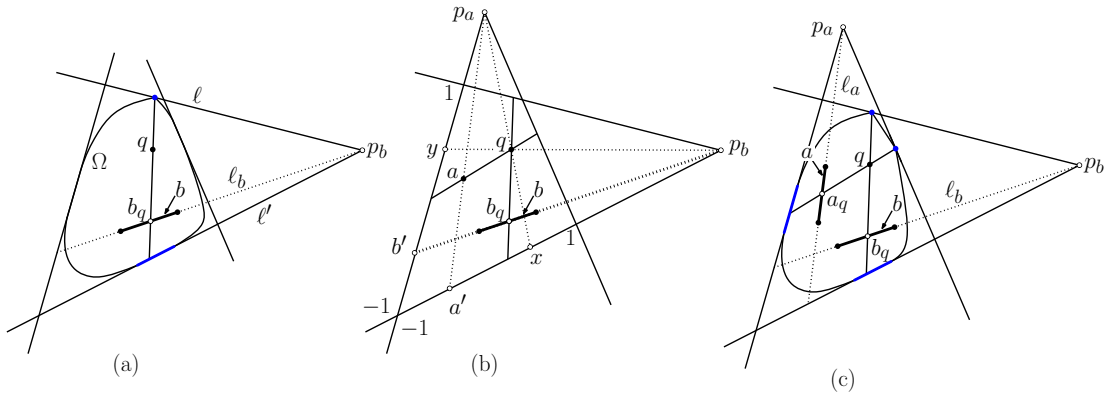
Shoot two rays originating at p_a passing through a and q , and let a' and x denote the respective points these rays hit on the opposite side of the quadrilateral (see Fig. 10(b)). Define b' and y analogously for rays originating at p_b through b . Let us parameterize points on these two edges so that -1 and 1 are the edge endpoints, and the points along the edge appear in the orders $\langle -1, a', x, 1 \rangle$ and $\langle -1, b', y, 1 \rangle$, respectively. We can now think of points $\{a', x, b', y\}$ as reals, where $-1 < a' \leq x < 1$ and $-1 < b' \leq y < 1$. (Due to space limitations, the proof appears in the arXiv version of the paper.)

► **Lemma 14.** *Given a convex polygon Ω and two sites $a, b \in \text{int}(\Omega)$, the bisector between a and b in the Hilbert metric is a piecewise conic curve.*

35:12 Voronoi Diagrams in the Hilbert Metric

Next, let us consider the case where a is a point site and $b = \overline{b_0b_1}$ is a line segment, where $a, b_0, b_1 \in \Omega$. As before, we wish to characterize the set of points q lying on the Hilbert bisector between a and b with respect to Ω . Let ℓ_b denote the linear extension of this line segment.

Recall from Lemma 6 that the closest point on ℓ_b is uniquely determined to be the point b_q such that there exist support lines to Ω at the endpoints of the chord $\chi(q, b_q)$ that intersect at a point on ℓ_b (see Fig. 11(a)).



■ **Figure 11** (a) Closest point on b to point q , (b) point-segment bisector, (c) segment-segment bisector.

Given the point p_b , we carry out the same construction as in the point-point bisector case, where the point p_a is defined exactly as before. We centrally project the points a and q through p_a until they hit the far side of the p_b -wedge, and symmetrically we centrally project the segment b and point q through p_b until they hit the far side of the p_a -wedge (see Fig. 11(b)). (Note that the entire segment b projects to the single point b' .) The values x and y , which determine curve containing the bisector, satisfy exactly the same conditions here as they do in the point-point case.

The segment-segment case is exactly analogous. Again, we find the point p_a through which the supporting lines pass, and obtain the same parameterization through central projections. The values x and y , which determine curve containing the bisector, satisfy exactly the same conditions here as they do in the point-point case.

It is interesting that unlike the Euclidean case, at a local level there is no fundamental difference between Hilbert bisectors in the point-point case and bisectors in the point-line or line-line cases. They are all conics, which may degenerate to lines in special configurations.

► **Lemma 15.** *Given a convex polygon Ω and two sites $a, b \in \text{int}(\Omega)$, either or both of which may be points or line segments, the bisector between a and b in the Hilbert metric is a piecewise conic curve.*

5 Randomized Incremental Algorithm

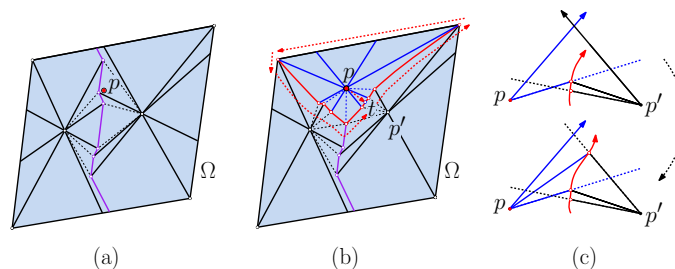
In this section we present two randomized incremental algorithms to compute the Hilbert Voronoi diagram of a set S of n sites. In both cases, the domain is an m -sided convex polygon Ω . First, we describe the algorithm in the case when S of n point sites. Second, we explain the changes when the sites are line segments. Both algorithms follow the structure of the randomized incremental algorithm for abstract Voronoi diagrams by Mehlhorn, Meiser, and Ó'Dúnlaing [13]. (A discussion on the relation of the Hilbert metric to abstract Voronoi diagrams can be found in the arXiv version of the paper.)

5.1 Point Sites

In this section we consider the construction of a Voronoi diagram for a set S of n point sites. The algorithm given in [13] starts by computing a bounding enclosure, but our body Ω serves that role. Let S be our set of sites. Consider an arbitrary iteration of the algorithm, and let R denote the set of sites already added to the diagram. To facilitate the constructing of our algorithm we maintain two data structures: $\text{Vor}(R)$ the current Voronoi diagram of R as a cell complex, and $G(R)$ the *conflict graph* of R with respect to S (defined below).

$\text{Vor}(R)$ may be stored in any standard data structure for planar subdivisions, such as a doubly connected edge list [9]. The vertices of $\text{Vor}(R)$ are the Voronoi vertices and the edges are the bisectors connecting them. To facilitate efficient tracing of bisectors, we augment each Voronoi cell with *spokes*, that is, line segments emanating from each site in the diagram to the segment vertices of the bisectors forming its Voronoi cell (see Fig. 12).

The conflict graph $G(R)$ is defined as follows. Its vertices consist of the edges of $\text{Vor}(R)$ together with the remaining sites $S \setminus R$. There is an edge between $e \in \text{Vor}(R)$ and $p \in S \setminus R$ if and only if the addition of p would result in edge e either being removed or trimmed.



■ **Figure 12** Inserting a new site in the randomized algorithm.

The algorithm randomly permutes the sites and inserts them one by one. The first two sites are inserted by brute force and the conflict graph is initialized based on the single bisector between these sites. Otherwise, let us assume that we have already inserted some subset R of sites and are now inserting the next site $p \in S \setminus R$. We need to (1) trace the boundary of p 's Voronoi cell in $\text{Vor}(R \cup \{p\})$ and (2) update the conflict graph. To trace a bisector between two sites we use the following method:

Let $V(p)$ denote the Voronoi cell of p after its insertion. We select any edge e that conflicts with p and we compute any point t on this edge that lies within p 's Voronoi cell. We proceed to walk along the bisector, continuously maintaining the distances to the associated sites until reaching a point on the boundary of $V(p)$. Starting at this point, we proceed to trace the boundary of $V(p)$ in counterclockwise order about p (see Figure 12(b)). This involves two types of traces:

Bisector Trace: We are tracing the bisector between p and some existing site p' (see Figure 12(c)). We consider the sectors of p and p' containing the current portion of the (p, p') -bisector. By applying the parameterization from Lemma 14, we can trace the bisector until (1) we encounter the boundary of either sector, (2) we encounter the bisector between p' and another site, or (3) we encounter the boundary of Ω .

In the first case, we create a new segment vertex here, add spokes to this vertex, erase the extension of the sector edge (shown as a broken line in Figure 12(c)) and continue the tracing in the new sector. In the second case, we create a new Voronoi vertex, add spokes to this vertex from its three defining sites, and continue the trace along the new

bisector. In the third case, we insert two spokes joining the point where the bisector encounters the boundary to p and p' , respectively. We then transition to the boundary trace described next.

Boundary Trace: We are tracing the Voronoi cell of p along the boundary of Ω . We walk along the boundary of Ω in counterclockwise order, considering each consecutive sector of the closest site p' , prior to p 's insertion. By applying the parameterization from Lemma 14, we determine whether the (p, p') -bisector intersects the boundary of Ω within the intersection of the current pair of sectors. If so, we identify this point on the $\partial\Omega$, add spokes to each of p and p' , and then resume tracing along the (p, p') -bisector. Otherwise, we encounter one of the two sector boundaries, and we continue the tracing the next sector.

Along the way, we erase spokes from the current diagram, and/or introduce new spokes connected to p . When we return to our starting point, the insertion is completed.

While we are tracing the bisector for the new site, we can simultaneously update the conflict graph. Whenever we arrive at a new Voronoi vertex in the diagram, we construct a Hilbert ball at this point whose radius is the distance to the newly added site p (recall Lemma 3). This includes Voronoi vertices “at infinity”, which arise whenever a bisector extends all the way to the boundary of Ω . Let p' and p'' be the two other sites of R that share this vertex. All the sites of $S \setminus R$ in this ball are added as conflict-graph neighbors of the two new Voronoi edges between p and p' and p and p'' .

The algorithm given in [13] runs in $O(n \log n)$ expected time, under the *update assumption* of Clarkson and Shor [8], which states that the time to insert a new site p is proportional to the number of objects that conflict with regions that conflict with p . In our context, we need to add an additional factor of $O(m)$. First, this is needed because bisectors are no longer of constant complexity, but consist of $O(m)$ segments. Our algorithm processes each new segment in $O(1)$ time. Second, when updating the conflict graph, in order to determine which points conflict with the newly created edges, we need to construct a Hilbert ball at each newly created Voronoi vertex, and this takes time $O(m)$. The overall expected running time is thus larger by a factor of $O(m)$, implying the following.

► **Theorem 16.** *Given an m -sided convex polygon Ω in \mathbb{R}^2 and a set of n point sites S in Ω , the randomized incremental algorithm computes $\text{Vor}_\Omega(S)$ in expected time $O(mn \log n)$ (where the expectation is over all possible insertion orders).*

5.2 Segment Sites

In this section, we present a sketch of the modifications necessary to generalize the algorithm from Section 5.1 to the case of line segments (due to space limitations, a more detailed account of the changes can be found in the arXiv version of the paper). Since the algorithm by Mehlhorn, Meiser, and Ó'Dúnlaing [13] is quite general, the approach applies here as well, but with a few changes.

First, we need to modify the notion of spokes to apply to segment objects. This is done by joining each bisector segment endpoint to its closest point on the segment (see Section 3.2). Second, the sector tracing process will be changed, but the complexity of the bisector is unaffected (see Lemma 15). Finally, the updating of the conflict graph is different, due to the fact that the circumcircles involved are now defined by combinations of point sets and segment sites.

Other than these changes, the algorithm and its analysis go through exactly as before. We suffer the same additional $O(m)$ in the running time due to complexity of Ω .

► **Theorem 17.** *Given an m -sided convex polygon Ω in \mathbb{R}^2 and a set of n point and line-segment sites S in Ω , the randomized incremental algorithm computes $\text{Vor}_\Omega(S)$ in expected time $O(mn \log n)$ (where the expectation is over all possible insertion orders).*

References

- 1 Ahmed Abdelkader, Sunil Arya, Guilherme Dias da Fonseca, and David M. Mount. Approximate nearest neighbor searching with non-Euclidean and weighted distances. In *Proc. 30th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 355–372, 2019. doi:10.1137/1.9781611975482.23.
- 2 Ahmed Abdelkader and David M. Mount. Economical Delone sets for approximating convex bodies. In *Proc. 16th Scand. Workshop Algorithm Theory*, pages 4:1–4:12, 2018.
- 3 Rahul Arya, Sunil Arya, Guilherme Dias da Fonseca, and David M. Mount. Optimal bound on the combinatorial complexity of approximating polytopes. *ACM Trans. Algorithms*, 18:1–29, 2022. doi:10.1145/3559106.
- 4 Sunil Arya, Guilherme Dias da Fonseca, and David M. Mount. Near-optimal ε -kernel construction and related problems. In *Proc. 33rd Internat. Sympos. Comput. Geom.*, pages 10:1–15, 2017.
- 5 Sunil Arya, Guilherme Dias da Fonseca, and David M. Mount. On the combinatorial complexity of approximating polytopes. *Discrete Comput. Geom.*, 58(4):849–870, 2017. doi:10.1007/s00454-016-9856-5.
- 6 Sunil Arya, Guilherme Dias da Fonseca, and David M. Mount. Optimal approximate polytope membership. In *Proc. 28th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 270–288, 2017.
- 7 Herbert Busemann. *The Geometry of Geodesics*. Academic Press, 1955.
- 8 Kenneth L. Clarkson and Peter W. Shor. Applications of random sampling in computational geometry, ii. *Discrete Comput. Geom.*, 4:387–421, 1989. doi:10.1007/BF02187740.
- 9 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 3rd edition, 2010.
- 10 Friedrich Eisenbrand, Nicolai Hähnle, and Martin Niemeier. Covering cubes and the closest vector problem. In *Proc. 27th Annu. Sympos. Comput. Geom.*, pages 417–423, 2011.
- 11 Friedrich Eisenbrand and Moritz Venzin. Approximate CVPs in time $2^{0.802n}$. *J. Comput. Sys. Sci.*, 124:129–139, 2021. doi:10.1016/j.jcss.2021.09.006.
- 12 D. Hilbert. Ueber die gerade Linie als kürzeste Verbindung zweier Punkte. *Math. Annalen*, 46:91–96, 1895.
- 13 K. Mehlhorn, St. Meiser, and C. Ó’Dúnlaing. On the construction of abstract Voronoi diagrams. *Discrete Comput. Geom.*, 6:211–224, 1991. doi:10.1007/BF02574686.
- 14 Márton Naszódi and Moritz Venzin. Covering convex bodies and the closest vector problem. *Discrete Comput. Geom.*, 67:1191–1210, 2022. doi:10.1007/s00454-022-00392-x.
- 15 Frank Nielsen and Laetitia Shao. On balls in a Hilbert polygonal geometry (multimedia contribution). In *Proc. 33rd Internat. Sympos. Comput. Geom.*, volume 77 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 67:1–67:4. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.SocG.2017.67.
- 16 Frank Nielsen and Ke Sun. Clustering in Hilbert’s projective geometry: The case studies of the probability simplex and the ellipotope of correlation matrices. In Frank Nielsen, editor, *Geometric Structures of Information*, pages 297–331. Springer Internat. Pub., 2019. doi:10.1007/978-3-030-02520-5_11.
- 17 Athanase Papadopoulos and Marc Troyanov. From Funk to Hilbert geometry. In *Handbook of Hilbert geometry*, volume 22 of *IRMA Lectures in Mathematics and Theoretical Physics*, pages 33–68. European Mathematical Society Publishing House, 2014. doi:10.4171/147-1/2.

35:16 Voronoi Diagrams in the Hilbert Metric

- 18 Athanase Papadopoulos and Marc Troyanov. *Handbook of Hilbert geometry*, volume 22 of *IRMA Lectures in Mathematics and Theoretical Physics*. European Mathematical Society Publishing House, 2014. doi:10.4171/147.
- 19 Thomas Rothvoss and Moritz Venzin. Approximate CVP in time $2^{0.802n}$ – Now in any norm! In *Proc. 23rd Internat. Conf. on Integ. Prog. and Comb. Opt. (IPCO 2022)*, pages 440–453, 2022. doi:10.1007/978-3-031-06901-7_33.
- 20 Marc Troyanov. Funk and Hilbert geometries from the Finslerian viewpoint. In *Handbook of Hilbert geometry*, volume 22 of *IRMA Lectures in Mathematics and Theoretical Physics*, pages 69–110. European Mathematical Society Publishing House, 2014. doi:10.4171/147-1/3.
- 21 Constantin Vernicos and Cormac Walsh. Flag-approximability of convex bodies and volume growth of Hilbert geometries. HAL Archive (hal-01423693i), 2016. URL: <https://hal.archives-ouvertes.fr/hal-01423693>.

Combinatorial Designs Meet Hypercliques: Higher Lower Bounds for Klee’s Measure Problem and Related Problems in Dimensions $d \geq 4$

Egor Gorbachev ✉

Saarbrücken Graduate School of Computer Science, Saarland Informatics Campus, Saarbrücken, Germany

Marvin Künnemann ✉

RPTU Kaiserslautern-Landau, Germany

Abstract

Klee’s measure problem (computing the volume of the union of n axis-parallel boxes in \mathbb{R}^d) is well known to have $n^{\frac{d}{2} \pm o(1)}$ -time algorithms (Overmars, Yap, SICOMP’91; Chan FOCS’13). Only recently, a conditional lower bound (without any restriction to “combinatorial” algorithms) could be shown for $d = 3$ (Künnemann, FOCS’22). Can this result be extended to a tight lower bound for dimensions $d \geq 4$?

In this paper, we formalize the technique of the tight lower bound for $d = 3$ using a combinatorial object we call *prefix covering design*. We show that these designs, which are related in spirit to combinatorial designs, directly translate to conditional lower bounds for Klee’s measure problem and various related problems. By devising good prefix covering designs, we give the following lower bounds for Klee’s measure problem in \mathbb{R}^d , the depth problem for axis-parallel boxes in \mathbb{R}^d , the largest-volume/max-perimeter empty (anchored) box problem in \mathbb{R}^{2d} , and related problems:

- $\Omega(n^{1.90476})$ for $d = 4$,
- $\Omega(n^{2.22222})$ for $d = 5$,
- $\Omega(n^{d/3 + 2\sqrt{d}/9 - o(\sqrt{d})})$ for general d ,

assuming the 3-uniform hyperclique hypothesis. For Klee’s measure problem and the depth problem, these bounds improve previous lower bounds of $\Omega(n^{1.777\dots})$, $\Omega(n^{2.0833\dots})$ and $\Omega(n^{d/3 + 1/3 + \Theta(1/d)})$ respectively.

Our improved prefix covering designs were obtained by (1) exploiting a computer-aided search using problem-specific insights as well as SAT solvers, and (2) showing how to transform combinatorial *covering designs* known in the literature to strong prefix covering designs. In contrast, we show that our lower bounds are close to best possible using this proof technique.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Fine-grained complexity theory, non-combinatorial lower bounds, computational geometry, clique detection

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.36

Related Version *Full Version*: <https://arxiv.org/abs/2303.08612>

Acknowledgements The second author thanks Karl Bringmann, Nick Fischer and Karol Węgrzycki for helpful discussions.

1 Introduction

For various problems in computational geometry, the best known algorithms display a running time of the form $n^{\Theta(d)}$ where d denotes the number of dimensions: Klee’s measure problem and the depth problem for axis-parallel boxes in \mathbb{R}^d can be solved in time $n^{d/2 \pm o(1)}$ [31, 12, 13], a recent algorithm [15] computes the largest-volume empty axis-parallel box among a given set of points in time $\tilde{O}(n^{(5d+2)/6})$, the star discrepancy can be computed in time $O(n^{d/2+1})$ [17],



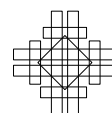
© Egor Gorbachev and Marvin Künnemann;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 36; pp. 36:1–36:14



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the maximum-weight rectangle problem can be solved in time $O(n^d)$ [6], to name few examples. Indeed, for all listed problems, it can be shown [12, 20, 6] that an $n^{o(d)}$ -time algorithm would refute the Exponential Time Hypothesis (ETH). Thus, the subsequent challenge is to determine running times $n^{f(d)}$ with $f(d) = \Theta(d)$ that are optimal under fine-grained complexity assumptions. By the nature of these running times (which quickly increase with d), it is particularly interesting to determine optimal time bounds for small dimensions such as $d \in \{2, 3, 4, 5\}$.

For some of these problems, strong conditional lower bounds are known: For Klee's measure problem and the depth problem, Chan [12] gives a tight conditional lower bound of $n^{d/2-o(1)}$ for *combinatorial* algorithms – roughly speaking, algorithms that avoid the algebraic techniques underlying fast matrix multiplication algorithms. When considering general algorithms (not only combinatorial ones), tight lower bounds are only known for *weighted* problems or small dimensions: For the weighted depth problem and the maximum-weight rectangle problem, tight lower bounds of $n^{d/2-o(1)}$ and $n^{d-o(1)}$, respectively, can be shown under the Weighted k -Clique Hypothesis [5]. Showing strong lower bounds for the simpler, *unweighted* problems appears to be more difficult, however. For Klee's measure problem and the unweighted depth problem, a recent result shows an $n^{d/(3-3/d)-o(1)}$ conditional lower bound under the 3-uniform hyperclique hypothesis [27], which yields a tight bound for $d = 3$, but not for $d \geq 4$.

Thus, the motivating question of this paper is the following:

Can we prove conditional optimality of known algorithms for Klee's measure problem, the depth problem and related problems for small dimensions $d \geq 4$, such as $d \in \{4, 5, 6\}$?

1.1 Our Results

As a starting point of this work, we formalize the approach used in [27] to obtain tight hardness for $d = 3$. To this end, we define the following combinatorial object, which we term *prefix covering designs* (due to its conceptual similarity to certain combinatorial designs¹).

In the following definition, let $\binom{S}{t}$ denote the set of t -element subsets of S .

► **Definition 1.** Let $d, K, \alpha \in \mathbb{N}$ with $d \geq 3$ and $K \geq 4$. A (d, K, α) -prefix covering design consists of d sequences s_1, \dots, s_d over $[K]$ with the following properties.

- **Triplet condition:** For every $\{a, b, c\} \in \binom{[K]}{3}$, there are $i, i', i'' \in [d]$ and $\ell, \ell', \ell'' \in \mathbb{N}_0$ such that
 - each element of $\{a, b, c\}$ is contained in $s_i[..\ell]$, $s_{i'}[.. \ell']$, or $s_{i''}[..\ell'']$. (Here, $s[..\ell]$ denotes the prefix of the first ℓ elements of s .)
 - $\ell + \ell' + \ell'' \leq \alpha$.
- **Singleton condition:** For every $x \in [K]$ occurring more than once in s_1, \dots, s_d , define $\ell_{\min}(x)$ ($\ell_{\max}(x)$) as the minimal (maximal) ℓ such that there is some i with $s_i[\ell] = x$. Then we have
 - $\ell_{\min}(x) + \ell_{\max}(x) \leq \alpha + 1$.

As an example, it is straightforward to see that for any d , the sequences $s_1 = (1, d+1)$, $s_2 = (2, d+1)$, \dots , $s_d = (d, d+1)$ constitute a $(d, d+1, 3)$ prefix covering design.²

¹ In fact, we will later establish a formal connection between these concepts.

² For the triplet condition, note that the triplet $\{a, b, c\} \in \binom{[d]}{3}$ is contained in the prefixes $s_a[..1], s_b[..1], s_c[..1]$ of total length $\alpha = 3$ and that any triplet $\{a, b, d+1\}$ with $\{a, b\} \in \binom{[d]}{2}$ is

Prefix covering designs constitute the core of the proof technique used in [27]. Specifically, we show that the existence of good prefix covering designs directly leads to strong lower bounds for several problems (these reductions are implicit in [27] or adapted to prefix covering designs from [20]).

► **Proposition 2.** *Let $d, K, \alpha \in \mathbb{N}$ such that there exists a (d, K, α) prefix covering design. Unless the 3-uniform Hyperclique Hypothesis fails, there is no $\varepsilon > 0$ such that there exists an $O(n^{\frac{K}{\alpha} - \varepsilon})$ -time algorithm for any of the following problems:*

- *Klee's Measure problem in \mathbb{R}^d ,*
- *Depth problem in \mathbb{R}^d ,*
- *Largest-Volume Empty Anchored Box problem in \mathbb{R}^{2d} ,*
- *Maximum-Perimeter Empty Anchored Box problem in \mathbb{R}^{2d} .*

Beyond these problems, similar reductions are also possible for related problems such as the Bichromatic Box problem in \mathbb{R}^{2d} (given sets of red and blue points, find the axis-parallel box containing the maximum number of blue points while avoiding any red point) and various related discrepancy problems such as the Star Discrepancy, see [20]. Note that there is a blow-up in the dimension for the Empty Anchored Box problems, which turns out to be unavoidable assuming the 3-uniform hyperclique hypothesis, as there are $O(n^{(1/2-\varepsilon)d})$ -algorithms for these problems (see below). At this point, we only give a rough sketch of the reduction, with the full proof deferred to the full version of this paper [22], where we also formally define all listed problems and discuss the 3-uniform hyperclique hypothesis.

Proof sketch for Proposition 2. For each problem, we give a reduction from the 3-uniform hyperclique problem: Given a 3-uniform hypergraph $G = (V, E)$ with $V = V^{(1)} \cup \dots \cup V^{(K)}$ and $|V^{(1)}| = \dots = |V^{(K)}| = n$, determine whether there are $v^{(1)} \in V^{(1)}, \dots, v^{(K)} \in V^{(K)}$ that form a clique in G . The 3-uniform hyperclique hypothesis states that this problem requires running time $n^{K-o(1)}$.

Intuitively, a special case of each of the problems listed above is to find an axis-parallel box Q satisfying certain properties. More specifically, any candidate box Q is given by choosing some value $v_i \in \{0, \dots, U-1\}$ for each dimension $i \in [d]$. We use a (d, K, α) prefix covering design s_1, \dots, s_d to interpret the values v_1, \dots, v_d as choices of vertices in $V^{(1)}, \dots, V^{(K)}$: Namely, with $s_i = (s_i[1], \dots, s_i[L])$, we think of any number $v_i \in \{0, \dots, U-1\}$ with $U = n^L$ as a base- n number $v_i = (v_i[1], \dots, v_i[L])$. We interpret $(v_i[1], \dots, v_i[L]) \in \{0, \dots, n-1\}^L$ as choosing the $(v_i[\ell] + 1)$ -st vertex in $V^{(s_i[\ell])}$ for all $1 \leq \ell \leq L$.

With this encoding fixed, it remains to ensure that the only true solutions Q encode a *clique* in G . This consists of two tasks: (1) ensuring that the candidate box Q chooses vertices consistently, i.e., for each $V^{(x)}$ such that x occurs in more than one s_i , we need to make sure that the same vertex is chosen in each occurrence, and (2) ensuring that the chosen vertices form a clique. Crucially, for both tasks, our geometric problems allow us to exclude candidate boxes Q where the v_i have certain prefixes. Specifically, due to the singleton condition, we only need to construct $O(n^\alpha)$ boxes to ensure consistency of the remaining candidate solutions Q . Likewise, the triplet condition is used to ensure that all candidate boxes Q that encode a non-clique (for which one of the triplets $\{v^{(a)}, v^{(b)}, v^{(c)}\}$ is not an edge in G) are excluded, using only $O(n^\alpha)$ additional boxes. In total, this creates an instance of size $O(n^\alpha)$ for the target problem, which yields an $n^{\frac{K}{\alpha} - o(1)}$ lower bound under the 3-uniform hyperclique hypothesis. ◀

contained in the prefixes $s_a[.2], s_b[.1]$ of total length $\alpha = 3$. The singleton condition only needs to be checked for $x = d + 1$, for which we note that $\ell_{\min}(d + 1) = \ell_{\max}(d + 1) = 2$ and thus $\ell_{\min}(d + 1) + \ell_{\max}(d + 1) = 4 \leq \alpha + 1$ for $\alpha = 3$.

From Proposition 2, we obtain the following direct corollary.

► **Corollary 3.** *For any $d \geq 3$, let $\gamma_d := \sup\{\frac{K}{\alpha} \mid \text{there is a } (d, K, \alpha) \text{ prefix covering design}\}$. Then for no $\varepsilon > 0$ there exists an $O(n^{\gamma_d - \varepsilon})$ -algorithm for any of the problems listed in Proposition 2, unless the 3-uniform hyperclique hypothesis fails.*

The tight conditional lower bound [27] for Klee's Measure problem and the depth problem in \mathbb{R}^3 follows from the following construction: For any $g \in \mathbb{N}$, we set $K = 3g$, write $[K] = \{a_1, \dots, a_g, b_1, \dots, b_g, c_1, \dots, c_g\}$ and observe that

$$s_1 = (a_1, \dots, a_g, b_g, \dots, b_1), s_2 = (b_1, \dots, b_g, c_g, \dots, c_1), s_3 = (c_1, \dots, c_g, a_g, \dots, a_1)$$

provide a $(3, 3g, 2g + 1)$ prefix covering design. Thus, we obtain $\gamma_3 \geq \lim_{g \rightarrow \infty} \frac{3g}{2g+1} = \frac{3}{2}$, establishing an $n^{\frac{3}{2} - o(1)}$ conditional lower bound for KMP in \mathbb{R}^3 and related problems.³

Given the direct applicability of prefix covering designs to Klee's measure problem, the depth problem and many related problems, it is only natural to ask what the highest obtainable lower bounds are using this technique. For one, designing better prefix covering designs gives stronger lower bounds. On the other hand, establishing limits for prefix covering designs may indicate potential for improved algorithms for KMP and related problems (such a phenomenon has been observed in other contexts, e.g., [11]).

Our first result is that prefix covering designs cannot establish a higher lower bound than $n^{\frac{d}{3} + O(\sqrt{d})}$. The following bound will be proved in Section 3.

► **Proposition 4.** *We have that $\gamma_d \leq \frac{d}{3(1-\sqrt{\frac{2}{d}})} = \frac{d}{3} + \sqrt{\frac{2}{9}} \cdot \sqrt{d} + o(\sqrt{d})$.*

However, as $\frac{d}{3(1-\sqrt{2/d})} \geq \frac{d}{2}$ for $d \leq 18$, this result does not rule out tight lower bounds for small dimensions. In fact, combining a computer-aided search with problem-specific insights, we give improved constructions for $d \in \{4, 5\}$, which give lower bounds that are surprisingly close to $\frac{d}{2}$.

► **Theorem 5.** *There is a $(4, 40, 21)$ prefix covering design, which yields $\gamma_4 \geq \frac{40}{21} > 1.90476$. There is a $(5, 40, 18)$ prefix covering design, which yields $\gamma_5 \geq \frac{40}{18} > 2.22222$.*

Proof. The following sequences give a $(4, 40, 21)$ prefix covering design:

$$\begin{aligned} s_1 &= (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 40, 19, 28, 37, 26), \\ s_2 &= (11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 30, 9, 38, 27, 36), \\ s_3 &= (21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 20, 39, 8, 7, 37), \\ s_4 &= (31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 10, 29, 18, 17, 27). \end{aligned}$$

The following sequences give a $(5, 40, 18)$ prefix covering design:

$$\begin{aligned} s_1 &= (1, 2, 3, 4, 5, 6, 7, 8, 24, 31, 38, 30, 14), \\ s_2 &= (9, 10, 11, 12, 13, 14, 15, 16, 32, 40, 6, 31, 22), \\ s_3 &= (17, 18, 19, 20, 21, 22, 23, 24, 8, 7, 39, 15, 30), \\ s_4 &= (25, 26, 27, 28, 29, 30, 31, 32, 40, 16, 23, 39, 6), \\ s_5 &= (33, 34, 35, 36, 37, 38, 39, 40, 16, 32, 15, 23). \end{aligned}$$

For the readers' convenience, we provide checker programs to verify the singleton and triplet conditions in [21] (see the full version of this paper [22] for details). ◀

³ It is not hard to prove that $\gamma_3 \leq \frac{3}{2}$, resulting in $\gamma_3 = \frac{3}{2}$. This raises the question whether we can find exact values of γ_d for $d \geq 4$.

For Klee's measure problem and the depth problem in \mathbb{R}^4 and \mathbb{R}^5 , the gap between the resulting conditional lower bound and the known upper bound is thus at most $O(n^{0.09524})$ and $O(n^{0.27778})$, respectively. This improves over previous hyperclique-based lower bounds of $\Omega(n^{1.777})$ and $\Omega(n^{2.0833})$, respectively.

These results may (re-)ignite hope that it might be possible to find prefix covering designs that establish tight lower bounds for $d = 4$ and $d = 5$. Alas, by a careful investigation of the limits of prefix covering designs, we refute this hope.

► **Theorem 6.** *We have $\gamma_4 < 2$.*

This result is proven via a careful analysis of the structure of prefix covering designs with quality $\frac{K}{\alpha}$ approaching 2: We show that certain *levels* (i.e., $s_1[\ell], \dots, s_d[\ell]$ for certain values of ℓ) must have a very rigid structure. Essentially, every element on such a level must have exactly a single copy on a corresponding other level. A detailed analysis of all possibilities displays a contradiction; we cannot get a quality $\frac{K}{\alpha}$ that is arbitrarily close to 2. The proof is in the full version of this paper [22]. It remains an interesting question to determine the precise value of γ_4 ; our results yield $1.90476 \leq \gamma_4 < 2$.

Connection to covering designs

Our previous results give evidence of the intricacy of designing good prefix covering designs. Unfortunately, designing optimized designs for small dimensions like $d = 4$ and $d = 5$ offers little insights into the asymptotics in d as well as the general structure of good prefix designs for larger dimensions.

We address this by providing general constructions that are applicable for all d and make use of the extensive literature on *combinatorial designs*. Specifically, we observe an interesting connection between so-called *covering designs* (see, e.g., the surveys [29, 24, 25, 23] and [14] for an algorithmic application in computational geometry) and prefix covering designs. A (v, k, t) covering design is a collection of k -sized subsets B_1, \dots, B_b – called *blocks* – of $[v]$ such that every t -element subset of $[v]$ is fully contained in some block B_i . These covering designs constitute a relaxation of balanced incomplete block designs.

Note that a (d, K, α) prefix covering design s_1, \dots, s_d where each s_i has length at most L is superficially similar to a (v, k, t) -covering design with $v = K$ elements, block size $k = L$, parameter $t = 3$ and d blocks: in both designs, we cover triplets among $v = K$ elements using d sequences/blocks. However, there are two key differences. (1) In covering designs, we cover each triplet in a single block, while in prefix covering designs, we may use prefixes from up to three sequences. (2) The sequences of prefix covering designs are inherently ordered (due to the *prefix* nature of the singleton and triplet conditions), while covering designs have unordered blocks. A priori, it is unclear whether there is a general way to use good covering designs to obtain good prefix covering designs or vice versa. Maybe surprisingly, we show how to use good (v, k, t) covering designs with $t = 2$ (rather than $t = 3$, which might appear as the more natural correspondence) to obtain strong prefix covering designs.

Specifically, for any such covering design satisfying a mild matching-like condition (which is satisfied by many constructions known in the literature), we obtain high-quality prefix covering designs. We will see below that by plugging in known constructions, we get prefix covering designs that are close to optimal when $d \rightarrow \infty$.

► **Theorem 7.** *Let $d \geq 3, k \in \mathbb{N}$ and v be a multiple of d such that there is a $(v, k, 2)$ covering design with d blocks with the following property: For every block B_i , there exists $U_i \subseteq B_i$ of size $\frac{v}{d}$ such that U_1, \dots, U_d partition $[v]$. Then $\gamma_d \geq \frac{d}{3-2\frac{v}{kd}}$.*

Let us give an example application of this theorem (see Sections 1.2 and the full version of this paper [22] for stronger consequences). It is well known that the projective plane of order q (where q is a prime power) yields a set of $v = q^2 + q + 1$ points, $d = q^2 + q + 1$ lines, with $k = q + 1$ points on each line, such that every pair of points is connected by a line. This yields a $(v, k, 2)$ -design with $d = v = q^2 + q + 1$ and $k = q + 1$. One can show that this design satisfies the matching-like condition (see the full version of this paper [22]). Thus, for infinitely many d , we obtain a lower bound of $\gamma_d \geq \frac{d}{3 - \frac{2}{q+1}}$. Since $q = O(\sqrt{d})$, we obtain $\gamma_d \geq \frac{d}{3 - \Omega(1/\sqrt{d})} = \frac{d}{3} + \Omega(\sqrt{d})$ for infinitely many d , improving over the lower bound of $\gamma_d \geq \frac{d}{3} + \frac{1}{3} + \frac{1}{3(d-1)}$ that is implicit in [27].

1.2 Consequences: Improved conditional lower bounds

Using Theorem 7, we may take any $(v, k, 2)$ covering design with d blocks that is known in the literature, check whether it satisfies the matching-like condition, and obtain the corresponding lower bound on γ_d . In Table 1, we list lower bounds on γ_d , $d \leq 10$ obtained this way, specifically, by using covering designs listed in the La Jolla Covering Repository [23] (see Section 2 for details). Notably, the resulting lower bounds improve over the constructions in [27] for $d \geq 4$.

We also provide a lower bound for *all* γ_d that is close to optimal when $d \rightarrow \infty$.

► **Theorem 8.** *There is some function $f(d) = d/3 + 2\sqrt{d}/9 - o(\sqrt{d})$ such that $\gamma_d \geq f(d)$ for all $d \geq 3$.*

This lower bound is obtained by showing how to extend the projective planes covering designs (in a suitable way) to obtain strong prefix covering designs for all values of d .

By the above theorem, we obtain a $n^{d/3+2/9\sqrt{d}-o(\sqrt{d})}$ conditional lower bound for Klee's measure problem and related problems. Note that Chan's reduction from K -clique [12] can be interpreted as a lower bound of $n^{(\omega/6)d-o(1)}$ assuming that current K -clique algorithms are optimal. If $\omega = 2$, this cannot give any higher lower bound than $n^{d/3-o(1)}$.

Table 1 also lists the corresponding upper bound of $O(n^{d/2})$ for Klee's measure problem and the depth problem for comparison. The gaps for the Largest-Volume/Maximum-Perimeter Empty (Anchored) Box problem in \mathbb{R}^d are a bit larger: Chan [15] obtains an upper bound⁴ for the anchored version of $\tilde{O}(n^{d/3+\lfloor d/2 \rfloor/6}) \leq \tilde{O}(n^{5d/12})$ for $d \geq 4$. In particular, this yields upper bounds of $\tilde{O}(n^{2.5})$, $\tilde{O}(n^{3.3334})$, and $\tilde{O}(n^{4.1667})$ for $d = 6$, $d = 8$ and $d = 10$, respectively, while we supply a conditional lower bound of $n^{\gamma_{d/2}-o(1)}$ for even $d \geq 6$, which yields lower bounds of $n^{1.5-o(1)}$, $n^{1.9047-o(1)}$ and $n^{2.2222-o(1)}$ for $d = 6$, $d = 8$ and $d = 10$, respectively. It is an interesting question whether we can prove a higher lower bound than $n^{d/4-o(1)}$ for any d or whether Chan's algorithms can be improved further.

Related Work. Klee's measure problem has been well-studied since the 1970s [26, 7, 19, 32, 31, 12, 13, 27], including algorithms beating $n^{d/2 \pm o(1)}$ for various special cases, e.g., [2, 8, 1, 9, 33, 10].

The depth problem for axis-parallel boxes is closely related to Klee's measure problem and often admits similar algorithmic ideas, see particularly [13].

Finding a largest-volume empty axis-parallel box has initially been mostly studied in two dimensions (see, e.g., [30, 16, 3]). In higher dimensions, Backer and Keil [4] give a $\tilde{O}(n^d)$ algorithm, which was recently improved to $\tilde{O}(n^{(5d+2)/6})$ by Chan [15]. Note that

⁴ While Chan focuses on the Largest-Volume Empty Box problem, he states that his algorithms for $d \geq 4$ also work for the Maximum-Perimeter version, see [15, Section 5].

■ **Table 1** The exponents of the upper and conditional lower bounds for Klee’s measure problem and the depth problem in \mathbb{R}^d for $d \leq 10$. The upper bound column is due to the $n^{d/2 \pm o(1)}$ -time algorithms [31, 12, 13], the conditional lower bounds are based on the 3-uniform hyperclique hypothesis and result from [27] (3rd column), Theorem 5 (4th column) and from combining Theorem 7 with covering designs found in the La Jolla Covering Repository maintained by D. Gordon [23] (5th column).

d	Upper bound	Previously known lower bound	SAT-solver lower bound	Covering designs lower bound
3	1.5	1.5		1.5
4	2	1.7777	1.9047	1.8461
5	2.5	2.0833	2.2222	2.1929
6	3	2.4		2.5714
7	3.5	2.7222		3
8	4	3.0476		3.3333
9	4.5	3.375		3.6818
10	5	3.7037		4.0540

our lower bounds are most interesting for the anchored version of the problem, which is solvable in faster running time $\tilde{O}(n^{5d/12})$ [15]. Approximation algorithms have been given in [18]. Giannopoulos et al. [20] give a reduction from d -clique, which can be understood as an $n^{(\omega/12)d - o(1)}$ lower bound assuming that current clique algorithms are optimal.

2 Constructions

In this section, we prove our general result transforming covering designs to prefix covering designs (Theorem 7). All remaining proofs and details on constructing prefix covering designs can be found in the full version of this paper [22].

For a (d, K, α) prefix covering design (PCD) with sequences s_1, s_2, \dots, s_d we call elements $s_1[i], s_2[i], \dots, s_d[i]$ the i -th level of the PCD.

When analyzing such prefix covering designs, it is helpful to distinguish between the “first” occurrence of some element, which we call the *primary* element, and all other occurrences, which we call *copies*. We call a pair (i, ℓ) a position if $1 \leq i \leq d, 1 \leq \ell$, and there exists ℓ -th element in s_i .

► **Definition 9.** For any prefix covering design s_1, \dots, s_d , we call a position (i, ℓ) the primary position of value x ($1 \leq x \leq K$) if and only if $s_i[\ell] = x$ and $s_{i'}[\ell'] \neq x$ for every other position (i', ℓ') such that (ℓ', i') precedes (ℓ, i) in the lexicographic ordering.

Every other occurrence (i', ℓ') with $s_{i'}[\ell'] = x$ is called a copy of x .

Note that if (i, ℓ) is a primary position of value x , then $\ell = \ell_{\min}(x)$.

► **Definition 10.** A (v, k, t) covering design where $v \geq 2, k \geq t \geq 1$ is a collection of k -element subsets (called blocks) of $[v]$ such that any t -element subset is contained in at least one block.

In the following proof, we will be extensively using (v, k, t) covering designs for $t = 2$. So, every pair of elements is contained in at least one block.

Proof of Theorem 7. Consider some $(v, k, 2)$ covering design consisting of d blocks where v is divisible by d and set $v' := \frac{v}{d} \in \mathbb{N}$. Define B_1, B_2, \dots, B_d as the blocks of this covering design. Assume there exist sets $U_1 \subseteq B_1, U_2 \subseteq B_2, \dots, U_d \subseteq B_d$ such that

s_1	8	9	10	1	2	3
s_2	11	12	13	4	1	5
s_3	14	15	16	7	1	6
s_4	17	18	19	6	2	4
s_5	20	21	22	2	5	7
s_6	23	24	25	3	4	7
s_7	26	27	28	5	3	6

■ **Figure 1** Example of a (7, 28, 10) PCD construction from a (7, 3, 2) covering design with 7 blocks.

$|U_1| = |U_2| = \dots = |U_d| = v'$ and U_1, U_2, \dots, U_d partition $[v]$. Then we will prove that for every $\varepsilon > 0$, there exist K and α such that $\frac{K}{\alpha} \geq \frac{d}{3-2\frac{v}{kd}} - \varepsilon$ and (d, K, α) PCD exists. From this we automatically get that $\gamma_d \geq \frac{d}{3-2\frac{v}{kd}}$ by going to the limit.

First, we present a slightly worse construction.

Order elements inside blocks of a given covering design in such a way that elements of U_i are located in the first v' positions of B_i , i.e., $\{B_i[j] \mid 1 \leq j \leq v'\} = U_i$. To construct sequences of our PCD, we take these blocks of the covering design and put kd new different elements in front of them by prepending k elements in each sequence. In other words, the resulting PCD has sequences s_1, s_2, \dots, s_d each of length $2k$ such that $s_i[j] = v + (i-1) \cdot k + j$ for $j \leq k$ and $s_i[j] = b_i[j-k]$ for $j > k$. An example for $d = 7$ is given in Figure 1. We will prove that this gives a (d, K, α) PCD with $K = (v' + k)d$ and $\alpha \leq 3k + v'$.

There are $v'd$ elements from a covering design and kd more unique elements that we added, so $K = (v' + k)d$. It remains to check that $\alpha \leq 3k + v'$.

First, we check the singleton condition. Due to our ordering of the covering design blocks, all primary positions of all elements are located in the first $k + v'$ levels, so $\ell_{\min}(x) \leq k + v'$ for every element x . At the same time, there are $2k$ elements in each sequence in total, so $\ell_{\max}(x) \leq 2k$. Thus, $\ell_{\min}(x) + \ell_{\max}(x) \leq (k + v') + 2k = 3k + v'$ for each $x \in [K]$.

Second, we check the triplet condition. Assume we chose three elements a, b and c . Define their primary positions as (i_a, ℓ_a) , (i_b, ℓ_b) and (i_c, ℓ_c) respectively. Without loss of generality, assume that $\ell_a \leq \ell_b \leq \ell_c$. Consider two cases.

1. If there is at most one element from the covering design among these three, then $\ell_a \leq k$, $\ell_b \leq k$ and $\ell_c \leq k + v'$, so we can cover them with prefixes $s_{i_a}[\dots \ell_a]$, $s_{i_b}[\dots \ell_b]$ and $s_{i_c}[\dots \ell_c]$ of total size $\ell_a + \ell_b + \ell_c \leq k + k + (k + v') = 3k + v'$.
2. If there are at least two elements from the covering design among these three, then b and c are in the covering design. By the definition of a covering design, there should be a sequence s_i that contains both b and c . Thus we can cover all three elements with two prefixes: $s_i[\dots 2k]$ (whole sequence) and $s_{i_a}[\dots \ell_a]$ of total size $2k + \ell_a \leq 2k + (k + v') = 3k + v'$.

This concludes the proof that $\alpha \leq 3k + v'$ and already gives a bound $\gamma_d \geq \frac{K}{\alpha} \geq \frac{(k+v')d}{3k+v'} = \frac{d}{3} \cdot \frac{3k+3v'}{3k+v'} = \frac{d}{3} \cdot \left(1 + \frac{2v'}{3k+v'}\right) = \frac{d}{3} \cdot \left(1 + \frac{2v}{3dk+v}\right)$.

To improve this construction we will replicate the covering design n times for some positive integer n . Define B_i^j for $1 \leq i \leq d$ and $1 \leq j \leq n$ as the i -th block of the j -th copy of the covering design. We want different copies of the covering design to be over different elements, so the v elements of B^j are $\{(j-1)v + 1, \dots, jv\}$. Define U_i^j as v' -element subsets of B_i^j such that $U_1^j, U_2^j, \dots, U_d^j$ partition $\{(j-1)v + 1, \dots, jv\}$. Define $R_i^j := B_i^j \setminus U_i^j$ as the remaining $k - v'$ elements of each block. Also, for every sequence of our PCD, we define

s_1	22	23	24	25	26	27	28	1	8	15	19	20	12	13	5	6
s_2	29	30	31	32	33	34	35	2	9	16	15	21	8	14	1	7
s_3	36	37	38	39	40	41	42	3	10	17	15	18	8	11	1	4
s_4	43	44	45	46	47	48	49	4	11	18	19	16	12	9	5	2
s_5	50	51	52	53	54	55	56	5	12	19	21	17	14	10	7	3
s_6	57	58	59	60	61	62	63	6	13	20	16	17	9	10	2	3
s_7	64	65	66	67	68	69	70	7	14	21	20	18	13	11	6	4

■ **Figure 2** Example of a $(7, 70, 24)$ prefix covering design obtained by a scaled construction with $v = 7$ ($v' = 1$), $k = 3$, $d = 7$ and $n = 3$.

$m := nk - (n - 1)v'$ unique elements that are put at the beginning of this sequence. Let these unique elements for sequence i be called A_i ($A_i = \{nv + (i - 1)m + 1, \dots, nv + im\}$). Now we are ready to construct the sequences s_1, \dots, s_d of our prefix covering design by

$$s_i = (A_i, U_i^1, U_i^2, \dots, U_i^n, R_i^n, R_i^{n-1}, \dots, R_i^1).$$

An example of such a construction is given in Figure 2.

We will prove that such a PCD has $K = (nk + v')d$ and $\alpha \leq 3nk - (2n - 3)v'$, similarly to the proof for the simpler construction. First, there are $v'd$ elements from every covering design, and there are n designs, so overall, there are $nv'd$ elements from covering designs. Additionally, there are $md = nk d - (n - 1)v'd$ more unique elements that we added, so $K = (nk + v')d$ indeed. It remains to check that $\alpha \leq T$ where $T := 3nk - (2n - 3)v'$. We will use that $T = 2m + nk + v' = 3m + nv'$.

First, we check the singleton condition. Due to our ordering of the covering design blocks, all primary positions of all elements are located in the first $m + nv'$ levels, so $\ell_{\min}(x) \leq m + nv'$ for every element x . If $\ell_{\min}(x) \leq m$, this element has only one occurrence, and we do not need to check the singleton condition for it. If $\ell_{\min}(x) = m + (n - i)v' + j$ for some $1 \leq i \leq n$ and $1 \leq j \leq v'$, then it means that element x belongs to the $(n - i + 1)$ -st covering design, and its other occurrences are located in the levels from $m + nv' + (i - 1)(k - v') + 1$ to $m + nv' + i(k - v')$. So $\ell_{\max}(x) \leq m + nv' + i(k - v')$. Consequently, $\ell_{\min}(x) + \ell_{\max}(x) \leq (m + (n - i)v' + j) + (m + nv' + i(k - v')) = 2m + 2nv' + i(k - 2v') + j \leq 2m + 2nv' + n(k - 2v') + v' = 2m + nk + v' = T < T + 1$ where we used the fact that $k - 2v' = k - 2\frac{v}{d} \geq 0$ due to the lemma below. We have even proved a slightly stronger inequality:

$$\ell_{\min}(x) + \ell_{\max}(x) \leq T. \tag{1}$$

▶ **Lemma 11.** For every $(v, k, 2)$ covering design with $d \geq 2$ blocks, $k \geq 2v/d$ holds.

Proof of Lemma. If $k < v$, then every element $x \in [v]$ should be located in at least two sets: otherwise, we would cover only $k - 1 < v - 1$ pairs involving x , which contradicts the fact that it is a covering design. But if every element is located in at least two sets, then the sum of all set sizes kd is at least $2v$. Dividing both numbers by d , we get the desired inequality.

If $k \geq v$, then $k \geq v \geq 2\frac{v}{d}$ because $d \geq 2$. ◀

Second, we check the triplet condition. Consider any three elements a, b and c . Define their primary positions as (i_a, ℓ_a) , (i_b, ℓ_b) and (i_c, ℓ_c) respectively. Without loss of generality, assume that $\ell_a \leq \ell_b \leq \ell_c$. Consider two cases.

1. If at most one element out of these three is from covering designs, we know that $\ell_a \leq m$, $\ell_b \leq m$ and $\ell_c \leq m + nv'$, so we can cover them with prefixes $s_{i_a}[\dots\ell_a]$, $s_{i_b}[\dots\ell_b]$ and $s_{i_c}[\dots\ell_c]$ with total size $\ell_a + \ell_b + \ell_c \leq m + m + (m + nv') = 3m + nv' = T$.

2. If at least two elements out of these three are from covering designs, then b and c are in the covering designs. By the definition of a covering design there should be a sequence s_i that contains both b and c .⁵ Then we can cover all three elements with two prefixes: $s_i[..\max(\ell_b^i, \ell_c^i)]$ and $s_{i_a}[..\ell_a]$ where ℓ_b^i and ℓ_c^i are positions of elements b and c , respectively, in the sequence i . We already know that elements b and c satisfy (1). It follows that $\ell_a + \ell_b^i \leq \ell_b + \ell_b^i \leq T$ and $\ell_a + \ell_c^i \leq \ell_c + \ell_c^i \leq T$. From this we can conclude that $\ell_a + \max(\ell_b^i, \ell_c^i) \leq T$, as desired.

This concludes the proof that $\alpha \leq T = 3nk - (2n - 3)v'$. This construction gives us a bound $\gamma_d \geq \frac{K}{\alpha} \geq \frac{(nk+v')d}{3nk-(2n-3)v'} = \frac{(k+\frac{v'}{n})d}{3k-(2-\frac{3}{n})v'}$ where $n \in \mathbb{N}$ can be chosen arbitrarily. When n approaches infinity, this value approaches $\frac{kd}{3k-2v'} = \frac{d}{3-\frac{2v'}{k}} = \frac{d}{3-2\frac{v}{kd}}$. Thus, for every $\varepsilon > 0$ there exists n such that such a construction gives $\frac{K}{\alpha} \geq \frac{d}{3-2\frac{v}{kd}} - \varepsilon$, as desired. ◀

We say that a $(v, k, 2)$ covering design with d blocks admits a multi-matching if for every block B_i we can choose a subset U_i of size v/d such that U_1, U_2, \dots, U_d partition $[v]$. The following observation shows that in Theorem 7 it is not a restriction to assume that v is divisible by d , since we can always suitably scale covering designs:

► **Observation 12.** *Every $(v, k, 2)$ covering design with d blocks can be transformed into a $(vd, kd, 2)$ covering design with d blocks by replacing each of v elements with d distinct elements. If this scaled covering design admits a multi-matching, we get a lower bound for γ_d of $\frac{d}{3-2\frac{(vd)}{(kd)d}} = \frac{d}{3-2\frac{v}{kd}}$.*

The bound we give depends on the existence of specific covering designs admitting multi-matchings. This lower bound can be transformed into a general lower bound depending only on d (see the full version of this paper [22] for details); one can also obtain lower bounds for specific values of d : for a fixed value of d , the lower bound can be obtained by finding a covering design that minimizes the value $\text{freq} := \frac{kd}{v}$ which we call *frequency* (for a fixed covering design, the frequency is the average number of occurrences of elements). We searched for covering designs in the La Jolla Covering Repository [23], fixing the number of blocks to d and choosing the ones with the smallest frequencies. Then we multiplied the number of elements and set sizes in these covering designs by d using Observation 12 (because Theorem 7 works only for covering designs with v divisible by d) and checked whether they admit multi-matching. Perhaps surprisingly, for all specific values of d that we checked, the obtained covering designs indeed admit a multi-matching. The covering designs used and their multi-matchings can be found in [21] along with a computer program that checks that provided constructions are indeed covering designs, and they indeed admit multi-matchings.

The question remains whether the frequency in some dimension d could be minimized by a covering design that does *not* admit a multi-matching. Indeed, one can construct covering designs that do not admit a multi-matching. However, since we aim to minimize the frequencies, we are considering covering designs that should have a relatively small degree of redundancy – otherwise, they probably could be improved. In the full version of this paper [22], we formulate the corresponding conjecture that “sufficiently good” covering designs always admit a multi-matching and discuss some evidence. We also provide weaker bounds obtained from covering designs not admitting multi-matchings.

⁵ b and c may belong to different copies of our covering design, but all copies are identical, so equivalent elements from all covering designs occur in the same sequences, so there indeed should exist such s_i .

3 Limits

In this section, we prove limits of prefix covering designs, i.e., upper bounds on $\gamma_d = \sup\{\frac{K}{\alpha} \mid \text{there exists a } (d, K, \alpha) \text{ prefix covering design}\}$. The proof that $\gamma_4 < 2$ is in the full version of this paper [22]. The following lemma formalizes the intuition that increasing the value of K can only lead to better (more precisely, not worse) prefix covering designs.

► **Lemma 13 (Scaling Lemma).** *For every (d, K, α) PCD and positive integer $\lambda \in \mathbb{N}$, there also exists a $(d, \lambda \cdot K, \lambda \cdot \alpha)$ PCD.*

The proof of this fact is in the full version of this paper [22].

Proof of Proposition 4. For a fixed (d, K, α) PCD define $g := \lceil \frac{K}{d} \rceil$. If $\alpha \geq 3g$ then $\frac{K}{\alpha} \leq \frac{K}{3g} \leq \frac{K}{3K/d} = \frac{d}{3}$ and the proposition statement holds. Otherwise define $a := \lceil g - \frac{\alpha}{3} \rceil \geq 1$, i.e., $3(g - a) \leq \alpha < 3(g - a + 1)$. We will prove that $a < \sqrt{\frac{2}{d}} + 2$. If $a = 1$, it is correct, so from now on we assume that $a \geq 2$.

Define B as the set of all elements x that have $\ell_{\min}(x) > g - a$. We claim that $|B| \geq d(a - 1)$: The number of (not necessarily distinct) elements in the first $(g - a)$ positions (over all s_i) is $d \cdot (g - a) = dg - da < d \cdot (\frac{K}{d} + 1) - da = K - d(a - 1)$. Since there are K distinct numbers in total, the claim follows.

To prove the proposition, we will define a graph G_B with vertex set B . We connect two elements $x, y \in B$ by an edge if and only if there is some sequence s_i containing both x and y . We obtain our result by proving both an upper and a lower bound on the number of edges.

For a lower bound on the number of edges, consider how triplets $\{a, b, c\} \in \binom{B}{3}$ are covered by prefixes: For any such triplet $\{a, b, c\}$, there are prefixes $s_i[..l]$, $s_{i'}[..l']$ and $s_{i''}[..l'']$ which contain a , b and c and satisfy $l + l' + l'' \leq \alpha$.

▷ **Claim 14.** Without loss of generality, we may assume that at least one of l , l' and l'' is zero.

Proof. If all of them are at least $g - a + 1$, then $l + l' + l'' \geq 3(g - a + 1) > \alpha$, which yields a contradiction. Otherwise, if at least one of them is at most $g - a$, then this prefix cannot contain any of a , b and c as $\ell_{\min}(a), \ell_{\min}(b), \ell_{\min}(c) > g - a$. We can set this prefix to the empty prefix without loss of generality. ◁

So indeed, we can imagine that triplets of elements in B must be covered by using only two prefixes, not three. In particular, for every triplet of elements from B , at least two of them must occur in the same sequence, i.e., they must have an edge in G_B . Put differently, the complement graph of G_B is triangle-free and thus contains at most $|B|^2/4$ edges by Mantel's Theorem [28] (a special case of Turan's Theorem). We conclude that G_B has at least $\binom{|B|}{2} - \frac{|B|^2}{4} = \frac{|B|^2 - 2|B|}{4} \geq \frac{(|B| - 2)^2}{4}$ edges because $|B| \geq 2$.

We now show that either the number of edges is at most $dg^2/2$ or $|B| \leq 2g$. We ask on which positions elements from B can be located in the sequences. We know that $\ell_{\min}(x) \geq g - a + 1$ for any $x \in B$. At the same time, if some element from B is located in position $\geq 2(g - a) + 3$ (in some sequence i), then this must be its only occurrence since otherwise, it would violate the singleton condition. Furthermore, any covering of a triplet with such an element cannot contain elements from B in other sequences because it would take a prefix of length at least $2(g - a) + 3$ in sequence i and a prefix of length at least $g - a + 1$ in some other sequence, which would violate the triplet condition. From this, we can conclude that if every triplet with this element and other elements in B is covered, all elements from B have to occur in sequence i . We can assume that all elements have indices at most α (otherwise, they are useless for coverings), so there are at most $\alpha - (g - a) \leq 2g$

36:12 Higher Lower Bounds for Klee's Measure Problem & Related Problems in \mathbb{R}^d , $d \geq 4$

elements from B in this sequence. This yields $|B| \leq 2g$. In the remaining case all $x \in B$ satisfy $\ell_{\max}(x) \leq 2(g-a) + 2$ and $\ell_{\min}(x) > (g-a)$, so there are at most $g-a+2 \leq g$ elements from B in each sequence. Thus, there are at most $d \cdot \binom{g}{2}$ pairs of elements from B that occur in the same sequence.

From the above lower and upper bounds on the number of edges, we derive that

$$\frac{(|B| - 2)^2}{4} \leq d \cdot \binom{g}{2} < \frac{dg^2}{2}.$$

Combining this with the fact that $d(a-1) \leq |B|$, we deduce that $d(a-1) - 2 \leq |B| - 2 < \sqrt{2dg}$. (Note that in the case $|B| \leq 2g$, the upper bound is trivially satisfied since $d \geq 2$.) Consequently,

$$a < \frac{\sqrt{2dg} + 2}{d} + 1 \leq \sqrt{\frac{2}{d}}g + 2$$

for $d \geq 2$. We plug this inequality into our initial inequality on α :

$$\alpha \geq 3(g-a) > 3g \left(1 - \sqrt{\frac{2}{d}} - \frac{2}{g}\right) \geq \frac{3K}{d} \left(1 - \sqrt{\frac{2}{d}} - \frac{2}{g}\right).$$

It follows that

$$\frac{K}{\alpha} \leq \frac{K}{\frac{3K}{d} \left(1 - \sqrt{\frac{2}{d}} - \frac{2}{g}\right)} = \frac{d}{3 \cdot \left(1 - \sqrt{\frac{2}{d}} - \frac{2}{g}\right)}.$$

Due to Scaling Lemma 13 we know that if there exists a (d, K, α) PCD then there also exists a $(d, K \cdot \lambda, \alpha \cdot \lambda)$ PCD for every positive integer λ . If we plug this covering design into the inequality above, we will get that

$$\frac{K}{\alpha} = \frac{\lambda \cdot K}{\lambda \cdot \alpha} \leq \frac{d}{3 \cdot \left(1 - \sqrt{\frac{2}{d}} - \frac{2}{g'}\right)}$$

where $g' := \lceil \frac{K \cdot \lambda}{d} \rceil$. If we take $\lambda \rightarrow +\infty$ then $\frac{2}{g'} \rightarrow 0$ and in the limit, we get the desired upper bound on $\frac{K}{\alpha}$:

$$\frac{K}{\alpha} \leq \frac{d}{3 \cdot \left(1 - \sqrt{\frac{2}{d}}\right)} = \frac{d}{3} \cdot \left(1 + \frac{\sqrt{\frac{2}{d}}}{1 - \sqrt{\frac{2}{d}}}\right) = \frac{d}{3} + \frac{\sqrt{2d}}{3 \left(1 - \sqrt{\frac{2}{d}}\right)} = \frac{d}{3} + \sqrt{\frac{2}{9}} \cdot \sqrt{d} + o(\sqrt{d}). \blacktriangleleft$$

4 Conclusion and Outlook

In this work, we make progress on obtaining tight conditional lower bounds for Klee's measure problem and related problems for $d \geq 4$. We give improved lower bounds that leave gaps of only $O(n^{0.09524})$, $O(n^{0.27778})$ and $O(n^{0.4286})$ for $d = 4$, $d = 5$ and $d = 6$, respectively. On the negative side, we prove that the proof technique via prefix covering designs and Proposition 2 – despite yielding a tight lower bound for $d = 3$ – cannot give tight lower bounds for $d \geq 4$, so that a novel reduction approach is needed for this task. Of course, it remains a tantalizing possibility that the $n^{d/2 \pm o(1)}$ running time for Klee's measure problem for large dimensions $d \geq 4$ can be broken.

We feel that the prefix covering designs formalized in this work are interesting in their own right. We establish a connection to the well-studied covering designs, by giving a framework that turns 2-covering designs into prefix covering designs. This connection leads to the asymptotic bound $\gamma_d = \frac{d}{3} + \Theta(\sqrt{d})$, leading to an $n^{d/3 + \Theta(\sqrt{d})}$ conditional lower bound for Klee's measure problem and related problems, improving over a previous bound of $n^{d/3 + 1/3 + \Omega(1/d)}$.

References

- 1 Pankaj K. Agarwal. An improved algorithm for computing the volume of the union of cubes. In David G. Kirkpatrick and Joseph S. B. Mitchell, editors, *Proceedings of the 26th ACM Symposium on Computational Geometry, Snowbird, Utah, USA, June 13-16, 2010*, pages 230–239. ACM, 2010. doi:10.1145/1810959.1811000.
- 2 Pankaj K. Agarwal, Haim Kaplan, and Micha Sharir. Computing the volume of the union of cubes. In Jeff Erickson, editor, *Proceedings of the 23rd ACM Symposium on Computational Geometry, Gyeongju, South Korea, June 6-8, 2007*, pages 294–301. ACM, 2007. doi:10.1145/1247069.1247121.
- 3 Alok Aggarwal and Subhash Suri. Fast algorithms for computing the largest empty rectangle. In D. Soule, editor, *Proceedings of the Third Annual Symposium on Computational Geometry, Waterloo, Ontario, Canada, June 8-10, 1987*, pages 278–290. ACM, 1987. doi:10.1145/41958.41988.
- 4 Jonathan Backer and J. Mark Keil. The mono- and bichromatic empty rectangle and square problems in all dimensions. In Alejandro López-Ortiz, editor, *LATIN 2010: Theoretical Informatics, 9th Latin American Symposium, Oaxaca, Mexico, April 19-23, 2010. Proceedings*, volume 6034 of *Lecture Notes in Computer Science*, pages 14–25. Springer, 2010. doi:10.1007/978-3-642-12200-2_3.
- 5 Arturs Backurs, Nishanth Dikkala, and Christos Tzamos. Tight hardness results for maximum weight rectangles. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPICs*, pages 81:1–81:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPICs.ICALP.2016.81.
- 6 Jérémy Barbay, Timothy M. Chan, Gonzalo Navarro, and Pablo Pérez-Lantero. Maximum-weight planar boxes in $o(n^2)$ time (and better). *Inf. Process. Lett.*, 114(8):437–445, 2014. doi:10.1016/j.ipl.2014.03.007.
- 7 J. L. Bentley. Algorithms for Klee’s rectangle problems. Department of Computer Science, Carnegie Mellon University, Unpublished notes, 1977.
- 8 Nicola Beume, Carlos M. Fonseca, Manuel López-Ibáñez, Luís Paquete, and Jan Vahrenhold. On the complexity of computing the hypervolume indicator. *IEEE Trans. Evol. Comput.*, 13(5):1075–1082, 2009. doi:10.1109/TEVC.2009.2015575.
- 9 Karl Bringmann. An improved algorithm for Klee’s measure problem on fat boxes. *Comput. Geom.*, 45(5-6):225–233, 2012. doi:10.1016/j.comgeo.2011.12.001.
- 10 Karl Bringmann. Bringing order to special cases of Klee’s measure problem. In Krishnendu Chatterjee and Jirí Sgall, editors, *Mathematical Foundations of Computer Science 2013 - 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings*, volume 8087 of *Lecture Notes in Computer Science*, pages 207–218. Springer, 2013. doi:10.1007/978-3-642-40313-2_20.
- 11 Karl Bringmann and Marvin Künnemann. Improved approximation for fréchet distance on c-packed curves matching conditional lower bounds. *Int. J. Comput. Geom. Appl.*, 27(1-2):85–120, 2017. doi:10.1142/S0218195917600056.
- 12 Timothy M. Chan. A (slightly) faster algorithm for Klee’s measure problem. *Comput. Geom.*, 43(3):243–250, 2010. doi:10.1016/j.comgeo.2009.01.007.
- 13 Timothy M. Chan. Klee’s measure problem made easy. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 410–419. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.51.
- 14 Timothy M. Chan. Orthogonal range searching in moderate dimensions: k-d trees and range trees strike back. *Discret. Comput. Geom.*, 61(4):899–922, 2019. doi:10.1007/s00454-019-00062-5.
- 15 Timothy M. Chan. Faster algorithms for largest empty rectangles and boxes. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational*

- Geometry, SoCG 2021, June 7-11, 2021, Buffalo, NY, USA (Virtual Conference)*, volume 189 of *LIPICs*, pages 24:1–24:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICs.SoCG.2021.24.
- 16 Bernard Chazelle, Robert L. (Scot) Drysdale III, and D. T. Lee. Computing the largest empty rectangle. *SIAM J. Comput.*, 15(1):300–315, 1986. doi:10.1137/0215022.
 - 17 David P. Dobkin, David Eppstein, and Don P. Mitchell. Computing the discrepancy with applications to supersampling patterns. *ACM Trans. Graph.*, 15(4):354–376, 1996. doi:10.1145/234535.234536.
 - 18 Adrian Dumitrescu and Minghui Jiang. On the largest empty axis-parallel box amidst n points. *Algorithmica*, 66(2):225–248, 2013. doi:10.1007/s00453-012-9635-5.
 - 19 Michael L. Fredman and Bruce W. Weide. On the complexity of computing the measure of $u[a_i, b_i]$. *Commun. ACM*, 21(7):540–544, 1978. doi:10.1145/359545.359553.
 - 20 Panos Giannopoulos, Christian Knauer, Magnus Wahlström, and Daniel Werner. Hardness of discrepancy computation and ε -net verification in high dimension. *J. Complex.*, 28(2):162–176, 2012. doi:10.1016/j.jco.2011.09.001.
 - 21 Egor Gorbachev and Marvin Künnemann. Codes and constructions for *Combinatorial Designs Meet Hypercliques: Higher Lower Bounds for Klee's Measure Problem and Related Problems in Dimensions $d \geq 4$* . <https://github.com/Peltorator/klees-measure-lower-bounds-repo>.
 - 22 Egor Gorbachev and Marvin Künnemann. Combinatorial designs meet hypercliques: Higher lower bounds for Klee's measure problem and related problems in dimensions $d \geq 4$, 2023. doi:10.48550/ARXIV.2303.08612.
 - 23 Daniel M. Gordon. La jolla covering repository. <https://www.dmgordon.org/cover/>. Accessed: 2022-11-27.
 - 24 Daniel M. Gordon, Greg Kuperberg, and Oren Patashnik. New constructions for covering designs. *Journal of Combinatorial Designs*, 3:269–284, 1995.
 - 25 Daniel M. Gordon and Douglas R. Stinson. Coverings. In Charles J. Colbourn and Jeffrey H. Dinitz, editors, *Handbook of Combinatorial Designs*. Chapman and Hall, 2006.
 - 26 Victor Klee. Can the measure of $\bigcup_1^n [a_i, b_i]$ be computed in less than $O(n \log n)$ steps? *The American Mathematical Monthly*, 84(4):284–285, 1977.
 - 27 Marvin Künnemann. A tight (non-combinatorial) conditional lower bound for Klee's measure problem in 3d. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS 2022)*, pages 555–566. IEEE, 2022. doi:10.1109/FOCS54457.2022.00059.
 - 28 W. Mantel. Problem 28: Solution by H. Gouwentak, W. Mantel, J. Teixeira de Mattes, F. Schuh and W. A. Wythoff. In *Wiskundige Opgaven 10*, pages 60–61. W. Mantel, 1907.
 - 29 W. H. Mills and R. C. Mullin. Coverings and packings. In J. H. Dinitz and D. R. Stinson, editors, *Contemporary Design Theory: A Collection of Surveys*, pages 371–399. Wiley, 1992.
 - 30 Amnon Naamad, D. T. Lee, and Wen-Lian Hsu. On the maximum empty rectangle problem. *Discret. Appl. Math.*, 8(3):267–277, 1984. doi:10.1016/0166-218X(84)90124-0.
 - 31 Mark H. Overmars and Chee-Keng Yap. New upper bounds in Klee's measure problem. *SIAM J. Comput.*, 20(6):1034–1045, 1991. doi:10.1137/0220065.
 - 32 Jan van Leeuwen and Derick Wood. The measure problem for rectangular ranges in d -space. *J. Algorithms*, 2(3):282–300, 1981. doi:10.1016/0196-6774(81)90027-4.
 - 33 Hakan Yildiz and Subhash Suri. On klee's measure problem for grounded boxes. In Tamal K. Dey and Sue Whitesides, editors, *Proceedings of the 28th ACM Symposium on Computational Geometry, Chapel Hill, NC, USA, June 17-20, 2012*, pages 111–120. ACM, 2012. doi:10.1145/2261250.2261267.

A Generalization of the Persistent Laplacian to Simplicial Maps

Aziz Burak Gülen ✉ 🏠

Department of Mathematics, The Ohio State University, Columbus, OH, USA

Facundo Mémoli ✉ 🏠

Department of Mathematics and Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA

Zhengchao Wan ✉ 🏠

Halıcıoğlu Data Science Institute, University of California San Diego, CA, USA

Yusu Wang ✉ 🏠

Halıcıoğlu Data Science Institute, University of California San Diego, CA, USA

Abstract

The (combinatorial) graph Laplacian is a fundamental object in the analysis of, and optimization on, graphs. Via a topological view, this operator can be extended to a simplicial complex K and therefore offers a way to perform “signal processing” on p -(co)chains of K . Recently, the concept of *persistent Laplacian* was proposed and studied for a pair of simplicial complexes $K \hookrightarrow L$ connected by an inclusion relation, further broadening the use of Laplace-based operators.

In this paper, we significantly expand the scope of the persistent Laplacian by generalizing it to a pair of weighted simplicial complexes connected by a weight preserving simplicial map $f : K \rightarrow L$. Such a simplicial map setting arises frequently, e.g., when relating a coarsened simplicial representation with an original representation, or the case when the two simplicial complexes are spanned by different point sets, i.e. cases in which it does not hold that $K \subset L$. However, the simplicial map setting is much more challenging than the inclusion setting since the underlying algebraic structure is much more complicated.

We present a natural generalization of the persistent Laplacian to the simplicial setting. To shed insight on the structure behind it, as well as to develop an algorithm to compute it, we exploit the relationship between the persistent Laplacian and the Schur complement of a matrix. A critical step is to view the Schur complement as a functorial way of restricting a self-adjoint positive semi-definite operator to a given subspace. As a consequence of this relation, we prove that the q th persistent Betti number of the simplicial map $f : K \rightarrow L$ equals the nullity of the q th persistent Laplacian $\Delta_q^{K,L}$. We then propose an algorithm for finding the matrix representation of $\Delta_q^{K,L}$ which in turn yields a fundamentally different algorithm for computing the q th persistent Betti number of a simplicial map. Finally, we study the persistent Laplacian on simplicial towers under weight-preserving simplicial maps and establish monotonicity results for their eigenvalues.

2012 ACM Subject Classification Mathematics of computing → Spectra of graphs; Mathematics of computing → Algebraic topology

Keywords and phrases combinatorial Laplacian, persistent Laplacian, Schur complement, persistent homology, persistent Betti number

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.37

Related Version *Full Version*: <https://arxiv.org/abs/2302.03771> [8]

Funding *Facundo Mémoli*: FM is partially supported by BSF 2020124, NSF CCF-1740761, NSF CCF-1839358, and NSF IIS-1901360.

Zhengchao Wan: ZW is partially supported by NSF CCF-2112665, NSF CCF-2217033.

Yusu Wang: YW is partially supported by NSF CCF-2112665, NSF CCF-2217033.



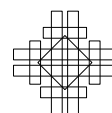
© Aziz Burak Gülen, Facundo Mémoli, Zhengchao Wan, and Yusu Wang;
licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 37; pp. 37:1–37:17

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

The graph Laplacian is an operator on the space of functions defined on the vertex set of a graph. It is one of the main tools in the analysis of and optimization on graphs. For example, the spectral properties of the graph Laplacian are extensively used in spectral clustering and other applications [2, 13, 17, 18, 21] and for efficiently solving systems of equations [12, 15, 19, 20].

As opposed to the traditional way of defining the graph Laplacian as the difference of the degree matrix and the adjacency matrix, it can also be defined from an algebraic topology perspective by considering the boundary operators and specific inner products defined on simplicial chain groups [2]. This point of view permits extending the graph Laplacian to operators on higher dimensional chain groups. Namely, this leads to the q th combinatorial Laplacian Δ_q^K on the q th chain group of a given simplicial complex K , in which the case $q = 0$ corresponds to the standard graph Laplacian [5, 6, 7, 10]. One fundamental property of the q th combinatorial Laplacian is that the q th Betti number of K equals the nullity of Δ_q^K .

By adopting the algebraic topology view, the q th persistent Laplacian $\Delta_q^{K,L}$ was independently introduced in [14, 22] for a pair of simplicial complexes $K \hookrightarrow L$ connected by an inclusion. The theoretical properties of $\Delta_q^{K,L}$ and algorithms to compute it have been extensively studied in [16]. One of these properties is that the nullity of $\Delta_q^{K,L}$ equals the *persistent Betti number* of the inclusion $K \hookrightarrow L$, which is a generalization of the corresponding property of the combinatorial Laplacian mentioned above.

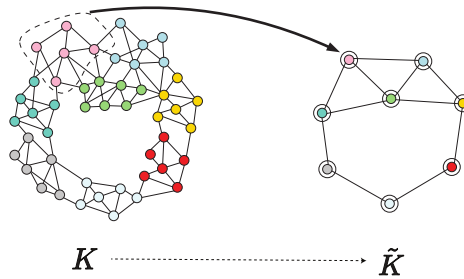


Figure 1 The 1-dimensional simplicial complex, i.e. graph, K is coarsened to produce the one on the right \tilde{K} . Vertices of the same color are “collapsed” to a “supernode” in \tilde{K} . This vertex map induces a simplicial map at the simplicial complex level.

Although the persistent Laplacian for a pair $K \hookrightarrow L$ has been used in some applications [1, 9, 11], the requirement that the complexes should be connected by an inclusion is restrictive and limits its applicability. Consider the scenario when we have two simplicial complexes $K \hookrightarrow L$ related by an inclusion so that their sizes are prohibitively large. Instead of tackling the direct computation of the persistent Betti numbers induced by the simplicial inclusion ι , practical needs may suggest that instead one *sparsifies* the complexes K and L to obtain (smaller) complexes and in the process one obtains a *simplicial map* connecting them (see Figure 1 for an illustration of the coarsening procedure in the case of graphs). This is the scenario described for example in [4, 3] and can be expressed through the following diagram where vertical arrows indicate the sparsification process:

$$\begin{array}{ccc}
 K & \xrightarrow{\iota} & L \\
 \downarrow & & \downarrow \\
 \tilde{K} & \xrightarrow{\varphi_\iota} & \tilde{L}
 \end{array}$$

This, therefore, motivates the study of persistent Laplacian for the setting where our input spaces (simplicial complexes) are connected by more general maps beyond inclusion, in particular, simplicial maps. This is the setting that we will study in this paper.

Contributions

We introduce a generalized version of the persistent Laplacian for weight preserving simplicial maps $f : K \rightarrow L$ between two weighted simplicial complexes K and L . Our work utilizes ideas from several different disciplines, including operator theory, spectral graph theory, and persistent homology. In more detail:

- In Section 2, we provide two equivalent definitions of the (up and down) persistent Laplacian for a weight preserving simplicial map $f : K \rightarrow L$. While one definition is more useful when proving some properties of the persistent Laplacian, the other definition provides a cleaner interpretation of the matrix representation of the persistent Laplacian. We also present one of the main properties of the persistent Laplacian, Theorem 7, which establishes that the nullity of $\Delta_q^{f:K \rightarrow L}$ equals the persistent Betti number of the (arbitrary) simplicial map $f : K \rightarrow L$, analogous to the nonpersistent and the inclusion-based persistent cases.
- In Section 3, we show that the Schur complement of a principal submatrix in a matrix can be viewed as a (Schur) restriction of a self-adjoint positive semi-definite operator to a subspace. In order to accomplish this, we find it useful to utilize some concepts and language from category theory. Viewing the set of self-adjoint positive semi-definite operators as the poset category of the Loewner order¹, we prove that Schur restriction is a right adjoint to the functor that extends an operator on a subspace to the whole space by composing with projection onto that subspace. We present our core observation about the Schur restriction, Theorem 11, which states that up and down persistent Laplacians can be obtained via Schur restrictions of the combinatorial up and down Laplacians.
- In Section 4, we present an algorithm to find a matrix representation of the persistent Laplacian for simplicial maps by the relation between up/down persistent Laplacians and the Schur restriction. We also analyze its complexity.
- In Section 5, we study the eigenvalues of up and down persistent Laplacians and prove monotonicity of these eigenvalues under the composition of simplicial maps.

Proofs of theorems and some extra details are available in the full version of this paper [8].

2 Persistent Laplacian for simplicial maps

2.1 Basics

Simplicial complexes and chain groups. An (abstract) simplicial complex K over a finite ordered vertex set V is a non-empty collection of non-empty subsets of V with the property that for every $\sigma \in K$, if $\tau \subseteq \sigma$, then $\tau \in K$. An element $\sigma \in K$ is called a q -simplex if the cardinality of σ is $q + 1$. We denote the set of q -simplices by S_q^K .

An oriented simplex, denoted $[\sigma]$, is a simplex $\sigma \in K$ whose vertices are ordered. As we start with an ordered vertex set, we always assume that the orientation on the simplices are inherited from the order on the vertex set. Let $\mathcal{S}_q^K := \{[\sigma] : \sigma \in K\}$.

¹ For two self-adjoint positive semi-definite operators L_1 and L_2 , the Loewner order is given by: $L_1 \succeq L_2$ if and only if $L_1 - L_2$ is positive semi-definite.

37:4 A Generalization of the Persistent Laplacian to Simplicial Maps

The q th chain group $C_q^K := C_q(K, \mathbb{R})$ of K is the vector space over \mathbb{R} with basis \mathcal{S}_q^K . Let $n_q^K := |\mathcal{S}_q^K| = \dim_{\mathbb{R}}(C_q^K)$.

The boundary operator $\partial_q^K : C_q^K \rightarrow C_{q-1}^K$ is defined by

$$\partial_q^K([v_0, \dots, v_q]) := \sum_{i=0}^q (-1)^i [v_0, \dots, \hat{v}_i, \dots, v_q] \quad (1)$$

for every q -simplex $\sigma = [v_0, \dots, v_q] \in \mathcal{S}_q^K$, where $[v_0, \dots, \hat{v}_i, \dots, v_q]$ denotes the omission of the i th vertex, and extended linearly to C_q^K .

A weight function on a simplicial complex K is any positive function $w^K : K \rightarrow (0, \infty)$. A simplicial complex is called weighted if it is endowed with a weight function. For every $q \in \mathbb{N}$, let $w_q^K := w^K|_{\mathcal{S}_q^K}$, the restriction of w^K onto \mathcal{S}_q^K . We define an inner product $\langle \cdot, \cdot \rangle_{w_q^K}$ on C_q^K as follows:

$$\langle [\sigma], [\sigma'] \rangle_{w_q^K} := \delta_{\sigma\sigma'} \cdot (w_q^K(\sigma))^{-1} \quad (2)$$

for all $\sigma, \sigma' \in \mathcal{S}_q^K$, where $\delta_{\sigma\sigma'}$ is the Kronecker delta.

Cochain groups as dual of chain groups. For clarification of some of our results/notations later, we also introduce certain concepts related to cochain groups. The cochain group C_K^q of K is the linear space consisting of all linear maps defined on C_q^K , i.e., $C_K^q := \text{hom}(C_q^K, \mathbb{R})$. The cochain group C_K^q also possesses a natural basis $\mathcal{S}_K^q := \{\chi_{[\sigma]} \mid [\sigma] \in \mathcal{S}_q^K\}$, where $\chi_{[\sigma]}$ is the linear map such that $\chi_{[\sigma]}([\tau]) = \delta_{[\sigma],[\tau]}$ for any $[\tau] \in \mathcal{S}_q^K$. We define an inner product $\langle\langle \cdot, \cdot \rangle\rangle_{w_q^K}$ on C_K^q as follows: for any $\chi_{[\sigma]}, \chi_{[\sigma']} \in \mathcal{S}_K^q$,

$$\langle\langle \chi_{[\sigma]}, \chi_{[\sigma']} \rangle\rangle_{w_q^K} := \delta_{\sigma\sigma'} \cdot w_q^K(\sigma). \quad (3)$$

Then, the map $j_q^K : C_q^K \rightarrow C_K^q$ sending a chain c to the linear map $\langle c, \cdot \rangle_{w_q^K}$ is an isometry w.r.t. the inner products of the two spaces. Moreover, the following diagram commutes:

$$\begin{array}{ccc} C_q^K & \xrightarrow{(\partial_{q+1}^K)^*} & C_{q+1}^K \\ j_q^K \downarrow & & \downarrow j_{q+1}^K \\ C_K^q & \xrightarrow{\delta_K^q} & C_K^{q+1} \end{array}$$

In this way, the adjoint $(\partial_{q+1}^K)^*$ of the boundary map ∂_{q+1}^K can be identified with the coboundary map δ_K^q . Similarly, $(\delta_K^q)^*$ can be identified with ∂_{q+1}^K . In the paper, we adopt the notation L^* to denote the adjoint of a linear map L between two inner product spaces.

Combinatorial Laplacian for simplicial complexes. Given a weighted simplicial complex K , one defines the q th combinatorial Laplacian Δ_q^K as follows:

$$\Delta_q^K := \partial_{q+1}^K \circ (\partial_{q+1}^K)^* + (\partial_q^K)^* \circ \partial_q^K : C_q^K \rightarrow C_q^K,$$

where $\Delta_{q,\text{up}}^K := \partial_{q+1}^K \circ (\partial_{q+1}^K)^*$ is called the q th up Laplacian and $\Delta_{q,\text{down}}^K := (\partial_q^K)^* \circ \partial_q^K$ is called the q th down Laplacian. Thanks to the renowned theorem by Eckmann [6], the combinatorial Laplacian is able to capture topological information of underlying simplicial complexes: the nullity of Δ_q^K agrees with the q th Betti number of K .

Simplicial maps. A simplicial map from a simplicial complex K into a simplicial complex L is a function from the vertex set of K to vertex set of L , $f : S_0^K \rightarrow S_0^L$, such that for every $\sigma \in K$, we have that $f(\sigma) \in L$. For every $q \in \mathbb{N}$, a simplicial map $f : K \rightarrow L$ induces a linear map $f_q : C_q^K \rightarrow C_q^L$ by the formula

$$f_q([v_0, \dots, v_q]) = \begin{cases} [f(v_0), \dots, f(v_q)] & \text{if } f(v_0), \dots, f(v_q) \text{ are distinct} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

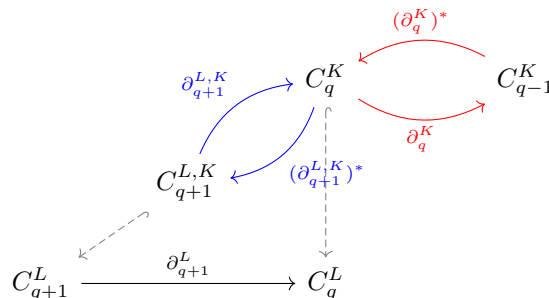
for every oriented q -simplex $[v_0, \dots, v_q] \in \mathcal{S}_q^K$. The linear map f_q does not have to preserve the orientation. That is, we could have that $f_q([\sigma]) = -[\tau]$ for some $[\sigma] \in \mathcal{S}_q^K$ and $[\tau] \in \mathcal{S}_q^L$. In this case, we write $\text{sgn}_{f_q}(\sigma) = -1$. We write $\text{sgn}_{f_q}(\sigma) = 1$ if $f_q([\sigma]) = [\tau]$.

► **Definition 1.** A simplicial map $f : K \rightarrow L$ between two weighted simplicial complexes is called weight preserving if for every $[\tau] \in \text{Im}(f_q)$ we have that

$$w_q^L(\tau) = \sum_{\substack{\sigma \in \mathcal{S}_q^K, \\ f_q([\sigma]) = \pm[\tau]}} w_q^K(\sigma). \tag{5}$$

2.2 The Persistent Laplacian for simplicial maps

The persistent Laplacian, whose definition we now recall, was initially defined only for inclusion maps. Given an inclusion map $\iota : K \hookrightarrow L$ between two simplicial complexes, we have the following commutative diagram



Here, $C_{q+1}^{L,K}$ denotes the subspace $C_{q+1}^{L,K} := \{c \in C_{q+1}^L \mid \partial_{q+1}^L(c) \in C_q^K\}$ of C_{q+1}^L , and $\partial_{q+1}^{L,K}$ denotes the restriction of ∂_{q+1}^L to $C_{q+1}^{L,K}$, i.e., $\partial_{q+1}^{L,K} := \partial_{q+1}^L|_{C_{q+1}^{L,K}} : C_{q+1}^{L,K} \rightarrow C_q^K$. Then, the q th up persistent Laplacian is defined as $\Delta_{q,\text{up}}^{K,L} := \partial_{q+1}^{L,K} \circ (\partial_{q+1}^{L,K})^*$, the q th down Laplacian is $\Delta_{q,\text{down}}^K = (\partial_q^K)^* \circ \partial_q^K$, and the q th persistent Laplacian is defined as

$$\Delta_q^{K,L} := \Delta_{q,\text{up}}^{K,L} + \Delta_{q,\text{down}}^K : C_q^K \rightarrow C_q^K. \tag{6}$$

Similarly to the case of the combinatorial Laplacian, the nullity of $\Delta_q^{K,L}$ recovers the persistent Betti number of the inclusion map $\iota : K \hookrightarrow L$ (cf. [16, Theorem 2.7]).

Re-examination of the persistent Laplacian for inclusion maps. Notice that (a) the definition of $C_{q+1}^{L,K}$ seems to depend on the fact that the map ι is an inclusion and (b) the down Laplacian part $\Delta_{q,\text{down}}^K$ does, a priori, not exhibit any dependence on L . However, the apparent dependence/independence mentioned in (a) and (b), respectively, are illusory. We now re-examine the definition above in order to motivate our extension of the notion of persistent Laplacian for simplicial maps.

37:6 A Generalization of the Persistent Laplacian to Simplicial Maps

First of all, we note that the expression $\partial_{q+1}^L(c) \in C_q^K$ in the definition of $C_{q+1}^{L,K}$ above is somewhat misleading. In fact, we are implicitly identifying C_q^K with its image $\iota_q(C_q^K)$ under the inclusion map $\iota_q : C_q^K \rightarrow C_q^L$ induced by ι . With this consideration, we rewrite $C_{q+1}^{L,K}$ in a more precise way:

$$C_{q+1}^{L,K} = \{c \in C_{q+1}^L \mid \partial_{q+1}^L(c) \in \iota_q(C_q^K)\}. \quad (7)$$

Expression (7) makes it clear that a certain set $\iota_q(C_q^K)$ is used in order to define the up Laplacian in the case of inclusions. This motivates us to consider the following dual construction which can be used to re-define the down Laplacian also in the case of inclusions

$$C_{q-1}^{K,L} := \{c \in C_{q-1}^K \mid (\partial_q^K)^*(c) \in (\iota_q)^*(C_q^L)\}. \quad (8)$$

As ι_q is injective, $(\iota_q)^*(C_q^L) = C_q^K$, and thus $C_{q-1}^{K,L} = C_{q-1}^K$. In this way, we see that using inclusion maps leads to concealing certain ‘‘persistence-like’’ structure inherent to the down part of the persistent Laplacian. An advantage of the formulation of the persistent Laplacian for general simplicial maps is that it will explicitly reveal this hidden structure.

Finally, we observe that for any $c \in C_{q+1}^{L,K}$, in fact, $\partial_{q+1}^L(c) \in \iota_q(\ker(\partial_q^K)) \subseteq \iota_q(C_q^K)$. This is simply due to the fact that $\partial_q^K \circ \partial_{q+1}^L(c) = \partial_q^L \circ \partial_{q+1}^L(c) = 0$. Here, we implicitly identify $\partial_{q+1}^L(c)$ with $\iota_q^{-1}(\partial_{q+1}^L(c))$ where ι_q^{-1} is the inverse of ι_q on its image. Hence, we have the following more refined expression for $C_{q+1}^{L,K}$:

$$C_{q+1}^{L,K} = \{c \in C_{q+1}^L \mid \partial_{q+1}^L(c) \in \iota_q(\ker(\partial_q^K))\}. \quad (9)$$

Integrating all these observations leads to our definition for the persistent Laplacian for general simplicial maps which we describe next.

Persistent Laplacian for simplicial maps. Suppose that we have a weight preserving simplicial map $f : K \rightarrow L$ and let $q \in \mathbb{N}$. Consider the subspaces

$$\mathfrak{C}_{q+1}^{L \leftarrow K} := \{c \in C_{q+1}^L \mid \partial_{q+1}^L(c) \in f_q(\ker(\partial_q^K))\},$$

$$\mathfrak{C}_{q-1}^{K \rightarrow L} := \{c \in C_{q-1}^K \mid (\partial_q^K)^*(c) \in \ker(f_q)^\perp\}.$$

Note that $\mathfrak{C}_{q+1}^{L \leftarrow K} \subseteq C_{q+1}^L$ and $\mathfrak{C}_{q-1}^{K \rightarrow L} \subseteq C_{q-1}^K$. Moreover, these spaces are natural generalizations of $C_{q+1}^{L,K}$ and $C_{q-1}^{K,L}$, respectively (cf. Equation (7) and Equation (8)), as $\ker(\iota_q)^\perp = C_q^K = (\iota_q)^*(C_q^L)$.

Let $\partial_{q+1}^{L,K}$ denote² the restriction of ∂_{q+1}^L to $\mathfrak{C}_{q+1}^{L \leftarrow K}$. Let $\delta_{q-1}^{K,L}$ denote³ the restriction of $(\partial_q^K)^*$ to $\mathfrak{C}_{q-1}^{K \rightarrow L}$. Furthermore, we let $\hat{f}_q : \ker(f_q)^\perp \rightarrow \text{Im}(f_q)$ denote the restriction of f_q onto $\ker(f_q)^\perp$. Before we proceed, we comment on some properties of \hat{f}_q and $\ker(f_q)^\perp$. We note that $\ker(f_q)^\perp$ possesses a canonical basis as follows. For every $[\tau] \in \text{Im}(f_q)$, we define

$$c_q^{\tau,f} := \sum_{\substack{\sigma \in S_q^K, \\ f_q([\sigma]) = \pm[\tau]}} \text{sgn}_{f_q}(\sigma) w_q^K(\sigma) [\sigma] \in C_q^K.$$

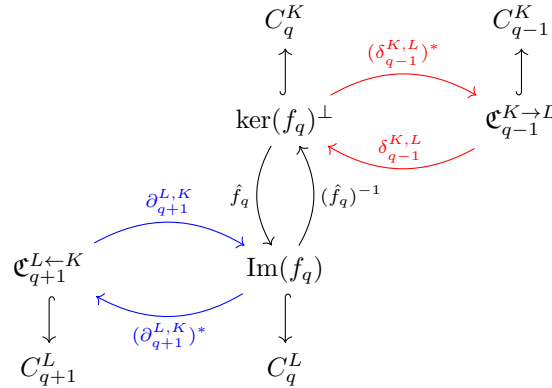
When the map f is clear from the content, we will simply write c_q^τ . We let $\mathcal{J} := \{c_q^\tau \mid [\tau] \in \text{Im}(f_q)\}$.

² The notation $\partial_{q+1}^{L,K}$ has been used before as the restriction of ∂_{q+1}^L to $C_{q+1}^{L,K}$. As $\mathfrak{C}_{q+1}^{L \leftarrow K}$ generalizes the space $C_{q+1}^{L,K}$, we stick to the same notation $\partial_{q+1}^{L,K}$ to denote the restriction of ∂_{q+1}^L to $\mathfrak{C}_{q+1}^{L \leftarrow K}$.

³ Recall that $(\partial_q^K)^*$ can be identified with the coboundary map δ_K^{q-1} in a sense specified in Subsection 2.1, hence we use $\delta_{q-1}^{K,L}$ to denote this restriction.

► **Lemma 2.** *The set \mathcal{J} is an orthogonal basis for $\ker(f_q)^\perp$. Moreover, the map $\hat{f}_q : \ker(f_q)^\perp \rightarrow \text{Im}(f_q)$ is an isometry between inner product spaces.*

Now, we consider the following diagram which contains all the notations we defined above:



We define up and down persistent Laplacian respectively as:

$$\Delta_{q,\text{up}}^{K \xrightarrow{f} L} := \partial_{q+1}^{L,K} \circ (\partial_{q+1}^{L,K})^* : \text{Im}(f_q) \rightarrow \text{Im}(f_q), \tag{10}$$

$$\Delta_{q,\text{down}}^{K \xrightarrow{f} L} := \hat{f}_q \circ \delta_{q-1}^{K,L} \circ (\delta_{q-1}^{K,L})^* \circ \hat{f}_q^{-1} : \text{Im}(f_q) \rightarrow \text{Im}(f_q). \tag{11}$$

As \hat{f}_q preserves inner product, we have that $\hat{f}_q^{-1} = \hat{f}_q^*$. Thus, both up and down persistent Laplacians are self-adjoint and non-negative operators on $\text{Im}(f_q)$. We then define the q -th persistent Laplacian $\Delta_q^{K \xrightarrow{f} L} : \text{Im}(f_q) \rightarrow \text{Im}(f_q)$ by:

$$\Delta_q^{K \xrightarrow{f} L} := \Delta_{q,\text{down}}^{K \xrightarrow{f} L} + \Delta_{q,\text{up}}^{K \xrightarrow{f} L}. \tag{12}$$

When the map $f : K \rightarrow L$ is clear, we will write $\Delta_q^{K,L}$ for the persistent Laplacian.

► **Remark 3.** By slightly abuse of notation, we also let f denote the simplicial map $f : K \rightarrow \text{Im}(f)$. Then, it follows from the definition of the down persistent Laplacian that $\Delta_{q,\text{down}}^{K \xrightarrow{f} L} = \Delta_{q,\text{down}}^{K \xrightarrow{f} \text{Im}(f)}$.

► **Remark 4.** When considering an inclusion $\iota : K \rightarrow L$, one can see that $\mathfrak{C}_{q-1}^{K \rightarrow L} = C_{q-1}^{K,L} = C_q^K$, $\mathfrak{C}_{q+1}^{L \leftarrow K} = C_{q+1}^{L,K}$ and $\iota_q : C_q^K \hookrightarrow C_q^L$ is an isometric embedding. Thus, our definition of persistent Laplacian generalizes the inclusion-based persistent Laplacian

► **Remark 5 (An alternative definition of the persistent Laplacian).** The weight preserving property of the simplicial map guarantees that $\ker(f_q)^\perp$ and $\text{Im}(f_q)$ are isometric, see Lemma 2. Thus, we could have, equivalently, defined the (up and down) persistent Laplacian as an operator on $\ker(f_q)^\perp$ instead of $\text{Im}(f_q)$ as follows:

$$\Delta_{q,\text{up}}^{K \xrightarrow{f} L} := \hat{f}_q^{-1} \circ \partial_{q+1}^{L,K} \circ (\partial_{q+1}^{L,K})^* \circ \hat{f}_q : \ker(f_q)^\perp \rightarrow \ker(f_q)^\perp,$$

$$\Delta_{q,\text{down}}^{K \xrightarrow{f} L} := \delta_{q-1}^{K,L} \circ (\delta_{q-1}^{K,L})^* : \ker(f_q)^\perp \rightarrow \ker(f_q)^\perp.$$

Note that when we have an inclusion $\iota : K \hookrightarrow L$, the (up/down) persistent Laplacian in [14, 16, 22] is defined on C_q^K , which is the same as $\ker(\iota_q)^\perp$ and isometrically isomorphic to $\text{Im}(\iota_q)$.

The two different definitions have their own advantages. Seeing the persistent Laplacian as an operator on $\text{Im}(f_q)$ increases the interpretability of this operator as the matrix representation can be computed using the canonical basis of $\text{Im}(f_q)$. On the other hand, seeing the persistent Laplacian on $\ker(f_q)^\perp$ helps us understanding some of its properties more easily. For example, see proof of Theorem 21.

► **Remark 6 (Cochain formulation of the persistent Laplacian).** Our generalization of the persistent Laplacian reveals a way to define a persistent Laplacian using the cochain spaces via dualization. If $f : K \rightarrow L$ is a simplicial map, then it induces a linear map in the cochain spaces $f^q : C_L^q \rightarrow C_K^q$, where $C_K^q = \text{hom}(C_K^K, \mathbb{R})$. Then, the following subspaces can be used to define a persistent Laplacian using cochains which in a sense extends the inclusion-based cochain formulation of the persistent (sheaf) Laplacian in [23]:

$$\begin{aligned} \mathfrak{C}_{L \leftarrow K}^{q+1} &:= \{c \in C_L^{q+1} \mid (\delta_q^L)^*(c) \in (f^q)^*(\ker(\delta_{q-1}^K)^*)\}, \\ \mathfrak{C}_{K \rightarrow L}^{q-1} &:= \{c \in C_K^{q-1} \mid \delta_{q-1}^K(c) \in \ker((f^q)^*)^\perp\}. \end{aligned}$$

It turns out that the operator defined via these spaces are the same as the persistent Laplacian defined using chains; see the full version of this paper for more details.

Let $\beta_q^{K \xrightarrow{f} L}$ denote the rank of the linear map $H_q(K) \rightarrow H_q(L)$ induced by f . $\beta_q^{K \xrightarrow{f} L}$ is called the persistent Betti number of the map $f : K \rightarrow L$. When the map $f : K \rightarrow L$ is clear from the content, we simply write $\beta_q^{K,L}$. With the machinery developed above together with several key observations that relates the (up and down) persistent Laplacians and Schur restriction of an operator, we have the following result.

► **Theorem 7 (Persistent Laplacians recover persistent Betti numbers).** *Let $f : K \rightarrow L$ be a simplicial map and $q \in \mathbb{N}$. Then, $\beta_q^{K,L} = \text{nullity}(\Delta_q^{K,L})$.*

► **Remark 8.** As the persistent Betti number does not depend on the weights on the simplicial complexes, weights can be assigned to K and L such that the simplicial map $f : K \rightarrow L$ is weight preserving. Then, one can use the persistent Laplacian to compute the persistent Betti number of f .

3 Schur Restriction and the Persistent Laplacian

One of the main contributions in [16] is a characterization of the up persistent Laplacian for inclusion maps via the so-called Schur complement. In this section, we establish that this characterization also holds in our setting of simplicial maps.

Let $M \in \mathbb{R}^{n \times n}$ be a block matrix $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ where $A \in \mathbb{R}^{(n-d) \times (n-d)}$ and $D \in \mathbb{R}^{d \times d}$.

The (generalized) Schur complement of D in M is $M/D := A - BD^\dagger C$, where D^\dagger is the Moore-Penrose generalized inverse of D .

A linear operator $L : V \rightarrow V$ on a finite dimensional real inner product space V is called *positive semi-definite* if $\langle L(v), v \rangle \geq 0$ for all $v \in V$, and it is called *self-adjoint* if $L^* = L$. The Schur complement, more generally, can be seen as a way of restricting a self-adjoint positive semi-definite operator on a real inner product space onto a subspace as follows. Assume that $L : V \rightarrow V$ is a self-adjoint positive semi-definite operator on V , where V is a finite dimensional ($\dim_{\mathbb{R}} V = n$) real inner product space. Let $W \subseteq V$ be a d -dimensional subspace and let W^\perp be its orthogonal complement. By choosing bases for W and W^\perp , we can represent L as a block matrix, say $[L] = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ where $A \in \mathbb{R}^{d \times d}$, $D \in \mathbb{R}^{(n-d) \times (n-d)}$.

Then, $[L]/D = A - BD^\dagger C$ can be interpreted as the restriction of L onto W , represented by the already chosen basis. We will see that the resulting operator represented by $[L]/D$ is independent of choice of basis (i.e. it is well-defined) and we call this operator the *Schur restriction of L onto W* , and denote it by $\mathbf{Sch}(L, W)$.

► **Proposition 9** (The Schur restriction is well-defined). *Let $L : V \rightarrow V$ be a self-adjoint positive semi-definite operator and let $W \subseteq V$ be a subspace. Then, $\mathbf{Sch}(L, W)$ is independent of choice of bases of W and W^\perp . More explicitly, if \mathcal{B}_1 and \mathcal{C}_1 are ordered bases for W and \mathcal{B}_2 and \mathcal{C}_2 are ordered bases for W^\perp , then the matrix representations of $\mathbf{Sch}(L, W)$ obtained from the ordered bases $\mathcal{B}_1 \cup \mathcal{B}_2$ and $\mathcal{C}_1 \cup \mathcal{C}_2$ are similar matrices via the change of basis matrix from \mathcal{B}_1 to \mathcal{C}_1 .*

As Proposition 9 guarantees that the Schur restriction of a self-adjoint positive semi-definite operator onto a subspace is well-defined, the next proposition reveals the recipe to acquire the Schur restriction and also justifies the name, “*Schur restriction*”.

► **Proposition 10.** *Let $f : \hat{V} \rightarrow V$ be a linear map between two finite dimensional real inner product spaces and let $L = f \circ f^* : V \rightarrow V$. Let $W \subseteq V$ be a subspace. Let $f_W : f^{-1}(W) \rightarrow W$ be the restriction of f on $f^{-1}(W)$ and the codomain is also restricted to W . Then, $\mathbf{Sch}(L, W) = f_W \circ f_W^*$.*

The proof we present for Proposition 10 heavily depends on the extremal characterization of Schur restrictions. It states that Schur restriction, as a functor, is a right adjoint. One of the most significant applications of Proposition 10 is the following theorem that establishes a relation between persistent Laplacians and the Schur restriction.

► **Theorem 11** (Up and down persistent Laplacians as Schur restrictions). *For a weight-preserving simplicial map $f : K \rightarrow L$, we have that*

$$\begin{aligned} \Delta_{q,\text{down}}^{K,L} &= \hat{f}_q \circ \mathbf{Sch}(\Delta_{q,\text{down}}^K, \ker(f_q)^\perp) \circ \hat{f}_q^{-1} \text{ and} \\ \Delta_{q,\text{up}}^{K,L} &= \iota_{\text{Im}(f_q)} \circ \mathbf{Sch}(\Delta_{q,\text{up}}^L, f_q(\ker(\partial_q^K))) \circ \text{proj}_{f_q(\ker(\partial_q^K))}, \end{aligned}$$

where $\iota_{\text{Im}(f_q)} : f_q(\ker(\partial_q^K)) \hookrightarrow \text{Im}(f_q)$ is the inclusion map and $\text{proj}_{f_q(\ker(\partial_q^K))} : \text{Im}(f_q) \rightarrow f_q(\ker(\partial_q^K))$ is the projection map.

4 Matrix Representation of Persistent Laplacian and an Algorithm

Based on the Schur restriction characterization of persistent Laplacians, i.e. Theorem 11, in the previous section, we now derive an algorithm for computing the matrix representation of persistent Laplacians.

4.1 Matrix Representation of Persistent Laplacian

Let $f : K \rightarrow L$ be a weight preserving simplicial map. Recall that for every oriented q -simplex $[\tau] \in \text{Im}(f_q)$, we defined the K q -chain

$$c_q^\tau := \sum_{\substack{\sigma \in S_q^K, \\ f([\sigma]) = \pm[\tau]}} \text{sgn}_{f_q}(\sigma) w_q^K(\sigma) [\sigma] \in C_q^K.$$

37:10 A Generalization of the Persistent Laplacian to Simplicial Maps

By Lemma 2, the set $\mathcal{J} = \{c_q^\tau \mid \tau \in \text{Im}(f_q)\}$ forms an orthogonal basis for $\ker(f_q)^\perp$. Assume that $\{[\tau_1], \dots, [\tau_n]\} \subseteq \text{Im}(f_q)$ is the set of all oriented q -simplices in L that are hit by f_q . Assume that for every $[\tau_i]$, $\{[\sigma_1^i], \dots, [\sigma_{d_i}^i]\} \subseteq \mathcal{S}_q^K$ is the set of all oriented q -simplices in K that are mapped to $\pm[\tau_i]$. Define

$$\sigma^{i,k} := \text{sgn}_{f_q}(\sigma_1^i)[\sigma_1^i] - \text{sgn}_{f_q}(\sigma_k^i)[\sigma_k^i]$$

for $i = 1, \dots, n$ and $k = 2, \dots, d_i$ for $d_i \geq 2$. Then, the set

$$\mathcal{B} = \{\sigma^{i,k} \mid 1 \leq i \leq n, 2 \leq k \leq d_i\} \cup \{[\sigma] \in \mathcal{S}_q^K \mid f_q([\sigma]) = 0\}$$

forms a basis for $\ker(f_q)$. Thus $\mathcal{J} \cup \mathcal{B}$ forms a basis for $C_q(K)$. Writing coordinates of basis elements of $\mathcal{J} \cup \mathcal{B}$ using the canonical basis \mathcal{S}_q^K as column vectors, we obtain the change of basis matrix $M_{\mathcal{J} \cup \mathcal{B} \rightarrow \mathcal{S}_q^K}$.

Matrix representation of down persistent Laplacian

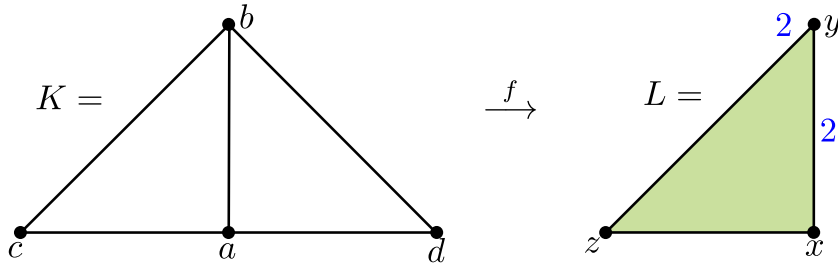
Let $[\Delta_{q,\text{down}}^K]$ be the matrix representation of $\Delta_{q,\text{down}}^K$ with respect to the canonical basis \mathcal{S}_q^K . Then, $N := (M_{\mathcal{J} \cup \mathcal{B} \rightarrow \mathcal{S}_q^K})^{-1} [\Delta_{q,\text{down}}^K] M_{\mathcal{J} \cup \mathcal{B} \rightarrow \mathcal{S}_q^K}$ is the matrix representation of $\Delta_{q,\text{down}}^K$ with respect to $\mathcal{J} \cup \mathcal{B}$. Given an integer m , let $[m]$ denote the set $[m] = \{1, 2, \dots, m\}$. The matrix N has dimension $n_q^K \times n_q^K$ where $n_q^K = |\mathcal{S}_q^K|$. Let $n := |\mathcal{J}| = \dim(\text{Im}(f_q)) = \dim(\ker(f_q)^\perp)$ and let

$$X = N([n], [n]), Y = N([n], [n_q^K] - [n]), Z = N([n_q^K] - [n], [n]), T = N([n_q^K] - [n], [n_q^K] - [n]). \quad (13)$$

Then, we can write N as a block matrix $N = \begin{pmatrix} X & Y \\ Z & T \end{pmatrix}$. Let $W_{\text{Im}(f_q)}$ denote the diagonal matrix $W_{\text{Im}(f_q)} = \text{diag}(w(\tau_1), w(\tau_2), \dots, w(\tau_n))$. Then, we are now ready to write the matrix representation of $\Delta_{q,\text{down}}^{K,L}$ with respect to the canonical basis $\{[\tau_1], \dots, [\tau_n]\}$ of $\text{Im}(f_q)$.

► **Proposition 12.** *With the notations above, the matrix representation of $\Delta_{q,\text{down}}^{K,L}$ with respect to the canonical basis $\{[\tau_1], \dots, [\tau_n]\}$ of $\text{Im}(f_q)$ is given by*

$$W_{\text{Im}(f_q)}(X - YT^\dagger Z)W_{\text{Im}(f_q)}^{-1}.$$



■ **Figure 2** A weight preserving simplicial map $f : K \rightarrow L$ between two weighted simplicial complexes K and L . K has all the weights equal to 1. In L , the edge xy and the vertex y has weights 2 and the rest of the simplicies have weight 1. The map f is given by $a \mapsto x, b \mapsto y, c \mapsto z, d \mapsto b$. And, ordering on the vertices are given by $a < b < c < d$ and $x < y < z$.

► **Example 13.** We will compute the matrix representation of the 1st down persistent Laplacian of the weight preserving simplicial map depicted in Figure 2. The 1st combinatorial down Laplacian of K is given by

$$[\Delta_{1,\text{down}}^K] = \begin{pmatrix} 2 & -1 & 1 & 1 & -1 \\ -1 & 2 & 1 & 0 & 1 \\ 1 & 1 & 2 & 1 & 0 \\ 1 & 0 & 1 & 2 & 1 \\ 1 & 1 & 0 & 1 & 2 \end{pmatrix}.$$

with respect to the canonical (ordered) basis $\mathcal{S}_1^K = \{[ab], [bc], [ac], [ad], [bd]\}$. Following the notation described above, we have that $\mathcal{J} = \{[ab] + [ad], [bc], [ac]\}$ and $\mathcal{B} = \{[ab] - [ad], [bd]\}$. Thus, we have the change of basis matrix as

$$M_{\mathcal{J} \cup \mathcal{B} \rightarrow \mathcal{S}_1^K} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Then, we compute

$$N = (M_{\mathcal{J} \cup \mathcal{B} \rightarrow \mathcal{S}_1^K})^{-1} [\Delta_{1,\text{down}}^K] M_{\mathcal{J} \cup \mathcal{B} \rightarrow \mathcal{S}_1^K} = \begin{pmatrix} 3 & -\frac{1}{2} & 1 & 0 & 0 \\ -1 & 2 & 1 & -1 & 1 \\ 2 & 1 & 2 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & 1 & -1 \\ 0 & 1 & 0 & -2 & 2 \end{pmatrix}.$$

Now, by extracting X, Y, Z , and T as described above in Equation (13), and realizing that $W_{\text{Im}(f_1)} = \text{diag}(2, 1, 1)$, we write the matrix representation of the 1st down persistent Laplacian $\Delta_{1,\text{down}}^{K,L}$ with respect to the basis $\{[xy], [yz], [xz]\}$ as follows

$$[\Delta_{1,\text{down}}^{K,L}] = W_{\text{Im}(f_q)}(X - YT^\dagger Z)W_{\text{Im}(f_q)}^{-1} = \begin{pmatrix} 3 & -1 & 2 \\ -\frac{1}{2} & \frac{3}{2} & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

Matrix representation of up persistent Laplacian

In order to write the matrix representation of up persistent Laplacian we need to choose bases \mathcal{B}_1 and \mathcal{B}_2 for $f_q(\ker(\partial_q^K))$ and $f_q(\ker(\partial_q^K))^\perp \subseteq \text{Im}(f_q)$ respectively, where $f_q(\ker(\partial_q^K))^\perp$ denotes the orthogonal complement of $f_q(\ker(\partial_q^K))$ inside the ambient space $\text{Im}(f_q)$. Let $\mathcal{D} = \{[\tau_{n+1}], \dots, [\tau_{n+l}]\} = \mathcal{S}_q^L - f_q(\pm \mathcal{S}_q^K)$. Then, $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D}$ is basis for $C_q(L)$. Writing the coordinates of this new basis elements with respect to the canonical basis \mathcal{S}_q^L as column vectors, we obtain the change of basis matrix

$$M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow \mathcal{S}_q^L} = \begin{pmatrix} R_1 & R_2 & 0_{n \times l} \\ 0_{l \times \text{rk} R_1} & 0_{l \times \text{rk} R_2} & \mathbb{I}_l \end{pmatrix}$$

where $R := (R_1 \ R_2)$ is the $n \times n$ change of basis matrix from $\mathcal{B}_1 \cup \mathcal{B}_2$ to the canonical basis of $\text{Im}(f_q)$, and \mathbb{I}_l is the $l \times l$ identity matrix.

37:12 A Generalization of the Persistent Laplacian to Simplicial Maps

Let $[\Delta_{q,\text{up}}^L]$ be the matrix representation of $\Delta_{q,\text{up}}^L$ with respect to the canonical basis of $C_q(L)$. Then, $Q = (M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow \mathcal{S}_q^L})^{-1} [\Delta_{q,\text{up}}^L] M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow \mathcal{S}_q^L}$ is the matrix representation of $\Delta_{q,\text{up}}^L$ with respect to $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D}$. Let $n_p = \dim(f_q(\ker(\partial_q^K)))$ and let $E = Q([n_p], [n_p])$. Thus we can write Q as a block matrix

$$Q = \begin{pmatrix} E & F \\ G & H \end{pmatrix}$$

where F, G, H are chosen appropriately to E . We are now ready to write the matrix representation of $\Delta_{q,\text{up}}^{K,L}$ with respect to the canonical basis of $\text{Im}(f_q)$.

► **Proposition 14.** *With the notations above, the matrix representation of $\Delta_{q,\text{up}}^{K,L}$ with respect to the canonical basis of $\text{Im}(f_q)$ is given by*

$$(R_1 \ R_2) \begin{pmatrix} E - FH^\dagger G & 0_{n_p \times (n-n_p)} \\ 0_{(n-n_p) \times n_p} & 0_{(n-n_p) \times (n-n_p)} \end{pmatrix} (R_1 \ R_2)^{-1}. \quad (14)$$

► **Example 15.** We will compute the matrix representation of the 1st up persistent Laplacian of the weight preserving simplicial map depicted in Figure 2. We will stick to the notation used above. We start by choosing bases \mathcal{B}_1 and \mathcal{B}_2 for $f_q(\ker(\partial_q^K))$ and $f_q(\ker(\partial_q^K))^\perp \subseteq \text{Im}(f_q)$ respectively. Observe that $f_q(\ker(\partial_q^K))$ is spanned by $[xy] + [yz] - [xz]$. So, we can choose $\mathcal{B}_1 = \{[xy] + [yz] - [xz]\}$ and $\mathcal{B}_2 = \{2[xy] - [yz], [yz] + [xz]\}$. As $f_1 : C_q^K \rightarrow C_1^L$ is surjective, we see that $\mathcal{D} = \emptyset$. Thus, $\mathcal{B}_1 \cup \mathcal{B}_2$ is a basis for C_1^L . Then, we have the change of basis matrix as

$$M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow \mathcal{S}_q^L} = M_{\mathcal{B}_1 \cup \mathcal{B}_2 \rightarrow \mathcal{S}_q^L} = \begin{pmatrix} 1 & 2 & 0 \\ 1 & -1 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

where $\mathcal{S}_1^L = \{[xy], [yz], [xz]\}$ is the canonical (ordered) basis of C_1^L . Moreover, we get that $(R_1 \ R_2) = M_{\mathcal{B}_1 \cup \mathcal{B}_2} = M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow \mathcal{S}_q^L}$. With respect to \mathcal{S}_1^L , the matrix representation of 1st combinatorial up Laplacian of L is given by

$$[\Delta_{1,\text{up}}^{K,L}] = \begin{pmatrix} \frac{1}{2} & 1 & -1 \\ \frac{1}{2} & 1 & -1 \\ -\frac{1}{2} & -1 & 1 \end{pmatrix}.$$

Now, we compute

$$Q = (M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow \mathcal{S}_q^L})^{-1} [\Delta_{q,\text{up}}^L] M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow \mathcal{S}_q^L} = \begin{pmatrix} \frac{5}{2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

and, we extract $E = (\frac{5}{2})$, $F = (0 \ 0)$, $G = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $H = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$. Thus, $E - FH^\dagger G = (\frac{5}{2})$.

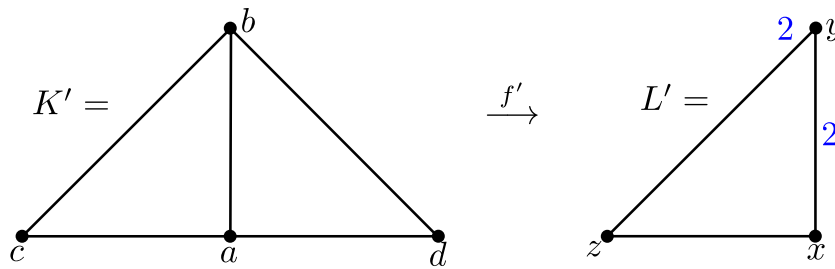
Thus, the matrix representation of $\Delta_{1,\text{up}}^{K,L}$ with respect to the basis $\mathcal{S}_1^L = \{[xy], [yz], [xz]\}$ is given by

$$[\Delta_{1,\text{up}}^{K,L}] = (R_1 \ R_2) \begin{pmatrix} E - FH^\dagger G & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} (R_1 \ R_2)^{-1} = \begin{pmatrix} \frac{1}{2} & 1 & -1 \\ \frac{1}{2} & 1 & -1 \\ -\frac{1}{2} & -1 & 1 \end{pmatrix}.$$

► Remark 16. By combining Example 13 and Example 15, we can see that the matrix representation of the 1st persistent Laplacian $\Delta_1^{K,L}$ is given by

$$[\Delta_1^{K,L}] = [\Delta_{1,\text{down}}^{K,L}] + [\Delta_{1,\text{up}}^{K,L}] = \begin{pmatrix} \frac{7}{2} & 0 & 1 \\ 0 & \frac{5}{2} & 0 \\ 1/2 & 0 & 3 \end{pmatrix}.$$

Then, we can justify Theorem 7 by observing that $\det([\Delta_1^{K,L}]) = 25 \neq 0$. That is, $\dim(\ker(\Delta_1^{K,L})) = 0 = \beta_1^{K,L}$.



■ **Figure 3** A weight preserving simplicial map $f' : K' \rightarrow L'$ between two weighted simplicial complexes K' and L' . K' has all the weights equal to 1. In L' , the edge xy and the vertex y has weights 2 and the rest of the simplicies have weight 1. The map f' is given by $a \mapsto x, b \mapsto y, c \mapsto z, d \mapsto b$. And, ordering on the vertices are given by $a < b < c < d$ and $x < y < z$.

► **Example 17.** Computing the matrix representation of the 1st persistent Laplacian of the map $f' : K' \rightarrow L'$ depicted in Figure 3 is similar to what we did for $f : K \rightarrow L$ in Example 13 and Example 15. Actually, $[\Delta_{1,\text{down}}^{K',L'}] = [\Delta_{1,\text{down}}^{K,L}]$ as $C_1^{K'} = C_1^{K,L}, C_1^{L'} = C_1^{L}$, and $f_1 = f'_1$. And, $[\Delta_{1,\text{up}}^{K',L'}] = 0_{3 \times 3}$ as $C_2^{L'} = \{0\}$. Thus,

$$[\Delta_1^{K',L'}] = [\Delta_{1,\text{down}}^{K',L'}] + [\Delta_{1,\text{up}}^{K',L'}] = [\Delta_{1,\text{down}}^{K,L}] + 0_{3 \times 3} = \begin{pmatrix} 3 & -1 & 2 \\ -\frac{1}{2} & \frac{3}{2} & 1 \\ 1 & 1 & 2 \end{pmatrix}.$$

Then, observe that $\dim(\ker(\Delta_1^{K',L'})) = 1 = \beta_1^{K',L'}$. Actually, the kernel of the matrix $[\Delta_1^{K',L'}]$ is generated by the vector $(1 \ 1 \ -1)^T$, which corresponds to the cycle $[xy] + [yz] - [xz]$ that can be seen as the image of the homology class that persists through the map f' .

4.2 An Algorithm for Computing the Persistent Laplacian

By Proposition 12 and Proposition 14, we have the matrix representations of up and down persistent Laplacians with respect to the canonical basis of $\text{Im}(f_q)$. So, simply adding them up, gives us the matrix representation of the persistent Laplacian $\Delta_q^{K,L}$ with respect to the canonical basis. In the process for finding these matrices, we use explicit bases $S_q^K, S_q^L, \mathcal{B} \cup \mathcal{J}$ and $\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D}$. However, we do not have an explicit basis for $f_q(\ker(\partial_q^K))$. Yet, we do not need to compute $\ker(\partial_q^K)$ in order to compute $f_q(\ker(\partial_q^K))$ by the following lemma.

► **Lemma 18.** $f_q(\ker(\partial_q^K)) = \ker(\Delta_{q,\text{down}}^{K,L})$.

■ **Algorithm 1** An algorithm for matrix representation of persistent Laplacian.

-
- 1: **Data:** $M_{\mathcal{J} \cup \mathcal{B} \rightarrow S_q^K}, [\Delta_{q,\text{down}}^K], [\Delta_{q,\text{up}}^L]$ and $W_{\text{Im}(f_q)}$
 - 2: **Result:** $[\Delta_q^{K,L}]$
 - 3: $N := M_{\mathcal{J} \cup \mathcal{B} \rightarrow S_q^K}^{-1} [\Delta_{q,\text{down}}^K] M_{\mathcal{J} \cup \mathcal{B} \rightarrow S_q^K}$
 - 4: $n := \dim(W_{\text{Im}(f_q)})$
 - 5: $[\Delta_{q,\text{down}}^{K,L}] := W_{\text{Im}(f_q)}(N/N([n_q^K] - [n], [n_q^K] - [n]))W_{\text{Im}(f_q)}^{-1}$
 - 6: Form $R = (R_1 \ R_2)$ by computing $\ker([\Delta_{q,\text{down}}^{K,L}])$
 - 7: Expand matrix R with the identity matrix to form $(n_q^L \times n_q^L)$ matrix $M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow S_q^L}$
 - 8: $Q := M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow S_q^L}^{-1} [\Delta_{q,\text{down}}^L] M_{\mathcal{B}_1 \cup \mathcal{B}_2 \cup \mathcal{D} \rightarrow S_q^L}$
 - 9: $n_p :=$ the number of columns of R_1
 - 10: **SchQ** := $Q/Q([n_q^L] - [n_p], [n_q^L] - [n_p])$
 - 11: Form the $n \times n$ matrix **PadSchQ** by zero padding to **SchQ**
 - 12: $[\Delta_{q,\text{up}}^{K,L}] = R^{-1} \text{PadSchQ} R$
 - 13:
 - 14: **return** $[\Delta_{q,\text{down}}^{K,L}] + [\Delta_{q,\text{up}}^{K,L}]$
-

4.2.1 Complexity

With the data we started in the Algorithm 1, we multiply matrices of dimension n_q^K and take Schur complement in a matrix of dimension n_q^K in order to compute $[\Delta_{q,\text{down}}^{K,L}]$. Thus, it takes $O((n_q^K)^3)$ time to compute $[\Delta_{q,\text{down}}^{K,L}]$. To compute $[\Delta_{q,\text{up}}^{K,L}]$, we compute kernel of a matrix of dimension $n < n_q^L$, take Schur complement in a matrix of dimension n_q^L , multiply matrices of dimension n_q^L and of dimension n . Hence, it takes $O((n_q^L)^3)$ time to compute $[\Delta_{q,\text{up}}^{K,L}]$. Therefore, it takes $O((n_q^K)^3) + (n_q^L)^3$ time to compute $[\Delta_q^{K,L}]$ in total.

It is important to note that the data we started in the Algorithm 1 also takes time to compute. Starting with boundary matrices and weight matrices, it takes $O((n_q^K)^2)$ time to compute $[\Delta_{q,\text{down}}^K]$ and it takes $O(n_{q+1}^L)$ to compute $[\Delta_{q,\text{up}}^L]$ as discussed in [16]. Thus, starting from scratch, Algorithm 1 computes $[\Delta_q^{K,L}]$ in $O((n_q^K)^3) + (n_q^L)^3 + n_{q+1}^L$ time.

Note that by Theorem 7, as a by-product, the above algorithm can also output the persistent Betti number for a simplicial map $f : K \rightarrow L$ in the same time complexity. This provides an alternative way to compute persistent Betti numbers for $f : K \rightarrow L$ that is different from the existing algorithm by Dey et al. [3] already in the literature.

5 Monotonicity of (up/down) persistent eigenvalues

For a simplicial map $f : K \rightarrow L$, the up and down persistent Laplacians are self-adjoint positive semi-definite operators. Therefore, they have non-negative eigenvalues. We denote them by $0 \leq \lambda_{q,\text{up},1}^{K,L} \leq \lambda_{q,\text{up},2}^{K,L} \leq \dots \leq \lambda_{q,\text{up},n}^{K,L}$, and $0 \leq \lambda_{q,\text{down},1}^{K,L} \leq \lambda_{q,\text{down},2}^{K,L} \leq \dots \leq \lambda_{q,\text{down},n}^{K,L}$, allowing repetition, where $n = \dim(\text{Im}(f_q))$. And, we call them the *up persistent eigenvalues* and the *down persistent eigenvalues*.

When the simplicial maps involved are inclusions, we have the following known monotonicity result for the up persistent Laplacian.

► **Theorem 19** ([16, Theorem 5.3]). *Let $f : K \hookrightarrow L$ and $g : L \hookrightarrow M$ be inclusion maps for simplicial complexes K, L and M . Then, for any $q \in \mathbb{N}$ and $k = 1, 2, \dots, n_q^K$,*

$$\lambda_{q,\text{up},k}^{K,M} \geq \lambda_{q,\text{up},k}^{L,M} \text{ and } \lambda_{q,\text{up},k}^{K,M} \geq \lambda_{q,\text{up},k}^{K,L}.$$

In Theorem 19, the monotonicity result of up persistent eigenvalues $\lambda_{q,\text{up},k}^{K,M} \geq \lambda_{q,\text{up},k}^{K,L}$ follows from the fact that $\Delta_{q,\text{up}}^{K,M} \succeq \Delta_{q,\text{up}}^{K,L}$. In the case of surjective maps, we present an analogous statement for the down persistent Laplacians as follows.

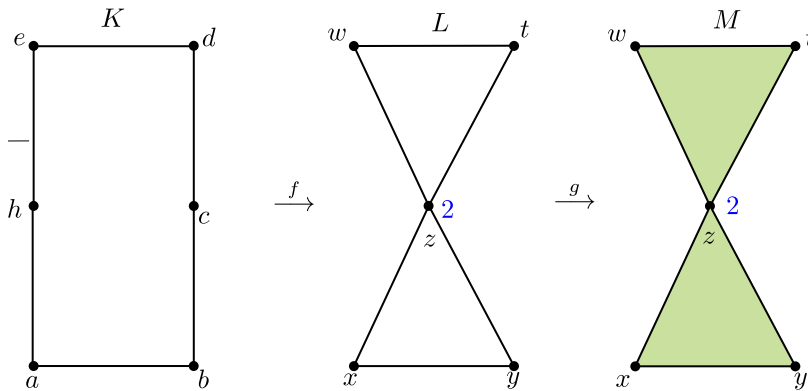
► **Proposition 20.** *Let $f : K \rightarrow L$ and $g : L \rightarrow M$ be weight preserving surjective simplicial maps. Then, $\Delta_{q,\text{down}}^{K,M} \succeq \Delta_{q,\text{down}}^{L,M}$.*

When the surjectivity assumption is removed, it is no longer guaranteed that the composition of two weight preserving maps is weight preserving. However, under the assumption that two maps and their composition are weight preserving, we get the monotonicity of the down persistent eigenvalues.

► **Theorem 21.** *Let $f : K \rightarrow L$ and $g : L \rightarrow M$ be weight preserving simplicial maps and assume that $g \circ f : K \rightarrow M$ is also weight preserving. Then, for any $q \in \mathbb{N}$ and $k = 1, 2, \dots, \dim(\text{Im}(g_q \circ f_q))$,*

$$\lambda_{q,\text{down},k}^{K,M} \geq \lambda_{q,\text{down},k}^{L,M} \quad \text{and} \quad \lambda_{q,\text{down},k}^{K,M} \geq \lambda_{q,\text{down},k}^{K,L}.$$

However, this type of monotonicity does not hold in general for up persistent eigenvalues even if we require weight preserving conditions for the involved simplicial maps as we did in Theorem 21. See the counterexample as follows.



■ **Figure 4** Composition of two weight preserving simplicial maps $f : K \rightarrow L$ and $g : L \rightarrow M$, where f is given by collapsing the vertices h and c to the same vertex z . And, g is given by the identity map on the vertices.

► **Example 22** (Up persistent eigenvalues are not monotonic). Considering the simplicial complexes K, L, M and the simplicial maps f, g depicted in Figure 4, we compute spectra of $\Delta_{1,\text{up}}^{K,M}$ and $\Delta_{1,\text{up}}^{L,M}$. It turns out that $\Delta_{1,\text{up}}^{K,M}$ has eigenvalues $0 \leq 0 \leq 0 \leq 0 \leq 0 \leq 3$ and $\Delta_{1,\text{up}}^{L,M}$ has eigenvalues $0 \leq 0 \leq 0 \leq 0 \leq 3 \leq 3$. So, $0 = \lambda_{1,\text{up},5}^{K,M} \not\geq \lambda_{1,\text{up},5}^{L,M} = 3$.

Recall from Theorem 11 that $\Delta_{q,\text{up}}^{K,L} = \iota_{\text{Im}(f_q)} \circ \text{Sch}(\Delta_{q,\text{up}}^L, f_q(\ker(\partial_q^K))) \circ \text{proj}_{f_q(\ker(\partial_q^K))}$. This formulation reveals that the up persistent Laplacian is obtained by extending the operator $\text{Sch}(\Delta_{q,\text{up}}^L, f_q(\ker(\partial_q^K)))$ defined on $f_q(\ker(\partial_q^K))$ to its superspace $\text{Im}(f_q)$ by “padding zeros”. This extension naturally introduces inevitable 0 eigenvalues to the up persistent Laplacian and we call them inevitable 0 eigenvalues. Considering again Example 22, we see that $g_1(f_1(\ker(\partial_1^K)))$ has dimension 1 and codimension 5 inside $\text{Im}(g_1 \circ f_1)$. Thus, $\Delta_{1,\text{up}}^{K,M}$ has 5 inevitable 0 eigenvalues. Similarly, $\Delta_{1,\text{up}}^{L,M}$ has 4 inevitable 0 eigenvalues as the codimension of $g_1(\ker(\partial_1^L))$ inside $\text{Im}(g_1)$ is 4. Disregarding these inevitable 0 eigenvalues from their

spectra, we see that $\Delta_{\text{up}}^{K,M}$ essentially has $\{3\}$ as its spectrum, while $\Delta_{1,\text{up}}^{L,M}$ essentially has $\{3, 3\}$ as its spectrum. Then, it seems that if we disregard inevitable 0 eigenvalues, we will obtain monotonicity for the eigenvalues of up persistent Laplacians. This is indeed the case:

We call $\text{Sch}(\Delta_{q,\text{up}}^L, f_q(\ker(\partial_q^K)))$ the *essential up persistent Laplacian*, whose spectrum is the same as the spectrum of $\Delta_{q,\text{up}}^{K,L}$ up to a difference in the multiplicity of the 0 eigenvalue. Then, we establish monotonicity of the eigenvalues of the essential up persistent Laplacian, which are denoted by $\lambda_{q,\text{up},k}^{K,L,\text{ess}}$, and are called *essential up persistent eigenvalues*.

► **Theorem 23.** *Let $f : K \rightarrow L$ and $g : L \rightarrow M$ be weight preserving simplicial maps. Then, for any $q \in \mathbb{N}$ and $k = 1, 2, \dots, \dim(g_q(f_q(\ker(\partial_q^K))))$, we have $\lambda_{q,\text{up},k}^{K,M,\text{ess}} \geq \lambda_{q,\text{up},k}^{L,M,\text{ess}}$.*

This monotonicity result on essential up persistent eigenvalues is stronger than the monotonicity result for inclusion maps (cf. Theorem 19) in that the latter is a direct consequence of the former.

6 Discussion

Once an invariant is associated to a simplicial filtration/tower, one of the most natural questions would be about its stability. So, it is highly desirable to explore the stability of the (up/down) persistent eigenvalues/eigenspaces that could potentially generalize the stability of up persistent eigenvalues in the inclusion-based persistent Laplacian [16, Theorem 5.10].

The persistent diagram of a Rips complex can be approximated by using simplicial towers obtained from the Rips complex such as sparsified Rips complex or graph induced complex as described in [4, 3]. Therefore, one might consider if the spectrum of the (up/down) persistent Laplacian can also be approximated via a similar sparsification process.

References

- 1 Jiahui Chen, Yuchi Qiu, Rui Wang, and Guo-Wei Wei. Persistent laplacian projected omicron ba. 4 and ba. 5 to become new dominating variants. *arXiv preprint arXiv:2205.00532*, 2022.
- 2 Fan Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- 3 Tamal Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. *Proceedings of the Annual Symposium on Computational Geometry*, 2012.
- 4 Tamal Krishna Dey, Fengtao Fan, and Yusu Wang. Graph induced complex on point data. In *Proceedings of the twenty-ninth annual symposium on Computational geometry*, pages 107–116, 2013.
- 5 Art Duval and Victor Reiner. Shifted simplicial complexes are Laplacian integral. *Transactions of the American Mathematical Society*, 354(11):4313–4344, 2002.
- 6 Beno Eckmann. Harmonische funktionen und randwertaufgaben in einem komplex. *Commentarii Mathematici Helvetici*, 17(1):240–255, 1944.
- 7 Timothy E Goldberg. Combinatorial Laplacians of simplicial complexes. *Senior Thesis, Bard College*, 2002.
- 8 Aziz Burak Gülen, Facundo Mémoli, Zhengchao Wan, and Yusu Wang. A generalization of the persistent laplacian to simplicial maps. *arXiv preprint arXiv:2302.03771*, 2023.
- 9 Renee S Hoekzema, Lewis Marsh, Otto Sumray, Thomas M Carroll, Xin Lu, Helen M Byrne, and Heather A Harrington. Multiscale methods for signal selection in single-cell data. *Entropy*, 24(8):1116, 2022.
- 10 Danijela Horak and Jürgen Jost. Spectra of combinatorial Laplace operators on simplicial complexes. *Advances in Mathematics*, 244:303–336, 2013.
- 11 Yuta Hozumi, Rui Wang, and Guo-Wei Wei. Cep: Correlated clustering and projection for dimensionality reduction. *arXiv preprint arXiv:2206.04189*, 2022.

- 12 Ioannis Koutis, Gary Miller, and Richard Peng. A fast solver for a class of linear systems. *Communications of the ACM*, 55, 2012.
- 13 James R Lee, Shayan Oveis Gharan, and Luca Trevisan. Multi-way spectral partitioning and higher-order Cheeger inequalities. In *Symposium on Theory of Computing (STOC)*, pages 1117–1130, 2012. doi:10.1145/2213977.2214078.
- 14 André Lieutier. Talk: Persistent harmonic forms. URL: <https://project.inria.fr/gudhi/files/2014/10/Persistent-Harmonic-Forms.pdf>.
- 15 Oren E Livne and Achi Brandt. Lean algebraic multigrid (lamg): Fast graph Laplacian linear solver. *SIAM Journal on Scientific Computing*, 34(4):B499–B522, 2012.
- 16 Facundo Mémoli, Zhengchao Wan, and Yusu Wang. Persistent laplacians: Properties, algorithms and implications. *SIAM Journal on Mathematics of Data Science*, 4(2):858–884, 2022.
- 17 Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in Neural Information Processing Systems*, 14(2):849–856, 2002.
- 18 Daniel Spielman. Spectral and algebraic graph theory. *Yale lecture notes, draft of December*, 4:47, 2019.
- 19 Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 81–90, 2004.
- 20 Nisheeth K Vishnoi. $Lx = b$: Laplacian solvers and their algorithmic applications. *Found. Trends Theor. Comput. Sci.*, 8(1-2):1–141, 2013. doi:10.1561/04000000054.
- 21 Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. doi:10.1007/s11222-007-9033-z.
- 22 Rui Wang, Duc Duy Nguyen, and Guo-Wei Wei. Persistent spectral graph. *International Journal for Numerical Methods in Biomedical Engineering*, page e3376, 2020.
- 23 Xiaoqi Wei and Guo-Wei Wei. Persistent sheaf Laplacians. *arXiv preprint arXiv:2112.10906*, 2021.

The Christoffel-Darboux Kernel for Topological Data Analysis

Pepijn Roos Hoefgeest ✉

Vrije Universiteit (VU) Amsterdam, The Netherlands

Lucas Slot ✉

ETH Zürich, Switzerland

Abstract

Persistent homology has been widely used to study the topology of point clouds in \mathbb{R}^n . Standard approaches are very sensitive to outliers, and their computational complexity depends badly on the number of data points. In this paper we introduce a novel persistence module for a point cloud using the theory of Christoffel-Darboux kernels. This module is robust to (statistical) outliers in the data, and can be computed in time linear in the number of data points. We illustrate the benefits and limitations of our new module with various numerical examples in \mathbb{R}^n , for $n = 1, 2, 3$. Our work expands upon recent applications of Christoffel-Darboux kernels in the context of statistical data analysis and geometric inference [13]. There, these kernels are used to construct a polynomial whose level sets capture the geometry of a point cloud in a precise sense. We show that the persistent homology associated to the sublevel set filtration of this polynomial is stable with respect to the Wasserstein distance. Moreover, we show that the persistent homology of this filtration can be computed in singly exponential time in the ambient dimension n , using a recent algorithm of Basu & Karisani [1].

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases Topological Data Analysis, Geometric Inference, Christoffel-Darboux Kernels, Persistent Homology, Wasserstein Distance, Semi-Algebraic Sets

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.38

Related Version *Full Version*: <https://arxiv.org/abs/2211.15489>

1 Introduction

Persistent homology is a central tool in the field of topological data analysis. It was developed in the early 2000s in order to extract topological and geometric information out of point-cloud data. Since discrete points in \mathbb{R}^n do not have any meaningful topological features in and of themselves, one needs to find a way to construct an “interesting” topological space out of them. An obvious approach is to look at the collection of balls of radius r centered around the data points. When the radius r is chosen correctly, these balls will intersect in ways that reflect the topology of the set the data is sampled from. However, it is not clear a priori which radius should be chosen, and in fact, a *single* “correct” choice need not even exist. The solution is to look at *all* radii $r \geq 0$, and to track which topological features *persist* over time as r increases. More concretely: if $r \leq r'$, then the balls of radius r include into those of radius r' , and this collection of inclusions forms what is called a *filtration*. Persistent homology tracks “birth-death events” of homology classes in such a filtration, see Figure 1. Classical approaches to obtain a filtration of topological spaces out of a point cloud, such as the Čech filtration (outlined above) and the Vietoris-Rips filtration [23] suffer from two main problems:

1. The complexity of computing the persistent homology depends badly on the number of data points.
2. The persistent homology of these filtrations is very sensitive to outliers in the data.



© Pepijn Roos Hoefgeest and Lucas Slot;

licensed under Creative Commons License CC-BY 4.0

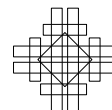
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 38; pp. 38:1–38:20

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



These issues have been addressed in the literature in several ways. *Alpha complexes* [9] are used to compute the persistent homology of the Čech filtration efficiently by first intersecting the metric balls with a Voronoi diagram. *Witness complexes* [18] build a small simplicial complex based on a subsample of the data, thus reducing computational complexity. Heuristically, these subsamples may be chosen to reduce sensitivity to outliers in the full data set, although this effect remains hard to quantify [19]. Chazal et al. [4] introduce the *distance-to-measure* function, which they apply to perform geometric inference of point clouds in \mathbb{R}^n . The key feature of this function is that it is stable with respect to the *Wasserstein distance*, implying robustness to (statistical) outliers in the data. Buchet et al. [3] use this property to construct a filtration which is also provably stable in this sense. However, it is hard to compute the associated persistent homology, and they therefore employ an approximation scheme.

In this paper, we propose a novel filtration based on so-called *Christoffel-Darboux kernels*. As we explain in more detail below, the resulting persistent homology can be computed in *linear time* in the number of data points, and is provably robust to statistical outliers. Christoffel-Darboux (CD) kernels have a long history in fundamental mathematics, with applications to orthogonal polynomials and in approximation theory (see [13] for an overview). They are the reproducing kernels $K_d^\mu : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ for the Hilbert space $\mathbb{R}[\mathbf{x}]_d$ of n -variate, real polynomials of degree $d \in \mathbb{N}$ with respect to the inner product $\langle p, q \rangle = \int pq d\mu$ induced by a finite measure μ on \mathbb{R}^n . Such reproducing kernels completely describe the inner product $\langle \cdot, \cdot \rangle$, and in our setting the kernels K_d^μ thus capture information about the underlying measure μ . For instance, the *Christoffel polynomial* $P_d^\mu(\mathbf{x}) := K_d^\mu(\mathbf{x}, \mathbf{x})$ can be used to estimate the support $\text{supp}(\mu) \subseteq \mathbb{R}^n$ of μ . Roughly speaking, $P_d^\mu(\mathbf{x})$ is small when $\mathbf{x} \in \text{supp}(\mu)$ and large when $\mathbf{x} \notin \text{supp}(\mu)$ (see Proposition 10 below). This property has recently been applied to perform geometric inference in a statistical setting by Lasserre, Pauwels and Putinar [12, 13, 15]. In these works, the authors consider CD kernels for the *empirical* measure $\mu_{\mathcal{X}}$ associated to a set of samples \mathcal{X} drawn according to some unknown measure μ . For fixed $d \in \mathbb{N}$, the polynomial $P_d^{\mu_{\mathcal{X}}}$ associated to \mathcal{X} is straightforward to compute. Moreover, the *sublevel set* $\{\mathbf{x} \in \mathbb{R}^n : P_d^{\mu_{\mathcal{X}}}(\mathbf{x}) \leq t\}$ captures the support of μ well for suitably selected $t \geq 0$, see Figure 2. However, a key issue of this approach is that the level $t \geq 0$ must be selected “by hand” based on heuristics, and the quality of geometric inference depends heavily on this choice. This problem motivates our use of persistent homology, which considers all sublevel sets simultaneously.

1.1 Contributions and outline

We propose a new scheme for topological data analysis of a finite point cloud $\mathcal{X} \subseteq [-1, 1]^n$, based on Christoffel-Darboux kernels. Our scheme unites recent applications of CD kernels in (statistical) data analysis with ideas from persistent homology. It consists of three steps:

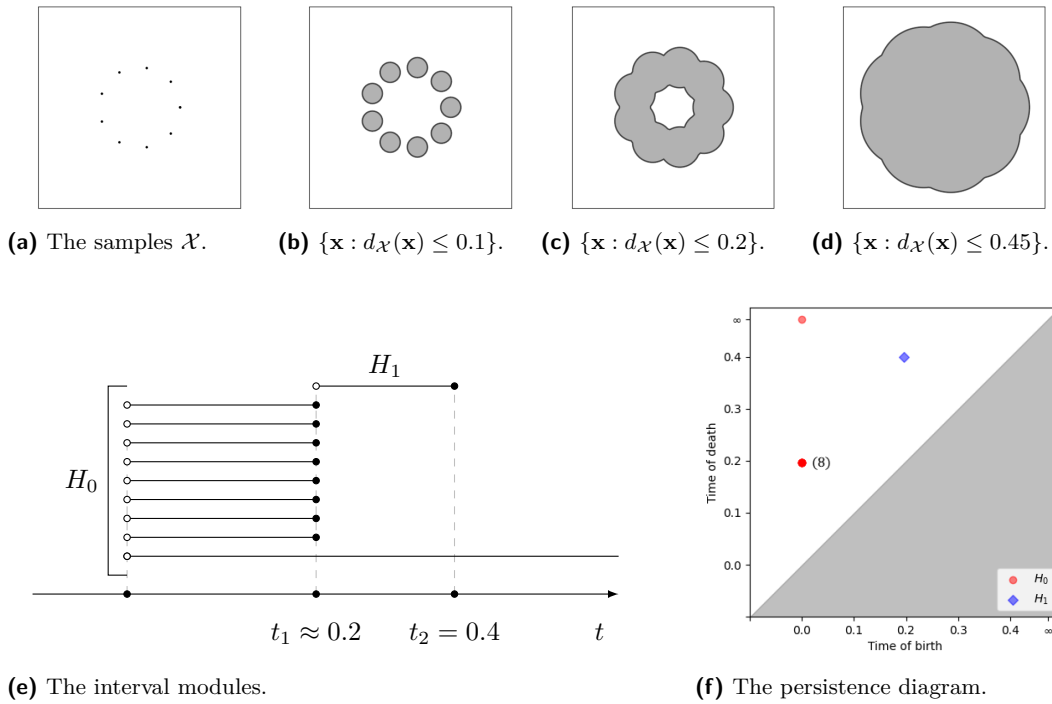
- Moment matrix.** Fix $d \in \mathbb{N}$. Choose a basis $\mathbf{b} = (b_\alpha)$ for the space $\mathbb{R}[\mathbf{x}]_d$ of n -variate polynomials of degree at most d . Compute the *moment matrix* $M_d(\mathbf{b})$ of size $\binom{n+d}{d}$, whose entries can be computed from \mathcal{X} in linear time via:

$$M_d(\mathbf{b})_{\alpha, \beta} := \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} b_\alpha(\mathbf{x}) b_\beta(\mathbf{x}) \quad (\alpha, \beta \in \mathbb{N}^n, |\alpha|, |\beta| \leq d).$$

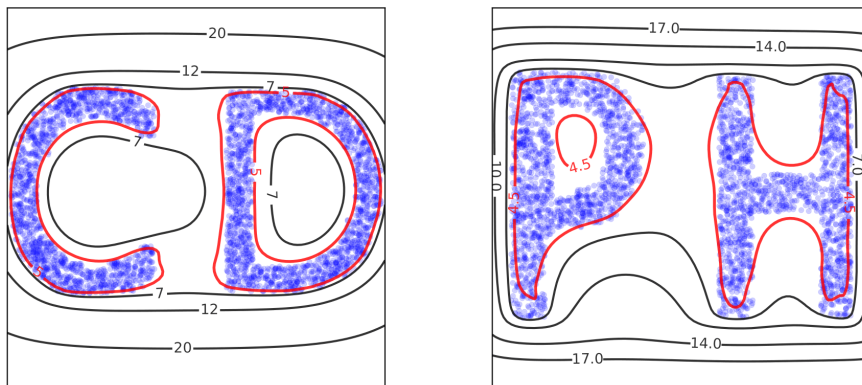
- Christoffel polynomial.** Invert the moment matrix to obtain the *Christoffel polynomial*:

$$P_d(\mathbf{x}) := \mathbf{b}(\mathbf{x})^\top (M_d(\mathbf{b}))^{-1} \mathbf{b}(\mathbf{x}),$$

whose sublevel sets are known to approximate the set \mathcal{X} , see Figure 2.



■ **Figure 1** A filtration of $[-1, 1]^2$ by the distance function $d_{\mathcal{X}} : x \mapsto \text{dist}(x, \mathcal{X})$ to a set of equidistant points \mathcal{X} on a circle of radius 0.4, and the corresponding persistence diagram. Note that there are 8 intervals that are born at $t = 0$ and die at $t \approx 0.2$, which show up as a single dot in the diagram. Throughout, we indicate the number of such overlapping dots in the diagram if necessary for clarity.



■ **Figure 2** The level sets of the Christoffel polynomial $x \mapsto P_{10}^{\mu_{\mathcal{X}}}(\mathbf{x})$ associated to the empirical measure $\mu_{\mathcal{X}}$ of two sample sets $\mathcal{X} \subseteq [-1, 1]^2$ (in blue). The level sets indicated in red capture the support of the underlying measure μ quite well.

3. **Persistence module.** Define the *sublevel set filtration*:

$$\mathbf{X}_t := \{\mathbf{x} \in [-1, 1]^n : \log P_d(\mathbf{x}) \leq t\} \quad (t \geq 0),$$

and compute its associated *persistence module*:

$$\mathbb{CD}(\mathcal{X}, d) := \text{PH}_*([-1, 1]^n, \log P_d).$$

Robustness to statistical outliers. We show that the module $\mathbb{CD}(\mathcal{X}, d)$ is stable and robust under perturbations of the input data \mathcal{X} . To be precise, we show *local* Lipschitz continuity of the function $\mathcal{X} \mapsto \mathbb{CD}(\mathcal{X}, d)$, in the *Bottleneck* and *Wasserstein* distance. We also give an estimate of the Lipschitz constant in terms of a concrete measure of algebraic degeneracy of the set \mathcal{X} . This is our main technical result, see Section 3.1.

Exact algorithm with linear dependence on the number of samples. We give an exact algorithm for computing the persistence module $\mathbb{CD}(\mathcal{X}, d)$ in Section 3.2, whose runtime is linear in the number of data points, but depends exponentially on the dimension n . This algorithm is a combination of 1) a known procedure to compute CD kernels and 2) the recent work [1], in which the authors propose an algorithm for computing the persistent homology of *semialgebraic* filtrations.

Numerical examples. We provide several numerical examples in Section 4 that illustrate the geometric properties of our scheme, and its potential benefits and downsides compared to existing methods. Unfortunately, there is no practical implementation available of the algorithm proposed in [1]. In order to perform numerical experiments, we therefore propose a simple scheme for approximating $\mathbb{CD}(\mathcal{X}, d)$ in Section 3.3, based on a triangulation of the sample space $[-1, 1]^n$. These experiments show that our novel persistence module is able to accurately capture underlying homological features of point clouds, even in the presence of outliers.

2 Background

Notations and conventions. Throughout, $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathbb{R}^n$ are n -dimensional variables. We denote by $\mathbb{R}[\mathbf{x}]$ the n -variate polynomial ring. We write $\mathbb{R}[\mathbf{x}]_d \subseteq \mathbb{R}[\mathbf{x}]$ for the subspace of polynomials of (total) degree at most d , which has (real) dimension $s(n, d) = \binom{n+d}{d}$. For ease of exposition, we assume throughout that sets of samples \mathcal{X}, \mathcal{Y} are contained in the box $[-1, 1]^n$, which can always be achieved by a rescaling.

2.1 Persistent homology

Persistent homology is a central tool in topological data analysis, and has received a lot of attention over recent years [8]. It serves to track homology classes through a diagram of spaces, typically arising from a filtration: Let \mathbf{X} be a filtered topological space, that is, for each $t \in \mathbb{R}$, there is a subspace $\mathbf{X}_t \subseteq \mathbf{X}$, such that if $s \leq t$, $\mathbf{X}_s \subseteq \mathbf{X}_t$. For convenience, we assume that the filtration is exhaustive, i.e. $\bigcup_t \mathbf{X}_t = \mathbf{X}$. Applying homology with coefficients in \mathbb{F} to each \mathbf{X}_t then yields a diagram of spaces $H_*(\mathbf{X}_t; \mathbb{F})$: For every $s \leq t$, the inclusion map $\iota_s^t : \mathbf{X}_s \rightarrow \mathbf{X}_t$ induces a map $h_s^t = (\iota_s^t)_* : H_*(\mathbf{X}_s; \mathbb{F}) \rightarrow H_*(\mathbf{X}_t; \mathbb{F})$, and the collection of maps $\{h_s^t\}$ satisfy:

1. For all $r \leq s \leq t$, $h_s^t \circ h_r^s = h_r^t$;
2. For all $t \in \mathbb{R}$, $h_t^t = \text{id}_{H_*(\mathbf{X}_t; \mathbb{F})}$.

This diagram of spaces is the *persistent homology* of \mathbf{X} , denoted by $\text{PH}_*(\mathbf{X}_t; \mathbb{F})$. Any \mathbb{R} -indexed collection of vector spaces with maps satisfying **1.** and **2.** above is called a *persistence module*. More succinctly put, a persistence module is a functor from the poset (\mathbb{R}, \leq) to the category of vector spaces over some field. The vector spaces can be taken over any field \mathbb{F} , but we always work over a finite field.

► **Example 1.** Let \mathbf{X} be a topological space, and let $f : \mathbf{X} \rightarrow \mathbb{R}$ be a continuous function. Then the sublevel set filtration of \mathbf{X} with respect to f is given by $\mathbf{X}_t = \{\mathbf{x} \in \mathbf{X} \mid f(\mathbf{x}) \leq t\}$. Applying homology to this filtration yields a persistence module, which we denote by $\text{PH}_*(\mathbf{X}, f)$.

If $H_*(\mathbf{X}_t)$ is finite dimensional for each $t \in \mathbb{R}$ (which is a mild requirement, and will always be satisfied in our setting), then $\text{PH}_*(\mathbf{X})$ is completely described by a set of intervals, denoted by $\text{Dgm}(\text{PH}_*(\mathbf{X}))$. The presence of an interval $[t_b, t_d) \in \text{Dgm}(\text{PH}_p(\mathbf{X}))$ tells us that a particular p -dimensional homology class is born at time t_b , and lives until time t_d , where it then dies. If $t_d = \infty$, this means that this homology class lives forever, and corresponds to a global homology class in $H_p(\mathbf{X})$. The diagram $\text{Dgm}(\text{PH}_*(\mathbf{X}))$ can be conveniently visualized, see Figure 1.

2.1.1 Stability

An important property of a persistence module one needs to verify before using it in an application, is that it is stable with respect to the input data. Intuitively, this means that small perturbations of the input data should result only in small perturbations in the obtained persistence diagrams. We make this precise below.

► **Definition 2.** A matching between two multi-sets A and B is a bijection χ between two subsets $A' \subset A$ and $B' \subset B$. We denote this by $\chi : A \rightarrow B$. If χ matches $a \in A$ to $b \in B$, we write $(a, b) \in \chi$. If $c \in A \cup B$ is unmatched by χ , we abuse notation and write $c \notin \chi$.

► **Definition 3.** Let $I = \langle t_{b_1}, t_{d_1} \rangle$ and $J = \langle t_{b_2}, t_{d_2} \rangle$ be two intervals in $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. Then the cost of I is given by

$$c(I) := (t_{d_1} - t_{b_1})/2$$

and the cost of the pair (I, J) is given by:

$$c(I, J) := \max\{|t_{b_1} - t_{b_2}|, |t_{d_1} - t_{d_2}|\}$$

Now let \mathcal{D}_1 and \mathcal{D}_2 be two multi-sets of intervals in $\overline{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. The cost of a matching $\chi : \mathcal{D}_1 \rightarrow \mathcal{D}_2$ is defined as

$$\text{cost}(\chi) := \max\left\{ \sup_{(I, J) \in \chi} c(I, J), \sup_{I \notin \chi} c(I) \right\}$$

Finally, the Bottleneck distance between \mathcal{D}_1 and \mathcal{D}_2 is given by:

$$d_B(\mathcal{D}_1, \mathcal{D}_2) := \inf_{\chi : \mathcal{D}_1 \rightarrow \mathcal{D}_2} \text{cost}(\chi)$$

The Bottleneck distance is the most widely used distance on the space of persistence diagrams, and it satisfies the following:

► **Theorem 4** ([6]). Suppose \mathbf{X} is a CW-complex, and $f, g : \mathbf{X} \rightarrow \mathbb{R}$ are two continuous functions on \mathbf{X} . Then

$$d_B(\text{Dgm}(\text{PH}_p(\mathbf{X}, f)), \text{Dgm}(\text{PH}_p(\mathbf{X}, g))) \leq \|f - g\|_\infty := \max_{\mathbf{x} \in \mathbf{X}} |f(\mathbf{x}) - g(\mathbf{x})|.$$

► **Definition 5.** Let \mathbf{X} and \mathbf{Y} be two subsets of a metric space (M, d) . Write $U_\varepsilon(\mathbf{X}) := \{m \in M \mid \text{dist}(m, \mathbf{X}) \leq \varepsilon\}$. Then the Hausdorff distance between \mathbf{X} and \mathbf{Y} is given by:

$$\begin{aligned} d_H(\mathbf{X}, \mathbf{Y}) &:= \inf\{\varepsilon \geq 0 \mid \mathbf{Y} \subseteq U_\varepsilon(\mathbf{X}) \text{ and } \mathbf{X} \subseteq U_\varepsilon(\mathbf{Y})\} \\ &= \max\left\{\sup_{\mathbf{x} \in \mathbf{X}} \text{dist}(\mathbf{x}, \mathbf{Y}), \sup_{\mathbf{y} \in \mathbf{Y}} \text{dist}(\mathbf{y}, \mathbf{X})\right\}. \end{aligned}$$

► **Example 6.** Let $\mathbf{X} \subseteq M$, be a subset of a metric space (M, d) . Then the distance function

$$d_{\mathbf{X}} : M \rightarrow \mathbb{R}, \quad \mathbf{x} \mapsto \text{dist}(\mathbf{x}, \mathbf{X})$$

is a continuous function on M , and defines a sublevel set filtration and a persistence module, which we suggestively denote by $\text{PH}_p(\check{C}(\mathbf{X}))$. Note that if \mathbf{Y} is another subset of M , then $\|d_{\mathbf{X}} - d_{\mathbf{Y}}\|_\infty = d_H(\mathbf{X}, \mathbf{Y})$, so it follows from Theorem 4 that

$$d_B\left(\text{Dgm}(\text{PH}_p(\check{C}(\mathbf{X}))), \text{Dgm}(\text{PH}_p(\check{C}(\mathbf{Y})))\right) \leq d_H(\mathbf{X}, \mathbf{Y}).$$

In the above example, \mathbf{X} is typically a finite subset of \mathbb{R}^n , and this is often used as one of the motivating examples for persistent homology. When \mathbf{X} is sampled from some unknown shape \mathfrak{X} inside of \mathbb{R}^n , its persistent homology can be used to estimate the homology of \mathfrak{X} . It can be computed using the Čech filtration of \mathbf{X} , which is a filtered simplicial complex whose homotopy type at each stage agrees with that of the sublevel set of $d_{\mathbf{X}}$ at the same scale. It is true, but not entirely straight-forward, that the two persistence modules arising from these constructions are isomorphic [2, 5].

2.1.2 Wasserstein distance

In Section 3.1, we will show stability results for our novel persistence module in terms of the *Wasserstein distance*. The Wasserstein distance is a metric on the space of probability measures supported on \mathbb{R}^n . It is commonly used in the context of optimal transport and (statistical) data analysis, see, e.g., [16]. The primary advantage of the Wasserstein distance over the Hausdorff distance is that it is much less sensitive to outliers, and therefore more suited to applications in statistics. For our purposes, it is enough to consider probability measures with *finite support*.

► **Definition 7.** Let $\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}$ be two probability measures with finite supports $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$, respectively. The Wasserstein distance $d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$ is then given by the optimum solution to the linear program:

$$d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) := \min_{\gamma} \sum_{\mathbf{x} \in \mathcal{X}, \mathbf{y} \in \mathcal{Y}} \gamma(\mathbf{x}, \mathbf{y}) \cdot \|\mathbf{x} - \mathbf{y}\|_2 \quad (1)$$

$$\text{s.t.} \quad \sum_{\mathbf{y} \in \mathcal{Y}} \gamma(\mathbf{x}, \mathbf{y}) = \mu_{\mathcal{X}}(\mathbf{x}) \quad (2)$$

$$\sum_{\mathbf{x} \in \mathcal{X}} \gamma(\mathbf{x}, \mathbf{y}) = \mu_{\mathcal{Y}}(\mathbf{y}) \quad (3)$$

$$\gamma : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}_{\geq 0}. \quad (4)$$

One can think of $d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$ as the amount of “work” required to transform the measure $\mu_{\mathcal{X}}$ into $\mu_{\mathcal{Y}}$. For instance, if $\mathcal{Y} = \{\mathbf{y}\}$ is a singleton, then $d_W(\mu_{\mathcal{X}}, \mu_{\{\mathbf{y}\}})$ is simply given by:

$$d_W(\mu_{\mathcal{X}}, \mu_{\{\mathbf{y}\}}) = \sum_{\mathbf{x} \in \mathcal{X}} \mu_{\mathcal{X}}(\mathbf{x}) \cdot \|\mathbf{x} - \mathbf{y}\|_2.$$

2.2 The Christoffel-Darboux kernel

In this section, we introduce some basic facts on Christoffel-Darboux kernels, with emphasis on the statistical setting. We refer to the book of Lasserre, Pauwels and Putinar [13] for a comprehensive treatment. Let μ be a finite, positive Borel measure on \mathbb{R}^n with compact, full-dimensional support. (In our setting, it is helpful to think of μ as the restriction of the Lebesgue measure to a sufficiently nice compact subset of \mathbb{R}^n). Then μ induces an inner product on the space $\mathbb{R}[\mathbf{x}]$ of n -variate, real polynomials via:

$$\langle p, q \rangle_\mu := \int p(\mathbf{x})q(\mathbf{x})d\mu(\mathbf{x}). \tag{5}$$

We can choose an orthonormal basis $\mathbf{b} = \{b_\alpha : \alpha \in \mathbb{N}^n\}$ for $\mathbb{R}[\mathbf{x}]$ with respect to $\langle \cdot, \cdot \rangle_\mu$, which we order so that $b_\alpha \in \mathbb{R}[\mathbf{x}]$ is of total degree $|\alpha| = \sum_{i=1}^n \alpha_i$ for each $\alpha \in \mathbb{N}^n$. That is, we have the orthogonality relations:

$$\langle b_\alpha, b_\beta \rangle_\mu = \int b_\alpha(\mathbf{x})b_\beta(\mathbf{x})d\mu(\mathbf{x}) = \delta_{\alpha\beta} \quad (\alpha, \beta \in \mathbb{N}^n). \tag{6}$$

Using this orthonormal basis, we can define the Christoffel-Darboux kernel.

► **Definition 8.** For $d \in \mathbb{N}$, the Christoffel-Darboux kernel $K_d^\mu : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ of degree d for the measure μ is defined as:

$$K_d^\mu(\mathbf{x}, \mathbf{y}) := \sum_{|\alpha| \leq d} b_\alpha(\mathbf{x})b_\alpha(\mathbf{y}). \tag{7}$$

The Christoffel-Darboux kernel K_d^μ is also called the *reproducing kernel* for the Hilbert space $(\mathbb{R}[\mathbf{x}]_d, \langle \cdot, \cdot \rangle_\mu)$, as we have the reproducing property:

$$\int K_d^\mu(\mathbf{x}, \mathbf{y})p(\mathbf{y})d\mu(\mathbf{y}) = \langle K_d^\mu(\mathbf{x}, \cdot), p(\cdot) \rangle_\mu = p(\mathbf{x}) \quad (p \in \mathbb{R}[\mathbf{x}]_d). \tag{8}$$

We note that the kernel K_d^μ is independent of our choice of basis $\{b_\alpha\}$, and it can be computed via the Gram-Schmidt procedure even if we do not have access to an explicit orthonormal basis.

► **Proposition 9** (Gram-Schmidt, see Prop. 4.1.2 in [13]). Let $d \in \mathbb{N}$ and let $\mathbf{b} = \{b_\alpha : |\alpha| \leq d\}$ be any basis for $\mathbb{R}[\mathbf{x}]_d$. For $\mathbf{x} \in \mathbb{R}^n$, write $\mathbf{b}_d(\mathbf{x}) = (b_\alpha(\mathbf{x}))_{|\alpha| \leq d} \in \mathbb{R}^{s(n,d)}$ and consider the matrix $M_d^\mu(\mathbf{b}) \in \mathbb{R}^{s(n,d) \times s(n,d)}$ given by the entrywise integral:

$$M_d^\mu(\mathbf{b}) := \int \mathbf{b}_d(\mathbf{x})\mathbf{b}_d(\mathbf{x})^\top d\mu(\mathbf{x}),$$

i.e., $(M_d^\mu(\mathbf{b}))_{\alpha,\beta} = \int b_\alpha(\mathbf{x})b_\beta(\mathbf{x})d\mu(\mathbf{x}) \quad (\alpha, \beta \in \mathbb{N}_d^n). \tag{9}$

The matrix $M_d^\mu(\mathbf{b})$ is strictly positive semidefinite, i.e., its eigenvalues are all strictly larger than 0. Moreover, we have:

$$K_d^\mu(\mathbf{x}, \mathbf{y}) = \mathbf{b}_d(\mathbf{x})^\top (M_d^\mu(\mathbf{b}))^{-1} \mathbf{b}_d(\mathbf{y}).$$

The Christoffel polynomial. For our purposes, we are mostly interested in the *Christoffel polynomial* $P_d^\mu : \mathbb{R}^n \rightarrow \mathbb{R}$, defined in terms of an orthonormal basis \mathbf{b} for $(\mathbb{R}[\mathbf{x}]_d, \langle \cdot, \cdot \rangle_\mu)$ as:

$$P_d^\mu(\mathbf{x}) := K_d^\mu(\mathbf{x}, \mathbf{x}) = \sum_{|\alpha| \leq d} b_\alpha(\mathbf{x})^2. \tag{10}$$

The Christoffel polynomial is a *sum of squares* of polynomials, implying immediately that $P_d^\mu(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in \mathbb{R}^n$. In fact, by definiteness of the inner product (5), it is strictly positive on \mathbb{R}^n . It has a remarkable alternative definition in terms of a *variational problem*:

$$\frac{1}{P_d^\mu(\mathbf{z})} = \min_{p \in \mathbb{R}[\mathbf{x}]_d} \left\{ \int p^2(\mathbf{x}) d\mu(\mathbf{x}) : p(\mathbf{z}) = 1 \right\} \quad (\mathbf{z} \in \mathbb{R}^n).$$

The Christoffel polynomial encodes information on the support of μ . Roughly speaking, $P_d^\mu(\mathbf{x})$ is rather *large* when $\mathbf{x} \notin \text{supp}(\mu)$, and rather *small* when $\mathbf{x} \in \text{supp}(\mu)$ (see Figure 2). This can be made precise in the regime $d \rightarrow \infty$.

► **Proposition 10** (see Sec. 4.3 of [13]). *Under certain assumptions on the measure μ , we have:*

$$\lim_{d \rightarrow \infty} P_d^\mu(\mathbf{x}) = \begin{cases} O(d^n) & \mathbf{x} \in \text{int}(\text{supp}(\mu)), \\ \Omega(\exp(\alpha d)) & \mathbf{x} \notin \text{supp}(\mu), \end{cases}$$

for any fixed $\mathbf{x} \in \mathbb{R}^n$. Here, the constant α is proportional to $\text{dist}(\mathbf{x}, \text{supp}(\mu))$.

When μ is the restriction of the Lebesgue measure to a sufficiently nice compact subset $\mathfrak{X} \subseteq \mathbb{R}^n$, a stronger result is shown by Lasserre and Pauwels [12].

► **Theorem 11** (reformulation of Thm. 7.3.2 in [13]). *Let $\mathfrak{X} \subseteq \mathbb{R}^n$ be a compact set satisfying the conditions of Assumption 7.3.1 in [13], and let μ be the restriction of the Lebesgue measure to \mathfrak{X} . Then there exist sequences $(t_k)_{k \in \mathbb{N}}$ and $(d_k)_{k \in \mathbb{N}}$ such that the sublevel sets $\mathbf{X}_k := \{\mathbf{x} \in \mathbb{R}^n : P_{d_k}^\mu(\mathbf{x}) \leq t_k\}$ satisfy:*

$$\lim_{k \rightarrow \infty} d_H(\mathbf{X}_k, \mathfrak{X}) = 0, \quad \text{and} \quad \lim_{k \rightarrow \infty} d_H(\partial \mathbf{X}_k, \partial \mathfrak{X}) = 0.$$

Here, $\partial \mathfrak{X}$ and $\partial \mathbf{X}_k$ denote the boundary of \mathfrak{X} and \mathbf{X}_k , respectively.

2.2.1 The empirical setting

Assume now that we do not have explicit knowledge of the measure μ , but are instead given a sequence $\mathcal{X} \subseteq \mathbb{R}^n$ of N samples $X_1, X_2, \dots, X_N \in \mathbb{R}^n$, drawn independently according to μ . These samples induce a probability measure $\mu_{\mathcal{X}}$ given by $\mu_{\mathcal{X}} = \frac{1}{N} \sum_{i=1}^N \delta_{X_i}$, which we call the empirical measure associated to \mathcal{X} . Under a non-degeneracy assumption (Assumption 13 below), the measure $\mu_{\mathcal{X}}$ induces an inner product of the form (5) on $\mathbb{R}[\mathbf{x}]_d$ by:

$$\langle p, q \rangle_{\mu_{\mathcal{X}}} := \int p(\mathbf{x})q(\mathbf{x}) d\mu_{\mathcal{X}}(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N p(X_i)q(X_i). \quad (11)$$

In light of (11) and Proposition 9, it is straightforward to compute the Christoffel-Darboux kernel $K_d^{\mu_{\mathcal{X}}}$ of degree d for the measure $\mu_{\mathcal{X}}$ (and thus to compute $P_d^{\mu_{\mathcal{X}}}$). Indeed, the entries of the matrix $M_d^{\mu_{\mathcal{X}}}(\mathbf{b})$ in (9) may each be computed in time $O(N)$, after which $M_d^{\mu_{\mathcal{X}}}(\mathbf{b})$ can be inverted in time $O(s(n, d)^3)$.

► **Proposition 12.** *The empirical Christoffel-Darboux kernel $K_d^{\mu_{\mathcal{X}}}$ of degree d for N samples in \mathbb{R}^n may be computed in time $O(Ns(n, d)^2 + s(n, d)^3)$.*

The above procedure only works when the matrix $M_d^{\mu_{\mathcal{X}}}(\mathbf{b})$ is invertible, or equivalently, when the “inner product” $\langle \cdot, \cdot \rangle_{\mu_{\mathcal{X}}}$ on $\mathbb{R}[\mathbf{x}]_d$ is definite. We shall make this assumption throughout.

► **Assumption 13.** We say a sample set \mathcal{X} is non-degenerate (up to degree d) if the inner product $\langle \cdot, \cdot \rangle_{\mu_{\mathcal{X}}}$ associated to the empirical measure $\mu_{\mathcal{X}}$ induced by \mathcal{X} via (11) is definite for polynomials up to degree d .

Assumption 13 is satisfied if and only if the samples \mathcal{X} are not contained in an algebraic hypersurface of degree d (i.e., the zero set of a polynomial of degree at most d). This implies in particular that $N \geq s(n, d) + 1$.

Under certain assumptions on μ , Lasserre and Pauwels [12] show that the (empirical) Christoffel polynomial $P_d^{\mu_{\mathcal{X}}}$ converges to the (population) Christoffel polynomial P_d^{μ} as the number of samples $N \rightarrow \infty$. The rate of this convergence can be quantified, see [22].

► **Theorem 14** ([12], see also [13]). Let $\mathcal{X} = (X_1, X_2, \dots, X_N)$ be sampled from μ as in the above. Then for each $\mathbf{x} \in [-1, 1]^n$, we have $\lim_{N \rightarrow \infty} |P_d^{\mu}(\mathbf{x}) - P_d^{\mu_{\mathcal{X}}}(\mathbf{x})| = 0$ almost surely.

3 A persistence module based on the Christoffel polynomial

Theorems 11 and 14 motivate the use of the Christoffel polynomial in (statistical) data analysis. They show that certain sublevel sets of the empirical Christoffel polynomial approximate the support of the underlying population measure μ well (in Hausdorff distance) as the number of samples grows. However, Theorem 11 gives very little explicit information on which (sub)level set to consider. This is the primary motivation for considering a persistent scheme instead, which we introduce now.

► **Definition 15.** Fix $d \in \mathbb{N}$, and let $\mathcal{X} \subseteq [-1, 1]^n$ be a set of samples whose associated empirical measure $\mu_{\mathcal{X}}$ satisfies Assumption 13. Let $P_d^{\mu_{\mathcal{X}}} : \mathbb{R}^n \rightarrow \mathbb{R}$ be the corresponding Christoffel polynomial (10). For $t \geq 0$, we consider the compact sublevel set

$$\mathbf{X}_t := \{\mathbf{x} \in [-1, 1]^n : \log P_d^{\mu_{\mathcal{X}}}(\mathbf{x}) \leq t\} = \{\mathbf{x} \in [-1, 1]^n : P_d^{\mu_{\mathcal{X}}}(\mathbf{x}) \leq 10^t\}, \tag{12}$$

which is well-defined as $P_d^{\mu_{\mathcal{X}}}(\mathbf{x}) \geq 1$ for all $\mathbf{x} \in \mathbb{R}^n$. By definition $(\mathbf{X}_t)_{t \geq 0}$ is a filtration, i.e. $\mathbf{X}_t \subseteq \mathbf{X}_{t'}$ for any $t \leq t'$. In light of Example 1, we can therefore define the persistence module

$$\mathbb{C}\mathbb{D}(\mathcal{X}, d) := \text{PH}_*([-1, 1]^n, \log P_d^{\mu_{\mathcal{X}}}). \tag{13}$$

Notably, we do not consider the level sets of $P_d^{\mu_{\mathcal{X}}}$, but rather those of $\log P_d^{\mu_{\mathcal{X}}}$. Before we motivate this choice, let us first observe that from a computational perspective, this logarithmic rescaling makes no difference. Indeed, one can obtain the persistence module of the filtration $([-1, 1]^n, \log P_d^{\mu_{\mathcal{X}}})$ by first computing the module associated to $([-1, 1]^n, P_d^{\mu_{\mathcal{X}}})$ and then rescaling all interval modules. We choose to work with $\log P_d^{\mu_{\mathcal{X}}}$ for two reasons. First, as we will see below, this choice allows us to prove a stronger and more elegant stability result for the module $\mathbb{C}\mathbb{D}(\mathcal{X}, d)$. Second, the logarithmic scaling produces persistence diagrams that better fit the underlying topology in practice. Proposition 10 provides a rather convincing theoretical argument for this observation. Indeed, if $\mathbf{x} \in [-1, 1]^n$ is a point outside of the support of the underlying measure μ , it tells us that $P_d^{\mu}(\mathbf{x}) \approx \exp(\text{dist}(\mathbf{x}, \text{supp}(\mu)) \cdot d)$, which is to say that $\log P_d^{\mu}(\mathbf{x})$ scales linearly in the distance $\text{dist}(\mathbf{x}, \text{supp}(\mu))$, an intuitively desirable property.

3.1 Stability and robustness

In this section, we show that the module $\mathbb{C}\mathbb{D}(\mathcal{X}, d)$ is locally stable under small perturbations of the sample set \mathcal{X} , measured in the Wasserstein distance (1). Namely, we show in Proposition 19 that:

$$d_B(\text{Dgm}(\mathbb{C}\mathbb{D}(\mathcal{X}, d)), \text{Dgm}(\mathbb{C}\mathbb{D}(\mathcal{Y}, d))) \leq \log(C_{\mathcal{X}} \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) + 1)$$

38:10 The Christoffel-Darboux Kernel for Topological Data Analysis

for fixed $\mathcal{X} \subseteq [-1, 1]^n$ and any $\mathcal{Y} \subseteq [-1, 1]^n$ for which $d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$ is sufficiently small. Here, the constant $C_{\mathcal{X}}$ depends on n, d , and the supremum norm

$$\|\log P_d^{\mu_{\mathcal{X}}}\|_{\infty} := \max_{\mathbf{x} \in [-1, 1]^n} |\log P_d^{\mu_{\mathcal{X}}}(\mathbf{x})|,$$

which we interpret as a ‘‘measure of algebraic degeneracy’’ of the set \mathcal{X} (see Assumption 13). For *arbitrary* sets \mathcal{X} and \mathcal{Y} , we show in Proposition 18 that

$$d_B(\text{Dgm}(\mathbb{CD}(\mathcal{X}, d)), \text{Dgm}(\mathbb{CD}(\mathcal{Y}, d))) \leq \log(C_{\mathcal{X}, \mathcal{Y}} \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) + 1), \quad (14)$$

where the constant $C_{\mathcal{X}, \mathcal{Y}}$ now additionally depends on $\|\log P_d^{\mu_{\mathcal{Y}}}\|_{\infty}$. If one restricts to ‘‘sufficiently non-degenerate’’ sample sets (i.e., those having $\|\log P_d^{\mu_{\mathcal{X}}}\|_{\infty}$ bounded from above), relation (14) may be read as a *global* stability result.

For the proof of these statements, note first that in light of Corollary 4, we have:

$$d_B(\text{Dgm}(\mathbb{CD}(\mathcal{X}, d)), \text{Dgm}(\mathbb{CD}(\mathcal{Y}, d))) \leq \|\log P_d^{\mu_{\mathcal{X}}} - \log P_d^{\mu_{\mathcal{Y}}}\|_{\infty} \quad (15)$$

and so it suffices to consider the quantity $\|\log P_d^{\mu_{\mathcal{X}}} - \log P_d^{\mu_{\mathcal{Y}}}\|_{\infty}$. We start by showing the following.

► **Theorem 16.** *Let $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$ be as in the above. Write $C_{n,d} := 4 \cdot s(n, d) \cdot d^2$, where $s(n, d) := \dim \mathbb{R}[\mathbf{x}]_d = \binom{n+d}{d}$. Then for all $\mathbf{x} \in \mathbb{R}^n$, we have that:*

$$|P_d^{\mu_{\mathcal{X}}}(\mathbf{x}) - P_d^{\mu_{\mathcal{Y}}}(\mathbf{x})| \leq C_{n,d} \cdot \|P_d^{\mu_{\mathcal{X}}}\|_{\infty} \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \cdot P_d^{\mu_{\mathcal{Y}}}(\mathbf{x}).$$

We prove Theorem 16 in [17, Appendix A] by adapting the techniques of Section 6.2 in [13] to our setting. This requires some small technical statements on Wasserstein distance and supremum norms of polynomials on compact domains, which we also give there.

We have the following immediate consequence:

► **Corollary 17.** *Let \mathcal{X}, \mathcal{Y} as in the above, and let $C_{n,d} > 0$ be the constant of Theorem 16. Then for all $\mathbf{x} \in [-1, 1]^n$, we have:*

$$|P_d^{\mu_{\mathcal{X}}}(\mathbf{x})/P_d^{\mu_{\mathcal{Y}}}(\mathbf{x}) - 1| \leq C_{n,d} \cdot \|P_d^{\mu_{\mathcal{X}}}\|_{\infty} \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}).$$

After taking logarithms (see [17, Appendix A] for details), we then obtain:

► **Proposition 18.** *Let $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}^n$ be as above, and let $C_{n,d} > 0$ be the constant of Theorem 16. Then we have:*

$$\|\log P_d^{\mu_{\mathcal{X}}} - \log P_d^{\mu_{\mathcal{Y}}}\|_{\infty} \leq \log(C_{n,d} \cdot \max\{\|P_d^{\mu_{\mathcal{X}}}\|_{\infty}, \|P_d^{\mu_{\mathcal{Y}}}\|_{\infty}\} \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) + 1). \quad (16)$$

Note that when the quantity $C := C_{n,d} \cdot \max\{\|P_d^{\mu_{\mathcal{X}}}\|_{\infty}, \|P_d^{\mu_{\mathcal{Y}}}\|_{\infty}\}$ is close to 0, the bound (16) tells us that $\|\log P_d^{\mu_{\mathcal{X}}} - \log P_d^{\mu_{\mathcal{Y}}}\|_{\infty} \leq C \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}})$. On the other hand, if $C \gg 1$, the bound on $\|\log(P_d^{\mu_{\mathcal{X}}}) - \log(P_d^{\mu_{\mathcal{Y}}})\|_{\infty}$ is exponentially smaller than in Theorem 16. Finally, we may get rid of the dependence on \mathcal{Y} in Corollary 17 after making an additional assumption.

► **Proposition 19.** *Let $\mathcal{X} \subseteq [-1, 1]^n$ be a (fixed) sample set satisfying Assumption 13. Let $C_{n,d} > 0$ be the constant of Theorem 16, and assume that $\mathcal{Y} \subseteq [-1, 1]^n$ is such that $C_{n,d} \cdot \|P_d^{\mu_{\mathcal{X}}}\|_{\infty} \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) \leq 1/2$. Then we have $\|P_d^{\mu_{\mathcal{Y}}}\|_{\infty} \leq 2\|P_d^{\mu_{\mathcal{X}}}\|_{\infty}$. In particular, the bound (16) then reads:*

$$\|\log P_d^{\mu_{\mathcal{X}}} - \log P_d^{\mu_{\mathcal{Y}}}\|_{\infty} \leq \log(C_{n,d} \cdot 2\|P_d^{\mu_{\mathcal{X}}}\|_{\infty} \cdot d_W(\mu_{\mathcal{X}}, \mu_{\mathcal{Y}}) + 1).$$

3.2 An exact algorithm for the persistence module

As we explain in Section 2.2 (see Proposition 12), the Christoffel polynomial $P_d^{\mu, \mathbf{x}}$ of degree d may be computed in time $O(s(n, d)^3 + Ns(n, d)^2)$, where $N = |\mathcal{X}|$ is the number of samples and $s(n, d) = \binom{n+d}{d}$ is the size of the moment matrix (9). Once we have access to $P_d^{\mu, \mathbf{x}}$, it remains to compute the persistent homology of the filtration $\{\mathbf{x} \in [-1, 1]^n : P_d^{\mu, \mathbf{x}}(\mathbf{x}) \leq t\}_{t \geq 0}$. The set $[-1, 1]^n$ is a particularly simple example of a basic (closed) *semialgebraic set*. That is, a subset of \mathbb{R}^n defined by a finite number of polynomial (in)equalities; namely:

$$[-1, 1]^n = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) := 1 - \mathbf{x}_i^2 \geq 0 \text{ for } i = 1, 2, \dots, n\}.$$

We may therefore use the following recent result of Basu and Karisani [1].

► **Theorem 20** (Basu, Karisani (2022)). *For fixed $p \in \mathbb{N}$, there is an algorithm that takes as input a description of a closed and bounded semialgebraic set $\mathbf{X} = \{\mathbf{x} \in \mathbb{R}^n : g_i(\mathbf{x}) \geq 0, i = 1, 2, \dots, m\}$, and a polynomial $P \in \mathbb{R}[\mathbf{x}]$, and outputs the persistence diagram associated to the filtration $\{\mathbf{x} \in \mathbf{X} : P(\mathbf{x}) \leq t\}_{t \geq 0}$ up to dimension p . The complexity of this algorithm is bounded by $(md)^{O(n)}$, where $d = \max\{\deg(g_i), \deg(P)\}$ is the largest degree amongst P and the polynomial inequalities defining \mathbf{X} .*

► **Corollary 21** (Exact algorithm). *For fixed $p \in \mathbb{N}$, there is an exact algorithm that computes the persistence diagram associated to $\mathbb{C}\mathbb{D}(\mathcal{X}, d)$ up to dimension p . Its runtime is bounded by $O(s(n, d)^3 + Ns(n, d)^2) + (nd)^{O(n)}$.*

3.3 An effective approximation scheme

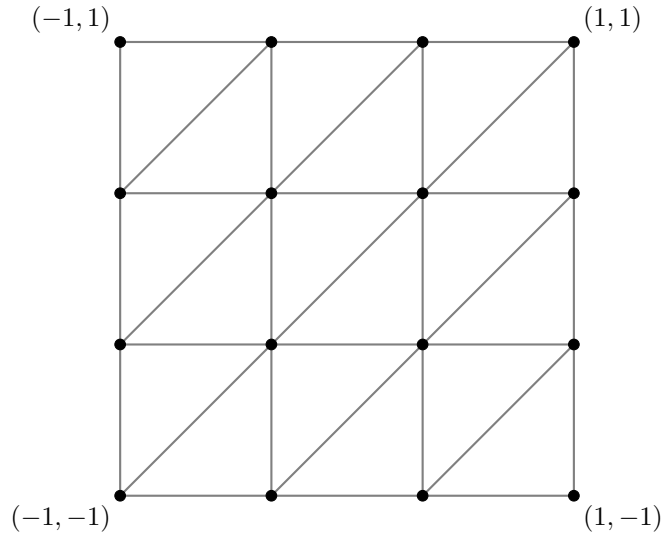
To the authors' knowledge, no implementation exists of the algorithm mentioned in Theorem 20 at the time of writing. In order to perform numerical experiments, we use a simple approximation scheme for $\mathbb{C}\mathbb{D}(\mathcal{X}, d)$. This method works in the more general case of approximating $\text{PH}_*([-1, 1]^n, f)$ for any Lipschitz continuous function $f : [-1, 1]^n \rightarrow \mathbb{R}$. Succinctly put, we first fix $m \in \mathbb{N}$, and construct the Freudenthal triangulation [10, 7] of $[-1, 1]^n$, with vertices equal to the lattice points of $\frac{2}{m} \cdot \mathbb{Z}^n$ contained in $[-1, 1]^n$. See Figure 3. We denote this triangulation by \mathcal{K}_m . Note that the diameter of any simplex in this triangulation is equal to $2\sqrt{n}/m$. We then evaluate f on each of the vertices, and compute the persistent homology of the lower-star filtration on \mathcal{K}_m induced by these function values. This persistence module, denoted by $\text{PH}_*(\mathcal{K}_m, f)$, approximates $\text{PH}_*([-1, 1]^n, f)$. The diagram of this module can be computed in polynomial time in the number of lattice points, which in our case is $(m + 1)^n$. The following proposition gives a guarantee on the quality of this approximation:

► **Proposition 22.** *Let $f : [-1, 1]^n \rightarrow \mathbb{R}$ be a Lipschitz continuous function with Lipschitz constant L_f , choose $m \in \mathbb{N}$, and let \mathcal{K}_m be as above. Then*

$$d_B(\text{Dgm}(\text{PH}_*([-1, 1]^n, f)), \text{Dgm}(\text{PH}_*(\mathcal{K}_m, f))) \leq L_f \cdot 2\sqrt{n}/m.$$

Since the function $\log P_d^{\mu, \mathbf{x}} : [-1, 1]^n \rightarrow \mathbb{R}$ is differentiable and $[-1, 1]^n$ is compact, $\log P_d^{\mu, \mathbf{x}}$ is Lipschitz continuous, and so the above proposition applies to our setting. For a proof and a more detailed discussion on this approximation scheme, we refer to [17, Appendix B].

► **Remark 23.** The stability results (14), (15) apply directly to $\text{PH}_*(\mathcal{K}_m, P_d^{\mu, \mathbf{x}})$. That is, the approximated modules are also stable in the Wasserstein distance, see [17, Appendix B].



■ **Figure 3** The Freudenthal triangulation of $[-1, 1]^2$ on the lattice $\frac{2}{m} \cdot \mathbb{Z}^2$ (with $m = 3$).

4 Numerical examples

In this section we illustrate our new scheme by computing the persistence diagram of $\mathbb{CD}(\mathcal{X}, d)$ for various toy examples $\mathcal{X} \subseteq [-1, 1]^n$ for $n = 1, 2, 3$. In each case, the sets \mathcal{X} are drawn from a measure supported $[-1, 1]^n$, after which some additional noise may be added. We compute the Christoffel polynomials using the method outlined in Section 2.2.1 and the linear algebra packages of NumPy [11] and Scipy [21]. To *approximate* the persistence of the resulting filtrations, we employ the method of Section 3.3, where we set the resolution $m = 250$ for $n = 1, 2$ and $m = 50$ for $n = 3$. See [17, Appendix B]. The persistence of the resulting lower-star filtration is computed using Gudhi [20]. We also use Gudhi to compute the (exact) persistent homology of the Čech filtration.

We add noise to our data sets \mathcal{X} in two ways. We say we add *uniform noise* when the noisy data $\tilde{\mathcal{X}}$ is obtained from \mathcal{X} by adding M points chosen uniformly at random from $[-1, 1]^n$. We say we add *Gaussian noise* (with standard deviation $\sigma \geq 0$), when $\tilde{\mathcal{X}}$ is obtained from \mathcal{X} by adding to each coordinate of each sample $\mathbf{x} \in \mathcal{X}$ an independently drawn perturbation $t \sim N(0, \sigma)$.

A univariate example. It is rather instructive to consider first a simple univariate example. Let $\mathcal{X} \subseteq [-1, 1]$ be drawn from a uniform measure μ supported on five disjoint intervals $I_1, I_2, \dots, I_5 \subseteq [-1, 1]$. The corresponding Christoffel polynomials ($d = 4, 8, 12$) and persistence diagrams for this situation are plotted in Figure 4. We would expect the persistence diagram of $\mathbb{CD}(\mathcal{X}, d)$ to reflect the simple topology of $\text{supp}(\mu)$; namely we expect $\mathbb{CD}(\mathcal{X}, d)$ to consist of five interval modules, each corresponding to one of the connected components of $\text{supp}(\mu)$. For $d = 8, 12$, we observe that this is indeed the case. For $d = 4$, however, we see there are only four interval modules. In the one-dimensional setting, this can be explained rather nicely. Indeed, “birth-events” correspond to (local) *minima* of $P_d^{\mu, \mathcal{X}}$, and “death-events” correspond to (local) *maxima*. As $P_4^{\mu, \mathcal{X}}$ is of degree 8, it can have at most 7 critical points, resulting in four interval modules (one of which is of infinite length).

The figure eight. In Figure 5, we plot $\log P_{12}^{\mu_{\mathcal{X}}}$ and $\mathbb{CD}(\mathcal{X}, 12)$ when \mathcal{X} is drawn from a measure supported on two circles in three different configurations: disjoint, just intersecting, and overlapping. We observe that $\mathbb{CD}(\mathcal{X}, 12)$ correctly captures the underlying topology in all three cases (for clarity, the number of overlapping points in the diagram is indicated).

Feature sizing. We consider the influence of feature size in the data on our persistence module. We draw samples \mathcal{X} from a configuration of two circles in $[-1, 1]^n$. In Figure 6, we depict the diagram of $\mathbb{CD}(\mathcal{X}, 10)$ as we decrease the *radius* of one of the circles. Similarly, in Figure 7, we depict the diagram of $\mathbb{CD}(\mathcal{X}, 10)$ as we decrease the *number of samples* drawn from one of the circles. In both cases, the decrease in feature size corresponds to a decrease in the length of the corresponding interval in $\mathbb{CD}(\mathcal{X}, 10)$. Interestingly, this is due to an earlier time of death in the former case, and due to a later time of birth in the latter.

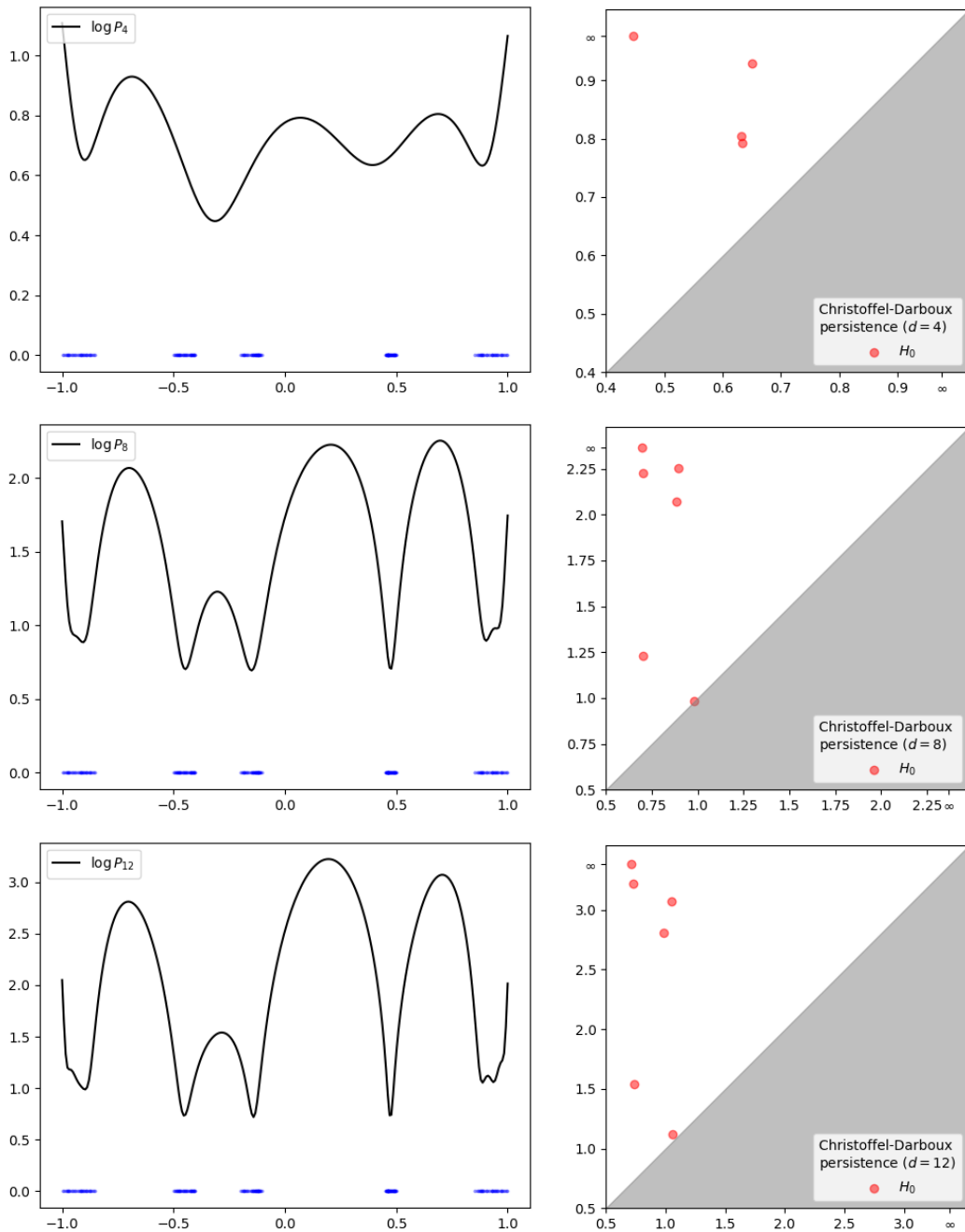
Comparison to the Čech module. We compare our new module $\mathbb{CD}(\mathcal{X}, 12)$ to Čech persistence in Figure 8, where \mathcal{X} is drawn from a measure supported on a ball, a triangle, and a square ($N = 10000$). We consider four cases: a pure sample \mathcal{X} , two samples with uniform noise ($M = 50$ and $M = 2500$) in $[-1, 1]^n$, and a sample with Gaussian noise ($\sigma = 0.03$). We observe that both the Christoffel-Darboux and Čech module are able to correctly capture the underlying topology for the pure sample and for the sample with Gaussian noise. However, only the Christoffel-Darboux module is able to do so for the samples with uniform noise, whereas the Čech module recovers no meaningful information there.

Stability under uniform noise. We consider a set \mathcal{X} consisting of evenly spaced points on the 1-skeleton of a cube¹ in \mathbb{R}^3 (50 points per edge) with edge-length 1.5. The significant persistent features of \mathcal{X} consist of a single interval (of infinite length) in H_0 ; five intervals in H_1 and one interval in H_2 , see Figure 9. We investigate how well the features in H_1 can be recovered after adding an increasing amount of uniform noise to \mathcal{X} , comparing Čech persistence to $\mathbb{CD}(\mathcal{X}, d)$, $d = 6, 8, 10$. To measure this, we follow [3] and use the *signal-to-noise ratio*; meaning the ratio between the size of the smallest interval in H_1 inherent to \mathcal{X} and the size of the largest interval (in H_1) induced by the noise. In Table 1, we report the median signal-to-noise ratios over 100 experiments. We reiterate that the Christoffel-Darboux persistence is computed *approximately*, which could affect these results. See [17, Appendix B] for a more detailed discussion. We observe that the ratios for the Christoffel-Darboux modules are much better than those for the Čech module. Furthermore, we note that the module of degree $d = 6$ outperforms the modules of degree $d = 8$ and $d = 10$. This is consistent with our stability results in Section 3.1, which are stronger for small values of d .

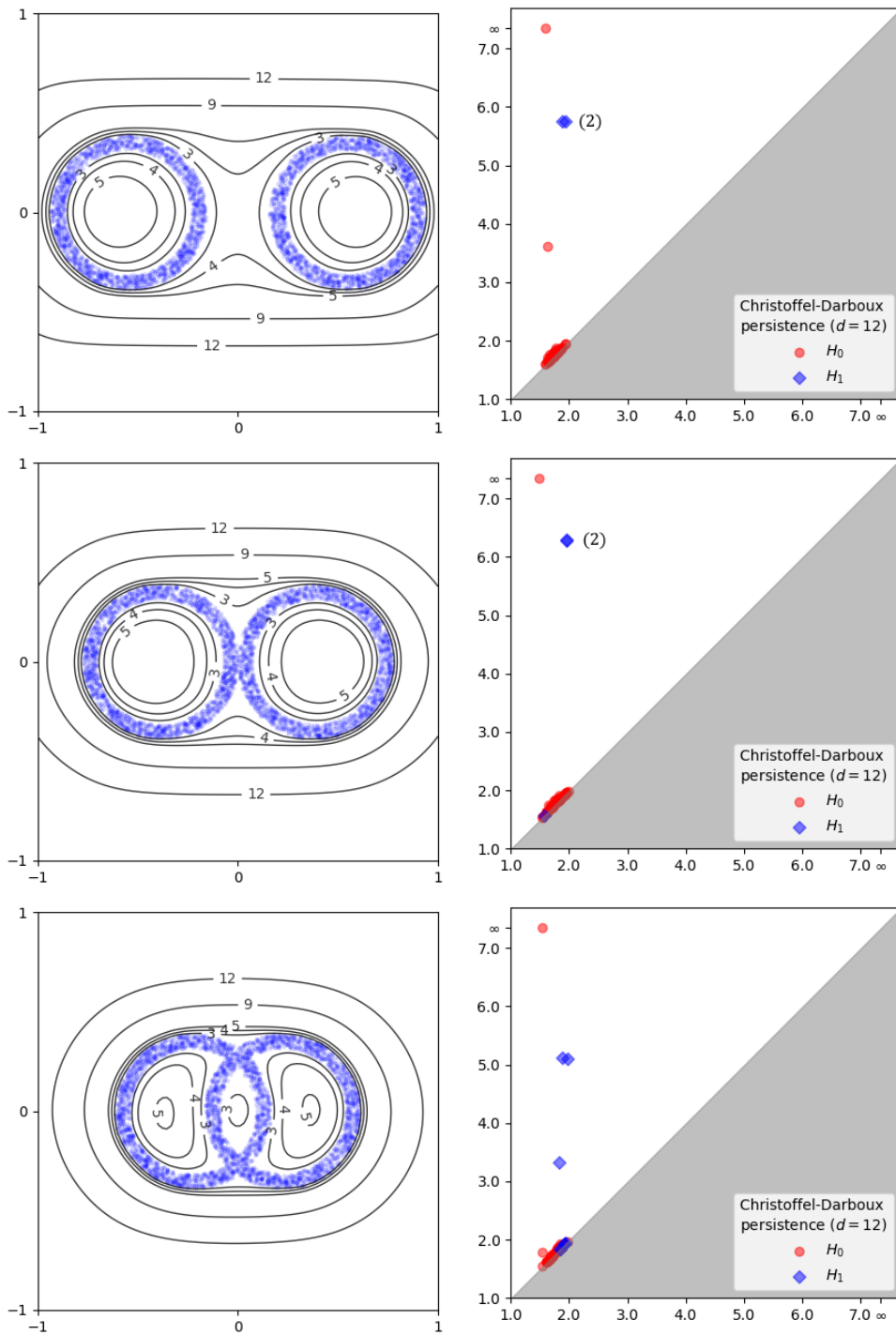
■ **Table 1** Signal-to-noise ratios for persistent homology in dimension 1 for the cube skeleton in the presence of uniform noise (median values over 100 experiments). See also Figure 9.

uniform noise (M)	baseline	25	50	100	250	500	1000
Čech	$\gg 10$	3.6	2.4	2.0	1.6	1.4	1.3
Christoffel-Darboux ($d = 6$)	$\gg 10$	7.8	5.7	5.3	3.7	2.3	1.2
Christoffel-Darboux ($d = 8$)	$\gg 10$	4.9	2.8	2.6	2.5	2.1	1.2
Christoffel-Darboux ($d = 10$)	$\gg 10$	8.8	4.0	2.3	1.9	1.8	1.3

¹ Because the cube skeleton is degenerate, we add a small amount of Gaussian noise ($\sigma = 0.025$) to \mathcal{X} to ensure the Christoffel polynomial for $\mu_{\mathcal{X}}$ is well-defined.

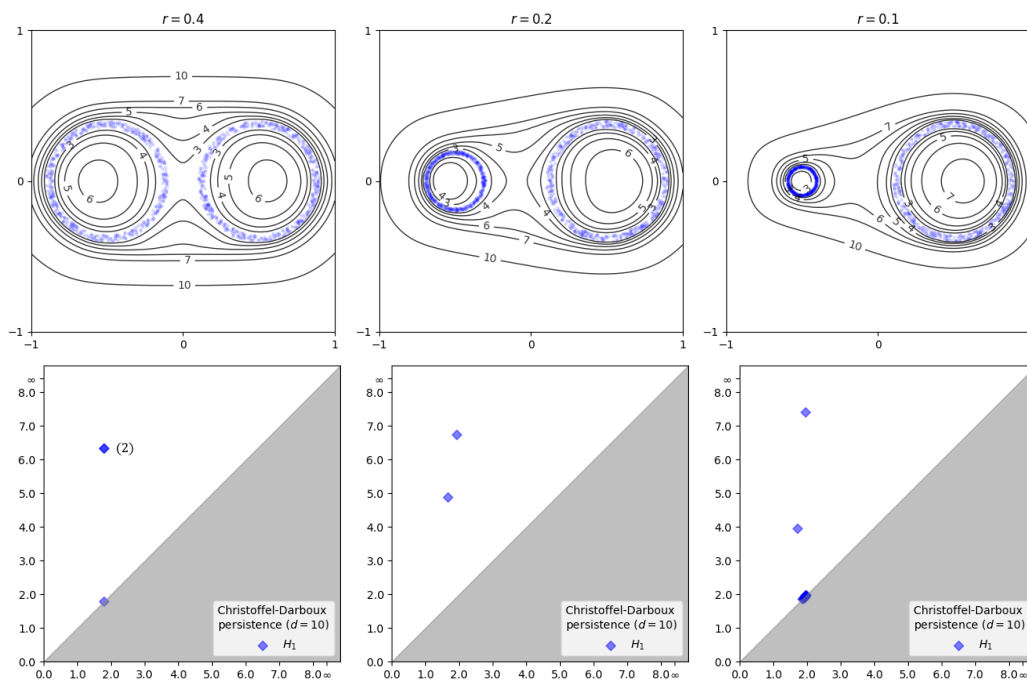


■ **Figure 4** Left: Christoffel polynomials ($d = 4, 8, 16$) for samples $\mathcal{X} \subseteq [-1, 1]$ drawn from a measure supported on five intervals in $[-1, 1]$ (in blue, $N = 500$). Right: diagrams of $\mathbb{CD}(\mathcal{X}, d)$.

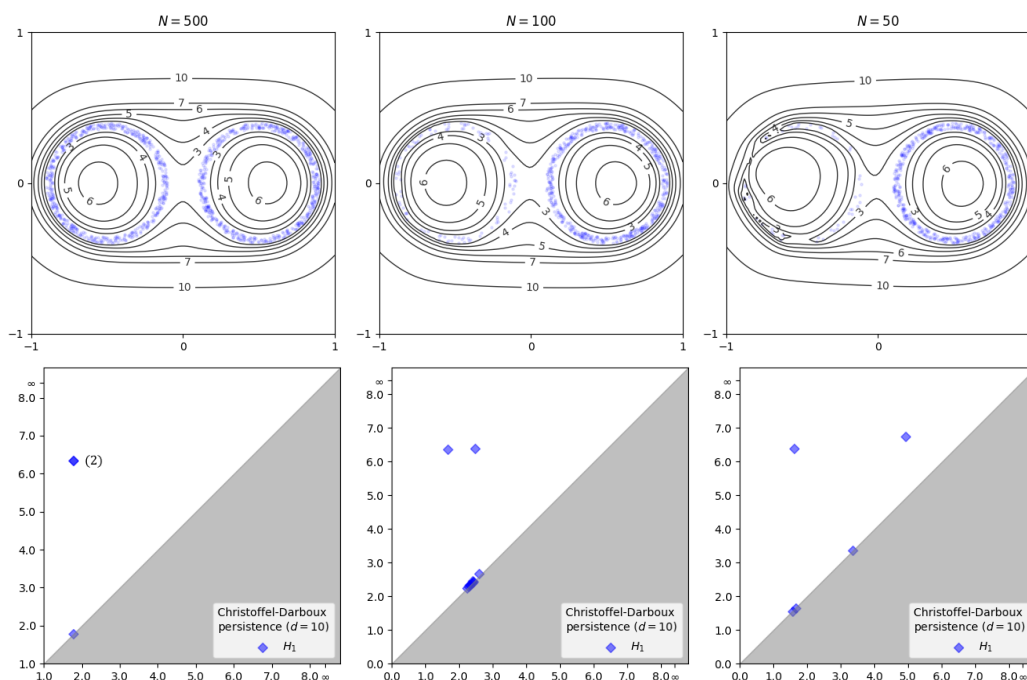


■ **Figure 5** Left: level sets (in black) of the Christoffel polynomial $P_{12}^{\mu, \mathcal{X}}$ of degree 12 for three sets of samples $\mathcal{X} \subseteq [-1, 1]^m$ (in blue, $N = 3000$). Right: the corresponding diagrams of $\mathbb{C}\mathbb{D}(\mathcal{X}, 12)$.

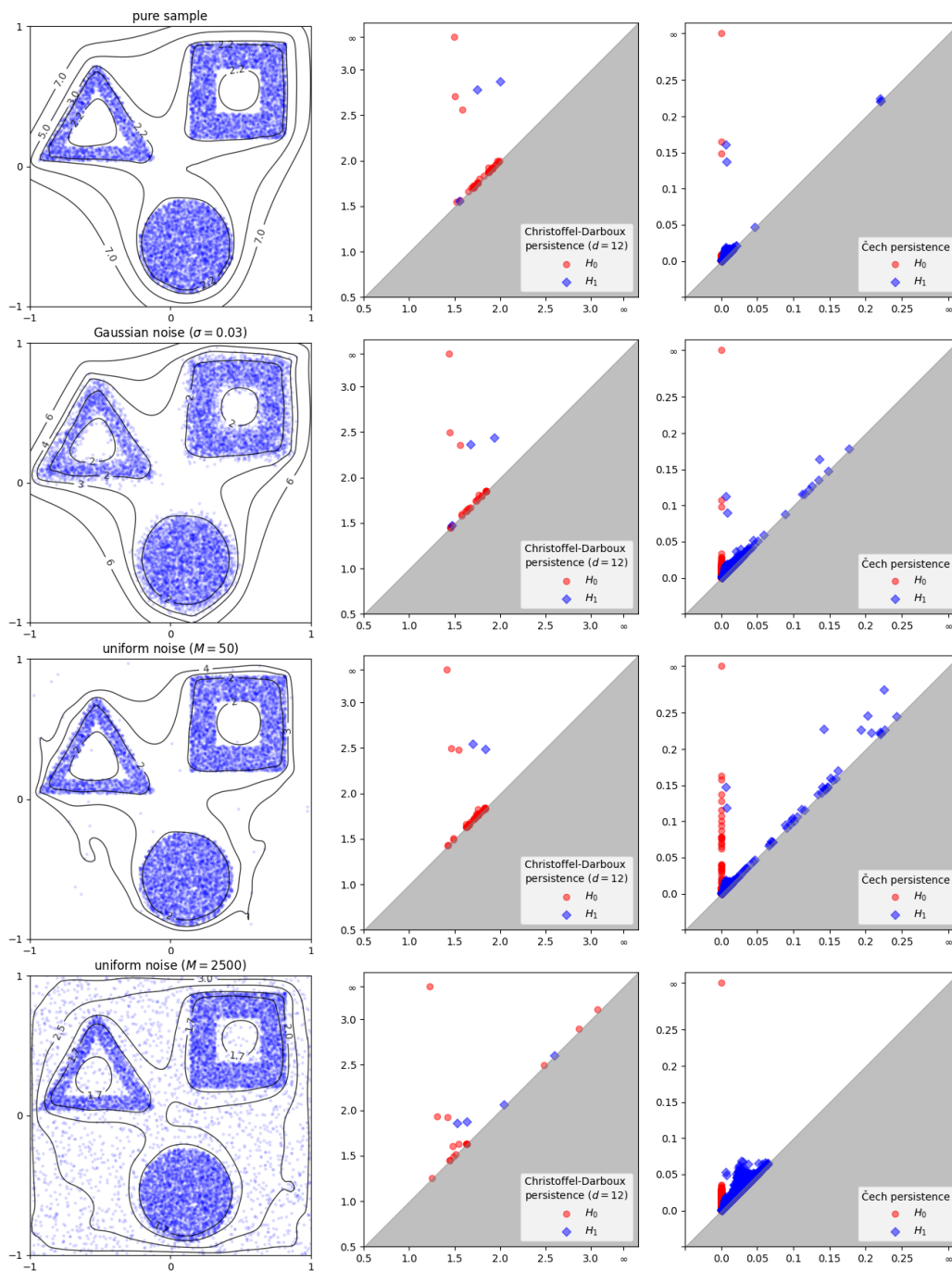
38:16 The Christoffel-Darboux Kernel for Topological Data Analysis



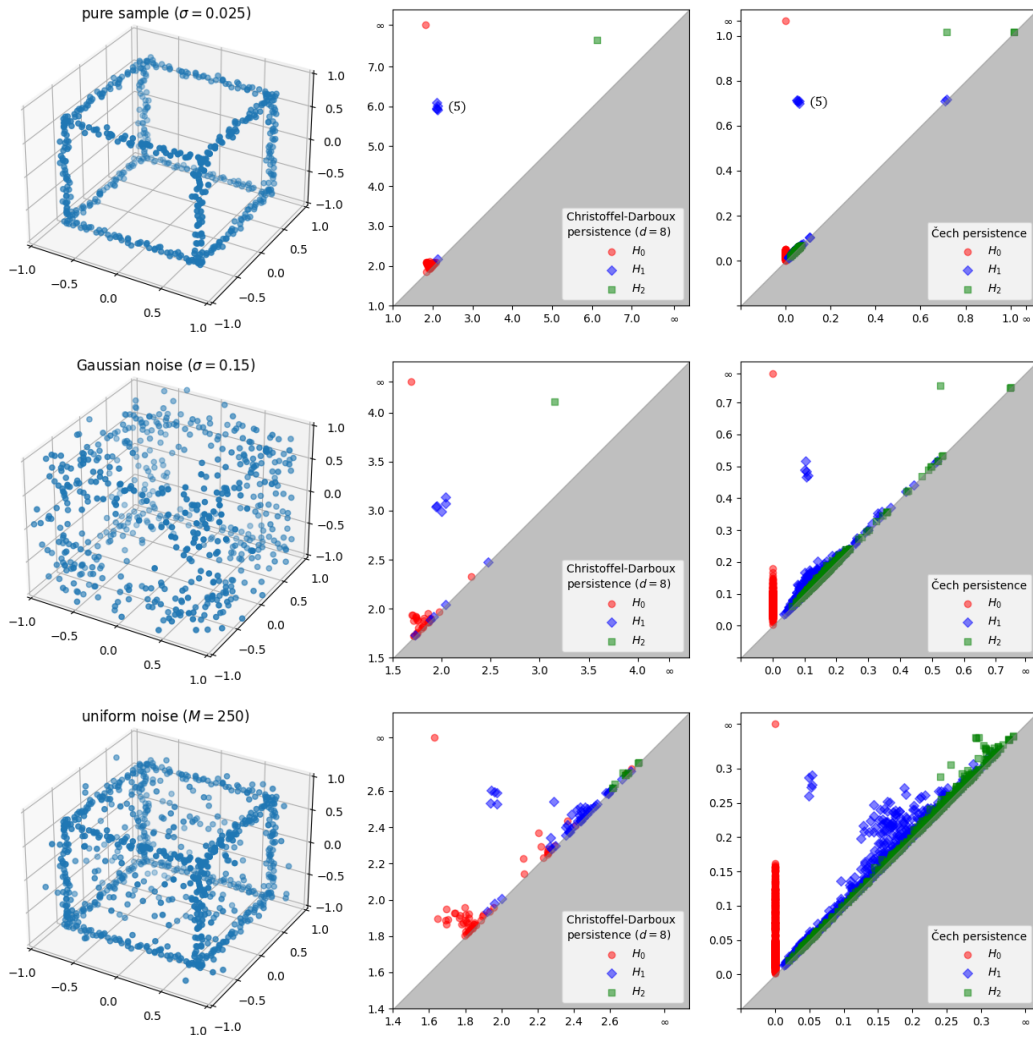
■ **Figure 6** Level sets of $\log P_{10}^{\mu_X}$ and diagrams of $\mathbb{CD}(\mathcal{X}, 10)$ for samples \mathcal{X} drawn from a measure supported on two circles. The radius r of the left circle decreases. Only degree 1 homology is shown.



■ **Figure 7** Level sets of $\log P_{10}^{\mu_X}$ and diagrams of $\mathbb{CD}(\mathcal{X}, 10)$ for samples \mathcal{X} drawn from a measure supported on two circles. The number of samples N drawn from the left circle decreases. Only degree 1 homology is shown.



■ **Figure 8** Left: level sets of $\log P_{12}^{\mu_{\mathcal{X}}}$ for sample sets \mathcal{X} in $[-1, 1]^2$ with different types of noise. Center: persistence diagrams of $\mathbb{CD}(\mathcal{X}, 12)$. Right: persistence diagrams for the Čech filtration.



■ **Figure 9** Left: sample sets obtained from the 1-skeleton of a cube in $[-1, 1]^3$ by adding Gaussian and uniform noise, respectively. Right: the corresponding Čech and Christoffel-Darboux persistence.

5 Discussion

We have introduced a new scheme for computing persistent homology of a point cloud in \mathbb{R}^n , based on the theory of Christoffel-Darboux kernels. Our scheme is stable w.r.t. the Wasserstein distance. It admits an exact algorithm whose runtime is linear in the number of samples, but depends rather heavily on the ambient dimension n and the degree d of the kernel. In several examples ($n = 1, 2, 3$), it was able to capture key topological features of the point cloud, even in the presence of uniform noise.

Computing the persistent homology. The persistence module $\mathbb{CD}(\mathcal{X}, d)$ arises from a particularly simple filtration of a semialgebraic set by a polynomial. This is what allows us to invoke the result of Basu & Karisani in Section 3.2 to compute its persistence diagram. Their result in fact applies in a much more general setting, but no practical implementation of the resulting algorithm exists. Our present work thus motivates the search for *effective*

exact algorithms in simple special cases. We instead rely in this work on the approximation scheme described in Section 3.3. We are only able to bound the error of this scheme in terms of the Lipschitz constant of $P_d^{\mu_{\mathcal{X}}}$ (which may be large, see also below). The practical performance of our scheme appears to be much better than this bound would suggest. It would therefore be desirable to prove further theoretical results that back this up.

Regularization. Our stability results of Section 3.1 depend on the “algebraic degeneracy” of the sample set \mathcal{X} . Such dependence is undesirable, and not present in stability results for most conventional persistence modules. This dependence can potentially be avoided by considering a *regularization* of the Christoffel polynomial, obtained by adding a small multiple of the identity to the moment matrix (9): $M \leftarrow M + \varepsilon \cdot \text{Id}$. One can also think of this as adding a small multiple of the uniform measure on $[-1, 1]^n$ to the empirical measure $\mu_{\mathcal{X}}$, which ensures that the corresponding inner product is definite. In [14], the authors already studied the impact of such modifications in the setting of functional approximation. There, it allows them to work over (near-)degenerate sets \mathcal{X} , while still accurately capturing the geometry of their problem. Preliminary experiments show this approach can be applied in our setting as well, and it would be very interesting to explore this further.

Selecting the degree d . Another important consideration is the selection of the degree d of the Christoffel polynomial $P_d^{\mu_{\mathcal{X}}}$. On the one hand, Proposition 10 and Theorem 11 suggest that the polynomial captures the support of the underlying measure μ better when d is large. On the other hand, Proposition 18 and Table 1 suggest that $\mathbb{CD}(\mathcal{X}, d)$ is more stable under perturbations of the data \mathcal{X} for smaller d . Furthermore, computing $P_d^{\mu_{\mathcal{X}}}$ rapidly becomes more costly as d grows. It is a hard open question what the “optimal” choice of d is w.r.t. n .

References

- 1 Saugata Basu and Negin Karisani. Persistent homology of semi-algebraic sets, 2022. URL: <https://arxiv.org/abs/2202.09591>.
- 2 Ulrich Bauer, Michael Kerber, Fabian Roll, and Alexander Rolle. A unified view on the functorial nerve theorem and its variations, 2022. URL: <https://arxiv.org/abs/2203.03571>.
- 3 Mickaël Buchet, Frédéric Chazal, Steve Y. Oudot, and Donald R. Sheehy. Efficient and robust persistent homology for measures. *Computational Geometry*, 58:70–96, 2016.
- 4 Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11:733–751, 2011.
- 5 Frédéric Chazal and Steve Oudot. Towards persistence-based reconstruction in euclidean spaces. *Proceedings of the Annual Symposium on Computational Geometry*, 2008. doi:10.1145/1377676.1377719.
- 6 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37:103–120, 2007.
- 7 B.C. Eaves. *A Course in Triangulations for Solving Equations with Deformations*. Lecture notes in economics and mathematical systems. Springer-Verlag, 1984.
- 8 Herbert Edelsbrunner and John Harer. Persistent homology—a survey. *Discrete & Computational Geometry - DCG*, 453, 2008. doi:10.1090/conm/453/08802.
- 9 Herbert Edelsbrunner and Ernst Mücke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13, 1994. doi:10.1145/147130.147153.
- 10 Hans Freudenthal. Simplizialzerlegungen von beschränkter flachheit. *Annals of Mathematics*, 43:580, 1942.
- 11 Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert

- Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.
- 12 Jean-Bernard Lasserre and Edouard Pauwels. The empirical Christoffel function with applications in data analysis. *Advances in Computational Mathematics*, 45(3):1439–1468, 2019.
 - 13 Jean-Bernard Lasserre, Edouard Pauwels, and Mihai Putinar. *The Christoffel–Darboux Kernel for Data Analysis*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2022. doi:10.1017/9781108937078.
 - 14 Swann Marx, Edouard Pauwels, Tillmann Weisser, Didier Henrion, and Jean-Bernard Lasserre. Semi-algebraic approximation using Christoffel–Darboux kernel. *Constructive Approximation*, 54:391–429, 2021.
 - 15 Edouard Pauwels, Mihai Putinar, and Jean-Bernard Lasserre. Data analysis from empirical moments and the Christoffel function. *Foundations of Computational Mathematics*, 21(1):243–273, 2021. doi:10.1007/s10208-020-09451-2.
 - 16 Gabriel Peyré and Marco Cuturi. Computational optimal transport: With applications to data science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607, 2019.
 - 17 Pepijn Roos Hoefgeest and Lucas Slot. The Christoffel-Darboux kernel for topological data analysis (full version). URL: <https://arxiv.org/abs/2211.15489>.
 - 18 Vin Silva and Gunnar Carlsson. Topological estimation using witness complexes. *Proc. Sympos. Point-Based Graphics*, 2004. doi:10.2312/SPBG/SPBG04/157-166.
 - 19 Bernadette J. Stolz. Outlier-robust subsampling techniques for persistent homology, 2021. URL: <https://arxiv.org/abs/2103.14743>.
 - 20 The GUDHI Project. *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015. URL: <http://gudhi.gforge.inria.fr/doc/latest/>.
 - 21 Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
 - 22 Mai Trang Vu, François Bachoc, and Edouard Pauwels. Rate of convergence for geometric inference based on the empirical Christoffel function. *ESAIM: PS*, 26:171–207, 2022.
 - 23 Afra Zomorodian and Gunnar Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33:249–274, 2005.

The Number of Edges in Maximal 2-Planar Graphs

Michael Hoffmann   

Department of Computer Science, ETH Zürich, Switzerland

Meghana M. Reddy   

Department of Computer Science, ETH Zürich, Switzerland

Abstract

A graph is *2-planar* if it has local crossing number two, that is, it can be drawn in the plane such that every edge has at most two crossings. A graph is *maximal 2-planar* if no edge can be added such that the resulting graph remains 2-planar. A 2-planar graph on n vertices has at most $5n - 10$ edges, and some (maximal) 2-planar graphs – referred to as *optimal 2-planar* – achieve this bound. However, in strong contrast to maximal planar graphs, a maximal 2-planar graph may have fewer than the maximum possible number of edges. In this paper, we determine the minimum edge density of maximal 2-planar graphs by proving that every maximal 2-planar graph on $n \geq 5$ vertices has at least $2n$ edges. We also show that this bound is tight, up to an additive constant. The lower bound is based on an analysis of the degree distribution in specific classes of drawings of the graph. The upper bound construction is verified by carefully exploring the space of admissible drawings using computer support.

2012 ACM Subject Classification Mathematics of computing → Combinatorics; Mathematics of computing → Graph theory; Human-centered computing → Graph drawings

Keywords and phrases k -planar graphs, local crossing number, saturated graphs, beyond-planar graphs

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.39

Related Version *Full Version*: <https://arxiv.org/abs/2303.08726v1> [14]

Supplementary Material *Software (Source Code)*: <https://github.com/meghanamreddy/Sparse-maximal-2-planar-graphs> [13]

archived at `swh:1:dir:a23d5245480bf9944c35362d4abc755c6148219f`

Funding Supported by the Swiss National Science Foundation within the collaborative DACH project *Arrangements and Drawings* as SNSF Project 200021E-171681.

1 Introduction

Maximal planar graphs a.k.a. (combinatorial) triangulations are a rather important and well-studied class of graphs with a number of nice and useful properties. To begin with, the number of edges is uniquely determined by the number of vertices, as every maximal planar graph on $n \geq 3$ vertices has $3n - 6$ edges. It is natural to wonder if a similar statement can be made for the various families of near-planar graphs, which have received considerable attention over the past decade; see, e.g. [11, 15].

In this paper we focus on k -planar graphs, specifically for $k = 2$. These are graphs with local crossing number at most k , that is, they admit a drawing in \mathbb{R}^2 where every edge has at most k crossings. The class of 1-planar graphs was introduced by Ringel [21] in the context of vertex-face colorings of planar graphs. Later, Pach and Tóth [20] used upper bounds on the number of edges in k -planar graphs to derive an improved version of the Crossing Lemma, which gives a lower bound on the crossing number of a simple (no loops or multi-edges) graph in terms of its number of vertices and edges. The class of k -planar graphs is not closed under edge contractions and already for $k = 1$ there are infinitely many minimal non-1-planar graphs, as shown by Korzhik [17].



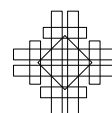
© Michael Hoffmann and Meghana M. Reddy;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 39; pp. 39:1–39:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



The maximum number of edges in a k -planar graph on n vertices increases with k , but the exact dependency is not known. A general upper bound of $O(\sqrt{kn})$ is known due to Ackerman and Pach and Tóth [1, 20] for graphs that admit a *simple* k -plane drawing, that is, a drawing where every pair of edges has at most one common point. Only for small k we have tight bounds. A 1-planar graph on n vertices has at most $4n - 8$ edges and there are infinitely many *optimal* 1-planar graphs that achieve this bound, as shown by Bodendiek, Schumacher, and Wagner [7]. A 2-planar graph on n vertices has at most $5n - 10$ edges and there are infinitely many *optimal* 2-planar graphs that achieve this bound, as shown by Pach and Tóth [20]. In fact, there are complete characterizations, for optimal 1-planar graphs by Suzuki [23] and for optimal 2-planar graphs by Bekos, Kaufmann, and Raftopoulou [6].

Much less is known about *maximal* k -planar graphs, that is, graphs for which adding any edge results in a graph that is not k -planar anymore. In contrast to planar graphs, where maximal and optimal coincide, it is easy to find examples of maximal k -planar graphs that are not optimal; a trivial example is the complete graph K_5 . In fact, the difference between maximal and optimal can be quite large for k -planar graphs, even – perhaps counterintuitively – maximal k -planar graphs for $k \geq 1$ may have fewer edges than maximal planar graphs on the same number of vertices. Hudák, Madaras, and Suzuki [16] describe an infinite family of maximal 1-planar graphs with only $8n/3 + O(1) \approx 2.667n$ edges. An improved construction with $45n/17 + O(1) \approx 2.647n$ edges was given by Brandenburg, Eppstein, Gleißner, Goodrich, Hanauer, and Reislhuber [8] who also established a lower bound by showing that every maximal 1-planar graph has at least $28n/13 - O(1) \approx 2.153n$ edges. Later, this lower bound was improved to $20n/9 \approx 2.22n$ by Barát and Tóth [4].

Maximal 2-planar graphs were studied by Auer, Brandenburg, Gleißner, and Hanauer [3] who constructed an infinite family of maximal 2-planar graphs with n vertices and $387n/147 + O(1) \approx 2.63n$ edges.¹ We are not aware of any nontrivial lower bounds on the number of edges in maximal k -planar graphs, for $k \geq 2$.

Results. In this paper, we give tight bounds on the minimum number of edges in maximal 2-planar graphs, up to an additive constant.

► **Theorem 1.** *Every maximal 2-planar graph on $n \geq 5$ vertices has at least $2n$ edges.*

► **Theorem 2.** *There exists a constant $c \in \mathbb{N}$ such that for every $n \in \mathbb{N}$ there exists a maximal 2-planar graph on n vertices with at most $2n + c$ edges.*

Related work. Maximality has also been studied for drawings of simple graphs. Let \mathcal{D} be a class of drawings. A drawing $D \in \mathcal{D}$ is *saturated* if no edge can be added to D so that the resulting drawing is still in \mathcal{D} . For the class of simple drawings, Kynčl, Pach, Radoičić and Tóth [18] showed that every saturated drawing on n vertices has at least $1.5n$ edges and there exist saturated drawings with no more than $17.5n$ edges. The upper bound was improved to $7n$ by Hajnal, Igamberdiev, Rote and Schulz [12]. Chaplick, Klute, Parada, Rollin, and Ueckerdt [9] studied saturated k -plane drawings, for $k \geq 4$, and obtained tight bounds linear in n , where the constant depends on k , for various types of crossing restrictions. For the class of 1-plane drawings, Brandenburg, Eppstein, Gleißner, Goodrich, Hanauer, and Reislhuber [8]

¹ Maximality is proven via uniqueness of the 2-plane drawing of the graph. However, there is no explicit proof of the uniqueness in this short abstract.

showed that there exist saturated drawings with no more than $7n/3 + O(1) \approx 2.33n$ edges. On the lower bound side, the abovementioned bound of $20n/9 \approx 2.22n$ edges by Barát and Tóth [4] actually holds for saturated 1-plane drawings. For the class of 2-plane drawings, Auer, Brandenburg, Gleißner, and Hanauer [3] describe saturated drawings with no more than $4n/3 + O(1) \approx 1.33n$ edges, and Barát and Tóth [5] show that every saturated 2-plane drawing on n vertices has at least $n - 1$ edges.

Although the general spirit is similar, saturated drawings are quite different from maximal abstract graphs. To obtain a sparse saturated drawing, one can choose both the graph and the drawing, whereas for sparse maximal graphs one can choose the graph only and needs to get a handle on all possible drawings. Universal lower bounds for saturated drawings carry over to the maximal graph setting, and existential upper bounds for maximal graphs carry over to saturated drawings. But bounds obtained in this fashion are far from tight usually; compare, for instance, the range of between n and $4n/3$ edges for saturated 2-plane drawings to our bound of $2n$ edges for maximal 2-planar graphs.

2 Preliminaries

A *drawing* of a graph $G = (V, E)$ is a map $\gamma : G \rightarrow \mathbb{R}^2$ that maps each vertex $v \in V$ to a point $\gamma(v) \in \mathbb{R}^2$ and each edge $uv \in E$ to a simple (injective) curve $\gamma(uv)$ with endpoints $\gamma(u)$ and $\gamma(v)$, subject to the following conditions: (1) γ is injective on V ; (2) for all $uv \in E$ we have $\gamma(uv) \cap \gamma(V) = \{\gamma(u), \gamma(v)\}$; and (3) for each pair $e_0, e_1 \in E$ with $e_0 \neq e_1$ the curves $\gamma(e_0)$ and $\gamma(e_1)$ have at most finitely many intersections, and each such intersection is either a common endpoint or a proper, transversal *crossing* (that is, no touching points between these curves). The connected components of $\mathbb{R}^2 \setminus \gamma(G)$ are the *faces* of γ . The *boundary* of a face f is denoted by ∂f .

To avoid notational clutter we will often identify vertices and edges with their geometric representations in a given drawing. A drawing is *simple* if every pair of edges has at most one common point. A drawing is *k-plane*, for $k \in \mathbb{N}$, if every edge has at most k crossings. A graph is *k-planar* if it admits a *k-plane* drawing. A graph is *maximal k-planar* if no edge can be added to it so that the resulting graph is still *k-planar*.

To analyze a *k-planar* graph one often analyzes one of its *k-plane* drawings. It is, therefore, useful to impose additional restrictions on this drawing if possible. One such restriction is to consider a *crossing-minimal k-plane* drawing, that is, a drawing that minimizes the total number of edge crossings among all *k-plane* drawings of the graph. For small k , such a drawing is always simple; for $k \geq 4$ this is not the case in general [22, Footnote 112].

► **Lemma 3** (Pach, Radoičić, Tardos, and Tóth [19, Lemma 1.1]). *For $k \leq 3$, every crossing-minimal k -plane drawing is simple.*

In figures, we use the following convention to depict edges: Uncrossed edges are shown green, singly crossed edges are shown purple, doubly crossed edges are shown blue, and edges for which the number of crossings is undetermined are shown black.

Connectivity. Next let us collect some basic properties of maximal *k-planar* graphs and their drawings. Some of these may be folklore, but for completeness we include the (simple) proofs in the full version [14].

► **Lemma 4.** *Let D be a crossing-minimal k -plane drawing of a maximal k -planar graph G , and let u and v be two vertices that lie on (the boundary of) a common face in D . Then uv is an edge of G and it is uncrossed in D .*

► **Lemma 5.** *Let D be a crossing-minimal k -plane drawing of a maximal k -planar graph on n vertices, for $k \leq 2 \leq n$. Then every vertex is incident to an uncrossed edge in D .*

► **Lemma 6.** *For $k \leq 2$, every maximal k -planar graph on $n \geq 3$ vertices is 2-connected.*

3 The Lower Bound

In this section we develop our lower bound on the edge density of maximal 2-planar graphs by analyzing the distribution of vertex degrees. As we aim for a lower bound of $2n$ edges, we want to show that the average vertex degree is at least four. Then, the density bound follows by the handshaking lemma. However, maximal 2-planar graphs may contain vertices of degree less than four. By Lemma 6 we know that the degree of every vertex is at least two. But degree-two vertices, so-called *hermits*, may exist, as well as vertices of degree three.

In order to lower bound the average degree by four, we employ a charging scheme where we argue that every *low-degree* vertex, that is, every vertex of degree two and three claims a certain number of halfedges at an adjacent *high-degree* vertex, that is, a vertex of degree at least five. Claims are exclusive, that is, every halfedge at a high-degree vertex can be claimed at most once. We use the term *halfedge* because the claim is not on the whole edge but rather on its incidence to one of its high-degree endpoints. The incidence at the other endpoint may or may not be claimed independently (by another vertex). For an edge uv we denote by \vec{uv} the corresponding halfedge at v and by \vec{vu} the corresponding halfedge at u . A halfedge \vec{uv} inherits the properties of its *underlying* edge uv , such as being crossed or uncrossed in a particular drawing. Vertices of degree four have a special role, as they are neither low- nor high-degree. However, a vertex of degree four that is adjacent to a hermit is treated like a low-degree vertex. More precisely, our charging scheme works as follows:

- (C1) Every hermit claims two halfedges at each high-degree neighbor.
- (C2) Every degree-three vertex claims three halfedges at some high-degree neighbor.
- (C3) Every degree four vertex that is adjacent to a hermit h claims two halfedges at some neighbor v of degree ≥ 6 . Further, the vertices h and v are adjacent, so h also claims two halfedges at v by (C1). If $\deg(v) = 6$, then v is adjacent to exactly one hermit.
- (C4) At most one vertex claims (one or more) halfedges at a degree five vertex.

The remainder of this section is organized as follows. First, we present the proof of Theorem 1 in Section 3.1. Then we prove the validity of our charging scheme along with some useful properties of low-degree vertices in Section 3.2–3.5. Specifically, we will use the following statements in the proof of Theorem 1 below.

► **Lemma 7.** *Let G be a maximal 2-planar graph on $n \geq 5$ vertices, let h be a hermit, and let x, y be the neighbors of h in G . Then we have $\deg(x) \geq 4$ and $\deg(y) \geq 4$.*

► **Lemma 8.** *Let G be a maximal 2-planar graph on $n \geq 5$ vertices. Then a vertex of degree i in G is adjacent to at most $\lfloor i/3 \rfloor$ hermits.*

3.1 Proof of Theorem 1

Let G be a maximal 2-planar graph on $n \geq 5$ vertices, and let m denote the number of edges in G . We denote by v_i the number of vertices of degree i in G . By Lemma 6 we know that G is 2-connected and, therefore, we have $v_0 = v_1 = 0$. Thus, we have

$$n = \sum_{i=2}^{n-1} v_i \quad \text{and by the handshaking lemma} \quad 2m = \sum_{i=2}^{n-1} i \cdot v_i. \quad (1)$$

Vertices of degree four or higher can be adjacent to hermits. Let v_i^{hj} denote the number of vertices of degree i incident to j hermits in G . By Lemma 8 we have

$$v_i = \sum_{j=0}^{\lfloor i/3 \rfloor} v_i^{hj} \quad \text{for all } i \geq 3. \tag{2}$$

By Lemma 7 both neighbors of a hermit have degree at least four. Thus, double counting the edges between hermits and their neighbors we obtain

$$2v_2 \leq v_4^{h1} + v_5^{h1} + v_6^{h1} + 2v_6^{h2} + v_7^{h1} + 2v_7^{h2} + 2v_8 + v_9^{h1} + 2v_9^{h2} + 3v_9^{h3} + \sum_{i=10}^{n-1} \lfloor i/3 \rfloor v_i. \tag{3}$$

If a vertex u claims halfedges at a vertex v , we say that v serves u . According to (C2), every vertex of degree three claims three halfedges at a high-degree neighbor. Every degree four vertex that is adjacent to a hermit together with this hermit claims four halfedges at a high-degree neighbor by (C3). We sum up the number of these claims and assess how many of them can be served by the different types of high-degree vertices.

In general, a high-degree vertex of degree $i \geq 5$ can serve at most $\lfloor i/3 \rfloor$ such claims. For $i \in \{5, 6, 7, 9\}$, we make a more detailed analysis, taking into account the number of adjacent hermits. Specifically, by (C3) and (C4) a degree five vertex serves at most one low-degree vertex, which is either a hermit or a degree-three vertex. A degree six vertex can serve two degree-three vertices but only if it is not adjacent to a hermit. If a degree six vertex serves a degree four vertex, it is adjacent to exactly one hermit by (C3). In particular, a degree six vertex that is adjacent to two hermits does not serve any degree three or degree four vertex. Altogether we obtain the following inequality:

$$v_3 + v_4^{h1} \leq v_5^{h0} + 2v_6^{h0} + v_6^{h1} + 2v_7^{h0} + 2v_7^{h1} + v_7^{h2} + 2v_8 + 3v_9^{h0} + 2v_9^{h1} + 2v_9^{h2} + v_9^{h3} + \sum_{i=10}^{n-1} \lfloor i/3 \rfloor v_i. \tag{4}$$

The combination $((3) + (4))/2$ together with (2) yields

$$v_2 + \frac{1}{2}v_3 \leq \frac{1}{2}v_5 + v_6 + \frac{3}{2}v_7 + 2v_8 + 2v_9 + \sum_{i=10}^{n-1} \lfloor i/3 \rfloor v_i. \tag{5}$$

Now, using these equations and inequalities, we can prove that $m - 2n \geq 0$, to complete the proof of Theorem 1. Let us start from the left hand side, using (1).

$$\begin{aligned} m - 2n &= \frac{1}{2} \sum_{i=2}^{n-1} i v_i - 2 \sum_{i=2}^{n-1} v_i = \sum_{i=2}^{n-1} \frac{i-4}{2} v_i \\ &= -v_2 - \frac{1}{2}v_3 + \frac{1}{2}v_5 + v_6 + \frac{3}{2}v_7 + 2v_8 + \frac{5}{2}v_9 + \sum_{i=10}^{n-1} \frac{i-4}{2} v_i \end{aligned}$$

By (5) the right hand side is nonnegative, quod erat demonstrandum.

3.2 Admissible Drawings

So far we have worked with the abstract graph G . In order to discuss our charging scheme, we also use a suitably chosen drawing of G . Specifically, we consider a maximal 2-planar graph G on $n \geq 5$ vertices and a crossing-minimal 2-plane drawing D of G that, among all such drawings, minimizes the number of doubly crossed edges. We refer to a drawing with these properties as an *admissible* drawing of G . By Lemma 3 we know that D is simple.

3.3 Hermits and degree four vertices

► **Lemma 9.** *Let h be a hermit and let x, y be its neighbors in G . Then x and y are adjacent in G and all three edges xy, hx, hy are uncrossed in D .*

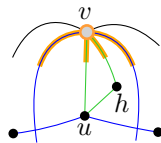
We refer to the edge xy as the *base* of the hermit h , which *hosts* h .

► **Lemma 10.** *Every edge of G hosts at most one hermit.*

By Lemma 7 both neighbors of a hermit have degree at least four. A vertex is of type T_4 -H if it has degree exactly four and it is adjacent to a hermit. The following lemma characterizes these vertices and ensures that every hermit has at least one high-degree neighbor.

► **Lemma 11.** *Let u be a T_4 -H vertex with neighbors h, v, w, x in G such that h is a hermit and v is the second neighbor of h . Then both uw and ux are doubly crossed in D , and the two faces of $D \setminus h$ incident to uv are triangles that are bounded by (parts of) edges incident to u and doubly crossed edges incident to v . Furthermore, we have $\deg(v) \geq 6$, and if $\deg(v) = 6$, then h is the only hermit adjacent to v in G .*

In our charging scheme, each hermit h claims two halfedges at each high-degree neighbor v : the halfedge \overrightarrow{hv} and the halfedge \overleftarrow{uv} , where uv denotes the edge that hosts h . Each T_4 -H vertex u claims the two doubly crossed halfedges at v that bound the triangular faces incident to uv in D .



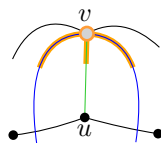
3.4 Degree-three vertices

We distinguish four different types of degree-three vertices in G , depending on their neighborhood and on the crossings on their incident edges in D . Consider a degree-three vertex u in G . By Lemma 5 every vertex is incident to at least one uncrossed edge in D .

T3-1: exactly one uncrossed edge. The two other edges incident to u are crossed.

► **Lemma 12.** *Let u be a T_3 -1 vertex with neighbors v, w, x in G such that the edge uv is uncrossed in D . Then the two faces of D incident to uv are triangles that are bounded by (parts of) edges incident to u and doubly crossed edges incident to v . Furthermore, we have $\deg(v) \geq 5$.*

In our charging scheme, each T_3 -1 vertex u claims three halfedges at its adjacent high-degree vertex v : the uncrossed halfedge \overrightarrow{uv} along with the two neighboring halfedges at v , which are doubly crossed by Lemma 12.

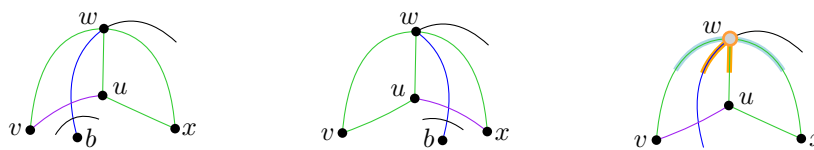


T3-2: exactly two uncrossed edges. The third edge incident to u is crossed.

► **Lemma 13.** *Let u be a T3-2 vertex in D with neighbors v, w, x such that uv is crossed at a point α . Then α is the only crossing of uv in D . Further, the edge that crosses uv is doubly crossed, it is incident to w or x , and its part between w or x and α is uncrossed.*

By the following lemma, we are free to select which of the two edges uv and ux incident to a T3-2 vertex are singly crossed; see Figure 1 (left and middle).

► **Lemma 14.** *Let u be a T3-2 vertex in D , and let the neighbors of u be v, w, x such that the edge uv is (singly) crossed by a (doubly crossed) edge wb . Then there exists an admissible drawing D' of G such that (1) D' is identical to D except for the edge wb and (2) the edge wb crosses the edge ux in D' .*



■ **Figure 1** Illustration of Lemma 14 (left and middle); halfedges claimed (marked orange) and assessed (marked lightblue) by a T3-2 vertex u (right).

► **Lemma 15.** *Let u be a T3-2 vertex with neighbors v, w, x s.t. the edge uv is singly crossed by a doubly crossed edge wb in D . Then $\deg(w) \geq 5$ and $\min\{\deg(v), \deg(x)\} \geq 4$.*

A halfedge \overrightarrow{wx} is *peripheral* for a vertex u of G if (1) u is a common neighbor of w and x ; (2) $\deg(w) \geq 5$; and (3) $\deg(x) \geq 4$. In our charging scheme, every T3-2 vertex u claims three halfedges at the adjacent high-degree vertex w : the halfedge \overrightarrow{uw} , the doubly crossed halfedge \overrightarrow{bw} , and one of the uncrossed peripheral halfedges \overrightarrow{vw} or \overrightarrow{xw} ; see Figure 1 (right). While the former two are closely tied to u , the situation is more complicated for the latter two halfedges. Eventually, we need to argue that u can exclusively claim (at least) one of the two peripheral halfedges. But for the time being we say that it *assesses* both of them.

T3-3: all three incident edges uncrossed. We say that such a vertex is of type T3-3. As an immediate consequence of Lemma 4 each T3-3 vertex u together with its neighbors $N(u)$ induces a plane K_4 in D . We further distinguish two subtypes of T3-3 vertices.

The first subtype accounts for the fact that there may be two adjacent T3-3 vertices in D . We refer to such a pair as an *inefficient hermit*. Observe that two T3-3 vertices z, z' that form an inefficient hermit have the same neighbors in $G \setminus \{z, z'\}$ by Lemma 4. A T3-3 vertex that is part of an inefficient hermit is called a *T3-3 hermit*.

► **Lemma 16.** *Let z be a T3-3 vertex in D , and let z' be a neighbor of z in G with $\deg(z') \leq 3$. Then z' is also a T3-3 vertex, that is, the pair z, z' forms an inefficient hermit in D .*

► **Lemma 17.** *Let z, z' be an inefficient hermit in D , and let x, y be their (common) neighbors in G . Then xy is an uncrossed edge in D , and the degree of x and y is at least five each.*

In particular, Lemma 17 implies that every T3-3 hermit is part of exactly one inefficient hermit. In our charging scheme, each T3-3 hermit claims three halfedges at one of its (two) adjacent high-degree vertices. More precisely, let z, z' be an inefficient hermit and let x, y be

its neighbors in G . Then the vertices x, y, z, z' induce a plane K_4 subdrawing Q of D . The vertex z claims the three halfedges of Q at x , and z' claims the three halfedges of Q at y .

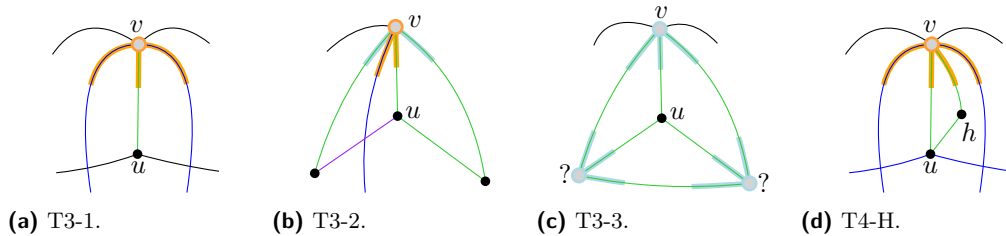
The second subtype is formed by those T3-3 vertices that are not T3-3 hermits; we call them *T3-3 minglers*. By Lemma 16 all neighbors of a T3-3 mingler have degree at least four.

► **Lemma 18.** *Let u be a T3-3 mingler in D , and let v, w, x be its neighbors. Then each of v, w, x has degree at least four. Further, at least one vertex among v, w, x has degree at least six, or at least two vertices among v, w, x have degree at least five.*

Let Q denote the plane K_4 induced by u, v, w, x in D . In our charging scheme, the T3-3 mingler u claims the three halfedges of Q at one of its high-degree neighbors. That is, the vertex u assesses all of its (up to six) peripheral halfedges at high-degree neighbors.

3.5 The charging scheme

In this section we argue that our charging scheme works out, that is, all claims made by low-degree vertices and T4-H vertices can be served by adjacent high-degree vertices. Figure 2 presents a summary of the different types of vertices and their claims.



■ **Figure 2** A vertex u with $\deg(u) \in \{3, 4\}$ and an adjacent high-degree vertex v at which u claims halfedges. Claimed halfedges are marked orange. Assessed halfedges are marked lightblue: A T3-2 vertex claims one of the two lightblue peripheral edges, and a T3-3 vertex claims a triple of halfedges at one of its high-degree neighbors.

For some halfedges it is easy to see that they are claimed at most once; these halfedges are shown orange in Figure 2. In particular, it is clear that a halfedge that is incident to the vertex that claims it is claimed at most once. We also need to consider the claims by hermits, which are not shown in the figure (except for the hermit adjacent to a T4-H vertex).

► **Lemma 19.** *Every halfedge claimed by a hermit is claimed by this hermit only.*

The next lemma settles the validity of our charging scheme for T3-1 and T4-H vertices.

► **Lemma 20.** *Every doubly crossed halfedge is claimed at most once.*

It remains to argue about the claims to peripheral halfedges by T3-2 and T3-3 vertices. Every T3-2 vertex assesses two peripheral halfedges of which it needs to claim one, and every T3-3 vertex assesses three pairs of halfedges of which it needs to claim one. In order to find a suitable assignment of claims for these vertices it is crucial that not too many vertices compete for the same sets of halfedges. Fortunately, we can show that this is not the case. We say that an edge of G is *assessed* by a low-degree vertex u if (at least) one of its corresponding halfedges is assessed by u .

► **Lemma 21.** *Every edge is assessed by at most two vertices.*

Proof. For a contradiction consider three vertices u_0, u_1, u_2 of type T3-2 or T3-3 that assess one of the halfedges of an edge uv . Then the edge uv is uncrossed in D , and all of u_0, u_1, u_2 are common neighbors of u and v in G . Moreover, we may suppose that all edges between u_0, u_1, u_2 and u, v are uncrossed in D : For T3-3 vertices all incident edges are uncrossed, anyway, and for T3-2 vertices this follows by Lemma 14. In other words, we have a plane $K_{2,3}$ subdrawing B in D between u_0, u_1, u_2 and u, v . Let ϕ_0, ϕ_1, ϕ_2 denote the three faces of B such that $\partial\phi_i = uu_i v u_{i\oplus 1}$, where \oplus denotes addition modulo 3.

Consider some $i \in \{0, 1, 2\}$. As $\deg(u_i) = 3$, there is exactly one vertex $x_i \notin \{u, v\}$ that is adjacent to u_i in G . The edge from u_i to x_i in D enters the interior of exactly one of ϕ_i or $\phi_{i\oplus 2}$. In other words, for exactly one of ϕ_i or $\phi_{i\oplus 2}$, no edge incident to u_i enters its interior. It follows that $G^- := G \setminus \{u, v\}$ is disconnected, in particular, the vertices u_0, u_1, u_2 are split into at least two components. Suppose without loss of generality that u_0 is separated from both u_1 and u_2 in G^- , and let C_0 denote the component of G^- that contains u_0 . Let D_0 denote the subdrawing of D induced by C_0 along with all edges between C_0 and u, v . Observe that uu_0v is an uncrossed path along the outer face of D_0 .

We remove D_0 from D and put it back right next to the uncrossed path uu_1v , in the face $(\phi_0$ or $\phi_1)$ incident to u_1 that is not entered by any edge incident to u_1 ; see Figure 3 for illustration. Furthermore, we flip D_0 with respect to u, v if necessary so as to ensure that the two uncrossed paths uu_1v and uu_0v appear consecutively in the circular order of edges incident to u and v , respectively, in the resulting drawing D' , effectively creating a quadrilateral face uu_1vu_0 . The drawing D' is an admissible drawing of G , to which we can add an uncrossed edge u_0u_1 in the face uu_1vu_0 , a contradiction to the maximality of G . Therefore, no such triple u_0, u_1, u_2 of vertices exists in G . ◀



■ **Figure 3** Redrawing in case that three vertices u_0, u_1, u_2 claim a halfedge of the edge uv .

Note that Lemma 21 settles the claims by T3-3 hermits, as they come in pairs that assess the same halfedges. By Lemma 21 no other vertex assesses these halfedges, so our scheme of assigning the halfedges at one endpoint to each works out. It remains to consider T3-2 vertices and T3-3 minglers. Let us start with the T3-2 vertices. Consider an edge or halfedge e that is assessed by a low-degree vertex u . We say that e is *contested* if there exists another low-degree vertex $u' \neq u$ that also assesses e . An edge or halfedge that is not contested is *uncontested*.

► **Lemma 22.** *The claims of all T3-2 vertices can be resolved in a greedy manner.*

Proof. Let u be a T3-2 vertex in D , and let \overrightarrow{wv} and \overleftarrow{xv} denote the halfedges that u assesses. We start a sequence of greedy selections for the claims of vertices u_1, u_2, \dots, u_k by letting $u_1 := u$ claim one of \overrightarrow{wv} and \overleftarrow{xv} arbitrarily, say, let u claim \overrightarrow{wv} (and withdraw its assessment of \overleftarrow{xv}). More generally, at the i -th step of our selection procedure we have a vertex u_i that has just claimed one of its assessed halfedges $\overrightarrow{w_i v_i}$. By Lemma 21 there is at most one other vertex u_{i+1} that also assesses $\overrightarrow{w_i v_i}$. If no such vertex u_{i+1} exists, then we are done and the selection procedure ends here with $i = k$. Otherwise, we consider two cases.

39:10 The Number of Edges in Maximal 2-Planar Graphs

Case 1: u_{i+1} is a T3-2 vertex. Then there is only one other (than $\overrightarrow{w_i v_i}$) halfedge that u_{i+1} assesses, denote it by $\overrightarrow{x_{i+1} v_i}$. We let u_{i+1} claim $\overrightarrow{x_{i+1} v_i}$ and proceed with the next step. \triangleleft

Case 2: u_{i+1} is a T3-3 mingler. Then u_{i+1} also assesses $\overrightarrow{v_i w_i}$, which is uncontested now, and it also assesses a second halfedge $\overrightarrow{x_{i+1} w_i}$ at w_i . We let u_{i+1} claim both $\overrightarrow{v_i w_i}$ and $\overrightarrow{x_{i+1} w_i}$ and then proceed with the next step. \triangleleft

For the correctness of the selection procedure it suffices to note that at every step exactly one halfedge is claimed that is (still) contested, and the claims of the (unique) vertex that assesses this halfedge are resolved in the next step. In particular, at the end of the procedure, all (still) assessed edges are unclaimed. As long as there exists another T3-2 vertex in D that has not claimed one of the two halfedges it requires, we start another selection procedure from there. Thus, eventually the claims of all T3-2 vertices are resolved. \blacktriangleleft

At this point it only remains to handle the claims of the remaining T3-3 minglers. They are more tricky to deal with compared to the T3-2 vertices because they require *two* halfedges at a single high-degree vertex. We may restrict our attention to a subclass of T3-3 minglers which we call *tricky*, as they assess a directed 3-cycle of contested halfedges. Consider a T3-3 mingler u , and let v, w, x be its neighbors in G . We say that u is *tricky* if (1) it assesses all six halfedges among its neighbors and (2) all of the halfedges $\overrightarrow{v w}, \overrightarrow{w x}, \overrightarrow{x v}$ or all of the halfedges $\overrightarrow{v x}, \overrightarrow{x w}, \overrightarrow{w v}$ (or both sets) are contested. A T3-3 mingler that is not tricky is *easy*.

► **Lemma 23.** *The claims of all easy T3-3 minglers can be resolved in a postprocessing step.*

Proof. Let M denote the set of easy T3-3 minglers in D . We remove M along with all the corresponding assessments from consideration, and let all other (that is, tricky) T3-3 minglers make their claims. We make no assumption about preceding claims, other than that every vertex (1) claims edges incident to one vertex only and (2) claims only edges it assesses. After all tricky T3-3 minglers have made their claims, we process the vertices from M , one by one, in an arbitrary order. In the following, the terms *assessed* and *(un)contested* refer to the initial situation, before any claims were made. The current state of a halfedge is described as either *claimed* or *unclaimed*.

Consider a vertex $u \in M$. If not all six halfedges are assessed by u , then not all of its neighbors are high-degree, in which case, at most one peripheral edge of u is contested. Thus, there always exists one pair of halfedges that is unclaimed and can be claimed by u . In the other case, let H denote the set of six halfedges that are assessed by u . By (2) every uncontested halfedge in H is unclaimed. By Lemma 21 every edge is assessed by at most two vertices. Thus, for each of the edges vx, xw, wv at most one vertex other than u assesses this edge. This other vertex may have claimed a corresponding halfedge, but by (1) for every edge vx, xw, wv at least one of its two corresponding halfedges is unclaimed.

As u is easy, at least one of $\overrightarrow{v w}, \overrightarrow{w x}, \overrightarrow{x v}$ and at least one of $\overrightarrow{v x}, \overrightarrow{x w}, \overrightarrow{w v}$ is uncontested. Suppose without loss of generality that $\overrightarrow{v w}$ is uncontested. We conclude with three cases.

Case 1: $\overrightarrow{v x}$ is uncontested. At least one of $\overrightarrow{x w}$ or $\overrightarrow{w x}$ is unclaimed. Thus, we can let u claim one of the pairs $\overrightarrow{x w}, \overrightarrow{v w}$ or $\overrightarrow{w x}, \overrightarrow{v x}$. \triangleleft

Case 2: $\overrightarrow{x w}$ is uncontested. Then we let u claim $\overrightarrow{v w}, \overrightarrow{x w}$. \triangleleft

Case 3: $\overrightarrow{w v}$ is uncontested. If one of $\overrightarrow{x w}$ or $\overrightarrow{x v}$ is unclaimed, then we let u claim it together with the matching halfedge of the edge vw , which is uncontested by assumption. Otherwise, both $\overrightarrow{x w}$ and $\overrightarrow{x v}$ are claimed. Then both $\overrightarrow{w x}$ and $\overrightarrow{v x}$ are unclaimed, and so u can safely claim these two halfedges. \blacktriangleleft

It remains to resolve the claims of tricky T3-3 minglers. Note that the classification tricky vs. easy depends on the other T3-3 minglers. For instance, a T3-3 mingler that is tricky initially may become easy after removing another easy T3-3 mingler. Here we have to deal with those T3-3 minglers only that remain tricky after all easy T3-3 minglers have been iteratively removed from consideration.

► **Lemma 24.** *The claims of all tricky T3-3 minglers can be resolved in a greedy manner.*

Proof. Let u be a tricky T3-3 mingler, and let v, w, x be its neighbors in G . As for each tricky T3-3 mingler all three peripheral edges are contested by other tricky T3-3 minglers, there exists a circular sequence $u = u_1, \dots, u_k$, with $k \geq 2$, of tricky T3-3 minglers that are neighbors of v in G and whose connecting edges appear in this order around v in D . We distinguish two cases, depending on the parity of k .

Case 1: k is even. Then we let each u_i , for i odd, claim the two halfedges at v that it assesses. This resolves the claims for all u_i , with i odd, and we claim that now all u_i , with i even, are easy. To see this, consider a vertex u_i , with i even. Both of its assessed halfedges at v are now claimed; denote these halfedges by $\overrightarrow{w_i v}$ and $\overrightarrow{x_i v}$. It follows that both $\overrightarrow{v w_i}$ and $\overrightarrow{v x_i}$ are unclaimed and u_i is the only vertex that still assesses them. As there is no directed 3-cycle of contested halfedges among the halfedges assessed by u_i anymore, the vertex u_i is easy. ◀

Case 2: k is odd. Then we let each u_i , for $i < k$ odd, claim the two halfedges at v that it assesses. This resolves the claims for these u_i and makes all u_i , with $i < k - 1$ even, easy, as in Case 1 above. It remains to argue about u_{k-1} and u_k . Let $x_i v$ denote the edge assessed by both u_i and u_{i+1} , for $1 \leq i < k$, and let $x_k v$ denote the edge assessed by both u_k and u_1 . As $\overrightarrow{x_k v}$ is claimed by u_1 , we can let u_k claim $\overrightarrow{v x_k}$ and $\overrightarrow{x_{k-1} x_k}$ along with it. This makes u_{k-1} easy, as both $\overrightarrow{v x_{k-2}}$ and $\overrightarrow{v x_{k-1}}$ are uncontested now. However, we still need to sort out the bold claim on $\overrightarrow{x_{k-1} x_k}$ by u_k . To this end, we apply the same greedy selection procedure as in the proof of Lemma 22, except that here we start with the selection of $\overrightarrow{x_{k-1} x_k}$, as the only contested halfedge that is claimed, and here we can only encounter (tricky) T3-3 minglers over the course of the procedure. ◀

We get rid of at least two tricky T3-3 minglers, either by resolving their claims or by making them easy. Thus, after a finite number of steps, no tricky T3-3 mingler remains. ◀

Our analysis of the charging scheme is almost complete now. However, we still need to justify our claim about degree five vertices in Property (C4). In principle it would be possible that two halfedges at a degree five vertex are claimed by a hermit and the remaining three by a degree-three vertex. But we can show that this is impossible.

► **Lemma 25.** *At most one low-degree vertex claims halfedges at a degree five vertex.*

4 The Upper Bound: Proof outline of Theorem 2

In this section we describe a construction for a family of maximal 2-planar graphs with few edges. We give a complete description of this family. But due to space constraints we give a very rough sketch only for the challenging part of the proof: to show that these graphs are *maximal* 2-planar. The full version [14] provides a complete version of this section, with all proofs.

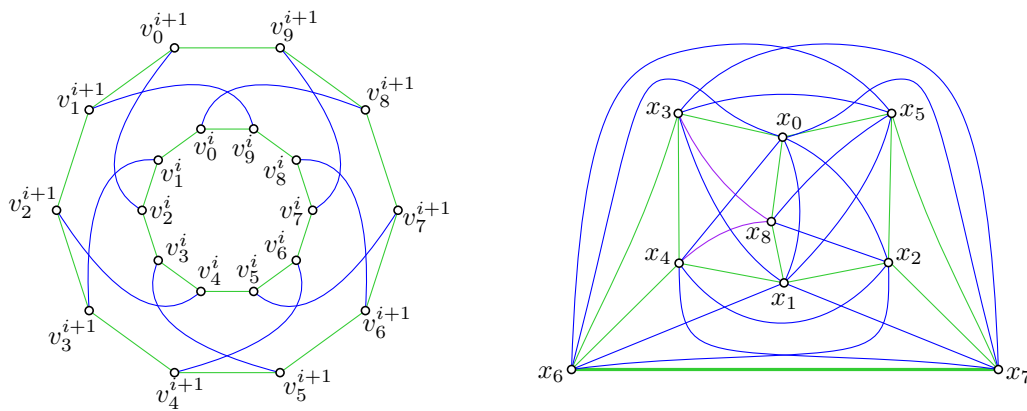
The graphs can roughly be described as braided cylindrical grids. More precisely, for a given $k \in \mathbb{N}$ we construct our graph G_k on $10k + 140$ vertices as follows.

39:12 The Number of Edges in Maximal 2-Planar Graphs

- Take k copies of C_{10} , the cycle on 10 vertices, and denote them by D_1, \dots, D_k . Denote the vertices of D_i , for $i \in \{1, \dots, 10\}$, by v_0^i, \dots, v_9^i so that the edges of D_i are $\{v_j^i v_{j \oplus 1}^i : 0 \leq j \leq 9\}$, where \oplus denotes addition modulo 10.
- For every $i \in \{1, \dots, k-1\}$, connect the vertices of D_i and D_{i+1} by a braided matching, as follows. For j even, add the edge $v_j^i v_{j \oplus 8}^{i+1}$ to G_k and for j odd, add the edge $v_j^i v_{j \oplus 2}^{i+1}$ to G_k . See Figure 4 (left) for illustration.
- To each edge of D_1 and D_k we attach a gadget $X \simeq K_9 \setminus (K_2 + K_2 + P_3)$ so as to forbid crossings along these edges. Denote the vertices of X by x_0, \dots, x_8 such that $\deg_X(x_0) = \deg_X(x_1) = 8$, $\deg_X(x_8) = 6$ and all other vertices have degree seven. Let x_6, x_7 be the non-neighbors of x_8 . To an edge e of D_1 and D_k we attach a copy of X so that e takes the role of the edge $x_6 x_7$ in this copy of X . As altogether there are 20 edges in D_1 and D_k and each copy of X adds seven more vertices, a total of $20 \cdot 7 = 140$ vertices are added to G_k with these gadgets.
- Finally, we add the edges $v_j^i v_{j \oplus 2}^{i+1}$, for all $0 \leq j \leq 9$ and $i \in \{1, k\}$.

This completes the description of the graph G_k . Note that G_k has $10k + 140$ vertices and $10k + 10(k-1) + 20 \cdot 31 + 2 \cdot 10 = 20k + 630$ edges. So to prove Theorem 2 asymptotically it suffices to choose $c \geq 630 - 2 \cdot 140 = 350$ and show that G_k is maximal 2-planar. Using some small local modifications we can then obtain the statement for all values of n .

To show that G_k is 2-planar it suffices to give a 2-plane drawing of it. Such a drawing can be deduced from Figure 4: (1) We nest the cycles D_1, \dots, D_k with their connecting edges using the drawing depicted in Figure 4 (left), (2) draw all copies of X attached to the edges of D_1 and D_k using the drawing depicted in Figure 4 (right), and (3) draw the remaining edges among the vertices of D_1 and D_k inside and outside D_1 and D_k , respectively.



■ **Figure 4** The braided matching between two consecutive ten-cycles in G_k , shown in blue (left); the gadget graph X that we attach to the edges of the first and the last ten-cycle of G_k (right).

It is much more challenging, though, to argue that G_k is *maximal* 2-planar. In fact, we do not know of a direct argument to establish this claim. Instead, we prove that G_k admits essentially only one 2-plane drawing, which is the one described above. Then maximality follows by just inspecting this drawing and observing that no edge can be added there because every pair of non-adjacent vertices is separated by a cycle of doubly-crossed edges.

This leaves us having to prove that G_k has a unique 2-plane drawing. We solve the problem in a somewhat brute-force way: by enumerating all 2-plane drawings of G_k , using computer support. Still, it is not immediately clear how to do this, given that (1) the space of (even 2-plane) drawings of a graph can be vast; (2) (G_k) is an infinite family; and (3) already for small k , even a single graph G_k is quite large.

First of all, the gadget graph X is of small constant size. So it can be analyzed separately, and there is no need to explicitly include these gadgets into the analysis of G_k . Instead, we account for the effect of the gadgets by considering the edges of D_1 and D_k as uncrossable. In this way, we also avoid counting all the variations of placing the attached copy of X on either side of the corresponding cycle as different drawings. In fact, the gadget X itself has a few formally different 2-plane drawings due to its automorphisms. But for our purposes of arguing about the maximality of G_k , these differences do not matter. However, these variations are the reason that the 2-plane drawing of G_k is *essentially* unique only.

We also disregard the length two edges along D_1 and D_k . Denote the resulting subgraph of G_i by G_i^- . We iteratively enumerate the 2-plane drawings of G_i^- , for all $i \in \mathbb{N}$, where only the edges of the first cycle D_1 are labeled uncrossable (but not the edges of the last cycle D_i). All drawings are represented as a doubly-connected edge list (DCEL) [10, Chapter 2]. As a base case, we use the unique (up to orientation, which we select to be counterclockwise, without loss of generality) plane drawing of $G_1^- = D_1$. For each drawing Γ computed, with a specific ten-cycle of vertices labeled as D , we consider all possible ways to extend Γ by adding another ten-cycle of new, labeled vertices and connect it to D using a braided matching, as in the construction of G_k and depicted in Figure 4 (left).

So in each iteration we have a partial drawing Γ and a collection H of vertices and edges still to be drawn. We then exhaustively explore the space of simple 2-plane drawings of $\Gamma \cup H$. Our approach is similar to the one used by Angelini, Bekos, Kaufmann, and Schneck [2] for complete and complete bipartite graphs. We consider the edges to be drawn in some order such that whenever an edge is considered, at least one of its endpoints is in the drawing already. When drawing an edge, we go over (1) all possible positions in the rotation at the source vertex and for each such position all options to (2) draw the edge with zero, one or two crossings. Each option to consider amounts to a traversal of some face incident to the source vertex, and up to two more faces in the neighborhood. At every step we ensure that the drawing constructed remains 2-plane and simple, and backtrack whenever an edge cannot be added or the drawing is complete (that is, it is a 2-plane drawing of G_i^- , for some $i \in \mathbb{N}$).

Every drawing for $\Gamma \cup H$ obtained in this fashion is then tested, as described below. If the tests are successful, then the drawing is added to the list of drawings to be processed, as a child of Γ , and such that the ten-cycle in H takes the role of D for future processing.

As for the testing a drawing Γ , we are only interested in a drawing that can – eventually, after possibly many iterations – be extended in the same way with an uncrossable ten-cycle D_k . In particular, all vertices and edges of D_k must lie in the same face of Γ . Hence, we test whether there exists a suitable *potential final* face in Γ where D_k can be placed; if not, then we discard Γ from further consideration. We also go over the faces of Γ and remove *irrelevant* faces and vertices that are too far from any potential final face to ever be able to interact with vertices and edges to be added in future iterations. Finally, we check whether the resulting *reduced* drawing has already been discovered by comparing it to all the already discovered drawings (by testing for an isomorphism that preserves the cycle D). If not, then we add it to the list of valid drawings.

For each drawing for which we found at least one child drawing, we also test whether there exists a similar extension where the cycle in H is uncrossable. Whenever such an extension is possible, we found a 2-plane drawing of G_i , for some $i \in \mathbb{N}$. The algorithm for G_i^- runs for about 1.5 days and discovers 86 simple 2-plane drawings of G_i^- . In only one of these drawings the last ten-cycle is uncrossed: the drawing described above (see Figure 4). The algorithm for the gadget X runs for about 3min. and discovers 32 simple 2-plane drawings, as expected. The full source code is available in our repository [13].

5 Conclusions

We have obtained tight bounds on the number of edges in maximal 2-planar graphs, up to an additive constant. Naturally, one would expect that our approach can also be applied to other families of near-planar graphs, specifically, to maximal 1- and 3-planar graphs. Intuitively, for k -planar graphs the challenge with increasing k is that the structure of the drawings gets more involved, whereas with decreasing k we aim for a higher bound.

References

- 1 Eyal Ackerman. On topological graphs with at most four crossings per edge. *Computational Geometry*, 85:101574, 2019. doi:10.1016/j.comgeo.2019.101574.
- 2 Patrizio Angelini, Michael A. Bekos, Michael Kaufmann, and Thomas Schneck. Efficient generation of different topological representations of graphs beyond-planarity. *J. Graph Algorithms Appl.*, 24(4):573–601, 2020. doi:10.7155/jgaa.00531.
- 3 Christopher Auer, Franz-Josef Brandenburg, Andreas Gleißner, and Kathrin Hanauer. On sparse maximal 2-planar graphs. In *Proc. 20th Int. Sympos. Graph Drawing (GD 2012)*, volume 7704 of *Lecture Notes Comput. Sci.*, pages 555–556. Springer, 2012. doi:10.1007/978-3-642-36763-2_50.
- 4 János Barát and Géza Tóth. Improvements on the density of maximal 1-planar graphs. *J. Graph Theory*, 88(1):101–109, 2018. doi:10.1002/jgt.22187.
- 5 János Barát and Géza Tóth. Saturated 2-planar drawings with few edges. *CoRR*, abs/2110.12781, 2021. arXiv:2110.12781.
- 6 Michael A. Bekos, Michael Kaufmann, and Chrysanthi N. Raftopoulou. On optimal 2- and 3-planar graphs. In *Proc. 33rd Internat. Sympos. Comput. Geom. (SoCG 2017)*, volume 77 of *LIPICs*, pages 16:1–16:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.SocG.2017.16.
- 7 Rainer Bodendiek, Heinz Schumacher, and Klaus Wagner. Bemerkungen zu einem Sechsfarbenproblem von G. Ringel. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 53:41–52, 1983. doi:10.1007/BF02941309.
- 8 Franz-Josef Brandenburg, David Eppstein, Andreas Gleißner, Michael T. Goodrich, Kathrin Hanauer, and Josef Reislhuber. On the density of maximal 1-planar graphs. In *Proc. 20th Int. Sympos. Graph Drawing (GD 2012)*, volume 7704 of *Lecture Notes Comput. Sci.*, pages 327–338. Springer, 2012. doi:10.1007/978-3-642-36763-2_29.
- 9 Steven Chaplick, Fabian Klute, Irene Parada, Jonathan Rollin, and Torsten Ueckerdt. Edge-minimum saturated k -planar drawings. In *Proc. 29th Int. Sympos. Graph Drawing (GD 2021)*, volume 12868 of *Lecture Notes Comput. Sci.*, pages 3–17. Springer, 2021. doi:10.1007/978-3-030-92931-2_1.
- 10 Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, Berlin, Germany, 3rd edition, 2008. doi:10.1007/978-3-540-77974-2.
- 11 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):1–37, 2020. doi:10.1145/3301281.
- 12 Péter Hajnal, Alexander Igamberdiev, Günter Rote, and André Schulz. Saturated simple and 2-simple topological graphs with few edges. *J. Graph Algorithms Appl.*, 22(1):117–138, 2018. doi:10.7155/jgaa.00460.
- 13 Michael Hoffmann and Meghana M Reddy. The number of edges in maximal 2-planar graphs—code repository. <https://github.com/meghanamreddy/Sparse-maximal-2-planar-graphs>.
- 14 Michael Hoffmann and Meghana M. Reddy. The number of edges in maximal 2-planar graphs, 2023. doi:10.48550/ARXIV.2303.08726.
- 15 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs*. Springer, Singapore, 2020. doi:10.1007/978-981-15-6533-5.

- 16 Dávid Hudák, Tomáš Madaras, and Yusuke Suzuki. On properties of maximal 1-planar graphs. *Discussiones Mathematicae*, 32:737–747, 2012. doi:10.7151/dmgt.1639.
- 17 Vladimir P. Korzhik. Minimal non-1-planar graphs. *Discrete Math.*, 308(7):1319–1327, 2008. doi:10.1016/j.disc.2007.04.009.
- 18 Jan Kynčl, János Pach, Radoš Radoičić, and Géza Tóth. Saturated simple and k-simple topological graphs. *Computational Geometry*, 48(4):295–310, 2015. doi:10.1016/j.comgeo.2014.10.008.
- 19 János Pach, Radoš Radoičić, Gábor Tardos, and Géza Tóth. Improving the crossing lemma by finding more crossings in sparse graphs. *Discrete Comput. Geom.*, 36(4):527–552, 2006. doi:10.1007/s00454-006-1264-9.
- 20 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
- 21 Gerhard Ringel. Ein Sechsfarbenproblem auf der Kugel. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 29:107–117, 1965. doi:10.1007/BF02996313.
- 22 Marcus Schaefer. The graph crossing number and its variants: A survey. *The Electronic Journal of Combinatorics*, 20, 2013. Version 7 (April 8, 2022). doi:10.37236/2713.
- 23 Yusuke Suzuki. Optimal 1-planar graphs which triangulate other surfaces. *Discr. Math.*, 310(1):6–11, 2010. doi:10.1016/j.disc.2009.07.016.

Worst-Case Deterministic Fully-Dynamic Biconnectivity in Changeable Planar Embeddings

Jacob Holm  

University of Copenhagen, Copenhagen, Denmark

Ivor van der Hoog  

Technical University of Denmark, Lyngby, Denmark

Eva Rotenberg  

Technical University of Denmark, Lyngby, Denmark

Abstract

We study dynamic planar graphs with n vertices, subject to edge deletion, edge contraction, edge insertion across a face, and the splitting of a vertex in specified corners. We dynamically maintain a combinatorial embedding of such a planar graph, subject to connectivity and 2-vertex-connectivity (biconnectivity) queries between pairs of vertices. Whenever a query pair is connected and not biconnected, we find the first and last cutvertex separating them.

Additionally, we allow local changes to the embedding by flipping the embedding of a subgraph that is connected by at most two vertices to the rest of the graph.

We support all queries and updates in deterministic, worst-case, $O(\log^2 n)$ time, using an $O(n)$ -sized data structure.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases dynamic graphs, planarity, connectivity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.40

Related Version *Full Version*: <https://arxiv.org/abs/2209.14079>

Funding *Ivor van der Hoog* and *Eva Rotenberg*: Partially supported by Independent Research Fund Denmark grants 2020-2023 (9131- 00044B) “Dynamic Network Analysis”.

Jacob Holm: Partially supported by the VILLUM Foundation grant 16582, “BARC”.

Ivor van der Hoog: This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 899987.

1 Introduction

In dynamic graph algorithms, the task is to efficiently update information about a graph that undergoes updates from a specified family of potential updates. Simultaneously, we want to efficiently support questions about properties of the graph or relations between vertices. Two vertices u and v are 2-vertex connected (i.e. biconnected) in a graph G , whenever after the removal of any vertex in G (apart from u and v) they are still connected in G . This work considers dynamically maintaining a combinatorial embedding of a graph that is planar, subject to biconnectivity queries between vertices. We show how to efficiently maintain G in $O(\log^2 n)$ time per update operation using linear space. We additionally support biconnectivity queries in $O(\log^2 n)$ time. The competitive parameters for dynamic algorithms include update time, query time, the class of allowed updates, the adversarial model, and whether times are worst-case or amortized. We present a deterministic algorithm: which means that all statements hold in the strictest adversarial model; against adaptive adversaries. Interestingly, for general graphs, there seems to be a large class of problems for which the deterministic amortized algorithms grossly outperform the deterministic worst-case time algorithms: for dynamic connectivity the state-of-the-art worst-case update time is of



© Jacob Holm, Ivor van der Hoog, and Eva Rotenberg;
licensed under Creative Commons License CC-BY 4.0

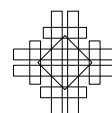
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 40; pp. 40:1–40:18

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the form $O(n^{o(1)})$ [14], whilst the state-of-the-art amortized update time is $\tilde{O}(\log^2 n)$ [21, 44]; for planarity testing, the best amortized solution has $O(\log^3 n)$ [26] update time, compared to $O(n^{2/3})$ worst-case [12] (in a restricted setting). For biconnectivity in general graphs the current best worst-case solution has update time $O(\sqrt{n})$ [6], while the best amortized update time is $\tilde{O}(\log^3 n)$ [21, 43, 27].

In this work, we provide algorithms for updating connectivity information of a combinatorially embedded planar graph, that is both deterministic, worst-case, and fully-dynamic.

► **Theorem 1.** *We maintain a planar combinatorial embedding in $O(\log^2 n)$ time subject to:*

- *delete(e): where e is an edge, deleting the edge e ,*
- *insert(u, v, f): where u, v are incident to the face f , inserting an edge uv across f ,*
- *find-face(u, v): returns some face f incident to both u and v , if any such face exists.*
- *contract(e): where e is an edge, contract the edge e ,*
- *split(v, c_1, c_2): where c_1 and c_2 are corners (corresponding to gaps between consecutive edges) around the vertex v , split v into two vertices v_{12} and v_{21} such that the edges of v_{12} are the edges of v after c_1 and before c_2 , and v_{21} are the remaining edges of v ,*
- *flip(v): for a vertex v : flip the orientation of the connected component containing v .*

We may answer the following queries in $O(\log^2 n)$ time:

- *connected(u, v), where u and v are vertices, answer whether they are connected,*
- *biconnected(u, v), where u and v are connected, answer whether they are biconnected.*

When not biconnected, we may report the separating cutvertex closest to u .

Our update time of $O(\log^2 n)$ should be seen in the light of the fact that even just supporting edge-deletion, insertion, and find-face(u, v), currently requires $O(\log^2 n)$ time [30]. We briefly review the concepts in this paper and the state-of-the-art.

Biconnectivity. For each connected component of a graph, the cutvertices are vertices whose removal disconnects the component. These cutvertices partition the edges of the graph into *blocks* where each block is either a single edge (a *bridge* or *cut-edge*), or a biconnected component. A pair of vertices are biconnected if they are incident to the same biconnected component, or, equivalently, if there are two vertex-disjoint paths connecting them. This notion generalises to k -connectivity where k objects of the graph are removed. While k -edge-connectivity is always an equivalence relation on the vertices, k -vertex-connectivity happens to be an equivalence relation for the edges only when $k \leq 2$.

Dynamic higher connectivity. Dynamic higher connectivity aims to facilitate queries to k -vertex-connectivity or k -edge-connectivity as the graph undergoes updates. For two-edge connectivity and biconnectivity in general graphs, there has been a string of work [11, 16, 6, 17, 21, 43, 27], and the current best deterministic results have $O(\log^2 n \log \log^2 n)$ amortized update time for 2-edge connectivity [27], and spend an additional amortized $\log(n)$ -factor for biconnectivity [21, 43]. Thus, the current state of the art for deterministic two-edge connectivity is $\log \log(n)$ -factors away from the best deterministic connectivity algorithm [44], while deterministic biconnectivity is $\log(n)$ -factors away. See [10, 19, 18, 21, 43, 31, 29, 44, 32, 35, 14] for more work on dynamic connectivity. For k -(edge-)connectivity with $k > 2$, only partial results have appeared, including incremental [39, 3, 38, 37, 25] and decremental [13, 42, 22, 1] results. The strongest lower bound is by Pătraşcu et al. [40], and implies that the update- and query time cannot both be $o(\log n)$ for any of the mentioned fully dynamic problems on general graphs, and this holds even for planar

embedded graphs. For special graph classes, such as planar graphs, graphs of bounded genus, and minor-free graphs, there has been a bulk of work on connectivity and higher connectivity, e.g. [9, 20, 13, 15, 5, 33, 34, 23, 22]. For dynamic planar embedded graphs, there are poly-logarithmic worst-case algorithms for two-edge [20] and two-vertex [16] connectivity, that assume a fixed planar embedding, allowing only edge-deletions and edge-insertions across a face. In this paper, we obtain the same $O(\log^2 n)$ time bound as in [20, 16], but our graphs are subject to a wider range of dynamic updates, including Whitney-flips, and edge-contractions.

An open question remains whether higher connectivity can generally be maintained in polylogarithmic worst-case time for dynamic planar graphs (k -connectivity and k -edge connectivity, $k > 2$). Particularly, this is highly motivated already when $k = 3$: In the quest for fully-dynamic planarity testing with worst-case polylogarithmic update times, an efficient algorithm for 3-vertex connectivity would be a major milestone. Namely, a 3-connected graph has a unique planar embedding (up to reflection). Much of the work on (dynamic) planarity testing [28, 36, 7, 25, 24] goes via understanding (changes to) the SPQR-tree; a tree over the 3-connected components and their interrelations. Given this, it is likely that any efficient worst-case fully-dynamic planarity testing algorithm would rely upon an efficient worst-case fully-dynamic 3-connectivity data structure. Note here that supporting changes to the embedding is crucial for this venture, since a deletion-insertion-sequence may require changes to the embedding, in order to remain planar. This paper presents the first step in this quest towards *worst-case* polylogarithmic fully-dynamic planarity testing, as we present a *worst-case* polylogarithmic data structure for 2-vertex connectivity subject to the required embedding-changing operations.

Techniques. Exploiting properties of planar graphs, we use the tree-cotree decomposition: a partitioning of edges into a spanning tree of the graph and a spanning tree of its dual. Using tree-cotree decompositions to obtain fast dynamic algorithms is a technique introduced by Eppstein [4], who obtains algorithms for dynamic graphs that have efficient genus-dependent running times. Note that the construction in [4] does not facilitate inserting edges in a way that minimises the resulting genus. Such queries are, however, allowed in the structure by Holm and Rotenberg [23], which also utilises the tree-cotree decomposition.

On this spanning tree and cotree, we use top-trees to handle local biconnectivity information. Much of our work concerns carefully choosing which biconnectivity information is relevant and sufficient to maintain, as top-tree clusters are merged and split. Note that the ideas for two-edge connectivity introduced by Hershberger et al. in [20], i.e. ideas of using topology trees on a vertex-split version of the graph to keep track of edge bundles, do not transfer to the problem at hand, since vertex-splitting changes the biconnectivity structure.

2 Preliminaries

We study a dynamic plane embedded graph $G = (V, E)$, where V has n vertices. We assume access to G and some *combinatorial embedding* [4] of G that specifies for every vertex in G the cyclical ordering of the edges incident to that vertex. Throughout the paper, we maintain some associated spanning tree $T_G = (V, E')$ over G . We study the combinatorial embedding subject to the update operations specified in Theorem 1. This is the same setting and includes the same updates as by Holm and Rotenberg [23].

Spanning and co- trees. If G is a connected graph, a spanning tree T_G is a tree where its vertices are V , and the edges of T_G are a subset of E such that T_G is connected. Given T_G , the cotree T_G^Δ has as vertices the faces in G , and as edges of T_G^Δ are all edges dual to those in $G \setminus T_G$. It is known that the cotree is a spanning tree of the dual graph of G [8].

Induced graph. We adopt the standard notion of (vertex) induced subgraphs: for any $V' \subseteq V$, $G[V']$ is the subgraph created by all edges $e \in E$ with both endpoints of e in V' . For any G and V' , we denote by $G \setminus G[V']$ the graph G minus all edges in $G[V']$. Observe that for (V_1, V_2) the set $G[V_1 \cup V_2]$ is not necessarily equal to $G[V_1] \cup G[V_2]$ (Figure 1).

Top trees. Our data structure maintains a specific variant of a *top tree* \mathcal{T}_G^{op} over the graph G [2, 41, 23]. This data structure (Figure 2) is a hierarchical decomposition of a planar, embedded, graph G based on a spanning tree T_G of G . Formally, for every connected subgraph S of T_G we define the *boundary vertices* of S as the vertices incident to an edge in $T_G \setminus S$. A *cluster* is a connected subgraph of T_G with at most 2 boundary vertices. A cluster with one boundary vertex is a *point cluster*; otherwise a *path cluster*. A top tree \mathcal{T}_G^{op} is a hierarchical decomposition of G (with depth $O(\log n)$) into point and path clusters that is structured as follows: the leaves of \mathcal{T}_G^{op} are the path and point clusters for each edge (u, v) in T_G (a leaf in \mathcal{T}_G^{op} is a point cluster if and only if the corresponding edge (u, v) is a leaf in T_G). Each inner node $\nu \in \mathcal{T}_G^{op}$ merges a constant number of *child* clusters sharing a single vertex into a new point or path cluster. The vertex set of ν is the union of those corresponding to its children. We refer to combining a constant number of nodes into a new inner node as a *merge*. We refer to its inverse as a *split*. Furthermore, for planar embedded graphs, we restrict our attention to *embedding-respecting* top trees; that is, given for each vertex a circular ordering of its incident edges, top trees that only allow merges of neighbouring clusters according to this ordering. In other words, if two clusters ν and μ share a boundary vertex b , and are mergable according to the usual rules of top trees, we only allow them to merge if furthermore they contain a pair of neighbouring edges $e_\mu \in \mu$ and $e_\nu \in \nu$ where e_μ is a neighbour of e_ν around b . Holm and Rotenberg [23] show how to dynamically maintain \mathcal{T}_G^{op} (and the spanning tree and cotree) with the following property:

► **Property 1.** Let $\nu \in \mathcal{T}_G^{op}$ be a point cluster with boundary vertex u . The graph $G[\nu]$ is a contiguous segment of the extended Euler tour of T_G .

► **Corollary 1.** Let $\nu \in \mathcal{T}_G^{op}$ be a point cluster with boundary vertex u . The edges of $G[\nu]$ that are incident to u form a connected interval in the clockwise order around u .

Edge division. For ease of exposition, we perform the trick of subdividing edges into paths of length three. We refer to G^o as the original and G as this edge-divided graph. Since G^o is planar, this does not asymptotically increase the number of vertices. We note:

1. Edge subdivision respects biconnectivity (since edge subdivision preserves the cycles in the graph; it preserves biconnectivity).



■ **Figure 1** (a) A planar graph G . (b) A set $\beta \subset V$ and $\gamma \subset V$. We show $G[\beta]$ and $G[\gamma]$. The set $G[\beta] \cup G[\gamma]$ contains no black edges. (c) We show $G \setminus G[\beta \cup \gamma]$.

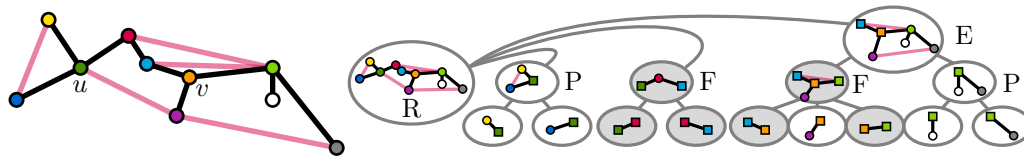


Figure 2 We recursively decompose G based on a spanning tree T_G . Square vertices are boundary vertices. We highlight path clusters. The root node has three children, where one is a path cluster that exposes $\{u, v\}$. The letters indicate the later defined merge type.

2. Any spanning tree of G^o can be transformed into a spanning tree of G where all non-tree edges have end points of degree two: for each non-tree edge in G , include exactly the first and last edge on its corresponding path in the spanning tree. This property can easily be maintained by any dynamic tree algorithm.
3. Dynamic operations in G^o easily transform to constantly many operations in G .

With this in place, our top tree structure automatically maintains more information about the endpoints of non-tree edges and their ordering around each endpoint.

Paper notation. We refer to vertices in G with Latin letters. We refer to nodes in the top tree \mathcal{T}_G^{op} with Greek letters. We refer indistinguishably to nodes $\nu \in \mathcal{T}_G^{op}$ and their associated vertex set. Vertices u and v are boundary vertices. For a path cluster $\nu \in \mathcal{T}_G^{op}$ with boundary vertices $\{u, v\}$ we call its *spine* $\pi(\nu)$ the path in T_G that connects u and v . For any path, its *internal vertices* exclude the two endpoints. For a point cluster with boundary vertex u , its spine $\pi(\nu)$ is u . We denote by $\mathcal{T}_G^{op}(\nu)$ the subtree rooted at ν .

Slim-path top trees over G . We use a variant of the top tree called a *slim-path top tree* by Holm and Rotenberg [23]. This variant of top trees upholds the *slim-path invariant*: for any path-cluster ν , all edges (of the spanning tree T_G) in the cluster that are incident to a boundary vertex belong to the spine. In other words: for every path cluster $\nu \in \mathcal{T}_G^{op}$, for each boundary vertex u , there is exactly one edge in the induced subgraph $G[\nu]$ that is connected to u . The root of this top tree is the merge between a path cluster with boundary vertices u and v , with at most two point clusters λ, μ , with $\pi(\lambda) = \{u\}$ and $\pi(\mu) = \{v\}$.¹ Holm and Rotenberg show how to obtain (and dynamically maintain) this top tree with four types of merges between clusters, illustrated by Figure 3 and 4. Our operations merge:

- (Root merge)** at most two point clusters and a path cluster to create the root node,
- (Point merge)** two point clusters μ, ν with $\pi(\mu) = \pi(\nu)$.
- (End merge)** a point and a path cluster that results in a point cluster, and
- (Four-way merge)** two path clusters μ, ν and at most two point clusters α, β , where their common intersection is one *central* vertex m . If there are two point clusters, they are not adjacent around m . This merge creates a path cluster.

Holm and Rotenberg [23] dynamically maintain the above data structure with at most $O(\log n)$ merges and splits per graph operation (where each merge or split requires $O(\log n)$ additional operations). Their data structure supports two additional critical operations:

Expose(u, v) selects two vertices u, v of G and ensures that for the unique path cluster ν of the root node, u and v are the two endpoints of $\pi(\nu)$; ($O(\log n)$ splits/merges).

¹ In the degenerate case where the graph is a star, we add one dummy edge to G to create a path cluster.

Meet(u, v, w) selects three vertices u, v, w of G and returns their meet in T_G , defined as the unique common vertex on all 3 paths between the vertices. Moreover, they also support this operation on the cotree T_G^Δ ; ($O(\log n)$ time).

When ν is a node in an embedding-respecting slim-path toptree, and G is formed from G° via edge-subdivisions, note that $G[\nu]$ has the following properties:

- For a point cluster ν with boundary vertex b that encompasses the tree-edges e_1 to e_l in the circular ordering around b , $G[\nu]$ corresponds to the sub-graph in G° induced by all vertices in ν , except edges to b that are not in e_1, \dots, e_l .
- When ν is a path cluster, $G[\nu]$ corresponds to the sub-graph in G° induced by all vertices in ν except non-tree edges incident to either of the two boundary vertices.

3 Dynamic biconnectivity queries and data structure

We want to maintain a slim-path top tree, subject to the aforementioned operations, that additionally supports biconnectivity queries in $O(\log^2 n)$ time. Our data structure consists of the slim-path top tree from [23] with three invariants which we formalise later:

1. For each cluster $\nu \in \mathcal{T}_G^{op}$, we store the biconnected components in $G[\nu]$ which are relevant for the exposed vertices (u, v) .
2. For each cluster ν , we store the information required to navigate through the top tree.
3. For each stored biconnected component, we store its “border” along the spine $\pi(\nu)$.

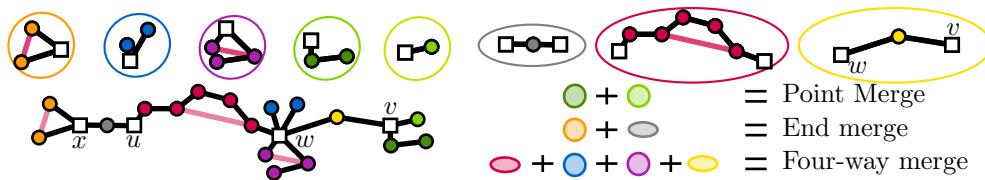
We show the technical details for our invariants. The full version contains the proofs and larger figures. We specify for each $\nu \in \mathcal{T}_G^{op}$, for each endpoint u of the spine, a *designated face*:

► **Lemma 1.** *Let $\nu \in \mathcal{T}_G^{op}$ be a path cluster. Each boundary vertex u (resp. v) is incident to a unique face $f_u^{\text{des}}(\nu)$ (resp. $f_v^{\text{des}}(\nu)$) of $G[\nu]$. Moreover, all edges in G that are not in $G[\nu]$ are contained in either $f_u^{\text{des}}(\nu)$ or $f_v^{\text{des}}(\nu)$.²*

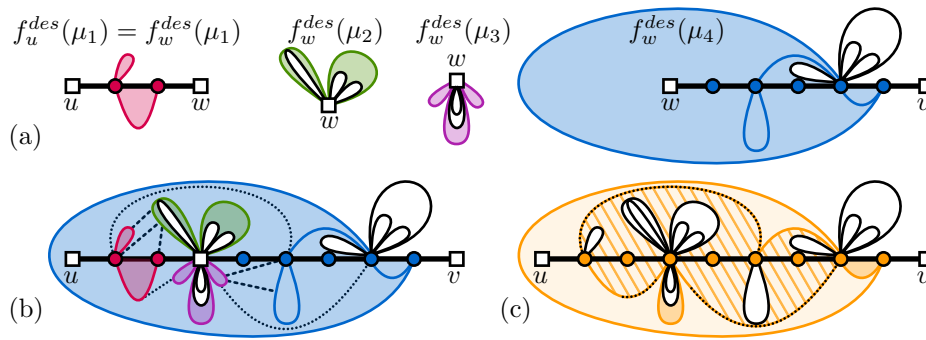
► **Lemma 2.** *Let $\nu \in \mathcal{T}_G^{op}$ be a point cluster with boundary vertex u . The subgraph $G[\nu]$ has a unique face $f_u^{\text{des}}(\nu)$ such that all edges in G that are not in $G[\nu]$ are contained in $f_u^{\text{des}}(\nu)$.*

► **Corollary 2.** *Let $\nu \in \mathcal{T}_G^{op}$ have a boundary vertex u , let μ be a descendent of ν and x be the boundary vertex of μ closest to u in T . Then $f_u^{\text{des}}(\nu) \subseteq f_x^{\text{des}}(\mu)$.*

² Formally, we can say that an edge, vertex, or face in G is contained in face f of a subgraph G' , if it is contained in f in any drawing of G that is consistent with the current combinatorial embedding.



■ **Figure 3** A graph G which we already split into five point clusters (circles) and three path clusters (ovals). We show the combinations of clusters that create three merge types. To obtain the root: execute the three suggested merges and merge the remaining components.



■ **Figure 4** (a) Nodes μ_i with faces $f_u^{des}(\mu_1)$ and $f_w^{des}(\mu_i)$. We color $BC_u(\mu_1, f_u^{des}(\mu_1))$ and $BC_w(\mu_i, f_w^{des}(\mu_i))$. (b) The four-way merge introduces a new node ν . Edges in $G[\mu_j]$ for $j \in \{1, 2, 3, 4\}$ are dashed/dotted. (c) Every dotted edge is part of a new biconnected component $B \in BC_u^*(\nu)$ which we show as tiled.

Lemmas 1 and 2 inspire the following definition: for all $\nu \in \mathcal{T}_G^{op}$, for each boundary vertex u (or v) of ν , there exists a unique face f which we call its *designated face* $f_u^{des}(\nu)$ (or $f_v^{des}(\nu)$). For biconnectivity between the exposed vertices, we are only interested in biconnected components that are edge-incident to $f_u^{des}(\nu)$ or $f_v^{des}(\nu)$. Let for a node ν with boundary vertex u , a biconnected component B be edge-incident to $f_u^{des}(\nu)$. Let μ be a descendant of ν and $B \subseteq G[\mu]$ then, by Corollary 2, B must be edge-incident to $f_x^{des}(\mu)$ where x is the boundary vertex of μ closest to u . This relation inspires us to define the *projected face* of u in μ as $\hat{f}_u^{des}(\mu) = f_x^{des}(\mu)$.

Relevant and alive biconnected components. Consider for a cluster ν , a biconnected component B of the induced subgraph $G[\nu]$. We say that B is *relevant* with respect to ν if B is vertex-incident to the spine $\pi(\nu)$. We say that B is *alive* with respect to a face f in $G[\nu]$ if B is edge-incident to f . We denote by $BC(\nu, f)$ the set of biconnected components in the induced subgraph $G[\nu]$ that are relevant with respect to ν and alive with respect to f . Intuitively, we want to keep track of the relevant and alive components (with respect $f_u^{des}(\nu)$ or $f_v^{des}(\nu)$). To save space, we store only the relevant biconnected components of ν that are not in its children. Formally (Figure 4), we define an invariant:

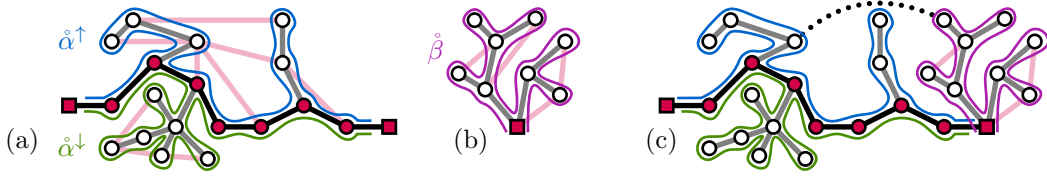
► **Invariant 1.** For each cluster $\nu \in \mathcal{T}_G^{op}$ (apart from the root) with children $\mu_1, \mu_2, \dots, \mu_s$ where u is a boundary vertex of ν , we store a unique object for each element in:

$$BC_u^*(\nu) := BC(\nu, f_u^{des}(\nu)) \setminus \bigcup_{i=1}^s BC(\mu_i, \hat{f}_u^{des}(\mu_i)).$$

Storing biconnected components in this way does not make us lose information:

► **Lemma 3.** Let $\nu \in \mathcal{T}_G^{op}$ with boundary vertex u and $B \in BC(\nu, f_u^{des}(\nu))$. There exists a unique node μ in $\mathcal{T}_G^{op}(\nu)$ where: $B \in BC_x^*(\mu)$ and x is the closest boundary vertex of μ to u .

In the remainder of this paper, we show that for each cluster $\nu \in \mathcal{T}_G^{op}$ with boundary vertex v , the set $BC_v^*(\nu)$ contains constantly many elements. Moreover, the root vertices (u, v) are biconnected at the root μ if and only if they share a biconnected components in $BC_u^*(\mu)$. In this section, we define two additional invariants so that we can maintain Invariant 1 in $O(\log n)$ additional time per split and merge. This will imply Theorem 1.



■ **Figure 5** (a) A path cluster α with $\pi(\alpha)$ in black and edges in T_G in black or grey. We show the tourpaths $\hat{\alpha}^\uparrow$ and $\hat{\alpha}^\downarrow$ in blue and green. (b) A point cluster β with the path $\hat{\beta}$. (c) Any edge in $G[\alpha \cup \beta] \setminus (G[\alpha] \cup G[\beta])$ must intersect one of $\{\hat{\alpha}^\uparrow, \hat{\alpha}^\downarrow\}$ and $\hat{\beta}$.

The core of our data structure is a slim-path top tree \mathcal{T}_G^{op} on G , that supports the expose operation in $O(\log n)$ splits and merges. In addition, it supports the meet operation in both the spanning tree T_G and its cotree T_G^Δ in $O(\log n)$ time. To maintain Invariant 1 in $O(\log n)$ time per split and merge, we add the following invariant for \mathcal{T}_G^{op} where:

▶ **Invariant 2.**

- (a) each node ν has pointers to its boundary vertices and parent node.
- (b) each path cluster ν stores: the length of $\pi(\nu)$ and its outermost spine edges.
- (c) each point cluster ν stores: the number of tree edges in ν incident to the boundary vertex.
- (d) each $x \in V$ points to the lowest common ancestor in \mathcal{T}_G^{op} where x is a boundary vertex.

Finally we add one final invariant which uses three additional concepts: slices of biconnected components, index orderings on the spine and an orientation on edges incident to a spine.

Biconnected component slices. For any node $\nu \in \mathcal{T}_G^{op}$ and any biconnected component $B \subseteq G[\nu]$, its *slice* is the interval $B \cap \pi(\nu)$ (which may be empty, or one vertex).

▶ **Lemma 4.** Let $\nu \in \mathcal{T}_G^{op}$. For all (maximal) biconnected components B in $G[\nu]$, if $B \cap \pi(\nu)$ is not empty, it is a path (possibly consisting of a single vertex).

Index orderings. For any $\nu \in \mathcal{T}_G^{op}$ and $w \in \pi(\nu)$, let w be the i 'th vertex on $\pi(\nu)$. We say that i is the *index* of w in $\pi(\nu)$. We can use Invariant 2 to obtain this index:

▶ **Lemma 5.** Given Invariant 2, a path cluster ν and a vertex $w \in \pi(\nu)$, we can compute for every path cluster β with $\pi(\beta) \subseteq \pi(\nu)$ the index of w in $\pi(\beta)$ in $O(\log n)$ total time.

Similarly in a point cluster ν , for each tree edge in ν incident to the boundary vertex u we define its *clockwise index* in ν as its index in the clockwise ordering of the edges around u .

▶ **Lemma 6.** Given Invariant 2, and an edge $e = (u, x) \in T_G$, we can compute the clockwise index of e in every point cluster $\nu \in \mathcal{T}_G^{op}$ that contains e and has u as boundary vertex, simultaneously, in worst case $O(\log n)$ total time.

Euler tour paths and endpoint orientations. Consider the Euler tour of T_G and an embedding of that Euler tour such that the Euler tour is arbitrarily close to the edges in T_G . Each edge e in G that is not in T_G must intersect the Euler tour twice. We classify each endpoint of e based on where it intersects this Euler tour. Formally, we define (Figure 5):

▶ **Definition 3.** For a point cluster ν with boundary vertex u , we denote by $\hat{\nu}$ its **tourpath** (the segment of the Euler tour in T_G from u to u that is incident to edges in $G[\nu]$).

► **Definition 4.** For a path cluster ν with boundary vertices u and v . We denote by $\hat{\nu}^\uparrow$ and $\hat{\nu}^\downarrow$ its two *tourpaths* (the two paths in the Euler tour in T_G from u to v).

► **Definition 5** (Figure 6). For any tourpath $\hat{\alpha}$ let e_1 be the first and e_2 be the last edge of T_G incident to $\hat{\alpha}$. We denote by $f^{\text{first}}(\hat{\alpha})$ the unique face in G (incident to e_1) whose interior contains the start of $\hat{\alpha}$. The face $f^{\text{last}}(\hat{\alpha})$ is defined analogously using e_2 .

► **Observation 6.** Let ν be a path cluster with the slim-path property. Any edge in $G[\nu]$ is either an edge in T_G or it must intersect one of $\{\hat{\nu}^\uparrow, \hat{\nu}^\downarrow\}$.

We introduce one last concept. Let $e = (x, y)$ be an edge of G not in T_G where e is an edge with an endpoint in $G[\alpha]$ and in $G[\beta]$ (for two clusters $\alpha, \beta \in \mathcal{T}_G^{\text{op}}$). We intuitively refer for an endpoint x of e , to the tourpath intersected by e “near” x . Let α be a path cluster. We say that the endpoint $x \in \alpha$ of e is a *northern* endpoint if e intersects $\hat{\alpha}^\uparrow$ near x , and a *southern* endpoint if e intersects $\hat{\alpha}^\downarrow$ near x . This distinction between north and south endpoints inspires the notion of biconnected component borders (Figure 7):

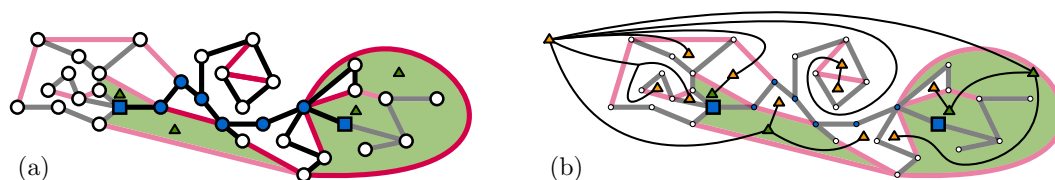
► **Definition 7** (Biconnected component borders). Let B be a subset of the edges in $G[\nu]$ that induces a biconnected subgraph such that $B \cap \pi(\nu)$ is a path (or singleton vertex).

- Let ν be a point cluster. Consider the clockwise ordering of edges in B incident to its boundary vertex u , starting from $f_u^{\text{des}}(\nu)$. The **border** of B is:
 - the vertex u together with its **eastern border**: the first edge of B in this ordering, and its **western border**: the last edge of B in this ordering.
- Let ν be a path cluster. Denote by a the “eastmost” vertex of $B \cap \pi(\nu)$ and by b its “westmost” vertex. If $a = b$ then the border is empty. Otherwise:
 - the **eastern border** of B is a , together with the first northern and last southern edge of B that is incident to a .
 - the **western border** of B is b , together with the last northern and first southern edge of B that is incident to b .

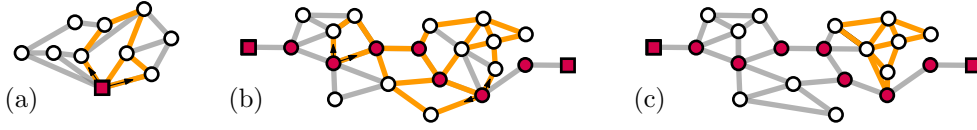
► **Invariant 3.** For any $\nu \in \mathcal{T}_G^{\text{op}}$ with boundary vertex u , for all $B \in BC_u^*(\nu)$ we store the **borders** of B in ν and their indices and clockwise indices in ν .

Using invariants for biconnectivity. Invariant 2 allows us to not only obtain the meet between vertices, but also the edges of their path to this meet (incident to the meet):

► **Theorem 2.** Given Invariant 2, let ν be a path cluster with boundary vertices u and v and $w \notin \pi(\nu)$ be a vertex in $G[\nu]$. We can obtain the meet $m = \text{meet}(u, v, w)$ and the last edge e^* in the path from w to m (in T_G) in $O(\log n)$ time.



■ **Figure 6** (a) A graph with the spine of a node ν shown in blue. We show the faces $(f^{\text{first}}(\hat{\alpha}^\uparrow), f^{\text{last}}(\hat{\alpha}^\uparrow), f^{\text{first}}(\hat{\alpha}^\downarrow), f^{\text{last}}(\hat{\alpha}^\downarrow))$ in green. (b) The cotree T_G^Δ . Vertices are triangles.



■ **Figure 7** Three times a cluster ν with $\pi(\nu)$ as red vertices and a yellow (not maximal) set of biconnected edges in $G[\nu]$. We show: (a) a border in a point cluster (b) a border in a path cluster and (c) a set of biconnected edges in $G[\nu]$ that has an empty border.

In our later analysis, we show that for each merge, the only edges e° which can be part of new relevant and alive biconnected components are the edges incident to some convenient meets in the dual graph. Given such an edge e° , we identify a convenient edge e^* of the newly formed biconnected component B . We identify the already stored biconnected components $B^* \in BC_u(\nu)$ which contain e^* (these components B^* get “absorbed” into B). We use Invariant 3 to identify all such B^* that contain e^* :

► **Theorem 3.** *Let $e^* \in T_G$ be an edge incident to a vertex u . Let k be the maximum over all u and ν of the number of elements in $BC_u^*(\nu)$. In $O(k \log n)$ total time we can, for all of the $O(\log n)$ nodes $\nu \in \mathcal{T}_G^{op}$ that contain e^* , for each $B^* \in BC_u^*(\nu)$, determine if e^* is in between the border of B^* in ν .*

Finally, Invariants 2 + 3 will suffice to maintain Invariant 1 in $O(\log n)$ time per split and merge (and thus, in $O(\log^2 n)$ time per update operation) which will prove Theorem 1.

4 Summary of the remainder of this paper

We present a high-level overview of how Theorem 1 is obtained in the remainder of this paper. At all times, we dynamically maintain a (combinatorial) embedding of some edge-divided graph G . We maintain the top tree \mathcal{T}_G^{op} by Holm and Rotenberg from [23] augmented with three aforementioned invariants. All updates the combinatorial embedding in Theorem 1 can be realized by $O(\log n)$ split and merge operations on the top tree (and co-tree). On a high level, we maintain all three invariants with $O(\log n)$ additional time per split and merge in the top tree (and co-tree). Thus, we have $O(\log^2 n)$ total update time.

4.1 Invariant 2: pointers in the top tree

Invariant 2 specifies that we want to store for each cluster ν in the top tree some “metadata”. This metadata can be stored using $O(1)$ space and $O(\log n)$ additional time per split and merge. Indeed, per split we simply delete the constant-complexity data. It may occur that for a vertex $x \in V$, we delete the lowest common ancestor in \mathcal{T}_G^{op} where x is the boundary vertex: Since a top tree is a balanced tree with $O(1)$ boundary vertices per node, we find the new lowest common ancestor in $O(\log n)$ time. For a merge, we simply compute components (a), (b) and (c) in $O(1)$ additional time. For the $O(1)$ boundary vertices, we test if there exists a vertex $x \in V$ for which Invariant 2(d) changes in $O(\log n)$ additional time.

4.2 Invariant 1: maintaining $BC_u^*(\nu)$

We define k as the maximum over all vertices u and clusters ν , of the size of $BC_u^*(\nu)$. During each split, a cluster ν with boundary vertex u is destroyed and we simply delete $BC_u^*(\nu)$ in $O(k)$ time. What remains is to show that when we merge clusters in our top trees to create a vertex ν with boundary vertex u , we can identify the new $BC_u^*(\nu)$.

Suppose we merge clusters α and β to create a cluster ν with boundary vertex u (Figure 8 (a)). We want to construct the set $BC_u^*(\nu)$ of “newly formed” relevant and alive biconnected components. We assumed that every such set contains at most k elements. Any $B \in BC_u^*(\nu)$ contains at least one edge e° in $G[\alpha \cup \beta] \setminus (G[\alpha] \cup G[\beta])$. Assume, for now, that there exist at most $O(k)$ such edges e° .

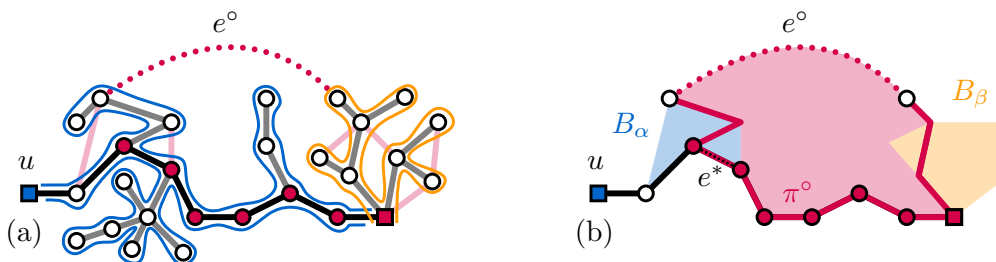
Any edge e° in $G[\alpha \cup \beta] \setminus (G[\alpha] \cup G[\beta])$ must be part of some new biconnected component B in $G[\nu]$. Indeed: e° has one endpoint x in $G[\alpha]$ and y in $G[\beta]$. The edge e° together with the path π° in the spanning tree connecting x and y must form a cycle. To determine whether B is a relevant and alive biconnected component, we want to (implicitly) compute the out-most cycle bounding B . This is not straightforward: as the biconnected component B contains the aforementioned cycle, but may also absorb biconnected components B_α in $G[\alpha]$ and B_β in $G[\beta]$ (Figure 8 (b)).

Testing whether B is relevant. The new biconnected component B is in $BC_u^*(\nu)$ whenever it (partially) coincides with at least one edge from the spine $\pi(\nu)$. We show that this occurs if and only if π° (partially) coincides with $\pi(\nu)$ and we test this in $O(\log n)$ time.

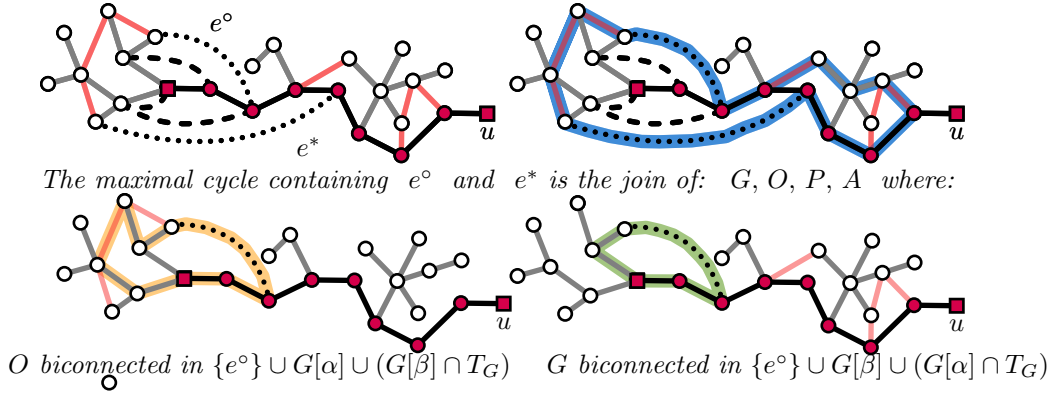
Testing whether B is alive. The newly formed biconnected component B is alive whenever it is incident to the face $f_u^{des}(\nu)$ in the graph $G[\nu]$. We want for a given pair (e°, B) test whether B is alive in $O(k \log n)$ time. In the full version we show how to do this efficiently. The core idea of this proof is (Figure 11 (a)) that B is alive if and only if one of two things is true:

1. $\{e^\circ\} \cup \pi^\circ$ incident to $f_u^{des}(\nu)$, or
2. B contains some (maximal) pre-stored biconnected B^* incident to $f_u^{des}(\nu)$.

We test case 1 with conventional methods in $O(\log n)$ time. For case 2, we identify a special edge e^* on π° . We show that such a pre-stored biconnected component B^* exists only if $e^* \in B^*$. Since the top tree has height $O(\log n)$, e^* may be contained in $O(k \log n)$ pre-stored biconnected components. We find these in $O(1)$ amortized constant time per biconnected component. I.e., our invariants allow us to apply Theorem 3 to identify such a B^* in $O(k \log n)$ worst case total time. Given the maximal B^* that contains e^* , we test whether B^* is incident to $f_u^{des}(\nu)$ using an additional $O(\log n)$ time (searching over its boundary).



■ **Figure 8** (a) Suppose that we merge a path cluster α (blue) with a point cluster β (yellow) to create a new point cluster ν (we call this an end merge). We are interested in all “new” biconnected components in $G[\nu]$. Every such new biconnected component must contain an edge e° in $G[\alpha \cup \beta] \setminus (G[\alpha] \cup G[\beta])$. (b) Consider the path π° along the spanning tree that connects the two endpoints of e° (red). This creates a cycle, and thus a new biconnected component B in the graph $G[\nu]$. The component B consists of this cycle, but may additionally “absorb” biconnected components B_α in $G[\alpha]$ and B_β in $G[\beta]$.



■ **Figure 9** A cluster ν with as children a point cluster α and a path cluster β . There may be many edges in $G[\alpha \cup \beta]$. These edges are all contained in some maximal cycle which we show in blue. For the edge e° , we show the biconnected component G in $\{e^\circ\} \cup G[\beta] \cup (T_G \cap G[\alpha])$ and O in $\{e^\circ\} \cup G[\alpha] \cup (T_G \cap G[\beta])$. Similarly, for the edge e^* we show the biconnected components P and A . On an intuitive level, the maximal blue cycle is their “join”.

Thus, we identify for every such edge e° in $O(k \log n)$ time whether it created a new relevant and alive biconnected component $B \in BC_u^*(\nu)$. If so, we create an object representing $B \in BC_u^*(\nu)$ in $O(1)$ additional time.

4.3 Invariant 3: storing the border of B

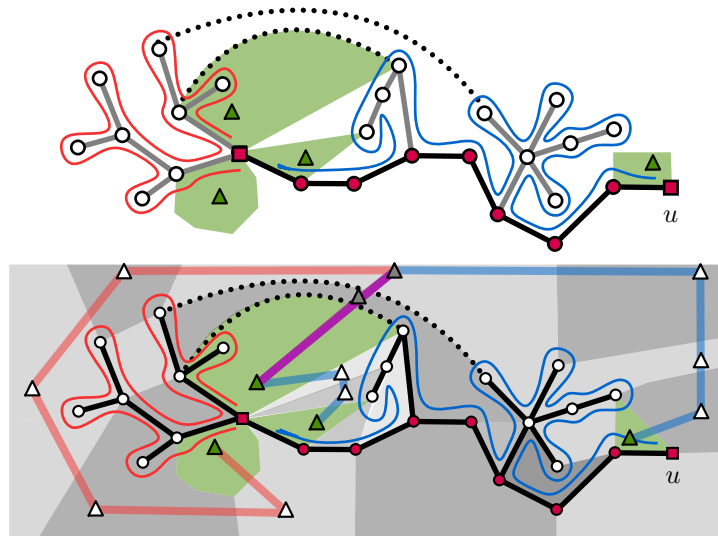
For each $B \in BC_u^*(\nu)$, we show in the full version that we can not only compute B but also its border to store in Invariant 3. The core idea (Figure 11 (b)) is that when merging two clusters α and β we can “project” e° onto $G[\alpha]$ and $G[\beta]$ to find the border of B in the respective graphs. However, two complications arise:

Firstly, we observed earlier that B may be the result of combining the cycle $e^\circ \cup \pi^\circ$ with a biconnected component B_α in $G[\alpha]$ (additionally, some biconnected component B_β in $G[\beta]$). Whenever that is the case, the eastern border of B is not the border of the path $\pi^\circ \cap \pi(\nu)$, but it rather gets “extended” to be the eastern border of B_α . This complication is relatively easy to solve: In the previous subsection we explained that we can find B_α . Since we merge the trees bottom-up, we have already restored Invariant 3 for the node α . Thus, we obtain the eastern border of B_α in $O(1)$ additional time and set it to be the eastern border of B .

Secondly, a merge can contain up to four clusters, not only two. We perform an extensive case analysis where we show that we can construct the border of B in $G[\nu]$ by pairwise joining projected borders (for an example, see Figure 9).

4.4 Finalising our argument

Up to this point, we showed that we can maintain our data structure and its invariants in $O(k^2 \log^2 n)$ time per operation in G . The integer k has two functions: first, it upper bounds the number of elements in $BC_u^*(\nu)$ for any $u \in V$ and $\nu \in \mathcal{T}_G^{op}$. Second we assumed that to maintain $BC_u^*(\nu)$, for each merge we need to inspect at most $O(k)$ edges $e^\circ \in G[\alpha \cup \beta] \setminus (G[\alpha] \cup G[\beta])$ in $O(k \log n)$ time each. In the full version we prove Theorem 1 by proving that such a k exists and that it is constant.



■ **Figure 10** An End merge between a point cluster α and a path cluster β to create a new path cluster ν . We show two Euler tours $\hat{\alpha}$ and $\hat{\beta}^\dagger$ in blue and red. The tour $\hat{\alpha}$ corresponds to the red path in the dual between two faces. The tour $\hat{\beta}^\dagger$ to the blue path. The purple path is their meet. Any edge $e^\circ \in G[\alpha \cup \beta] \setminus (G[\alpha] \cup G[\beta])$ intersects both $\hat{\alpha}$ and $\hat{\beta}^\dagger$ (or $\hat{\alpha}$ and $\hat{\beta}^\dagger$) and must thus lie on the purple path (or an alternative meet in the dual). The first edge on this path is e^* , as any further edge cannot be incident to the face $f_u^{des}(\nu)$.

Proving k exists and that it is a constant. Consider any edge $e^\circ \in G[\alpha \cup \beta] \setminus (G[\alpha] \cup G[\beta])$. We observe that e° must intersect a tourpath of α and a tourpath of β . For any fixed pair of tourpaths $(\hat{\alpha}, \hat{\beta})$ we consider our co-tree (i.e., the spanning tree on the dual of G). The Euler tour around $\hat{\alpha}$ is a path in the dual. Similarly, the Euler tour around $\hat{\beta}$ is a path and their common intersection is a meet in the dual (Figure 10). All edges that intersect both $\hat{\alpha}$ and $\hat{\beta}$ must lie on this meet (this concept is similar to the *edge bundles* by Laporte et al. in [30]). Thus, by Theorem 2, we can obtain for each pair $(\hat{\alpha}, \hat{\beta})$ this meet in $O(\log n)$ time.

We show that we may restrict our attention to the first edge e^* of this bundle (i.e. the first edge encountered on the meet). Indeed, the cycle formed by T_G and e^* encloses all other edges e° of the bundle in a face f^* . We are only interested in biconnected components that are alive (incident to the face $f_u^{des}(\nu)$). For all other edges e° their respective biconnected components B either include e^* , or are contained in f^* and can therefore not be incident to $f_u^{des}(\nu)$. It follows that whenever we create a new node ν with boundary vertex u , for every pair of tourpaths of its children, there is a unique edge e^* which can form a new biconnected component in $BC_u^*(\nu)$. Each merge involves at most 4 children that collectively have at most 6 tourpaths, and thus k is upper bound by 6 choose 2 (which is 15).

One special case. The above proof strategy applies to almost all our merge types. There exists however, one special case. During a four-way merge, whenever α and β are path clusters around a central vertex m , there exists no such “maximal” edge e^* (Figure 11 (c)). Thus, we cannot identify the biconnected components created by the edge bundle between $G[\alpha]$ and $G[\beta]$. We observe that any such component is only useful for answering biconnectivity queries between u and v if it connects the edges e_1 and e_2 of $\pi(\nu)$ incident to m . Indeed, if removing m separates e_1 and e_2 then it must also separate u and v (which are the boundary vertices of the root). We test if removing m separates e_1 and e_2 in G in $O(\log n)$ time. Note

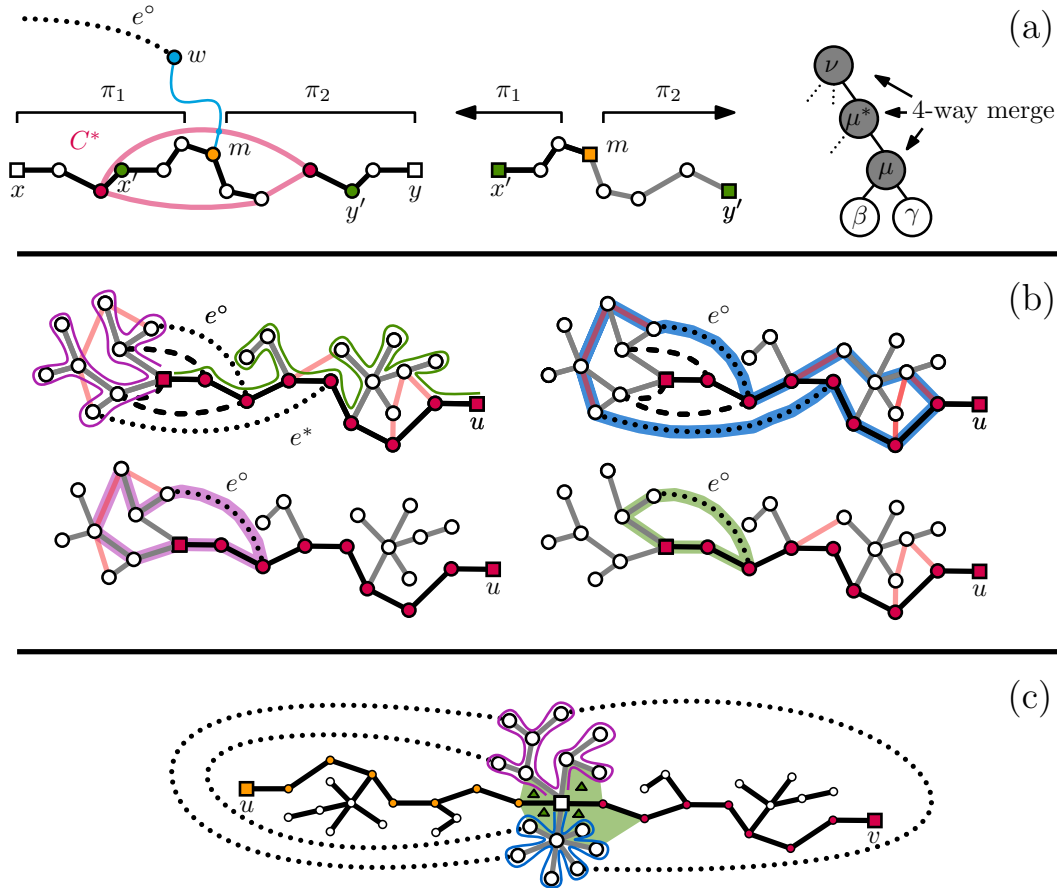
that testing if removing a vertex separates two other vertices is already possible in $O(\log^2 n)$ time using [23] (by splitting m along the right corners and testing for connectivity). However to have $O(\log^2 n)$ total update time, we want $O(\log n)$ update time per merge. In the full version we open their black box slightly to test this in $O(\log n)$ time instead.

4.5 Conclusion

We have presented an efficient data structure for 2-vertex connectivity in dynamic planar graphs subject to edge-insertions, edge-deletions, contractions, splits, and local changes to the embedding in the form of flips. In this process, and with this result, we may have taken a first step towards worst-case deterministic fully-dynamic planarity testing in polylogarithmic time; one of the fundamental research questions in dynamic graph drawing.

Our technique is to consider the planar top-tree, and our contribution includes insights into important features and information to store in the top-tree clusters. Top-tree clusters can be seen as sketches of subgraphs, and thus, these insights about subgraph features for this computational problem, may have independent interest. Indeed, the concepts of *designated face* and *alive*, may be useful when constructing a dynamic data structure for fully-dynamic planar 3-vertex connectivity, or even for higher vertex connectivity in dynamic planar graphs.

Looking forward, there is a multitude of planar graph problems which would be interesting to examine in the worst-case deterministic fully-dynamic setting. Examples of such problems include $k \geq 3$ -vertex connectivity, k -edge connectivity, arboricity decomposition, and various questions about constrained planar drawings of directed graphs. Approaching any of these related questions would require new ideas and techniques.



■ **Figure 11** Three challenges that are encountered in our paper and described in our overview.
 (a) Let e° have some endpoint w and consider the path in T_G to some vertex m . Let B be the biconnected component of $G[\nu]$ that contains e° . There exists some child μ of ν where m is the central vertex of the merge. If B contains some pre-stored biconnected component B^* (red) then B includes either the edge e_1 or e_2 in $G[\mu]$ incident to m .
 (b) Consider an End Merge and an edge e° intersecting the purple and green Euler tours. The edge e° is part of a biconnected component with the blue cycle as its outer cycle. We find for e° , however, only the purple and green cycles in $G[\alpha]$ and $G[\beta]$ separately. We smartly join these cycles together with the cycles for e^* to get the out-most cycle bounding the new biconnected component B^* .
 (c) In a four-way merge, the edges incident to the outer face of the embedding may be arbitrary edges in the edge bundle between the two path clusters. Neither edges incident to the outer face are incident to $f_u^{des}(\nu)$ or $f_v^{des}(\nu)$. Since we have no techniques for finding these edges, we instead test if the central vertex separates (u, v) .

References

- 1 Anders Aamand, Adam Karczmarz, Jakub Lacki, Nikos Parotsidis, Peter M. R. Rasmussen, and Mikkel Thorup. Optimal decremental connectivity in non-sparse graphs. *ArXiv*, 2021.
- 2 Stephen Alstrup, Jacob Holm, Kristian De Lichtenberg, and Mikkel Thorup. Maintaining information in fully dynamic trees with top trees. *Acm Transactions on Algorithms (TALG)*, 2005. doi:10.1145/1103963.1103966.
- 3 Giuseppe Di Battista and Roberto Tamassia. On-line maintenance of triconnected components with spqr-trees. *Algorithmica*, 1996. doi:10.1007/BF01961541.
- 4 David Eppstein. Dynamic generators of topologically embedded graphs. In *ACM-SIAM Symposium on Discrete algorithms (SODA)*, 2003. doi:10.5555/644108.644208.
- 5 David Eppstein, Zvi Galil, Giuseppe Italiano, and Thomas Spencer. Separator-based sparsification ii: Edge and vertex connectivity. *SIAM Journal on Computing*, 1999. doi:10.1137/S0097539794269072.
- 6 David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Amnon Nissenzweig. Sparsification - a technique for speeding up dynamic graph algorithms. *Journal of the ACM (JACM)*, 1997. doi:10.1145/265910.265914.
- 7 David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Thomas H. Spencer. Separator based sparsification. i. planary testing and minimum spanning trees. *Journal of Computer and System Sciences (JCSS)*, 1996. doi:10.1006/jcss.1996.0002.
- 8 David Eppstein, Giuseppe F Italiano, Roberto Tamassia, Robert Tarjan, Jeffery Westbrook, and Moti Yung. Maintenance of a minimum spanning forest in a dynamic plane graph. *Journal of Algorithms*, 1992. doi:10.1016/0196-6774(92)90004-V.
- 9 David Eppstein, Giuseppe F. Italiano, Roberto Tamassia, Robert E. Tarjan, Jeffery R. Westbrook, and Moti Yung. Maintenance of a minimum spanning forest in a dynamic planar graph. *Journal of Algorithms*, 1992.
- 10 Greg Frederickson. Data structures for on-line updating of minimum spanning trees, with applications. *SIAM Journal on Computing*, 1985.
- 11 Greg Frederickson. Ambivalent data structures for dynamic 2-edge-connectivity and k smallest spanning trees. *SIAM Journal on Computing*, 1997. doi:10.1137/S0097539792226825.
- 12 Zvi Galil, Giuseppe F. Italiano, and Neil Sarnak. Fully dynamic planarity testing with applications. *Journal of the ACM (JACM)*, 1999. doi:10.1145/300515.300517.
- 13 Dora Giammarresi and Giuseppe F. Italiano. Decremental 2- and 3-connectivity on planar graphs. *Algorithmica*, 1996. doi:10.1007/BF01955676.
- 14 Gramoz Goranci, Harald Räcke, Thatchaphol Saranurak, and Zihan Tan. The expander hierarchy and its applications to dynamic graph algorithms. In Dániel Marx, editor, *ACM-SIAM Symposium on Discrete algorithms (SODA)*, 2021. doi:10.1137/1.9781611976465.132.
- 15 Jens Gustedt. Efficient union-find for planar graphs and other sparse graph classes. *Theoretical Computer Science (TCS)*, 1998. doi:10.1016/S0304-3975(97)00291-0.
- 16 Monika R. Henzinger and Han La Poutré. Certificates and fast algorithms for biconnectivity in fully-dynamic graphs. In *European Symposium on Algorithms (ESA)*, 1995.
- 17 Monika Rauch Henzinger and Valerie King. Fully dynamic 2-edge connectivity algorithm in polylogarithmic time per operation, 1997.
- 18 Monika Rauch Henzinger and Valerie King. Randomized fully dynamic graph algorithms with polylogarithmic time per operation. *Journal of the ACM (JACM)*, 1999. doi:10.1145/320211.320215.
- 19 Monika Rauch Henzinger and Mikkel Thorup. Sampling to provide or to bound: With applications to fully dynamic graph algorithms. *Random Structures and Algorithms*, 1997. doi:10.1002/(SICI)1098-2418(199712)11:4<369::AID-RSA5>3.0.CO;2-X.
- 20 John Hershberger, Monika Rauch, and Subhash Suri. Data structures for two-edge connectivity in planar graphs. *Theoretical Computer Science (TCS)*, 1994. doi:10.1016/0304-3975(94)90156-2.

- 21 Jacob Holm, Kristian de Lichtenberg, and Mikkel Thorup. Poly-logarithmic deterministic fully-dynamic algorithms for connectivity, minimum spanning tree, 2-edge, and biconnectivity. *Journal of the ACM (JACM)*, 2001. doi:10.1145/502090.502095.
- 22 Jacob Holm, Giuseppe Italiano, Adam Karczmarz, Jakub Lacki, and Eva Rotenberg. Decremental SPQR-trees for Planar Graphs. In *European Symposium on Algorithms (ESA)*, 2018. doi:10.4230/LIPIcs.ESA.2018.46.
- 23 Jacob Holm and Eva Rotenberg. Dynamic planar embeddings of dynamic graphs. *Theory of Computing Systems (TCS)*, 2017. doi:10.1007/s00224-017-9768-7.
- 24 Jacob Holm and Eva Rotenberg. Fully-dynamic planarity testing in polylogarithmic time. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *PACM Symposium on Theory of Computing (STOC)*, 2020. doi:10.1145/3357713.3384249.
- 25 Jacob Holm and Eva Rotenberg. Worst-case polylog incremental SPQR-trees: Embeddings, planarity, and triconnectivity. In Shuchi Chawla, editor, *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2020. doi:10.1137/1.9781611975994.146.
- 26 Jacob Holm and Eva Rotenberg. Good r-divisions imply optimal amortised decremental biconnectivity. *Symposium on Theoretical Aspects of Computer Science (STACS)*, 2021. doi:10.4230/LIPIcs.STACS.2021.42.
- 27 Jacob Holm, Eva Rotenberg, and Mikkel Thorup. Dynamic bridge-finding in $\tilde{O}(\log^2 n)$ amortized time. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2018. doi:10.1137/1.9781611975031.3.
- 28 John E. Hopcroft and Robert Endre Tarjan. Efficient planarity testing. *Journal of the ACM (JACM)*, 1974. doi:10.1145/321850.321852.
- 29 Shang-En Huang, Dawei Huang, Tsvi Kopelowitz, and Seth Pettie. Fully dynamic connectivity in $O(\log n(\log \log n)^2)$ amortized expected time. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2017. doi:10.1137/1.9781611974782.32.
- 30 Giuseppe F. Italiano, Johannes A. La Poutré, and Monika Rauch. Fully dynamic planarity testing in planar embedded graphs (extended abstract). In Thomas Lengauer, editor, *European Symposium on Algorithms (ESA)*, 1993. doi:10.1007/3-540-57273-2_57.
- 31 Bruce Kapron, Valerie King, and Ben Mountjoy. Dynamic graph connectivity in polylogarithmic worst case time. In *ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2013. doi:10.1137/1.9781611973105.81.
- 32 Casper Kejlberg-Rasmussen, Tsvi Kopelowitz, Seth Pettie, and Mikkel Thorup. Faster Worst Case Deterministic Dynamic Connectivity. In *European Symposium on Algorithms (ESA)*, 2016. doi:10.4230/LIPIcs.ESA.2016.53.
- 33 Jakub Lacki and Piotr Sankowski. Min-cuts and shortest cycles in planar graphs in $O(n \log \log n)$ time. In *European Symposium on Algorithms (ESA)*, 2011. doi:10.1007/978-3-642-23719-5_14.
- 34 Jakub Lacki and Piotr Sankowski. Optimal decremental connectivity in planar graphs. In *Symposium on Theoretical Aspects of Computer Science, (STACS)*, 2015. doi:10.4230/LIPIcs.STACS.2015.608.
- 35 Danupon Nanongkai, Thatchaphol Saranurak, and Christian Wulff-Nilsen. Dynamic minimum spanning forest with subpolynomial worst-case update time. In *Symposium on Foundations of Computer Science (FOCS)*, 2017. doi:10.1109/FOCS.2017.92.
- 36 Johannes A. La Poutré. Alpha-algorithms for incremental planarity testing (preliminary version). In Frank Thomson Leighton and Michael T. Goodrich, editors, *ACM Symposium on Theory of Computing (STOC)*, 1994. doi:10.1145/195058.195439.
- 37 Johannes A. La Poutré. Maintenance of 2- and 3-edge-connected components of graphs II. *SIAM Journal of Computing*, 2000. doi:10.1137/S0097539793257770.
- 38 Johannes A. La Poutré, Jan van Leeuwen, and Mark H. Overmars. Maintenance of 2- and 3-edge- connected components of graphs I. *Discrete Mathematics*, 1993. doi:10.1016/0012-365X(93)90376-5.

- 39 Johannes A. La Poutré and Jeffery R. Westbrook. Dynamic 2-connectivity with backtracking. *SIAM Journal of Computing*, 1998. doi:10.1137/S0097539794272582.
- 40 Mihai Pătraşcu and Erik D Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM Journal on Computing*, 2006. doi:10.1137/S0097539705447256.
- 41 Robert Endre Tarjan and Renato Fonseca F Werneck. Self-adjusting top trees. In *ACM-SIAM Symposium on Discrete algorithms (SODA)*, 2005. doi:10.5555/1070432.1070547.
- 42 Mikkel Thorup. Decremental dynamic connectivity. In *ACM-SIAM Symposium on Discrete algorithms (SODA)*, 1997.
- 43 Mikkel Thorup. Near-optimal fully-dynamic graph connectivity. In *ACM Symposium on Theory of Computing (STOC)*, 2000. doi:10.1145/335305.335345.
- 44 Christian Wulff-Nilsen. Faster deterministic fully-dynamic graph connectivity. In *Encyclopedia of Algorithms*. Springer Berlin Heidelberg, 2016. doi:10.1137/1.9781611973105.126.

Disjoint Faces in Drawings of the Complete Graph and Topological Heilbronn Problems

Alfredo Hubard ✉

LIGM, Université Gustave Eiffel, CNRS, ESIEE Paris, F-77454 Marne-la-Vallée, France

Andrew Suk ✉

Department of Mathematics, University of California San Diego, CA, USA

Abstract

Given a complete simple topological graph G , a k -face generated by G is the open bounded region enclosed by the edges of a non-self-intersecting k -cycle in G . Interestingly, there are complete simple topological graphs with the property that every odd face it generates contains the origin. In this paper, we show that every complete n -vertex simple topological graph generates at least $\Omega(n^{1/3})$ pairwise disjoint 4-faces. As an immediate corollary, every complete simple topological graph on n vertices drawn in the unit square generates a 4-face with area at most $O(n^{-1/3})$. Finally, we investigate a \mathbb{Z}_2 variant of Heilbronn's triangle problem for not necessarily simple complete topological graphs.

2012 ACM Subject Classification Mathematics of computing \rightarrow Combinatorics

Keywords and phrases Disjoint faces, simple topological graphs, topological Heilbronn problems

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.41

Funding *Alfredo Hubard*: Supported by the projects SOS ANR-17-CE40-0033 and Min Max ANR-19-CE40-0033. This work was mostly done in a year of *délegation* funded by the CNRS at the Laboratoire Solomon Lefschetz (LaSol) at the Instituto de Matematicas-UNAM-Ciudad Universitaria. *Andrew Suk*: Supported by NSF CAREER award DMS-1800746 and NSF award DMS-1952786.

Acknowledgements AH thanks Dominic Dotterer for sharing his ideas on Heilbronn's triangle problem.

1 Introduction

A *topological graph* is a graph drawn in the plane such that its vertices are represented by points and its edges are represented by non-self-intersecting arcs connecting the corresponding points. The arcs are not allowed to pass through vertices different from their endpoints, and if two edges share an interior point, then they must properly cross at that point in common. A topological graph is *simple* if every pair of its edges intersect at most once, either at a common endpoint or at a proper crossing point. A topological graph is called *plane* if there are no two crossing edges. If the edges are drawn as straight-line segments, then the graph is said to be *geometric*. Simple topological graphs have been extensively studied [1, 22, 13, 5, 8], and are sometimes referred to as simple drawings [8, 2]. In this paper, we study the crossing pattern of the faces generated by a simple topological graph.

If $\gamma \subset \mathbb{R}^2$ is a *Jordan curve* (i.e. non-self-intersecting closed curve), then by the Jordan curve theorem, $\mathbb{R}^2 \setminus \gamma$ has two connected components one of which is bounded. For any Jordan curve $\gamma \subset \mathbb{R}^2$, we refer to the bounded open region of $\mathbb{R}^2 \setminus \gamma$ given by the Jordan curve theorem as the *face inside* of γ . We refer to the *area* of γ as the area of the face inside of γ , which we denote by $\text{area}(\gamma)$.

It is known that every complete simple topological graph G of n vertices contains many non-self-intersecting k -cycles, for $k = (\log n)^{1/4 - o(1)}$ (e.g. see [13, 22, 14, 12]). A k -face generated by G is the face inside of a non-self-intersecting k -cycle in G . For simplicity, we



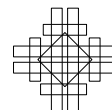
© Alfredo Hubard and Andrew Suk;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

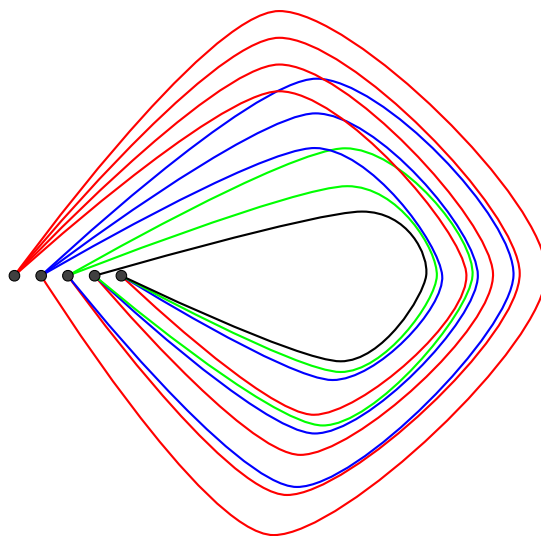
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 41; pp. 41:1–41:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** The complete twisted graph on 5 vertices.

say that a k -face is *in* G , if G generates it, and we call it an *odd* (even) face if k is odd (even). Let us remark that a k -face in G may contain other vertices and edges from G . Moreover, notice that if G is simple then every 3-cycle in G must be non-self-intersecting, so for convenience, we call 3-faces *triangles*.

Surprisingly, one cannot guarantee two disjoint 3-faces in complete simple topological graphs. In the next section, we will show that the well-known construction due to Harborth and Mengerson [7], known as the twisted graph and depicted in Figure 1, shows the following.

► **Proposition 1.** *For every $n \geq 1$, there exists a complete n -vertex simple topological graph such that every odd face it generates contains the origin.*

See Figure 1. However, the main result in this paper shows that we can guarantee many pairwise disjoint 4-faces.

► **Theorem 2.** *Every n -vertex complete simple topological graph generates at least $\Omega(n^{1/3})$ pairwise disjoint 4-faces.*

We apply the results mentioned above to a topological variant of Heilbronn's triangle problem. Over 70 years ago, Heilbronn asked: What is the smallest $h(n)$ such that any set of n points in the unit square spans a triangle whose area is at most $h(n)$? A simple triangulation argument shows that $h(n) \leq O(\frac{1}{n})$. This was improved several times by Roth and Schmidt [19, 16, 17, 18, 21], and currently, the best known upper bound is $\frac{1}{n^{8/7-o(1)}}$ due to Komlós, Pintz, and Szemerédi [9]. Heilbronn conjectured that $h(n) = \Theta(\frac{1}{n^2})$, which was later disproved by Komlós, Pintz, and Szemerédi [10], who showed that $h(n) \geq \Omega(\frac{\log n}{n^2})$. Erdős [4] conjectured that this new bound is asymptotically tight.

Here, we study Heilbronn's problem for topological graphs. A simple variant of Proposition 1 shows that one cannot guarantee a small triangle in a complete simple topological graph drawn in the unit square.

► **Proposition 3.** *For every $n \geq 1$ and $\varepsilon > 0$, there exists a complete n -vertex simple topological graph in the unit square such that every odd face it generates has area at least $1 - \varepsilon$.*

On the other hand, as an immediate corollary to Theorem 2, we have the following.

► **Corollary 4.** *Every n -vertex complete simple topological graph drawn in the unit square generates a 4-face with area at most $O(\frac{1}{n^{1/3}})$.*

In the other direction, a construction due to Lefmann [11] shows that the complete n -vertex geometric graph can be drawn in the unit square such that every 4-face has area at least $\Omega\left(\frac{\log^{1/2} n}{n^{3/2}}\right)$. It would be interesting to see if one can improve this bound for simple topological graphs. Lastly, let us mention that Heilbronn's triangle problem has been studied for k -gons, and we refer the interested reader to [11] for more results.

Our paper is organized as follows. In Section 2, we establish Propositions 1 and 3. In Section 3, we establish a lemma on finding 4-faces in complete simple topological graph. In Section 4, we use this lemma to prove Theorem 2. Finally in Section 5, we consider Heilbronn's triangle problem for (not necessarily simple) topological graphs.

2 The complete twisted graph

The *complete twisted graph* on n vertices is a complete simple topological graph with vertices labelled 1 to n which we will draw on the horizontal axis from left to right, with the property that two edges intersect if their indices are nested, i.e., edges (i, j) and (k, ℓ) , with $i < j$, $k < \ell$, intersect if and only if $i < k < \ell < j$ or $k < i < j < \ell$. See Figure 1. The complete twisted graph was introduced by Harborth and Mengerson [7] as an example of a complete simple topological graph with no subgraph that is weakly isomorphic¹ to the complete convex geometric graph on five vertices. See also [13, 22] for more applications.

► **Proposition 5.** *There exists a common point in the interior of all the odd faces generated by the complete twisted graph. Moreover, for every $\varepsilon > 0$, the complete twisted graph can be drawn in the unit square such that every odd face has area at least $1 - \varepsilon$.*

We will need the following claim, which is essentially equivalent to the Jordan curve theorem for piecewise smooth curves. In what follows, a *ray* is a straight, semi-infinite arc.

► **Lemma 6.** *Let γ be a piecewise smooth Jordan closed curve in \mathbb{R}^2 . Let \vec{v} be a direction such that every line parallel to \vec{v} intersects γ in a finite number of points. Then $\mathbb{R}^2 \setminus \gamma$ has two path-connected components, one bounded and one unbounded. Let p be a point not on γ . Then p is in the bounded region of $\mathbb{R}^2 \setminus \gamma$ if and only if the ray α emanating from p in direction \vec{v} properly crosses γ in an odd number of points.*

By Sard's lemma (see for instance [6]), given a smooth curve γ , almost every \vec{v} satisfies the assumption, and in the proof below, the directions that satisfy this assumption will be referred to as *generic*. A differentiable geometry proof of Lemma 6 in the case of γ smooth can be found in [6] (see exercise 12 in Chapter 2.5). The proof we give below is an adaptation of a well-known elementary proof for the case when γ is piecewise linear (polygons), which can be found in Chapter 5.3 of [3] for instance.

¹ Two simple topological graphs G and H are *weakly isomorphic* if there is an incidence preserving bijection between G and H such that two edges of G cross if and only if the corresponding edges in H cross as well.



■ **Figure 2** Parity of proper crossings under a perturbation of the arc, a local picture.

Proof of Lemma 6. Partition γ into a finite number of smooth arcs. If two points p and p' are connected by a segment that doesn't intersect γ , and the rays α and α' have direction \vec{v} and emanate from p and p' respectively, then they satisfy $|\gamma \cap \alpha| \pmod{2} = |\gamma \cap \alpha'| \pmod{2}$. Indeed, as we move α to α' , the only moments where the number of intersections between α and γ might change is when the tangent of γ is parallel to \vec{v} , or when we pass a singular point, in which, the ray locally leaves the two smooth arcs of γ on the same side. In these cases, the number of proper intersections between α and γ changes by two. Proper crossings can only appear or disappear in pairs when α is perturbed parallel to itself. See Figure 2. By a similar argument, changing \vec{v} for fixed p does not change the parity of the number of intersections.

In what follows, we denote by $w_2(p, \gamma)$ the parity of the number of intersections between γ and any ray α emanating from p in a generic direction. Notice that if p and q are connected by a piecewise linear arc that avoids γ , by the aforementioned argument for each straight segment of the arc, we obtain $w_2(p, \gamma) = w_2(q, \gamma)$.

Consider for every pair of points p and q in the same path-connected component of $\mathbb{R}^2 \setminus \gamma$, two rays α_p and α_q that emanate in the generic direction \vec{v} from p and q respectively. Then modify α_p and α_q by stopping each ray just before the first proper crossing it has with γ . We then extend α_p to a piecewise linear arc by following γ very closely without ever intersecting γ . Since γ is piecewise smooth, if p and q are in the same path-connected component, then the extension of α_p can be chosen so that it eventually reaches the end point of the segment α_q . This is a piecewise linear arc connecting p and q that avoids γ . Furthermore, two points that lie near γ and on opposite sides γ , have different parity so we can conclude that the two path-connected components of $\mathbb{R}^2 \setminus \gamma$ can be identified with the two possible values of $w_2(\cdot, \gamma)$. Finally, observe that for any point p sufficiently far from γ there exists a ray that doesn't intersect γ , hence $w_2(p, \gamma) = 0$, and we can conclude that a point is in the bounded component of $\mathbb{R}^2 \setminus \gamma$ if and only if $w_2(p, \gamma) = 1$. ◀

Proof of Proposition 5. Consider the complete n -vertex twisted graph such vertex v_i is placed at $(i, 0)$. See Figure 1. Let $p = (n + 1, 0)$ and consider a ray emanating out of p that passes just above the vertices. This ray intersects each edge of the twisted drawing exactly once. Hence, for any non-self intersecting odd cycle γ in G , $w_2(p, \gamma) = 1$. By Lemma 6, p lies in the face of γ . To upgrade this drawing so that each odd face has large area, we can apply a homeomorphism ϕ to the plane such that the drawing lies in the unit square, all the vertices cluster around the origin, and each face that contains $\phi(p)$ has area at least $1 - \varepsilon$. ◀

3 Finding a 4-face inside a large face

In this section, we establish several lemmas that will be used in the proof of Theorem 2. First, let us clarify some terminology. Given a *planar* graph H drawn in the plane with no crossing edges, the components of the complement of H are called the *faces* of H . Let G be a complete simple topological graph and let T be a triangle in G . We let $V(T)$ denote the set of vertices of the 3-cycle in G that generates T . We say that T is *incident* to vertex $v \in V(G)$, if $v \in V(T)$. We say that triangle T is *empty*, if there is no vertex from G that lies in T . We will repeatedly use the following lemma due to Ruiz-Vargas (see also [5]).

► **Lemma 7** ([20]). *Let G be a complete simple topological graph and H be a connected plane subgraph of G with at least two vertices. Let v be a vertex of G that is not in H , and let F be the face of H that contains v . Then there exist two edges of G emanating out of v to the boundary of F such that their interior lies complete inside of F .*

If the plane subgraph $H \subset G$ in Lemma 7 contains a single edge incident to vertex v , then by deleting this edge and applying Lemma 7 to v and the remaining plane subgraph, we obtain the following.

► **Lemma 8.** *Let G be a complete simple topological graph and H be a connected plane subgraph of G with at least two vertices. Let v be a vertex of H with degree one, and let F be the face of H whose boundary contains v . Then there exist an edge of G emanating out of v to the boundary of F such that its interior lies complete inside of F .*

We will also need the following lemma, which is a simple consequence of Lemma 7.

► **Lemma 9.** *Let G be a complete simple topological graph on four vertices, and let T be a triangle in G with a vertex $v \in V(G)$ inside of it. Then G generates a 4-face that lies inside of the triangle T .*

Lastly, we will need following key lemma, which can be considered as a generalization of Lemma 9. Given a plane graph H and a face F in H , the *size* of F , denoted by $|F|$, is the total length of the closed walk(s) in H bounding the face F . Given two vertices u, v along the boundary of F , the *distance* between u and v is the length of the shortest walk from u to v along the boundary of F .

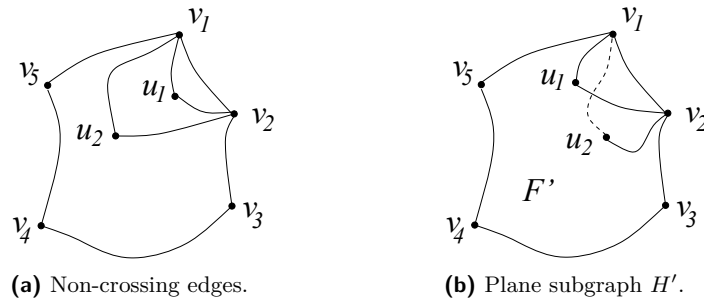
► **Lemma 10.** *Let $k \geq 5$ and G be a complete simple topological graph and H be a connected plane subgraph of G with minimum degree two. Let F be a face of H such that $|F| = k$ and F contains at least $6(k - 4)$ vertices of G in its interior. Then G generates a 4-face that lies inside of F .*

Proof. We proceed by induction on k , the size of F . For the base case $k = 5$, since H has minimum degree two, the boundary of F must be a simple 5-cycle. Let v_1, \dots, v_5 be the vertices along the boundary of F appearing in clockwise order. Let u_1, \dots, u_6 be the vertices of G in the interior of F . By applying Lemma 7 to u_i and the plane graph H , we obtain two edges emanating out of u_i to the boundary of F , whose interior lies completely inside of F . If the endpoints of these edges have distance more than one along the boundary of F , then we have generated a 4-face inside of F and we are done. Therefore, we can assume that for each u_i , the two edges emanating out of it obtained from Lemma 7 have endpoints at distance one (consecutive) along the boundary of F .

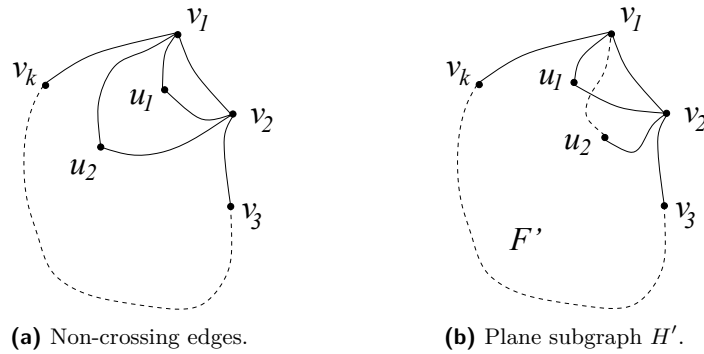
Since $|F| = 5$, by the pigeonhole principle, there are two vertices, say u_1 and u_2 , such that the two edges emanating out of u_1 and u_2 obtained from Lemma 7 go to the same two consecutive vertices, say v_1, v_2 . If these 4 edges are non-crossing, then we obtain a triangle with a vertex inside of it. See Figure 3a. By Lemma 9, we obtain a 4-face inside of F and we are done. Therefore, without loss of generality, we can assume that edges u_2v_1 and u_1v_2 cross.

Let $H' = H \cup \{u_1v_1, u_1v_2, u_2v_2\}$, and let F' be the face such that u_2 lies on the boundary of F' . See Figure 3b. Since u_2 has degree one in H' , we apply Lemma 8 to u_2 and H' to obtain an edge u_2v_i emanating out of u_2 to the boundary of F' , whose interior lies in F' .

If $v_i = v_3$, then we obtain a 4-face inside of F by following the sequence of vertices (v_3, u_2, v_1, v_2) in G . If $v_i = v_4$, then we obtain a 4-face inside of F by following the sequence vertices (v_4, u_2, v_2, v_3) in G . If $v_i = v_5$, then we obtain the 4-face inside of F by following sequence vertices (v_5, v_1, v_2, u_2) in G . Finally, if $v_i = u_1$, then by following the sequence vertices (u_2, u_1, v_1, v_2) in G , we obtain a 4-face inside of F .



■ **Figure 3** Finding a 4-face inside a 5-face.



■ **Figure 4** Finding a 4-face inside a face of size k .

For the inductive step, assume that the statement holds for all $k' < k$. Let F be a face of H such that $|F| = k$, and let $(v_1, v_2, \dots, v_k, v_1)$ be the closed walk(s) along the entire boundary of F . Set $t = 6(k - 4)$, and let u_1, \dots, u_t be vertices of G that lie in the interior of F . For each u_i , we apply Lemma 7, with respect to H , to obtain two edges emanating out of u_i to the boundary of F , such that their interior lies inside of F . The proof now falls into the following cases.

Case 1. Suppose there is a u_i such that the two edges emanating out of u_i obtained from Lemma 7 have endpoints at distance two along the boundary of F . Then we have created a 4-face inside of F and we are done.

Case 2. Suppose there is a vertex u_i such that the two edges emanating out of u_i obtained from Lemma 7 have endpoints at distance at least 3. Then these two edges emanating out of u_i partition F into two parts, F_s and F_r , such that $|F_s| = s, |F_r| = r, 5 \leq s, r \leq k - 1$ and $s + r = k + 4$. By the pigeonhole principle, G has at least $6(s - 4)$ vertices inside of F_s or $6(r - 4)$ vertices inside F_r . Indeed, otherwise the total number of vertices inside of F (including vertex u_i) is at most

$$6(s - 4) - 1 + 6(r - 4) - 1 + 1 = 6(k - 4) - 1,$$

contradiction. Hence, we can apply induction to F_s or F_r to obtain a 4-face inside of F and we are done.

Case 3. Assume for each u_i , the two edges emanating out of u_i obtained from Lemma 7 have endpoints that have distance one along the boundary of F (consecutive vertices along F). Since $t = 6(k - 4) > k$, by the pigeonhole principle, there are two vertices, say u_1 and u_2 ,

such that the two edges emanating out of u_1 and u_2 obtained from Lemma 7 go to the same two vertices, say v_1, v_2 . If these four edges are noncrossing, then we have a triangle with a vertex inside. By Lemma 9, we obtain a 4-face inside of F and we are done. See Figure 4a. Therefore, without loss of generality, we can assume that edges u_1v_2 and u_2v_1 cross.

Let $H' = H \cup \{u_1v_1, u_1v_2, u_2v_2\}$, which implies that u_2 has degree one in H' . Let F' be the face that contains u_2 on its boundary. See Figure 4b. We apply Lemma 8 to u_2 and the plane graph H' , to obtain an edge u_2v_i whose interior lies inside of F' and v_i lies on the boundary of F' . If $v_i = v_3$, then we obtain a 4-face inside of F by following the sequence of vertices (u_2, v_1, v_2, v_3) in G . If $v_i = u_1$, then we obtain a 4-face inside of F by following the sequence of vertices (u_2, u_1, v_1, v_2) in G . If $v_i = v_k$, then again, we obtain a 4-face inside of F by following the sequence of vertices (u_2, v_k, v_1, v_2) in G .

Finally, if $v_i \neq v_k, u_1, v_3$, then at least one of $u_2v_2 \cup u_2v_i$ or $u_2v_1 \cup u_2v_i$ partitions F into two parts, F_s and F_r , such that $|F_s| = s, |F_r| = r$, where $5 \leq s, r \leq k - 1$ and $s + r = k + 4$. By following the arguments in Case 2, we can apply induction on F_s or F_r to obtain a 4-face inside of F . This completes the proof. ◀

4 Pairwise disjoint 4-faces in simple drawings

In this section, we prove Theorem 2. Roughly speaking, we follow the arguments of Fulek and Ruiz-Vargas [5] by constructing a large planar subgraph $H \subset G$ using Lemma 7. Then, by combining the pigeonhole principle with Dilworth's theorem, H will contain either

1. a planar $K_{2,t}$ for t large, or
2. many nested triangles, or
3. many interior disjoint triangles.

Here, *large* and *many* means $\Omega(n^{\frac{1}{3}})$. In the first case, it is easy to find many pairwise disjoint 4-faces. In the second case, we use Lemma 10 to find them. In the last case however, the set of interior disjoint triangles may not give rise to many pairwise disjoint 4-faces, as it is possible that the triangles are empty. In order to rectify this, we carefully construct our planar subgraph H using Lemma 11 below. We now flesh out the details of the proof.

Proof of Theorem 2. Let $G = (V, E)$ be a complete n -vertex simple topological graph. We can assume that $n \geq 40$ since otherwise the statement is trivial. Notice that the edges of G divide the plane into several cells (regions), one of which is unbounded. We can assume that there is a vertex $v_0 \in V$ such that v_0 lies on the boundary of the unbounded cell. Indeed, otherwise we can project G onto a sphere, then choose an arbitrary vertex v_0 and then project G back to the plane such that v_0 lies on the boundary of the unbounded cell. Moreover, the new drawing is isomorphic to the original one as topological graphs.

Consider the topological edges emanating out from v_0 in clockwise order, and label their endpoints v_1, \dots, v_{n-1} . For convenience, we write $v_i \prec v_j$ if $i < j$. Given subsets $U, W \subset \{v_1, \dots, v_{n-1}\}$, we write $U \prec W$ if $u \prec w$ for all $u \in U$ and $w \in W$. We start by partitioning our vertex set

$$\mathcal{P} : V(G) = V_0 \cup V_1 \cup \dots \cup V_{\lfloor \frac{n-1}{5} \rfloor},$$

such that for $j < \lfloor \frac{n-1}{5} \rfloor$, we have

$$V_j = \{v_{5j+1}, v_{5j+2}, v_{5j+3}, v_{5j+4}, v_{5j+5}\},$$

and $|V_{\lfloor \frac{n-1}{5} \rfloor}| \leq 5$. Let $H \subset G$ be a plane subgraph of G , and let T, T' be two triangles in H that are incident to v_0 . We say that T and T' are *adjacent* if $V(T) = \{v_0, v_i, v_j\}$

and $V(T') = \{v_0, v_j, v_k\}$ such that $v_i \prec v_j \prec v_k$, and the edges v_0v_i, v_0v_j, v_0v_k appear consecutively in clockwise order among the edges emanating out of v_0 in H (not in G). See Figures 6c and 7a for an example.

In what follows, we will construct a plane subgraph $H \subset G$ so that, at each step, we use Lemma 7 to add at least one edge within the vertex set $\{v_1, \dots, v_{n-1}\}$. The goal at each step is to add an edge without creating any empty triangles incident to v_0 . If we are forced to create such an empty triangle, we then create another triangle incident to v_0 that is adjacent to it, so that we obtain a 4-face. We now give the details of this process.

► **Lemma 11.** *For each $i \in \{0, 1, \dots, \lfloor n/12 \rfloor\}$, there is a plane subgraph $H_i \subset G$ such that $V(H_i) = V(G)$ and H_i satisfies the following properties.*

1. H_i has at least i edges with both endpoints in the vertex set $\{v_1, \dots, v_{n-1}\}$.
2. The number of parts $V_j \in \mathcal{P}$ with the property that each vertex in V_j has degree one in H_i is at least $\lfloor (n-1)/5 \rfloor - 2i$.
3. If the vertex set $\{v_0, v_k, v_\ell\}$ induces an empty triangle T in H_i , then both vertices v_k, v_ℓ must lie in the same part $V_j \in \mathcal{P}$ and $\ell = k + 1$. Moreover, given that such an empty triangle T exists, there must be another triangle T' adjacent to T in H_i , such that $V(T') = \{v_0, v_t, v_{t'}\}$ and $v_t, v_{t'} \in V_j$.
4. If the edge v_0v_t is not in H_i , then v_t is an isolated vertex in H_i .

Proof. We start by setting H_0 as the plane subgraph of G consisting of all edges emanating out of v_0 . Clearly, H_0 satisfies the properties above. For $i < n/12$, having obtained H_i with the properties described above, we obtain H_{i+1} as follows.

Fix a part $V_j \in \mathcal{P}$ such that each vertex in V_j has degree one in H_i and $|V_j| = 5$. For simplicity, set $u_i = v_{5j+i}$, for $i \in \{1, \dots, 5\}$, which implies $V_j = \{u_1, u_2, u_3, u_4, u_5\}$. Since

$$\left\lfloor \frac{n-1}{5} \right\rfloor - 2i \geq \frac{n-1}{5} - \frac{n}{6} > 1,$$

such a part $V_j \in \mathcal{P}$ exists. Clearly, all vertices in V_j lie on the boundary of a face F in the plane graph H_i . We then apply Lemma 8 to the plane graph H_i and the vertex u_3 , and obtain edge u_3v_k , whose interior lies within F and v_k is on the boundary of F . We now consider the following cases.

Case 1. Suppose $v_k \neq u_2, u_4$. See Figure 5a. We then set $H_{i+1} = H_i \cup \{u_3v_k\}$. Clearly, H_{i+1} does not contain two crossing edges. Moreover, the number of edges in H_{i+1} within the vertex set $\{v_1, \dots, v_{n-1}\}$ is at least $i + 1$. Also, the only vertices that no longer have degree one in H_{i+1} are u_3 and v_k . Hence, the number of parts $V_\ell \in \mathcal{P}$ with the property that all vertices in V_ℓ have degree one in H_{i+1} is at least

$$\left\lfloor \frac{n-5}{2} \right\rfloor - 2i - 2 = \left\lfloor \frac{n-5}{2} \right\rfloor - 2(i+1).$$

Since $v_k \neq u_2, u_4$, no empty triangles incident to v_0 were created. Also, no edge emanating out of v_0 was deleted from H_i . Thus, H_{i+1} satisfies the conditions described above.

Case 2. Suppose $v_k = u_2$ or $v_k = u_4$. Without loss of generality, we can assume $v_k = u_4$, since otherwise a symmetric argument would follow. If there is a vertex of G inside the triangle $T = \{v_0, u_3, u_4\}$, then we set $H_{i+1} = H_i \cup \{u_3u_4\}$. By the same arguments as above, H_{i+1} satisfies the properties described above.

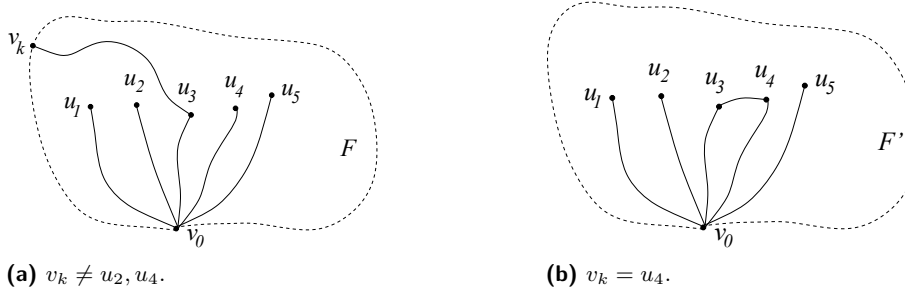


Figure 5 Cases 1 and 2 in Lemma 11.

Hence, we can assume that the triangle T , where $V(T) = \{v_0, u_3, u_4\}$, is empty in G . Set $H' = H_i \cup \{u_3u_4\}$, and let F' be the face in H' whose boundary contains u_2 . See Figure 5b. We apply Lemma 8 to H' and u_2 and obtain another edge u_2v_ℓ whose interior lies inside F' . If $v_\ell = u_3$, then we set $H_{i+1} = H_i \cup \{u_3u_4, u_2u_3\}$, which implies that the empty triangle T is adjacent to triangle T' , where $V(T') = \{v_0, u_2, u_3\}$. Clearly, H_{i+1} has at least $i + 2 > i + 1$ edges within the vertex set $\{v_1, \dots, v_{n-1}\}$. The number of parts $V_\ell \in \mathcal{P}$ with the property that all vertices in V_ℓ have degree one in H_{i+1} is at least

$$\left\lfloor \frac{n-5}{2} \right\rfloor - 2i - 1 > \left\lfloor \frac{n-5}{2} \right\rfloor - 2(i+1).$$

If $v_\ell \neq u_1, u_3$, then edge u_2v_ℓ does not create any empty triangles incident to v_0 and we set $H_{i+1} = H_i \cup \{u_2v_\ell\}$. By the same argument as above, H_{i+1} satisfies the desired properties.

Finally, let us consider the case that $v_\ell = u_1$. If the triangle T' is not empty, where $V(T') = \{v_0, u_1, u_2\}$, we set $H_{i+1} = H_i \cup \{u_1u_2\}$ and we are done by the arguments above. Therefore, we can assume that the triangle T' is also empty.

Let $H'' = (H_i \cup \{u_1u_2\}) \setminus \{u_3\}$. Let F'' be the face whose boundary contains u_4 in H'' . See Figure 6a. We apply Lemma 8 to H'' and the vertex u_4 to obtain edge u_4v_t whose interior lies inside F'' . We now examine $H_i \cup \{u_1u_2, u_3u_4, u_4v_t\}$. The proof now falls into the following cases.

Case 2.a. Suppose edge u_4v_t crosses edge v_0u_3 . If $v_t = u_5$, then $\{v_0, u_4, u_5\}$ induces a non-empty triangle in G , so we set $H_{i+1} = H_i \cup \{u_4u_5\} \setminus \{v_0v_3\}$. Then u_3 is an isolated vertex in H_{i+1} and we did not create any empty triangles incident to v_0 , and we are done. See Figure 6b. If $v_t = u_2$, then we set $H_{i+1} = H_i \cup \{u_1u_2, u_2u_4\} \setminus \{v_0u_3\}$. Then the empty triangle on $\{v_0, u_1, u_2\}$ is adjacent to the triangle on $\{v_0, u_2, u_4\}$ in H_{i+1} , u_3 is an isolated vertex, and we are done. See Figure 6c. Otherwise, if $v_t \neq u_2, u_5$, we set $H_{i+1} = (H_i \cup \{u_4v_t\}) \setminus \{v_0u_3\}$. Then u_3 is an isolated vertex, we do not create any empty triangles incident to v_0 , and we are done. See Figure 6d.

Case 2.b. Suppose edges v_0u_3 and u_4v_t do not cross. If $v_t = u_5$, then we set $H_{i+1} = H_i \cup \{u_3u_4, u_4u_5\}$, and the empty triangle on the vertex set $\{v_0, u_3, u_4\}$ is adjacent to the triangle on $\{v_0, u_4, u_5\}$, and we are done. See Figure 7a. If $v_t \neq u_5$, then we set $H_{i+1} = H_i \cup \{u_4v_t\}$. Since we do not create any empty triangles incident to v_0 , we are done. See Figure 7b. This completes the proof of the statement. ◀

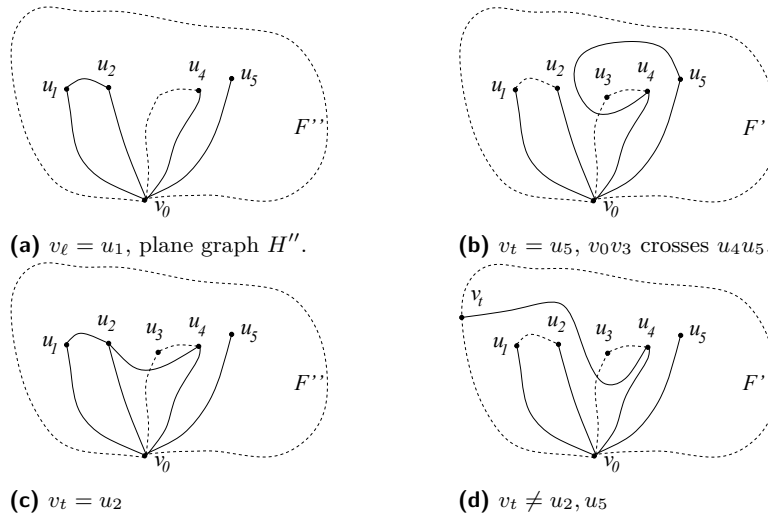


Figure 6 Cases 2.a in Lemma 11. Edge $u_4 v_t$ crosses $v_0 v_3$.

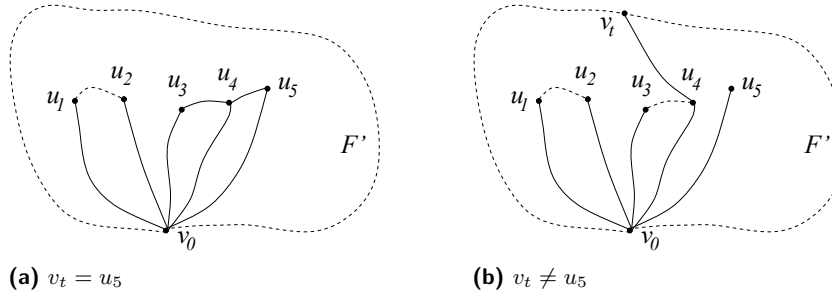


Figure 7 Case 2.b in Lemma 11. Edge $u_4 v_t$ does not cross $v_0 v_3$.

Set $H = H_{\lfloor n/12 \rfloor}$. We now will use the plane graph H and the vertices of G to find many pairwise disjoint 4-faces. If there is a vertex $v_j \in \{v_1, \dots, v_{n-1}\}$ with degree at least $n^{1/3}$ in H , then together with v_0 , we have a plane drawing of $K_{2, \lfloor n^{1/3} \rfloor}$. Indeed, recall that in H , every vertex is either connected to v_0 , or an isolated vertex. This gives rise to $\Omega(n^{1/3})$ pairwise disjoint 4-faces and we are done.

Hence, we can assume that every vertex $v_i \in \{v_1, \dots, v_{n-1}\}$ has degree at most $n^{1/3}$. Since there are at least $n/12$ edges induced on the vertex set $\{v_1, \dots, v_{n-1}\}$ in the plane graph H , there is a plane matching M on $\{v_1, \dots, v_{n-1}\}$ of size at least $n^{2/3}/16$. Notice that there is a natural partial ordering \prec^* on M . Given two edges $v_i v_j, v_k v_\ell \in M$, we write $v_k v_\ell \prec^* v_i v_j$ if $v_i \prec v_k \prec v_\ell \prec v_j$. By Dilworth's theorem, M contains either a chain or antichain of length at least $n^{1/3}/4$ with respect to the partial ordering \prec^* . The proof now falls into two cases.

Case 1. Suppose we have an antichain M' of size $n^{1/3}/4$. Let

$$M' = \{v_{\ell_1} v_{r_1}, v_{\ell_2} v_{r_2}, \dots, v_{\ell_t} v_{r_t}\},$$

where $t = n^{1/3}/4$ and $\ell_i < r_i$ for all i . Since H is a plane drawing, and every non-isolated vertex is connected to v_0 , we have

$$\{v_{\ell_1}, v_{r_1}\} \prec \{v_{\ell_2}, v_{r_2}\} \prec \dots \prec \{v_{\ell_t}, v_{r_t}\}.$$

See Figure 8a. If at least half of the edges in M' give rise to a non-empty triangle incident to v_0 , then we apply Lemma 9 to each such triangle to obtain $\Omega(n^{1/3})$ pairwise disjoint 4-faces. Hence, we can assume at least half of these triangles are empty. By construction of H , each such empty triangle has another triangle adjacent to it. Since the three edges emanating out of v_0 of two adjacent triangles must be consecutive in H (by definition), this corresponds to $\Omega(n^{1/3})$ pairwise disjoint 4-faces. See Figure 8b.

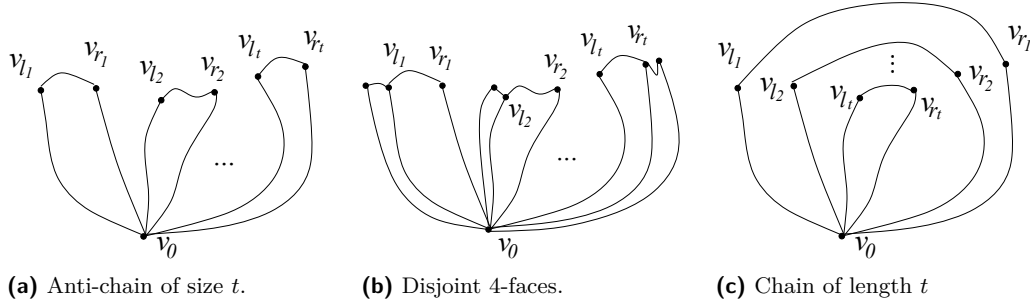


Figure 8 Large antichain and chain.

Case 2. Suppose we have a chain $M' \subset M$ of size $n^{1/3}/4$. Hence,

$$M' = \{v_{l_1}v_{r_1}, v_{l_2}v_{r_2}, \dots, v_{l_t}v_{r_t}\},$$

where $t = n^{1/3}/4$ and we have

$$v_{l_t}v_{r_t} \prec^* \dots \prec^* v_{l_2}v_{r_2} \prec^* v_{l_1}v_{r_1}.$$

See Figure 8c. Set $M'' \subset M'$ such that $M'' = \{v_{l_{7j}}v_{r_{7j}}\}_j$. Hence, $|M''| \geq \Omega(n^{1/3})$. Let us consider edges $v_{l_7}v_{r_7}$ and $v_{l_{14}}v_{r_{14}}$ from M'' , and the region F enclosed by the six edges.

$$v_{l_7}v_{r_7}, v_{l_{14}}v_{r_{14}}, v_0v_{l_7}, v_0v_{l_{14}}, v_0v_{r_7}, v_0v_{r_{14}}.$$

See Figure 9. Let H' be the plane subgraph on the vertex set $\{v_0, v_{l_{14}}, v_{r_{14}}, v_{r_7}, v_{l_7}\}$ and the six edges listed above. By construction of M'' , we know that there are at least 12 vertices of $V(G)$ inside F . Since $|F| = 6$, we can apply Lemma 10 to find a 4-face inside of F . By repeating this argument for each consecutive pair of edges in the matching M'' with respect to the partial order \prec^* , we obtain $\Omega(n^{1/3})$ pairwise disjoint 4-faces. ◀

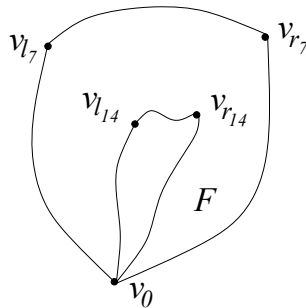


Figure 9 Face F of size 6.

5 \mathbb{Z}_2 -cycles in topological graphs

Now we pass to a variant of Heilbronn's triangle problem for not necessarily simple topological graphs. Specifically, if γ is piecewise smooth closed curve with transverse self intersections, then one can consider Lemma 6, from Section 2, as a definition of the \mathbb{Z}_2 -inside of γ . That is, p is in the interior of γ if any arc with one endpoint at p and the other outside a large disk containing γ , intersects γ an odd number of times at proper crossings.

Does every complete topological graph drawn inside the unit square contain a cycle whose \mathbb{Z}_2 -inside has small area? More generally, we will consider this question for the group of \mathbb{Z}_2 -cycles instead of graph cycles. The result of this section is a negative answer to this question. Using a simple probabilistic construction, we show that there are complete topological graphs in the unit square in which every cycle has constant area.

5.1 Chain complexes

Let us recall the basic objects of cellular homology, refer to [15] for a gentle introduction. If X is a cell complex, for each i , the group of i -th chain group, denoted by $C_i(X, \mathbb{Z}_2)$ is the group of formal linear combinations of the i -dimensional cells. An element of $C_i(X, \mathbb{Z}_2)$ has the form $\sum_{\sigma \in F_i(X)} a_\sigma \sigma$, where σ is an element of F_i , the set of i -dimensional cells, and a_σ is an element of \mathbb{Z}_2 , the field with two elements. The boundary operator is a linear map $\partial C_i(X, \mathbb{Z}_2) \rightarrow C_{i-1}(X, \mathbb{Z}_2)$, which can be succinctly described using a pair of basis, one for $C_i(X, \mathbb{Z}_2)$ and one for $C_{i-1}(X, \mathbb{Z}_2)$ which have an element for each cell, then the boundary map of a cell σ is the linear combinations of the $(i-1)$ -cells that are incident to σ . The kernel of the boundary operator is the group of cycles $Z_i(X, \mathbb{Z}_2)$ and its image is the group of boundaries $B_{i-1}(X, \mathbb{Z}_2)$, the quotient group $Z_i(X, \mathbb{Z}_2)/B_{i-1}(X, \mathbb{Z}_2)$ is the i -th cellular homology group $H_i(X, \mathbb{Z}_2)$. In the following we will use that a two dimensional disk has trivial homology. This is the case because homology is invariant under homotopy equivalences and a disk can be contracted to a point which can be modeled with a cell complex that has no higher dimensional cells.

Consider K_n as a simplicial complex, in other words, $F_1(K_n)$ is the set of edges and $F_0(K_n)$ is the set of vertices of the complete graph.

In this case the boundary $\partial: C_1(K_n, \mathbb{Z}_2) \rightarrow C_0(K_n, \mathbb{Z}_2)$ is defined as follows: if $e = (i, j)$ is an edge, then the chain $1e$ is mapped to $\partial(e) = 1i + 1j$. The kernel of ∂ is the group of 1-cycles of K_n , $Z_1(K_n) := \ker \partial$. Elements in $Z_1(K_n)$ can be identified with (possibly disjoint) graphs in which every vertex has even degree.

Let us consider the planar graph induced by G by introducing a vertex at every intersection between two edges, and let \hat{G} be the cell decomposition of the smallest closed topological disk that contains G . More precisely, every intersection between edges of G is a vertex of \hat{G} (including the vertices of G). Two consecutive intersections along an edge of G share an edge in \hat{G} . The regions of $\mathbb{R}^2 \setminus G$ are the 2 dimensional cells of \hat{G} . Consider the chain groups $C_i(\hat{G}, \mathbb{Z}_2)$, and observe that for $i = 0, 1$ there exists linear maps $f_i: C_i(K_n, \mathbb{Z}_2) \rightarrow C_i(\hat{G}, \mathbb{Z}_2)$. For example, for a given edge $e \in E(K_n)$, $f_1(e)$ is the linear combination of the edges in \hat{G} that support the arc representing e , and similarly, for the vertices.

It is not hard to see that this chain map induces a well defined map between cycle groups, $f_1: Z_1(K_n) \rightarrow Z_1(\hat{G})$. Now, since the homology group $H_1(\hat{G}, \mathbb{Z}_2)$ is trivial, for any cycle $z \in Z_1(\hat{G})$ there exists a 2 chain $c \in C_2(\hat{G}, \mathbb{Z}_2)$ such that $\partial(c) = z$. On the other hand, if some other chain $c' \neq c$ satisfied $\partial c' = z$, then $\partial(c + c') = 0$, hence $c + c'$ would be a two dimensional cycle, but since there are no 3 dimensional faces, this would imply that the homology group $H_2(\hat{G}, \mathbb{Z}_2) \neq 0$, which is absurd. So there exists a unique chain c such that $\partial c = z$, and the interior of its support corresponds to the set of points $\{p \in \mathbb{R}^2 : w_2(p, z) = 1\}$.



■ **Figure 10** A possible edge in the construction of proposition 12 before re-scaling and perturbing.

5.2 A topological graph without \mathbb{Z}_2 -cycles of small area

► **Proposition 12.** *There exists a drawing of the complete graph inside $[0, 1]^2$ such that the \mathbb{Z}_2 -inside of every \mathbb{Z}_2 -cycle of the complete graph has area at least $\frac{1}{4}$.*

We begin describing a random construction. Consider a rectangle of size $m \times 1$, with corners at $\{(0, 0), (0, 1), (m, 0), (m, 1)\}$ where m will be a large number with respect to n that we will define later on. We perform the area analysis for this drawing, but notice that by applying the linear transformation $(x, y) \rightarrow (\frac{x}{m}, y)$, we can transform it back to the unit square.

We place all the points in general position on a small neighbourhood of the lower corner $(0,0)$ of the rectangle. The drawing will be random and at the end it will be perturbed by an arbitrary small amount so that it is in general position. To refer to this small perturbation we use the word “near” in the description below. Notice that one could perturb each edge so that it stays piecewise linear or one could smooth each edge, as long as areas of cycles do not change too much and every intersection is a proper crossing (in the language of differentiable topology this corresponds to the curves being transverse and in PL topology to general position).

Each edge will go all the way to near $(m, 0)$ and come back near $(0, 0)$. Choose two vertices i, j , the edge $e = (i, j)$ will be represented by an arc that begins at the vertex i and is a concatenation of almost vertical and almost horizontal arcs. More precisely, for each $k \in \{0, 1, 2, \dots, m - 1\}$ assume that we have constructed a path $\alpha_{ij}(k)$ that begins at i (near $(0, 0)$) and ends at (k, Y_k) with $Y_k \in \{0, 1\}$, let Y_{k+1} be a Bernoulli random variable with probability $\frac{1}{2}$, and extend the arc $\alpha_{ij}(k)$ by concatenating it with the segment $\{(k + t, y_k) : t \in [0, 1]\}$ if $Y_{k+1} = Y_k$, and by the concatenation of the segments $\{(k, t) : t \in [0, 1]\}$ followed by $\{(k + t, Y_{k+1}) : t \in [0, 1]\}$ if $Y_{k+1} \neq Y_k$.

When we reach $x = m$, if $y = 1$, we concatenate it to $(m, 0)$. In both cases $y = 0, 1$, we end the arc by concatenating all the way back to the vertex j near $(0, 0)$ with a long near horizontal arc close to the x -axis. Finally, we perturb what we have constructed a very small amount so that the intersections between any two such edges is a finite set of points where they cross properly, and we re-scale the x -axis so that the whole picture is contained in the unit square.

Proof of Proposition 12. We work with the rectangle and make some observations about the re-scaling and perturbing at the end of the proof. Using the random construction described above, to compute the expected area of a cycle z , consider a point p in the interior of the rectangle, say that p has coordinates $(k + \frac{1}{2}, \frac{1}{2})$, and consider the horizontal segment of a fixed edge e that joins (k, Y_k) with $(k + 1, Y_{k+1})$. The vertical ray emanating up from p , intersects this edge with probability $\frac{1}{2}$. Conditioned on all the other edges of a cycle z containing e , the square $\{(k, 0), (k, 1), (k + 1, 1), (k + 1, 0)\}$ is \mathbb{Z}_2 -inside z with probability $\frac{1}{2}$ and \mathbb{Z}_2 -outside z with probability $\frac{1}{2}$. This implies that the area of every cycle is a sum of m independent Bernoulli random variables with probability $\frac{1}{2}$, so $\mathbb{E}[\text{area}(z)] \geq \frac{m}{2}$, and Chernoff bound yields:

$$\Pr(\text{area}(z) < \frac{m}{3}) \leq e^{-\frac{m}{128}}.$$

There are exactly $2^{\binom{n-1}{2}} - 1$ non-zero elements in $Z_1(K_n)$, while the areas of two different cycles z and z' that share some edge are dependent random variables, if we let $m = 64n^2$, the union bound yields

$$\Pr(\exists z \in Z_1(K_n), \text{area}(z) < \frac{m}{3}) \leq 2^{-\frac{n}{2}}$$

Since this probability is strictly smaller than 1, there exists some drawing such that the area of every cycle is at least $\frac{m}{3}$, which after perturbing and re-scaling by $\frac{1}{m}$, corresponds to all cycles having area at least $\frac{1}{3} - \epsilon$, for any given $\epsilon > 0$. ◀

► **Remark 13.** If we only cared about graph cycles, i.e. connected subgraphs of K_n in which each vertex has degree two, then it is enough to take $m = O(n \log n)$.

► **Remark 14.** There is nothing special about $\frac{1}{4}$ or about $\frac{1}{3}$, at the cost of making m larger, we can force all cycles to have area at least $\frac{1}{2} - \epsilon$ for any $\epsilon > 0$. It is easy to see that for any complete topological graph there exists $z \in Z_1(G)$ with $\text{area}(z) \leq \frac{1}{2}$.

In the aforementioned construction, for two fixed edges e, e' , and a fixed integer i , there is a constant probability that e and e' cross near the vertical line at $\{(i, x) : x \in \mathbb{R}^1\}$, hence the expected number of crossings is $\Omega(n^6)$

► **Problem 15.** For a fixed k , is there a function $\epsilon_k(n)$ with $\epsilon_k(n) \rightarrow 0$ when $n \rightarrow \infty$, such that for every drawing of K_n in which every pair of edges intersect at most k times, we can find a cycle of area at most $\epsilon_k(n)$?

References

- 1 O. Aichholzer, A. García, J. Tejel, B. Vogtenhuber, and A. Weinberger. Twisted ways to find plane structures in simple drawings of complete graphs. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry, SoCG 2022, June 7-10, 2022, Berlin, Germany*, volume 224 of *LIPICs*, pages 5:1–5:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- 2 A. Arroyo, M. Derka, and I. Parada. Extending simple drawings. In *27th International Symposium on Graph Drawing and Network Visualization (GD)*, volume 11904 of *Lecture Notes in Computer Science*, pages 230–243. Springer, 2019. doi:10.1007/978-3-030-35802-0_18.
- 3 R. Courant and H. Robbins. *What is Mathematics?* Oxford University Press, 1941.
- 4 P. Erdos. Problems and results in combinatorial geometry. *Annals of the New York Academy of Sciences*, 440(1):1–11, 1985.
- 5 R. Fulek and A.J. Ruiz-Vargas. Topological graphs: empty triangles and disjoint matchings. In *Proceedings of the 29th Annual Symposium on Computational Geometry (SoCG'13)*, pages 259–266, 2013.
- 6 V. Gullemin and A. Pollock. *Differentiable Geometry*. American Mathematical Society. Chelsea Publishing, 1974.
- 7 H. Harborth and I. Mengersen. Drawings of the complete graph with maximum number of crossings. *Congr. Numer.*, 88:225–228, 1992.
- 8 M. Hoffmann, CH. Liu, M.M. Reddy, and C.D. Tóth. Simple topological drawings of k -planar graphs. In *Graph Drawing and Network Visualization*, volume 12590 of *Lecture Notes in Computer Science*. Springer, 2020.
- 9 J. Komlós, J. Pintz, and E. Szemerédi. On Heilbronn’s triangle problem. *J. London Math. Soc.*, 24:385–396, 1981.
- 10 J. Komlós, J. Pintz, and E. Szemerédi. A lower bound for Heilbronn’s problem. *J. London Math. Soc.*, 25:13–24, 1982.
- 11 H. Lefmann. Distributions of points in the unit square and large k -gons. *European J. Combin.*, 29:946–965, 2008.

- 12 A. Marcus and G. Tardos. Intersection reverse sequences and geometric applications. *J. Comb. Theory, Ser. A*, 113:675–691, 2006.
- 13 J. Pach, J. Solymosi, and G. Tóth. Unavoidable configurations in complete topological graphs. *Disc. Comput. Geom.*, 30:311–320, 2003.
- 14 R. Pinchasi and R. Radoičić. On the number of edges in geometric graphs with no self-intersecting cycle of length 4. *Towards a Theory of Geometric Graphs, Contemporary Mathematics (J. Pach, ed.)*, 342, 2004.
- 15 V. V. Prasolov. *Elements of homology theory*. American Mathematical Society, Vol. 81, 2007.
- 16 K. F. Roth. On a problem of Heilbronn. *J. London Math. Soc.*, 26:198–204, 1951.
- 17 K. F. Roth. On a problem of Heilbronn II. *J. London Math. Soc.*, 25:193–212, 1972.
- 18 K. F. Roth. On a problem of Heilbronn III. *J. London Math. Soc.*, 25:543–549, 1972.
- 19 K. F. Roth. Developments in Heilbronn’s triangle problem. *Adv. Math.*, 22:364–385, 1976.
- 20 A.J. Ruiz-Vargas. Empty triangles in complete topological graphs. *Discrete Comput. Geom.*, 53:703–712, 2015.
- 21 W. M. Schmid. On a problem of Heilbronn. *J. London Math. Soc.*, 2:545–550, 1971/72.
- 22 A. Suk and J. Zeng. Unavoidable patterns in complete simple topological graphs. *Graph Drawing and Network Visualization, GD 2022*, 2022.

On the Width of Complicated JSJ Decompositions

Kristóf Huszár   

Univ Lyon, CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR5668, France

Jonathan Spreer   

School of Mathematics and Statistics F07, The University of Sydney, NSW 2006 Australia

Abstract

Motivated by the algorithmic study of 3-dimensional manifolds, we explore the structural relationship between the JSJ decomposition of a given 3-manifold and its triangulations. Building on work of Bachman, Derby-Talbot and Sedgwick, we show that a “sufficiently complicated” JSJ decomposition of a 3-manifold enforces a “complicated structure” for all of its triangulations. More concretely, we show that, under certain conditions, the treewidth (resp. pathwidth) of the graph that captures the incidences between the pieces of the JSJ decomposition of an irreducible, closed, orientable 3-manifold \mathcal{M} yields a linear lower bound on its treewidth $\text{tw}(\mathcal{M})$ (resp. pathwidth $\text{pw}(\mathcal{M})$), defined as the smallest treewidth (resp. pathwidth) of the dual graph of any triangulation of \mathcal{M} .

We present several applications of this result. We give the first example of an infinite family of bounded-treewidth 3-manifolds with unbounded pathwidth. We construct Haken 3-manifolds with arbitrarily large treewidth – previously the existence of such 3-manifolds was only known in the non-Haken case. We also show that the problem of providing a constant-factor approximation for the treewidth (resp. pathwidth) of bounded-degree graphs efficiently reduces to computing a constant-factor approximation for the treewidth (resp. pathwidth) of 3-manifolds.

2012 ACM Subject Classification Mathematics of computing → Geometric topology; Theory of computation → Problems, reductions and completeness; Theory of computation → Fixed parameter tractability

Keywords and phrases computational 3-manifold topology, fixed-parameter tractability, generalized Heegaard splittings, JSJ decompositions, pathwidth, treewidth, triangulations

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.42

Related Version *Full Version:* <http://arxiv.org/abs/2303.06789> [25]

Funding *Kristóf Huszár:* Supported by the French National Research Agency through the 3IA Côte d’Azur (ANR-19-P3IA-0002), AlgoKnot (ANR-20-CE48-0007), GrR (ANR-18-CE40-0032), and TWIN-WIDTH (ANR-21-CE48-0014) projects. A previous version of this manuscript was written during the author’s visit at Institut Henri Poincaré (IHP). The author acknowledges the hospitality of IHP (UAR 839 CNRS–Sorbonne Université) and LabEx CARMIN (ANR-10-LABX-59-01).

Jonathan Spreer: Supported by the Australian Research Council’s Discovery funding scheme (project no. DP220102588).

Acknowledgements We are grateful to Arnaud de Mesmay and to Clément Maria for their interest in our work and for inspiring discussions. We thank the anonymous referees for several suggestions to improve this article.

1 Introduction

Manifolds in geometric topology are often studied through the following two-step process. Given a piecewise linear d -dimensional manifold \mathcal{M} , first find a “suitable” triangulation of it, i.e., a decomposition of \mathcal{M} into d -simplices with “good” combinatorial properties. Then apply algorithms on this triangulation to reveal topological information about \mathcal{M} .



© Kristóf Huszár and Jonathan Spreer;

licensed under Creative Commons License CC-BY 4.0

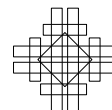
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 42; pp. 42:1–42:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



The work presented in this article is motivated by this process in dimension $d = 3$. Here every manifold can be triangulated [41] and questions about them typically admit algorithmic solutions [32, 36, 50].¹ At the same time, the feasibility of a particular computation can greatly depend on structural properties of the triangulation in use. Over the past decade, this phenomenon was recognized and exploited in various settings, leading to *fixed-parameter tractable* (FPT) algorithms for several problems in low-dimensional topology, some of which are even known to be **NP**-hard [13, 14, 15, 16, 17].² Although these algorithms have exponential running time in the worst case, for input triangulations with *dual graph* of bounded *treewidth* they always terminate in polynomial (in most cases, linear) time.³ Moreover, some of them have implementations that are highly effective in practice, providing useful tools for researchers in low-dimensional topology [11, 12].

The theoretical efficiency of the aforementioned FPT algorithms crucially depends on the assumption that the dual graph of the input triangulation has small treewidth. To understand their scope, it is thus instructive to consider the *treewidth* $\text{tw}(\mathcal{M})$ of a compact 3-manifold \mathcal{M} , defined as the smallest treewidth of the dual graph of any triangulation of \mathcal{M} . Indeed, the relation between the treewidth and other quantities associated with 3-manifolds has recently been investigated in various contexts [22, 23, 24, 26, 39]. For instance, in [26] together with Wagner we have shown that the treewidth of a *non-Haken* 3-manifold is always bounded below in terms of its *Heegaard genus*. Combined with earlier work of Agol [1] – who constructed non-Haken 3-manifolds with arbitrary large Heegaard genus – this implies the existence of 3-manifolds with arbitrary large treewidth. Despite the fact that, asymptotically, most triangulations of most 3-manifolds must have dual graph of large treewidth [26, Appendix A], this collection described by Agol has remained, to this date, the only known family of 3-manifolds with arbitrary large treewidth.

The main result. In this work we unravel new structural links between the triangulations of a given 3-manifold and its *JSJ decomposition* [28, 29, 30]. Employing the machinery of *generalized Heegaard splittings* [46], the results developed in [26], and building on the work of Bachman, Derby-Talbot and Sedgwick [4, 5], we show that, under suitable conditions, the dual graph of any triangulation of a given 3-manifold \mathcal{M} inherits structural properties from the *decomposition graph* that encodes the incidences between the pieces of the JSJ decomposition of \mathcal{M} . More precisely, in Section 4 we prove the following theorem.

► **Theorem 1** (Width inheritance). *For any closed, orientable and irreducible 3-manifold \mathcal{M} with sufficiently complicated⁴ torus gluings in its JSJ decomposition \mathcal{D} , the treewidth and pathwidth of \mathcal{M} and that of the decomposition graph $\Gamma(\mathcal{D})$ of \mathcal{D} satisfy*

$$\text{tw}(\Gamma(\mathcal{D})) \leq 18 \cdot (\text{tw}(\mathcal{M}) + 1) \quad (1) \quad \text{and} \quad \text{pw}(\Gamma(\mathcal{D})) \leq 4 \cdot (3 \text{pw}(\mathcal{M}) + 1). \quad (2)$$

An algorithmic construction. Much work in 3-dimensional topology has been devoted to the study of 3-manifolds constructed by pasting together simpler pieces along their boundary surfaces via “sufficiently complicated” gluing maps, and to understand how different

¹ In higher dimensions none of these statements is true in general. See, e.g., [38], [40] or [42, Section 7].

² See [7] for an FPT algorithm checking tightness of (weak) pseudomanifolds in arbitrary dimensions.

³ The running times are given in terms of the *size* of the input triangulation, i.e., its number of tetrahedra.

⁴ The notion of “sufficiently complicated” under which we establish Theorem 1 is discussed in Section 4.

decompositions of the same 3-manifold interact under various conditions, see, e.g., [3, 4, 5, 6, 33, 37, 44, 49]. Theorem 1 allows us to leverage these results to construct 3-manifolds, where we have tight control over the treewidth and pathwidth of their triangulations [22]. By combining Theorem 1 and [3, Theorem 5.4] (cf. [5, Appendix]), in Section 5 we prove

- **Theorem 2.** *There is a polynomial-time algorithm that, given an n -node graph G with maximum node-degree Δ , produces a triangulation \mathcal{T}_G of a closed 3-manifold \mathcal{M}_G , such that*
1. *the triangulation \mathcal{T}_G contains $O_\Delta(\text{pw}(G) \cdot n)$ tetrahedra,⁵*
 2. *the JSJ decomposition \mathcal{D} of \mathcal{M}_G satisfies $\Gamma(\mathcal{D}) = G$, and*
 3. *there exist universal constants $c, c' > 0$, such that*
 - a. *$(c/\Delta) \text{tw}(\mathcal{M}_G) \leq \text{tw}(G) \leq 18 \cdot (\text{tw}(\mathcal{M}_G) + 1)$, and*
 - b. *$(c'/\Delta) \text{pw}(\mathcal{M}_G) \leq \text{pw}(G) \leq 4 \cdot (3 \text{pw}(\mathcal{M}_G) + 1)$.*

Applications. In Section 6 we present several applications of Theorem 2. First, we construct a family of bounded-treewidth 3-manifolds with unbounded pathwidth (Corollary 12). Second, we exhibit Haken 3-manifolds with arbitrary large treewidth (Corollary 13). To our knowledge, no such families of 3-manifolds had been known before. Third, we show that the problem of providing a constant-factor approximation for the treewidth (resp. pathwidth) of bounded-degree graphs reduces in polynomial time to computing a constant-factor approximation for the treewidth (resp. pathwidth) of 3-manifolds (Corollary 14). This reduction, together with previous results [43, 51, 52], suggests that this problem may be computationally hard.

Outline of the proof of Theorem 1. We now give a preview of the proof of our main result. As the arguments for showing (1) and (2) are analogous, we only sketch the proof of (1). To show that $\text{tw}(\Gamma(\mathcal{D})) \leq 18(\text{tw}(\mathcal{M}) + 1)$, we start with a triangulation \mathcal{T} of \mathcal{M} whose dual graph has minimal treewidth, i.e., $\text{tw}(\Gamma(\mathcal{T})) = \text{tw}(\mathcal{M})$. Following [26, Section 6], we construct from \mathcal{T} a generalized Heegaard splitting \mathcal{H} of \mathcal{M} , together with a *sweep-out* $\Sigma = \{\Sigma_x : x \in H\}$ along a tree H , such that the genus of any level surface Σ_x is at most $18 \cdot (\text{tw}(\Gamma(\mathcal{T})) + 1)$. If \mathcal{H} is not already *strongly irreducible*, we repeatedly perform *weak reductions* until we get a strongly irreducible generalized Heegaard splitting \mathcal{H}' with associated *sweep-out* $\Sigma' = \{\Sigma'_x : x \in H\}$ along the same tree H . Crucially, weak reductions do not increase the genera of level surfaces [46, Section 5.2], thus $18 \cdot (\text{tw}(\Gamma(\mathcal{T})) + 1)$ is still an upper bound on those in Σ' . Now, by the assumption of the JSJ decomposition of \mathcal{M} being “sufficiently complicated,” each *JSJ torus* can be isotoped in \mathcal{M} to coincide with a connected component of some *thin level* of \mathcal{H}' . This implies that, after isotopy, each level set Σ'_x is incident to at most $18 \cdot (\text{tw}(\Gamma(\mathcal{T})) + 1) + 1$ JSJ pieces of \mathcal{M} . Sweeping along H , we can construct a *tree decomposition* of $\Gamma(\mathcal{D})$ where each *bag* contains at most $18 \cdot (\text{tw}(\Gamma(\mathcal{T})) + 1) + 1$ nodes of $\Gamma(\mathcal{D})$. Hence $\text{tw}(\Gamma(\mathcal{D})) \leq 18 \cdot (\text{tw}(\Gamma(\mathcal{T})) + 1) = 18 \cdot (\text{tw}(\mathcal{M}) + 1)$. ◀

Organization of the paper. In Section 2 we review the necessary background on graphs and 3-manifolds. Section 3 contains a primer on generalized Heegaard splittings, which provides us with the indispensable machinery for proving our main result (Theorem 1) in Section 4. In Section 5 we describe the algorithmic construction of 3-manifolds that “inherit” their combinatorial width from that of their JSJ decomposition graph (Theorem 2). Then, in Section 6 we present the aforementioned applications of this construction (Corollaries 12–14). We conclude the paper with a discussion and some open questions in Section 7.

⁵ Similar to the standard *big-O notation*, $O_\Delta(x)$ means “a quantity bounded above by x times a constant depending on Δ .” To ensure that 3a is satisfied, but not necessarily 3b, $O_\Delta(\text{tw}(G) \cdot n)$ tetrahedra suffice.

2 Preliminaries

2.1 Graphs

A *(multi)graph* $G = (V, E)$ is a finite set $V = V(G)$ of *nodes*⁶ together with a multiset $E = E(G)$ of unordered pairs of not necessarily distinct nodes, called *arcs*. The *degree* d_v of a node $v \in V$ equals the number of arcs containing it, where loop arcs are counted twice. G is *k-regular*, if $d_v = k$ for all $v \in V$. A *tree* is a connected graph with n nodes and $n - 1$ arcs. A *tree decomposition* of a graph $G = (V, E)$ is a pair $(\mathcal{X} = \{B_i : i \in I\}, T = (I, F))$ with *bags* $B_i \subseteq V$ and a tree $T = (I, F)$, such that **1.** $\bigcup_{i \in I} B_i = V$ (node coverage), **2.** for all arcs $\{u, v\} \in E$, there exists $i \in I$ such that $\{u, v\} \subseteq B_i$ (arc coverage), and **3.** for all $v \in V$, $T_v = \{i \in I : v \in B_i\}$ spans a connected sub-tree of T (sub-tree property).

The *width* of a tree decomposition equals $\max_{i \in I} |B_i| - 1$, and the *treewidth* $\text{tw}(G)$ is the smallest width of any tree decomposition of G . Replacing all occurrences of “tree” with “path” in the definition of treewidth yields the notion of *pathwidth* $\text{pw}(G)$. We have $\text{tw}(G) \leq \text{pw}(G)$.

2.2 Manifolds

A *d-dimensional manifold* is a topological space \mathcal{M} , where each point $x \in \mathcal{M}$ has a neighborhood homeomorphic to \mathbb{R}^d or to the closed upper half-space $\{(x_1, \dots, x_d) \in \mathbb{R}^d : x_d \geq 0\}$. The latter type of points of \mathcal{M} constitute the *boundary* $\partial\mathcal{M}$ of \mathcal{M} . A compact manifold is said to be *closed* if it has an empty boundary. We consider manifolds up to homeomorphism (“continuous deformations”) and write $\mathcal{M}_1 \cong \mathcal{M}_2$ for homeomorphic manifolds \mathcal{M}_1 and \mathcal{M}_2 .

3-Manifolds and surfaces

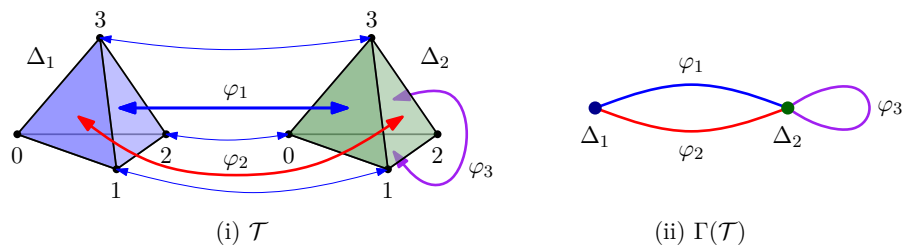
The main objects of study in this paper are 3-dimensional manifolds, or *3-manifolds* for short. Here we give a brief introduction to 3-manifolds tailored to our purposes. We refer the reader to [48] (and the references therein) for more details. All 3-manifolds and surfaces encountered in this article are compact and orientable. We let \mathbb{S}_g denote the closed, connected, orientable surface of genus g . We also refer to the d -dimensional torus and sphere as \mathbb{T}^d and \mathbb{S}^d , respectively (hence $\mathbb{S}_0 = \mathbb{S}^2$ and $\mathbb{S}_1 = \mathbb{T}^2$). The genus $g(\mathcal{S})$ of a (not necessarily connected) surface \mathcal{S} is defined to be the sum of the genera of its connected components.

Triangulations and the treewidth of 3-manifolds. A *triangulation* \mathcal{T} of a given 3-manifold \mathcal{M} is a finite collection of abstract tetrahedra glued together along pairs of their triangular faces, such that the resulting space is homeomorphic to \mathcal{M} . Unpaired triangles comprise a triangulation of the boundary of \mathcal{M} . Note that the face gluings may also identify several tetrahedral edges (or vertices) in a single *edge* (or *vertex*) of \mathcal{T} . Every compact 3-manifold admits a triangulation [41] (cf. [9]). Given a triangulation \mathcal{T} , its *dual graph* $\Gamma(\mathcal{T})$ is the multigraph whose nodes correspond to the tetrahedra in \mathcal{T} , and arcs to face gluings (Figure 1).

For a compact 3-manifold \mathcal{M} , its *treewidth* $\text{tw}(\mathcal{M})$ (resp. *pathwidth* $\text{pw}(\mathcal{M})$) is defined as the smallest treewidth (resp. pathwidth) of the dual graph of any triangulation of \mathcal{M} .

Incompressible surfaces and essential disks. Given a 3-manifold \mathcal{M} , a surface $\mathcal{S} \subset \mathcal{M}$ is said to be *properly embedded* in \mathcal{M} if $\partial\mathcal{S} \subset \partial\mathcal{M}$ and $(\mathcal{S} \setminus \partial\mathcal{S}) \subset (\mathcal{M} \setminus \partial\mathcal{M})$. Given a properly embedded surface $\mathcal{S} \subset \mathcal{M}$, an embedded disk $D \subset \mathcal{M}$ with $\text{int}(D) \cap \mathcal{S} = \emptyset$ and $\partial D \subset \mathcal{S}$ a

⁶ Throughout this paper we use the terms *edge* and *vertex* to refer to an edge or vertex in a 3-manifold triangulation, whereas the terms *arc* and *node* denote an edge or vertex in a graph, respectively.



■ **Figure 1** (i) Example of a triangulation \mathcal{T} with two tetrahedra Δ_1 and Δ_2 , and three face gluing maps φ_1 , φ_2 and φ_3 . The map φ_1 is specified to be $\Delta_1(123) \longleftrightarrow \Delta_2(103)$. (ii) The dual graph $\Gamma(\mathcal{T})$ of the triangulation \mathcal{T} . Reproduced from [23, Figure 1].

curve not bounding a disk on \mathcal{S} is called a *compressing disk*. If such a disk exists, \mathcal{S} is said to be *compressible* in \mathcal{M} , otherwise – and if \mathcal{S} is not a 2-sphere – it is called *incompressible*. A 3-manifold \mathcal{M} is said to be *irreducible*, if every embedded 2-sphere bounds a 3-ball in \mathcal{M} . A disk $D \subset \mathcal{M}$ properly embedded in a 3-manifold \mathcal{M} is called *inessential* if it cuts off a 3-ball from \mathcal{M} , otherwise D is called *essential*. A compact, orientable, irreducible 3-manifold is called *Haken* if it contains an orientable, properly embedded, incompressible surface, and otherwise is referred to as *non-Haken*.

Heegaard splittings of closed 3-manifolds. A *handlebody* \mathcal{H} is a connected 3-manifold homeomorphic to a thickened graph. The *genus* $g(\mathcal{H})$ of \mathcal{H} is defined as the genus of its boundary surface $\partial\mathcal{H}$. A *Heegaard splitting* of a closed, orientable 3-manifold \mathcal{M} is a decomposition $\mathcal{M} = \mathcal{H} \cup_{\mathcal{S}} \mathcal{H}'$ where \mathcal{H} and \mathcal{H}' are homeomorphic handlebodies with $\mathcal{H} \cup \mathcal{H}' = \mathcal{M}$ and $\mathcal{H} \cap \mathcal{H}' = \partial\mathcal{H} = \partial\mathcal{H}' = \mathcal{S}$ called the *splitting surface*. Introduced in [21], the *Heegaard genus* $\mathfrak{g}(\mathcal{M})$ of \mathcal{M} is the smallest genus $g(\mathcal{S})$ over all Heegaard splittings of \mathcal{M} .

The JSJ decomposition. A central result by Jaco–Shalen [28, 29] and Johannson [30] asserts that every closed, irreducible and orientable 3-manifold \mathcal{M} admits a collection \mathbf{T} of pairwise disjoint embedded, incompressible tori, where each piece of the complement $\mathcal{M} \setminus \mathbf{T}$ is either Seifert fibered⁷ or atoroidal⁸. A minimal such collection of tori is unique up to isotopy and gives rise to the so-called *JSJ decomposition* (or *torus decomposition*) of \mathcal{M} [20, Theorem 1.9]. We refer to this collection of incompressible tori as the *JSJ tori* of \mathcal{M} . The graph with nodes the JSJ pieces, and an arc for each JSJ torus (with endpoints the two nodes corresponding to its two adjacent pieces) is called the *dual graph* $\Gamma(\mathcal{D})$ of the *JSJ decomposition* \mathcal{D} of \mathcal{M} .

3 Generalized Heegaard Splittings

A Heegaard splitting of a closed 3-manifold is a decomposition into two identical handlebodies along an embedded surface. Introduced by Scharlemann and Thompson [47], a generalized Heegaard splitting of a compact 3-manifold \mathcal{M} (possibly with boundary) is a decomposition of \mathcal{M} into several pairs of compression bodies along a family of embedded surfaces, subject to certain rules. Following [46, Chapters 2 and 5],⁹ here we give an overview of this framework.

⁷ See [48, Section 3.7] or [20, p. 18] for an introduction to Seifert fibered spaces.

⁸ An irreducible 3-manifold \mathcal{M} is called *atoroidal* if every incompressible torus in \mathcal{M} is boundary-parallel.

⁹ For an open-access version, see [45, Sections 3.1 and 4].

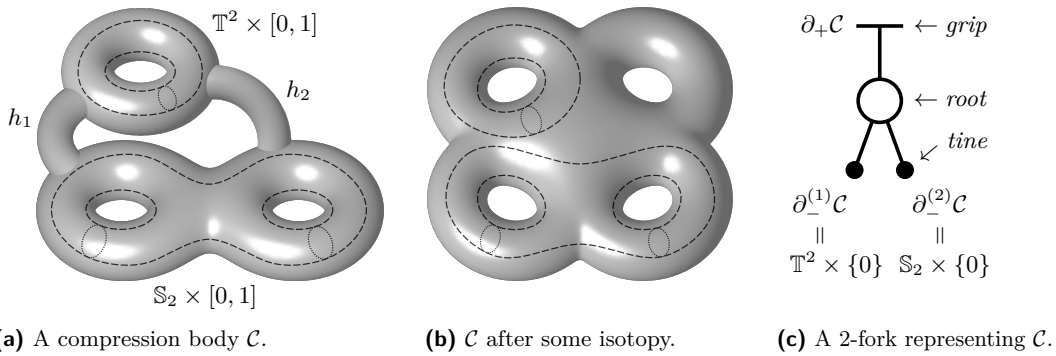
3.1 Compression bodies and forks

A *compression body* \mathcal{C} is a 3-manifold with boundary obtained by the following procedure.

1. Consider the thickening $\mathcal{S} \times [0, 1]$ of a closed, orientable, possibly disconnected surface \mathcal{S} ,
2. optionally attach some 1-*handles*, each being of the form $D \times [0, 1]$ (thickened edge, where the disk D is the cross-section), to $\mathcal{S} \times \{1\}$ along $D \times \{0\} \cup D \times \{1\}$, and
3. optionally fill in some collection of 2-sphere components of $\mathcal{S} \times \{0\}$ with 3-balls.

We call $\partial_- \mathcal{C} = \mathcal{S} \times \{0\} \setminus \{\text{filled-in 2-sphere components}\}$ the *lower boundary* of \mathcal{C} and $\partial_+ \mathcal{C} = \partial \mathcal{C} \setminus \partial_- \mathcal{C}$ its *upper boundary*. By construction, $g(\partial_+ \mathcal{C}) \geq g(\partial_- \mathcal{C})$. Note that, if $\partial_- \mathcal{C}$ is empty, then \mathcal{C} is a handlebody. We allow compression bodies to be disconnected.

A *fork*, more precisely an *n-fork*, F is a tree with $n + 2$ nodes $V(F) = \{\rho, \gamma, \tau_1, \dots, \tau_n\}$, where ρ , called the *root*, is of degree $n + 1$, and the other nodes are leaves. One of them, denoted γ , is called the *grip*, and the remaining leaves τ_1, \dots, τ_n are called *tines*. A fork can be regarded as an abstraction of a connected compression body \mathcal{C} , where the grip corresponds to $\partial_+ \mathcal{C}$ and each tine corresponds to a connected component of $\partial_- \mathcal{C}$, see, e.g., Figure 2.



■ **Figure 2** The compression body \mathcal{C} is obtained by first thickening the disconnected surface $\mathbb{T}^2 \cup \mathbb{S}_2$ to $(\mathbb{T}^2 \cup \mathbb{S}_2) \times [0, 1]$, then attaching two 1-handles (h_1 and h_2) between $\mathbb{T}^2 \times \{1\}$ and $\mathbb{S}_2 \times \{1\}$. For the lower boundary of \mathcal{C} we have $\partial_- \mathcal{C} = (\mathbb{T}^2 \cup \mathbb{S}_2) \times \{0\}$, and for its upper boundary $\partial_+ \mathcal{C} = \partial \mathcal{C} \setminus \partial_- \mathcal{C} \cong \mathbb{S}_4$.

Non-faithful forks. In certain situations (notably, in the proof of Theorem 1, cf. Section 4) it is useful to also take a simplified view on a generalized Heegaard splitting. To that end, one may represent several compression bodies by a single *non-faithful* fork, where the grip and the tines may correspond to collections of boundary components. To distinguish faithful forks from non-faithful ones, we color the roots of the latter with gray, see Figure 3.



■ **Figure 3** Two faithful forks bundled into a non-faithful fork. The colors show the grouping of the tines.

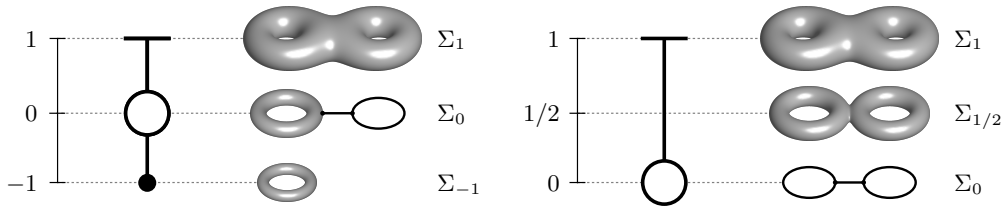
■ **Figure 4** Spines of a compression body \mathcal{C} and of a handlebody \mathcal{H} .

Spines and sweep-outs of compression bodies. A graph Γ embedded in a compression body \mathcal{C} is called a *spine* of \mathcal{C} , if every node of Γ that is incident to $\partial_- \mathcal{C}$ is of degree one, and $\mathcal{C} \setminus (\Gamma \cup \partial_- \mathcal{C}) \cong \partial_+ \mathcal{C} \times (0, 1]$, see Figure 4. Assume first that $\partial_- \mathcal{C} \neq \emptyset$. A *sweep-out* of \mathcal{C}

along an interval, say $[-1, 1]$, is a continuous map $f: \mathcal{C} \rightarrow [-1, 1]$, such that $f^{-1}(\pm 1) = \partial_{\pm}\mathcal{C}$,

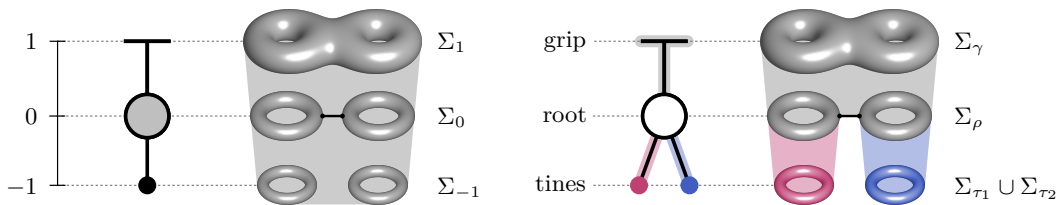
$$f^{-1}(t) \cong \begin{cases} \partial_+\mathcal{C}, & \text{if } t \in (0, 1), \\ \Gamma \cup \partial_-\mathcal{C}, & \text{if } t = 0, \text{ and} \\ \partial_-\mathcal{C}, & \text{if } t \in (-1, 0). \end{cases} \quad (3)$$

Writing $\Sigma_t = f^{-1}(t)$, we get a 1-parameter family of surfaces (except for Σ_0) “sweeping through” \mathcal{C} . For handlebodies, the definition of a sweep-out is similar, but it “ends at 0” with the spine: $f^{-1}(1) = \partial\mathcal{H}$, $f^{-1}(t) \cong \partial\mathcal{H}$ if $t \in (0, 1)$, and $f^{-1}(0) = \Gamma$, see Figure 5.



■ **Figure 5** Sweep-outs of the compression body \mathcal{C} and of the handlebody \mathcal{H} shown in Figure 4.

► **Remark 3 (Sweep-out along a fork).** It is straightforward to adapt the definition of sweep-out in a way that, instead of an interval, we sweep a compression body \mathcal{C} along a (faithful or non-faithful) fork F . Such a sweep-out is a continuous map $f: \mathcal{C} \rightarrow \|F\|$ (where $\|F\|$ denotes a geometric realization of F) that satisfies very similar requirements to those in (3), however, the components of the lower boundary $\partial_-\mathcal{C}$ may appear in different level sets, see Figure 6.



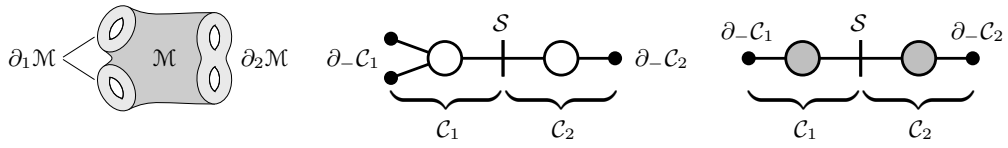
■ **Figure 6** Sweep-out of a compression body along $[-1, 1]$ (left) and along its faithful fork (right).

3.2 Generalized Heegaard splittings and fork complexes

Heegaard splittings revisited. Let \mathcal{M} be a 3-manifold, and $\{\partial_1\mathcal{M}, \partial_2\mathcal{M}\}$ be a partition of the components of $\partial\mathcal{M}$. A *Heegaard splitting* of $(\mathcal{M}, \partial_1\mathcal{M}, \partial_2\mathcal{M})$ is a triplet $(\mathcal{C}_1, \mathcal{C}_2, \mathcal{S})$, where \mathcal{C}_1 and \mathcal{C}_2 are compression bodies with $\mathcal{C}_1 \cup \mathcal{C}_2 = \mathcal{M}$, $\mathcal{C}_1 \cap \mathcal{C}_2 = \partial_+\mathcal{C}_1 = \partial_+\mathcal{C}_2 = \mathcal{S}$, $\partial_-\mathcal{C}_1 = \partial_1\mathcal{M}$, and $\partial_-\mathcal{C}_2 = \partial_2\mathcal{M}$. The *genus* of the Heegaard splitting $(\mathcal{C}_1, \mathcal{C}_2, \mathcal{S})$ is the genus $g(\mathcal{S})$ of the *splitting surface* \mathcal{S} . The *Heegaard genus* $\mathfrak{g}(\mathcal{M})$ of \mathcal{M} is the smallest genus of any Heegaard splitting of $(\mathcal{M}, \partial_1\mathcal{M}, \partial_2\mathcal{M})$, taken over all partitions $\{\partial_1\mathcal{M}, \partial_2\mathcal{M}\}$ of $\partial\mathcal{M}$.

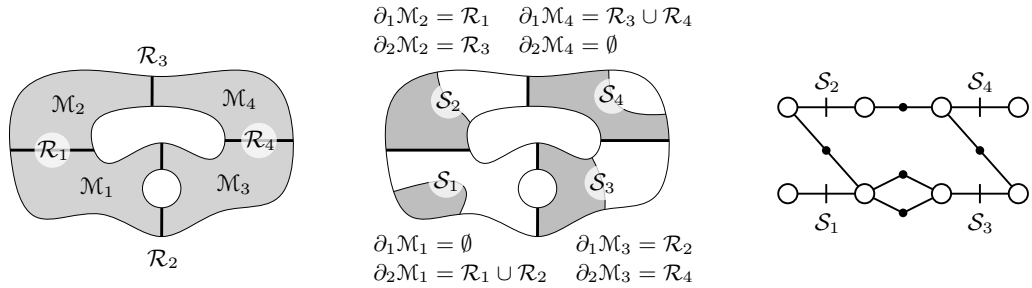
► **Proposition 4** ([46, Theorem 2.1.11], cf. [23, Appendix B]). *For any partition $\{\partial_1\mathcal{M}, \partial_2\mathcal{M}\}$ of the boundary components of \mathcal{M} , there exists a Heegaard splitting of $(\mathcal{M}, \partial_1\mathcal{M}, \partial_2\mathcal{M})$.*

Generalized Heegaard splittings. A *generalized Heegaard splitting* \mathcal{H} of a 3-manifold \mathcal{M} consists of **1.** a decomposition $\mathcal{M} = \bigcup_{i \in I} \mathcal{M}_i$ into submanifolds $\mathcal{M}_i \subseteq \mathcal{M}$ intersecting along closed surfaces (Figure 8a), **2.** for each $i \in I$ a partition $\{\partial_1\mathcal{M}_i, \partial_2\mathcal{M}_i\}$ of the components



■ **Figure 7** Schematic of a 3-manifold M with a partition of its boundary components (left). Faithful (center) and non-faithful (right) *fork complexes* representing a Heegaard splitting of $(M, \partial_1 M, \partial_2 M)$.

of ∂M_i that together satisfy an *acyclicity condition*: there is an ordering, i.e., a bijection $\ell: I \rightarrow \{1, \dots, |I|\}$, such that, for each $i \in I$ the components of $\partial_1 M_i$ (resp. $\partial_2 M_i$) not belonging to ∂M are only incident to submanifolds M_j with $\ell(j) < \ell(i)$ (resp. $\ell(j) > \ell(i)$), and **3.** a choice of a Heegaard splitting $(C_1^{(i)}, C_2^{(i)}, S_i)$ for each $(M_i, \partial_1 M_i, \partial_2 M_i)$. Such a choice of *splitting surfaces* S_i ($i \in I$) is said to be *compatible* with ℓ , see, e.g., Figure 8b.



(a) A decomposition of M into four submanifolds M_1, \dots, M_4 intersecting along (possibly disconnected) closed surfaces R_i .

(b) An admissible choice of splitting surfaces S_i for the M_i that is compatible with the trivial ordering $\ell(i) \mapsto i$.

(c) The faithful fork complex that represents the generalized Heegaard splitting shown in the center (Figure 8b).

■ **Figure 8** Schematics of a generalized Heegaard splitting (based on figures from [23, Section 2.4]).

Just as compression bodies can be represented by forks, (generalized) Heegaard splittings can be visualized via *fork complexes*, see Figures 7 and 8c (cf. [46, Section 5.1] for details).

Sweep-outs of 3-manifolds. A generalized Heegaard splitting \mathcal{H} of a 3-manifold M induces a *sweep-out* $f: M \rightarrow \|\mathcal{F}\|$ of M along any fork complex \mathcal{F} that represents \mathcal{H} (here $\|\mathcal{F}\|$ denotes a drawing, i.e., a *geometric realization* of the abstract fork complex \mathcal{F}) by concatenating the corresponding sweep-outs of the compression bodies that comprise \mathcal{H} (cf. Remark 3). We also refer to a sweep-out $f: M \rightarrow \|\mathcal{F}\|$ by the ensemble $\Sigma = \{\Sigma_x : x \in \|\mathcal{F}\|\}$ of its *level sets*, where $\Sigma_x = f^{-1}(x)$.

The width of a generalized Heegaard splitting. For a generalized Heegaard splitting \mathcal{H} , the surfaces S_i ($i \in I$) are also called the *thick levels*, and the lower boundaries $\partial_- C_1^{(i)}, \partial_- C_2^{(i)}$ are called the *thin levels* of \mathcal{H} . The *width* $w(\mathcal{H})$ of \mathcal{H} is the sequence obtained by taking a non-increasing ordering of the multiset $\{g(S_i) : i \in I\}$ of the genera of the thick levels.

A generalized Heegaard splitting \mathcal{H} of a 3-manifold M for which $w(\mathcal{H})$ is minimal with respect to the lexicographic order ($<$) among all splittings of M is said to be in *thin position*.

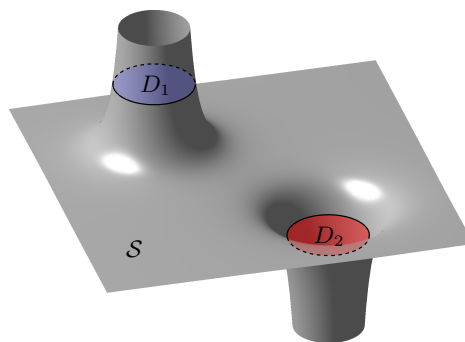
3.3 Weak reductions

A Heegaard splitting $(\mathcal{C}_1, \mathcal{C}_2, \mathcal{S})$ of a connected 3-manifold is said to be *weakly reducible* [18], if there are essential disks $D_i \subset \mathcal{C}_i$ ($i = 1, 2$)¹⁰ with $\partial D_1 \cap \partial D_2 = \emptyset$, see Figure 9. In this case we also say that the splitting surface \mathcal{S} is weakly reducible. A generalized Heegaard splitting \mathcal{H} is *weakly reducible*, if at least one of its splitting surfaces is weakly reducible; otherwise \mathcal{H} is called *strongly irreducible*. Every 3-manifold possesses a strongly irreducible generalized Heegaard splitting and this fact can be exploited in various contexts (e.g., in the proof of Theorem 1). The usefulness of such splittings is mainly due to the following seminal result.

► **Theorem 5** ([46, Lemma 5.2.4], [47, Rule 5]). *Let \mathcal{H} be a strongly irreducible generalized Heegaard splitting. Then every connected component of every thin level of \mathcal{H} is incompressible.*

Given a weakly reducible generalized Heegaard splitting \mathcal{H} with a weakly reducible splitting surface¹¹ \mathcal{S} and essential disks D_1 and D_2 as above, one can execute a *weak reduction* at \mathcal{S} . This modification amounts to performing particular cut-and-paste operations on \mathcal{S} guided by D_1 and D_2 , and decomposes each of the two compression bodies adjacent to \mathcal{S} into a pair of compression bodies. Importantly, this operation results in another generalized Heegaard splitting \mathcal{H}' of the same 3-manifold with $w(\mathcal{H}') < w(\mathcal{H})$.

For an example, we refer to the [25, Appendix A.1], and for further details to [46, Proposition 5.2.3] (notably, Figures 5.8–5.13 therein, but also Lemma 5.2.2, Figures 5.6 and 5.7, and Proposition 5.2.4), including an exhaustive list of instances of weak reductions.¹²



■ **Figure 9** Local picture of a portion of a weakly reducible splitting surface \mathcal{S} .

4 The Main Result

In this section we prove Theorem 1. The inequalities (1) and (2) are deduced in the same way, thus we only show the proof of (1) in detail, and then explain how it can be adapted to that of (2). First, we specify what we mean by a “sufficiently complicated” JSJ decomposition.

► **Definition 6.** *Given $\delta > 0$, the JSJ decomposition of an irreducible 3-manifold \mathcal{M} is δ -complicated, if any incompressible or strongly irreducible Heegaard surface $\mathcal{S} \subset \mathcal{M}$ with genus $g(\mathcal{S}) \leq \delta$ can be isotoped to be simultaneously disjoint from all the JSJ tori of \mathcal{M} .*

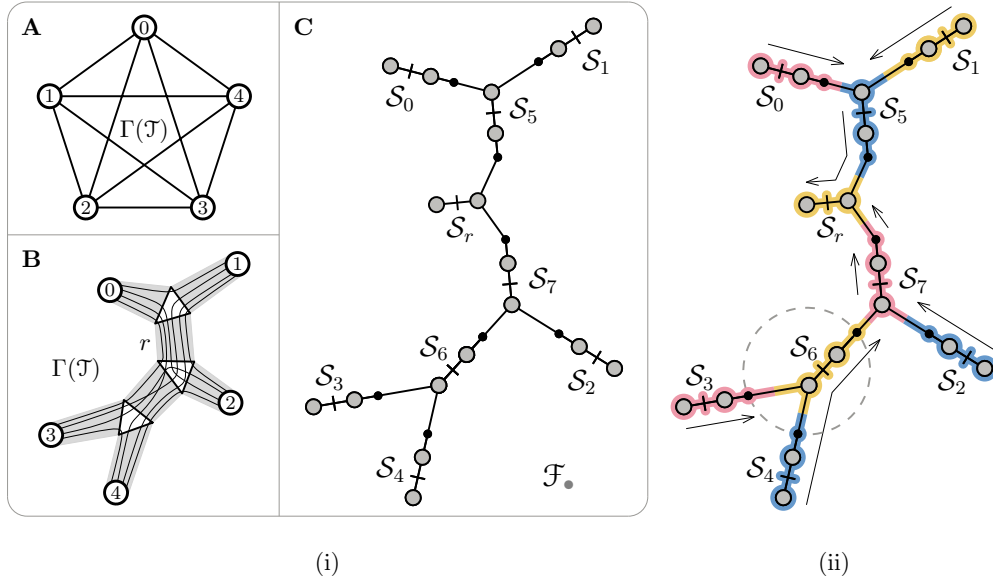
¹⁰The assumption that D_i is essential in the compression body \mathcal{C}_i implies that $\partial D_i \subset \partial_+ \mathcal{C}_i = \mathcal{S}$ ($i = 1, 2$).

¹¹For future reference we remind the reader that splitting surfaces, also called thick levels, correspond to grips in the faithful fork complex that represents the generalized Heegaard splitting \mathcal{H} .

¹²In the open-access version [45] these are Proposition 4.2.3 and Figures 87–92 (as well as Lemma 4.2.2, Figures 85 and 86, and Lemma 4.2.4).

Proof of inequality (1). Our goal is to prove that $\text{tw}(\Gamma(\mathcal{D})) \leq 18(\text{tw}(\mathcal{M}) + 1)$, where $\Gamma(\mathcal{D})$ is the dual graph of the δ -complicated JSJ decomposition of the irreducible 3-manifold \mathcal{M} . To this end, we set $\delta = 18(\text{tw}(\mathcal{M}) + 1)$ and fix a triangulation \mathcal{T} of \mathcal{M} , whose dual graph $\Gamma(\mathcal{T})$ has minimal treewidth, i.e., $\text{tw}(\Gamma(\mathcal{T})) = \text{tw}(\mathcal{M})$. Now, (1) is established in four stages.

1. Setup. By invoking the construction in [26, Section 6], from \mathcal{T} we obtain a generalized Heegaard splitting \mathcal{H} of \mathcal{M} together with a sweep-out $f: \mathcal{M} \rightarrow \|\mathcal{F}_\bullet\|$ along a non-faithful fork complex \mathcal{F}_\bullet representing \mathcal{H} . By construction, \mathcal{F}_\bullet is a tree with all of its nodes having degree one or three (Figure 10), moreover all non-degenerate level surfaces $\Sigma_x = f^{-1}(x)$ have genus bounded above by $18(\text{tw}(\mathcal{M}) + 1)$. Let \mathcal{F}_\circ be the faithful fork complex representing \mathcal{H} . Note that \mathcal{F}_\circ is obtained from \mathcal{F}_\bullet by replacing every non-faithful fork $F \in \mathcal{F}_\bullet$ with the collection \mathcal{C}_F of faithful forks that accurately represents the (possibly disconnected) compression body \mathcal{C} corresponding to F . The inverse operation, i.e., for each $F \in \mathcal{F}_\bullet$ bundling all faithful forks in \mathcal{C}_F into F (see Figure 3), induces a projection map $\pi: \|\mathcal{F}_\circ\| \rightarrow \|\mathcal{F}_\bullet\|$ between the drawings (cf. Figures 11(i)–(ii)). Note that every fork, tine, or grip in \mathcal{F}_\bullet corresponds to a collection of the corresponding items in \mathcal{F}_\circ and so the projection map is well-defined. \triangleleft



■ **Figure 10** (i) **A.** The dual graph $\Gamma(\mathcal{T})$ of some triangulation \mathcal{T} of a 3-manifold \mathcal{M} . **B.** Low-congestion routing of $\Gamma(\mathcal{T})$ along a host tree with marked root arc r . **C.** Drawing of a non-faithful fork complex \mathcal{F}_\bullet that represents the generalized Heegaard splitting of \mathcal{M} induced by the routing of $\Gamma(\mathcal{T})$. The genera of all (possibly disconnected) thick levels \mathcal{S}_i is bounded above by $18(\text{tw}(\mathcal{M}) + 1)$. (ii) Color-coded segmentation of $\|\mathcal{F}_\bullet\|$ in preparation for the next stage of the proof (cf. Figure 11).

2. Weak reductions. In case \mathcal{H} is weakly reducible, we repeatedly perform weak reductions until we obtain a strongly irreducible generalized Heegaard splitting \mathcal{H}' of \mathcal{M} . Since weak reductions always decrease the width of a generalized Heegaard splitting, this process terminates after finitely many iterations. Throughout, we maintain that the drawings of the associated faithful fork complexes follow $\|\mathcal{F}_\bullet\|$. Let \mathcal{F}'_\circ be the faithful fork complex representing the final splitting \mathcal{H}' , $f': \mathcal{M} \rightarrow \|\mathcal{F}'_\circ\|$ be the sweep-out of \mathcal{M} induced by \mathcal{H}' , and $\pi': \|\mathcal{F}'_\circ\| \rightarrow \|\mathcal{F}_\bullet\|$ be the associated projection map (Figure 11(iii)). Due to the nature of weak reductions, $18(\text{tw}(\mathcal{M}) + 1)$ is still an upper bound on the genus of any (possibly

disconnected) level surface of $\pi' \circ f' : \mathcal{M} \rightarrow \|\mathcal{F}_\bullet\|$. As \mathcal{M} is irreducible, we may assume that no component of such a level surface is homeomorphic to a sphere (cf. [34, p. 337]). It follows that the number of components of any level set of $\pi' \circ f'$ is at most $18(\text{tw}(\mathcal{M}) + 1)$. \triangleleft

3. Perturbation and isotopy. We now apply a level-preserving perturbation $\pi' \rightsquigarrow \pi''$ on π' , after which each thin level of \mathcal{H}' lies in different level sets of $\pi'' \circ f' : \mathcal{M} \rightarrow \|\mathcal{F}_\bullet\|$ (Figure 11(iv)). Since \mathcal{H}' is strongly irreducible, its thick levels are strongly irreducible (by definition), and its thin levels are incompressible (by Theorem 5). Moreover, all these surfaces have genera at most $18(\text{tw}(\mathcal{M}) + 1)$. Hence, as the JSJ decomposition of \mathcal{M} is assumed to be δ -complicated with $\delta = 18(\text{tw}(\mathcal{M}) + 1)$, we may isotope all JSJ tori to be disjoint from all the thick and thin levels of \mathcal{H}' . Then, by invoking [4, Corollary 4.5] we can isotope each JSJ torus of \mathcal{M} to coincide with a component of some thin level of \mathcal{H}' (Figure 11(v)). As a consequence, every compression body of the splitting \mathcal{H}' is contained in a unique JSJ piece of \mathcal{M} . \triangleleft

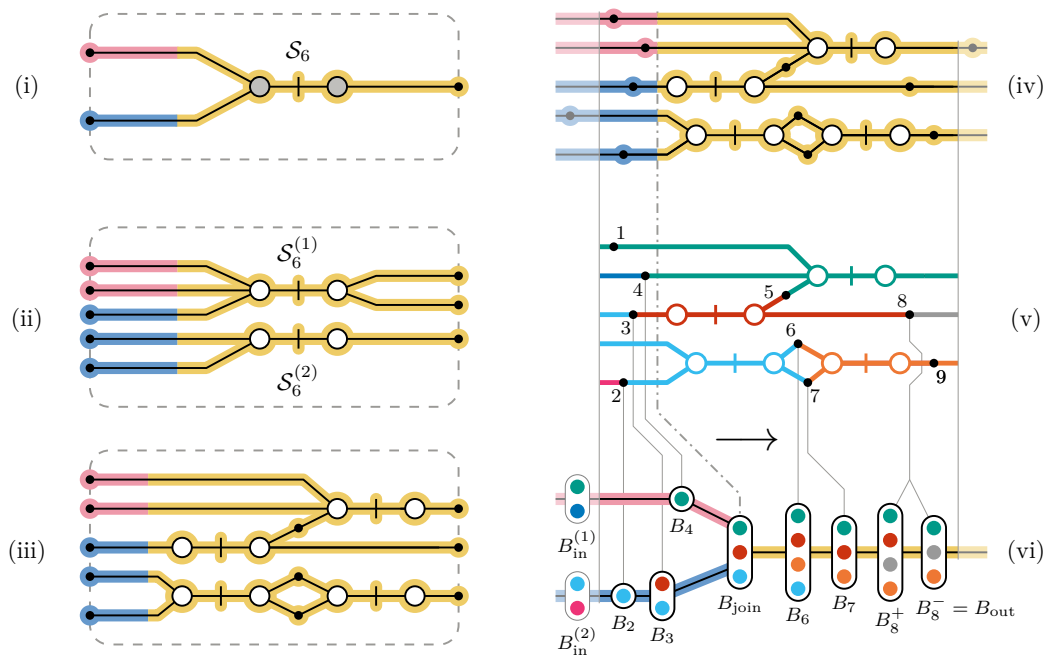


Figure 11 Overview of the proof of inequality (1). The figure shows the circled area in Figure 10(ii). Stage 1: Construction of initial fork complex (i) and split into faithful fork complex (ii). Stage 2: Weak reductions (iii), see [46, Proposition 5.2.3, Figures 5.8–5.13] for a complete list and their effects on the underlying fork complexes. Stage 3: perturbations and isotopy (iv). Stage 4: Construction of tree decomposition (v) and (vi).

4. The tree decomposition of $\Gamma(\mathcal{D})$. First note that every level set $(\pi'' \circ f')^{-1}(x)$ is incident to at most $18(\text{tw}(\mathcal{M}) + 1) + 1 = 18 \text{tw}(\mathcal{M}) + 19$ JSJ pieces of \mathcal{M} . The “plus one” appears, because if $(\pi'' \circ f')^{-1}(x)$ contains a JSJ torus, then this torus is incident to two JSJ pieces of \mathcal{M} . Also note that, because of the perturbation performed in the previous stage, each level set $(\pi'' \circ f')^{-1}(x)$ can contain at most one JSJ torus of \mathcal{M} .

We now construct a tree decomposition (\mathcal{X}, T) of $\Gamma(\mathcal{D})$ of width $18(\text{tw}(\mathcal{M}) + 1)$. Eventually, T will be a subdivision of \mathcal{F}_\bullet (which is a tree) with nodes corresponding to the bags in \mathcal{X} , which we now describe. By [26, Section 6 (p. 86)], each leaf l of \mathcal{F}_\bullet corresponds to a spine of a handlebody \mathcal{H}_l . We define a bag that contains the unique node of $\Gamma(\mathcal{D})$ associated with

the JSJ piece containing \mathcal{H}_l . As we sweep through $\|\mathcal{F}_\bullet\|$ (cf. the arrows on Figure 10(2)), whenever we pass through a point $x \in \|\mathcal{F}_\bullet\|$ such that the level set $(\pi'')^{-1}(x)$ contains a tine of $\|\mathcal{F}'_\bullet\|$, one of four possible events may occur (illustrated in Figure 11(v)–(vi)): **1.** A new JSJ piece appears. In this case we take a copy of the previous bag, and add the corresponding node of $\Gamma(\mathcal{D})$ into the bag. **2.** A JSJ piece disappears. Then we delete the corresponding node of $\Gamma(\mathcal{D})$ from a copy of the previous bag. **3.** Both previous kinds of events happen simultaneously. In this case we introduce two new bags. The first to introduce the new JSJ piece, the second to delete the old one. **4.** If neither a new JSJ piece is introduced, nor an old one is left behind, we do nothing. Whenever we arrive at a merging point in the sweep-out (i.e., a degree-three node of $\|\mathcal{F}_\bullet\|$), we introduce a new bag, which is the union of the two previous bags. Note that the two previous bags do not necessarily need to be disjoint. \triangleleft

It remains to verify that (\mathcal{X}, T) is, indeed, a tree decomposition of $\Gamma(\mathcal{D})$ of width at most $18 \text{tw}(\mathcal{M}) + 18$. *Node coverage:* Every node of $\Gamma(\mathcal{D})$ must be considered at least once, since we sweep through the entirety of $\|\mathcal{F}_\bullet\|$. *Arc coverage:* we must ensure that all pairs of nodes of $\Gamma(\mathcal{D})$ with their JSJ pieces meeting at a tine are contained in some bag. This is always the case for **1**, because a new JSJ piece appears while all other JSJ pieces are still in the bag. It is also the case for **2**, because a JSJ piece disappears but the previous bag contained all the pieces. In **3** a new JSJ piece appears and at the same time another JSJ piece disappears. However, in this case we first introduce the new piece, thus making sure that adjacent JSJ pieces always occur in at least one bag. *Sub-tree property:* This follows from the fact that a JSJ piece, once removed from all bags, must be contained in the part of $\|\mathcal{F}_\bullet\|$ that was already swept. Now, every JSJ piece incident to a given level set $(\pi'' \circ f')^{-1}(x)$ of the sweep-out must contribute a positive number to its genus. Hence, it follows that every bag can contain at most $18 \text{tw}(\mathcal{M}) + 19$ elements (with equality only possible where a tine simultaneously introduces and forgets a JSJ piece). This proves inequality (1). \blacktriangleleft

Proof of inequality (2). We start with the results from [26, Section 5] yielding a fork-complex \mathcal{F}_\bullet whose underlying space $\|\mathcal{F}_\bullet\|$ is a path, and the genus of the level sets of the associated sweep-out is bounded above by $4(3 \text{pw}(\mathcal{M}) + 1)$. Setting $\delta = 4(3 \text{pw}(\mathcal{M}) + 1)$, the remainder of the proof is analogous with the proof of inequality (1). \blacktriangleleft

5 An algorithmic construction

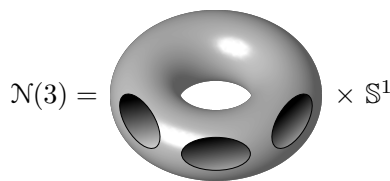
Here we establish Theorem 2 that paves the way to the applications in Section 6. In what follows, Δ denotes an arbitrary, but fixed, positive integer. Let $G = (V, E)$ be a graph with $|V| = n$ and maximum degree Δ . Theorem 2 asserts that, in $\text{poly}(n)$ time one can construct a triangulation \mathcal{T}_G of a closed, irreducible 3-manifold \mathcal{M}_G , such that the dual graph $\Gamma(\mathcal{D})$ of its JSJ decomposition \mathcal{D} equals G , moreover, the pathwidth (resp. treewidth) of G determines the pathwidth (resp. treewidth) of \mathcal{M}_G up to a constant factor.

Our proof of Theorem 2 rests on a synthesis of work by Lackenby [35] and by Bachman, Derby-Talbot and Sedgwick [5]. In [35, Section 3] it is shown that the homeomorphism problem for closed 3-manifolds is at least as hard as the graph isomorphism problem. The proof relies on a simple construction that, given a graph G , produces a closed, orientable, triangulated 3-manifold whose JSJ decomposition \mathcal{D} satisfies $\Gamma(\mathcal{D}) = G$. This gives the blueprint for our construction as well. In particular, we use the same building blocks that are described in [35, p. 591]. However, as opposed to Lackenby, we paste together these building blocks via *high-distance torus gluings* akin to the construction presented in [5, Section 4].

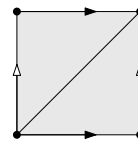
This ensures that we can apply Theorem 1 for the resulting 3-manifold \mathcal{M}_G and deduce the right-hand-side inequalities of **3a** and **3b**. The left-hand-side inequalities are shown by inspecting \mathcal{T}_G . We now elaborate on the ingredients of the proof of Theorem 2.

The building blocks. We first recall the definition of the building blocks from [35, p. 591]. Let $k \in \mathbb{N}$ be a positive integer. Consider the 2-dimensional torus $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$, and let \mathbb{T}_k^2 denote the compact surface obtained from \mathbb{T}^2 by the removal of k pairwise disjoint open disks. We define $\mathcal{N}(k) = \mathbb{T}_k^2 \times \mathbb{S}^1$. Note that the boundary $\partial\mathcal{N}(k)$ of $\mathcal{N}(k)$ consists of k tori, which will be the gluing sites. See Figure 12 for the example of $\mathcal{N}(3)$.

We choose a triangulation $\mathcal{T}(k)$ for $\mathcal{N}(k)$ that induces the minimal 2-triangle triangulation of the torus (cf. Figure 13) at each boundary component. Note that $\mathcal{T}(k)$ can be constructed from $O(k)$ tetrahedra. An explicit description of $\mathcal{T}(k)$ is given in [25, Appendix B].



■ **Figure 12** Illustration of $\mathcal{N}(3)$.



■ **Figure 13** Minimal triangulation of the torus \mathbb{T}^2 .

Overview of the construction of \mathcal{M}_G . We now give a high-level overview of constructing \mathcal{M}_G from the above building blocks. Given a graph $G = (V, E)$, for each node $v \in V$ we pick a block $\mathcal{N}_v \cong \mathcal{N}(d_v)$, where d_v denotes the degree of v . Next, for each arc $e = \{u, v\} \in E$, we pick a homeomorphism $\phi_e: \mathbb{T}^2 \rightarrow \mathbb{T}^2$ of sufficiently high *distance* (we discuss this notion below) and use it to glue together a boundary torus of \mathcal{N}_u with one of \mathcal{N}_v . After performing all of these gluings, we readily obtain the 3-manifold \mathcal{M}_G (cf. Figure 14(i)–(ii)).

▷ **Claim 7** (Based on [35, p. 591]). The JSJ decomposition \mathcal{D} of \mathcal{M}_G satisfies $\Gamma(\mathcal{D}) = G$.

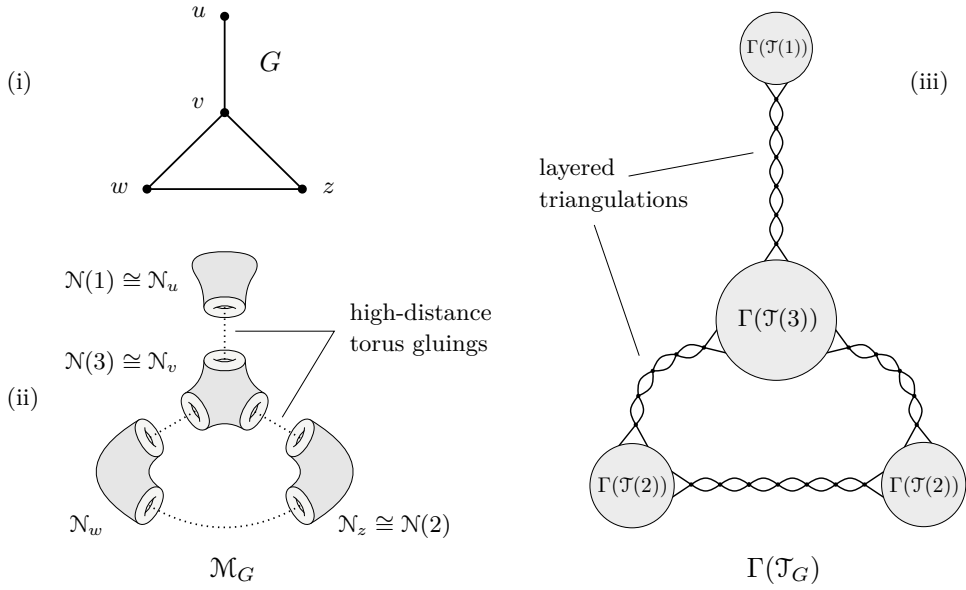
For a proof of Claim 7 we refer to [25, Appendix A.2].

High-distance torus gluings. As already mentioned, to ensure that we can apply Theorem 1 to \mathcal{M}_G , we use torus homeomorphisms of “sufficiently high distance” to glue the building blocks together. This notion of *distance*, which is defined through the *Farey distance*, is somewhat technical. Thus, we refer to [5, Section 4.1 and Appendix] for details, and rather recall a crucial result that makes the usefulness of distance in the current context apparent.

► **Theorem 8** ([5, Appendix], [3, Theorem 5.4]¹³). *There exists a computable constant K , depending only on the homeomorphism types of the blocks, so that if any set of blocks are glued with maps of distance at least $K\delta$ along their torus boundary components to form a closed 3-manifold \mathcal{M} , then the JSJ decomposition of \mathcal{M} is δ -complicated (cf. Definition 6).*

Triangulating the gluing maps. We have already discussed that the block $\mathcal{N}(k)$ admits a triangulation $\mathcal{T}(k)$ with $O(k)$ tetrahedra, where $\mathcal{T}(k)$ induces a minimal, 1-vertex triangulation at each torus boundary of $\mathcal{N}(k)$. It is shown in [5, Section 4.2] that the gluings beading these blocks together can be realized as *layered triangulations* [27]. These triangulations manifest as “daisy chains” in the dual graph $\Gamma(\mathcal{T}_G)$ of the final triangulation \mathcal{T}_G , see Figure 14(iii).

¹³The notation and the statement of Theorem 8 have been adapted to match the present context.



■ **Figure 14** Schematic overview of the construction underlying Theorem 2.

► **Lemma 9** ([5, Lemma 4.6]). *There exist torus gluings with distance at least D , that can be realized as layered triangulations using $2D$ tetrahedra.*

▷ **Claim 10.** There exist universal constants $c, c' > 0$ such that

$$(c/\Delta) \text{tw}(\mathcal{M}_G) \leq \text{tw}(G) \quad \text{and} \quad (c'/\Delta) \text{pw}(\mathcal{M}_G) \leq \text{pw}(G).$$

Proof. Since every node of G has degree at most Δ , the construction of \mathcal{M}_G only uses building blocks homeomorphic to $\mathcal{N}(1), \dots, \mathcal{N}(\Delta)$. Hence, for each $v \in V$ the triangulation $\mathcal{J}(d_v)$ of the block \mathcal{N}_v contains $O(\Delta)$ tetrahedra. This, together with the above discussion on triangulating the gluing maps implies that (upon ignoring multi-arcs and loop arcs, which are anyway not “sensed” by treewidth or pathwidth), the dual graph $\Gamma(\mathcal{J}_G)$ is obtained from G by **1.** replacing each node $v \in V$ with a copy of the graph $\Gamma(\mathcal{J}(d_v))$ that contains $O(\Delta)$ nodes, and by **2.** possibly subdividing each arc $e \in E$ several times.

Now, the first operation increases the treewidth (resp. pathwidth) at most by a factor of $O(\Delta)$, while the arc-subdivisions keep these parameters basically the same, cf. Lemma 11. Hence $\text{tw}(\Gamma(\mathcal{J}_G)) \leq O(\Delta \text{tw}(G))$ and $\text{pw}(\Gamma(\mathcal{J}_G)) \leq O(\Delta \text{pw}(G))$, and the claim follows. ◁

► **Lemma 11** (Folklore, cf. [8, Lemma A. 1]). *Let $G = (V, E)$ be a graph. If G' is a graph obtained from G by subdividing a set $F \subseteq E$ of arcs an arbitrary number of times. Then*

$$\text{pw}(G') \leq \text{pw}(G) + 2 \quad \text{and} \quad \text{tw}(G') \leq \max\{\text{tw}(G), 3\}.$$

Finishing the proof of Theorem 2. We have already shown **2** (Claim 7) and the left-hand-sides of the inequalities **3a** and **3b** (Claim 10). To prove the remaining parts of Theorem 2, let $\delta = \max\{18(\text{tw}(G) + 1), 4(3 \text{pw}(G) + 1)\} = O(\text{pw}(G))$. By Theorem 8, there is a computable constant K_Δ depending only on $\mathcal{N}(1), \dots, \mathcal{N}(\Delta)$ and hence only on Δ , so that if we glue together the blocks via maps of distance at least $K_\Delta \delta$, then the JSJ decomposition of \mathcal{M}_G is δ -complicated. By Lemma 9, each such gluing map can be realized as a layered triangulation consisting of $2K_\Delta \delta$ tetrahedra. Since G has at most $\Delta n/2$ arcs, these layered triangulations contain at most $2K_\Delta \delta \Delta n/2 = \Delta K_\Delta \delta n = O(\Delta K_\Delta \text{pw}(G) \cdot n) = O_\Delta(\text{pw}(G) \cdot n)$ tetrahedra

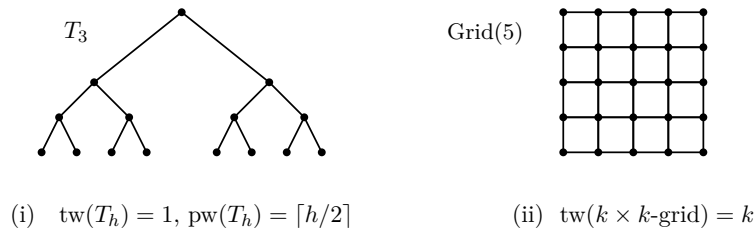
altogether. Since the triangulated blocks $\mathcal{T}(d_v)$ contain $O(\Delta \cdot n)$ tetrahedra in total, the triangulation \mathcal{T}_G of the manifold \mathcal{M}_G can be built from $O_\Delta(\text{pw}(G) \cdot n) \leq O_\Delta(n^2)$ tetrahedra. Last, as it follows from [5, Section 4], the construction can be executed in quadratic time. ◀

6 Applications

► **Corollary 12.** *There exist 3-manifolds $(\mathcal{M}_h)_{h \in \mathbb{N}}$ with $\text{tw}(\mathcal{M}_h) \leq 2$ and $\text{pw}(\mathcal{M}_h) \xrightarrow{h \rightarrow \infty} \infty$.*

► **Corollary 13.** *There exist Haken 3-manifolds $(\mathcal{N}_k)_{k \in \mathbb{N}}$ with $\text{tw}(\mathcal{N}_k) \xrightarrow{k \rightarrow \infty} \infty$.*

Proof of Corollaries 12 and 13. Using Theorem 2, the construction of bounded-treewidth (Haken) 3-manifolds with arbitrarily large pathwidth follows by taking $\mathcal{M}_h = \mathcal{M}_{T_h}$ (with the notation of Theorem 2), where T_h is the *complete binary tree of height h* . The construction of Haken 3-manifolds of arbitrary large treewidth is deduced by setting $\mathcal{N}_k = \mathcal{M}_{\text{Grid}(k)}$, where $\text{Grid}(k)$ denotes the $k \times k$ grid graph. See Figure 15. Obtained as JSJ decompositions, where the JSJ tori are two-sided, incompressible surfaces, all of these manifolds are Haken. ◀



■ **Figure 15** (i) The complete binary tree T_h of height h has pathwidth $\lceil h/2 \rceil$, cf. [10, Theorem 67]. (ii) The $k \times k$ grid graph is known to have pathwidth and treewidth both equal to k .

► **Corollary 14.** *Approximating the treewidth (resp. pathwidth) of closed, orientable 3-manifolds up to a constant factor is at least as hard as giving a constant-factor approximation of the treewidth (resp. pathwidth) of bounded-degree graphs.*

Proof. The argument for treewidth and pathwidth is the same. Given a graph G with maximum vertex degree Δ , we use the polynomial-time procedure from Theorem 2 to build a 3-manifold \mathcal{M} with $\text{tw}(\mathcal{M})$ within a constant factor of $\text{tw}(G)$. An oracle for a constant-factor approximation of $\text{tw}(\mathcal{M})$ hence gives us a constant-factor approximation of $\text{tw}(G)$ as well. ◀

► **Remark 15.** Computing a constant-factor approximation of treewidth (resp. pathwidth) for arbitrary graphs is known to be conditionally **NP**-hard under the Small Set Expansion Hypothesis [43, 51, 52]. For proving Corollary 14, however, we rely on the assumption that the graph has bounded degree. Thus the conditional hardness of approximating the treewidth (resp. pathwidth) of a 3-manifold does not directly follow. Establishing such a hardness result would add to the growing, but still relatively short list of algorithmic problems in low-dimensional topology that are known to be (conditionally) hard, cf. [2, 5, 19, 31, 35].

7 Discussion and Open Problems

We have demonstrated that 3-manifolds with JSJ decompositions with dual graphs of large treewidth (resp. pathwidth) and “sufficiently complicated” gluing maps cannot admit triangulations of low treewidth (resp. pathwidth). This provides a technique to construct

a wealth of families of 3-manifolds with unbounded tree- or pathwidth that hopefully will prove to be useful for future research in the field.

One obvious limitation of our construction is a seemingly heavy restriction on the JSJ gluing maps in order to deduce a connection between the treewidth of a 3-manifold and that of the dual graph of its JSJ decomposition. Hence, a natural question to ask is, how much this restriction on gluing maps may be relaxed while still allowing meaningful structural results about treewidth. In particular, we have the following question.

► **Question 1.** Given a 3-manifold \mathcal{M} with JSJ decomposition \mathcal{D} and no restrictions on its JSJ gluings. Is there a lower bound for $\text{tw}(\mathcal{M})$ in terms of $\text{tw}(\Gamma(\mathcal{D}))$?

Note that the assumption that we are considering the JSJ decomposition of \mathcal{M} is necessary: Consider a graph $G = (V, E)$ and a collection of 3-manifolds $\{\mathcal{M}_v\}_{v \in V}$, where \mathcal{M}_v has $\deg(v)$ torus boundary components. Assume that we glue the manifolds \mathcal{M}_v along the arcs of G to obtain a closed 3-manifold \mathcal{M} . Without restrictions on how these pieces are glued together, this cannot result in a lower bound $\text{tw}(\mathcal{M})$ in terms of $\text{tw}(G)$: we can construct Seifert fibered spaces \mathcal{M} in this way, even if $G = (V, E)$ is the complete graph with $|V|$ arbitrarily large. At the same time, Seifert fibered spaces have constant treewidth, see [24].

► **Question 2.** What is the complexity of computing the treewidth of a 3-manifold?

We believe that this should be at least as hard as computing the treewidth of a graph.



References

- 1 I. Agol. Small 3-manifolds of large genus. *Geom. Dedicata*, 102:53–64, 2003. doi:10.1023/B:GEOM.0000006584.85248.c5.
- 2 I. Agol, J. Hass, and W. Thurston. The computational complexity of knot genus and spanning area. *Trans. Am. Math. Soc.*, 358(9):3821–3850, 2006. doi:10.1090/S0002-9947-05-03919-X.
- 3 D. Bachman. Stabilizing and destabilizing Heegaard splittings of sufficiently complicated 3-manifolds. *Math. Ann.*, 355(2):697–728, 2013. doi:10.1007/s00208-012-0802-4.
- 4 D. Bachman, R. Derby-Talbot, and E. Sedgwick. Heegaard structure respects complicated JSJ decompositions. *Math. Ann.*, 365(3-4):1137–1154, 2016. doi:10.1007/s00208-015-1314-9.
- 5 D. Bachman, R. Derby-Talbot, and E. Sedgwick. Computing Heegaard genus is NP-hard. In *A Journey Through Discrete Mathematics: A Tribute to Jiří Matoušek*, pages 59–87. Springer, Cham, 2017. doi:10.1007/978-3-319-44479-6_3.
- 6 D. Bachman, S. Schleimer, and E. Sedgwick. Sweepouts of amalgamated 3-manifolds. *Algebr. Geom. Topol.*, 6:171–194, 2006. doi:10.2140/agt.2006.6.171.
- 7 B. Bagchi, B. A. Burton, B. Datta, N. Singh, and J. Spreer. Efficient algorithms to decide tightness. In *32nd Int. Symp. Comput. Geom. (SoCG 2016)*, volume 51 of *LIPICs. Leibniz Int. Proc. Inf.*, pages 12:1–12:15. Schloss Dagstuhl–Leibniz-Zent. Inf., 2016. doi:10.4230/LIPICs.SoCG.2016.12.
- 8 R. Belmonte, T. Hanaka, M. Kanzaki, M. Kiyomi, Y. Kobayashi, Y. Kobayashi, M. Lampis, H. Ono, and Y. Otachi. Parameterized complexity of (A, ℓ) -path packing. *Algorithmica*, 84(4):871–895, 2022. doi:10.1007/s00453-021-00875-y.
- 9 R. H. Bing. An alternative proof that 3-manifolds can be triangulated. *Ann. Math. (2)*, 69:37–65, 1959. doi:10.2307/1970092.
- 10 H. L. Bodlaender. A partial k -arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.
- 11 B. A. Burton. Computational topology with Regina: algorithms, heuristics and implementations. In *Geometry and Topology Down Under*, volume 597 of *Contemp. Math.*, pages 195–224. Am. Math. Soc., Providence, RI, 2013. doi:10.1090/conm/597/11877.
- 12 B. A. Burton, R. Budney, W. Pettersson, et al. Regina: Software for low-dimensional topology, 1999–2022. Version 7.2. URL: <https://regina-normal.github.io>.

- 13 B. A. Burton and R. G. Downey. Courcelle's theorem for triangulations. *J. Comb. Theory, Ser. A*, 146:264–294, 2017. doi:10.1016/j.jcta.2016.10.001.
- 14 B. A. Burton, T. Lewiner, J. Paixão, and J. Spreer. Parameterized complexity of discrete Morse theory. *ACM Trans. Math. Softw.*, 42(1):6:1–6:24, 2016. doi:10.1145/2738034.
- 15 B. A. Burton, C. Maria, and J. Spreer. Algorithms and complexity for Turaev–Viro invariants. *J. Appl. Comput. Topol.*, 2(1–2):33–53, 2018. doi:10.1007/s41468-018-0016-2.
- 16 B. A. Burton and W. Pettersson. Fixed parameter tractable algorithms in combinatorial topology. In *Proc. 20th Int. Conf. Comput. Comb. (COCOON 2014)*, pages 300–311, 2014. doi:10.1007/978-3-319-08783-2_26.
- 17 B. A. Burton and J. Spreer. The complexity of detecting taut angle structures on triangulations. In *Proc. 24th Annu. ACM-SIAM Symp. Discrete Algorithms (SODA 2013)*, pages 168–183, 2013. doi:10.1137/1.9781611973105.13.
- 18 A. J. Casson and C. McA. Gordon. Reducing Heegaard splittings. *Topology Appl.*, 27(3):275–283, 1987. doi:10.1016/0166-8641(87)90092-7.
- 19 A. de Mesmay, Y. Rieck, E. Sedgwick, and M. Tancer. The unbearable hardness of unknotting. *Adv. Math.*, 381:Paper No. 107648, 36, 2021. doi:10.1016/j.aim.2021.107648.
- 20 A. Hatcher. Notes on basic 3-manifold topology. Available at <https://pi.math.cornell.edu/~hatcher/3M/3Mfds.pdf> (accessed: February 18, 2023).
- 21 P. Heegaard. Sur l' "Analysis situs". *Bull. Soc. Math. France*, 44:161–242, 1916. doi:10.24033/bsmf.968.
- 22 K. Huszár. *Combinatorial width parameters for 3-dimensional manifolds*. PhD thesis, IST Austria, June 2020. doi:10.15479/AT:ISTA:8032.
- 23 K. Huszár. On the pathwidth of hyperbolic 3-manifolds. *Comput. Geom. Topol.*, 1(1):1–19, 2022. doi:10.57717/cgt.v1i1.4.
- 24 K. Huszár and J. Spreer. 3-Manifold triangulations with small treewidth. In *35th Int. Symp. Comput. Geom. (SoCG 2019)*, volume 129 of *LIPICs. Leibniz Int. Proc. Inf.*, pages 44:1–44:20. Schloss Dagstuhl–Leibniz-Zent. Inf., 2019. doi:10.4230/LIPICs.SoCG.2019.44.
- 25 K. Huszár and J. Spreer. On the width of complicated JSJ decompositions, 2023. 22 pages, 19 figures. arXiv:2303.06789.
- 26 K. Huszár, J. Spreer, and U. Wagner. On the treewidth of triangulated 3-manifolds. *J. Comput. Geom.*, 10(2):70–98, 2019. doi:10.20382/jogc.v10i2a5.
- 27 W. Jaco and J. H. Rubinstein. Layered-triangulations of 3-manifolds, 2006. 97 pages, 32 figures. arXiv:math/0603601.
- 28 W. Jaco and P. B. Shalen. A new decomposition theorem for irreducible sufficiently-large 3-manifolds. In *Algebraic and Geometric Topology*, volume 32, part 2 of *Proc. Sympos. Pure Math.*, pages 71–84. Am. Math. Soc., Providence, RI, 1978. doi:10.1090/pspum/032.2/520524.
- 29 W. H. Jaco and P. B. Shalen. Seifert fibered spaces in 3-manifolds. *Mem. Am. Math. Soc.*, 21(220):viii+192, 1979. doi:10.1090/memo/0220.
- 30 K. Johannson. *Homotopy equivalences of 3-manifolds with boundaries*, volume 761 of *Lect. Notes Math.* Springer, Berlin, 1979. doi:10.1007/BFb0085406.
- 31 D. Koenig and A. Tsvietkova. NP-hard problems naturally arising in knot theory. *Trans. Am. Math. Soc. Ser. B*, 8:420–441, 2021. doi:10.1090/btran/71.
- 32 G. Kuperberg. Algorithmic homeomorphism of 3-manifolds as a corollary of geometrization. *Pacific J. Math.*, 301(1):189–241, 2019. doi:10.2140/pjm.2019.301.189.
- 33 M. Lackenby. The Heegaard genus of amalgamated 3-manifolds. *Geom. Dedicata*, 109:139–145, 2004. doi:10.1007/s10711-004-6553-y.
- 34 M. Lackenby. Heegaard splittings, the virtually Haken conjecture and property (τ) . *Invent. Math.*, 164(2):317–359, 2006. doi:10.1007/s00222-005-0480-x.
- 35 M. Lackenby. Some conditionally hard problems on links and 3-manifolds. *Discrete Comput. Geom.*, 58(3):580–595, 2017. doi:10.1007/s00454-017-9905-8.

- 36 M. Lackenby. Algorithms in 3-manifold theory. In I. Agol and D. Gabai, editors, *Surveys in 3-manifold topology and geometry*, volume 25 of *Surv. Differ. Geom.*, pages 163–213. Int. Press Boston, 2020. doi:10.4310/SDG.2020.v25.n1.a5.
- 37 T. Li. Heegaard surfaces and the distance of amalgamation. *Geom. Topol.*, 14(4):1871–1919, 2010. doi:10.2140/gt.2010.14.1871.
- 38 C. Manolescu. Lectures on the triangulation conjecture. In *Proc. 22nd Gökova Geom.-Topol. Conf. (GGT 2015)*, pages 1–38. Int. Press Boston, 2016. URL: <https://gokovagt.org/proceedings/2015/manolescu.html>.
- 39 C. Maria and J. Purcell. Treewidth, crushing and hyperbolic volume. *Algebr. Geom. Topol.*, 19(5):2625–2652, 2019. doi:10.2140/agt.2019.19.2625.
- 40 A. Markov. The insolubility of the problem of homeomorphy (in Russian). *Dokl. Akad. Nauk SSSR*, 121:218–220, 1958.
- 41 E. E. Moise. Affine structures in 3-manifolds. V. The triangulation theorem and Hauptvermutung. *Ann. Math. (2)*, 56:96–114, 1952. doi:10.2307/1969769.
- 42 B. Poonen. Undecidable problems: a sampler. In *Interpreting Gödel*, pages 211–241. Cambridge Univ. Press, 2014. doi:10.1017/CB09780511756306.015.
- 43 P. Raghavendra and D. Steurer. Graph expansion and the unique games conjecture. In *Proc. 2010 ACM Int. Symp. Theor. Comput. (STOC'10)*, pages 755–764. ACM, New York, 2010.
- 44 M. Scharlemann and J. Schultens. Comparing Heegaard and JSJ structures of orientable 3-manifolds. *Trans. Am. Math. Soc.*, 353(2):557–584, 2001. doi:10.1090/S0002-9947-00-02654-4.
- 45 M. Scharlemann, J. Schultens, and T. Saito. Lecture notes on generalized Heegaard splittings, 2005. Open-access version of [46] with slightly different structure. arXiv:math/0504167.
- 46 M. Scharlemann, J. Schultens, and T. Saito. *Lecture Notes on Generalized Heegaard Splittings*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2016. doi:10.1142/10019.
- 47 M. Scharlemann and A. Thompson. Thin position for 3-manifolds. In *Geometric topology (Haifa, 1992)*, volume 164 of *Contemp. Math.*, pages 231–238. Am. Math. Soc., Providence, RI, 1994. doi:10.1090/conm/164/01596.
- 48 J. Schultens. *Introduction to 3-Manifolds*, volume 151 of *Grad. Stud. Math.* Am. Math. Soc., Providence, RI, 2014. doi:10.1090/gsm/151.
- 49 J. Schultens and R. Weidmann. Destabilizing amalgamated Heegaard splittings. In *Workshop on Heegaard Splittings*, volume 12 of *Geom. Topol. Monogr.*, pages 319–334. Geom. Topol. Publ., Coventry, 2007. doi:10.2140/gtm.2007.12.319.
- 50 P. Scott and H. Short. The homeomorphism problem for closed 3-manifolds. *Algebr. Geom. Topol.*, 14(4):2431–2444, 2014. doi:10.2140/agt.2014.14.2431.
- 51 Y. Wu, P. Austrin, T. Pitassi, and D. Liu. Inapproximability of treewidth, one-shot pebbling, and related layout problems. *J. Artificial Intelligence Res.*, 49:569–600, 2014. doi:10.1613/jair.4030.
- 52 K. Yamazaki. Inapproximability of rank, clique, Boolean, and maximum induced matching-widths under small set expansion hypothesis. *Algorithms (Basel)*, 11(11):Paper No. 173, 10, 2018. doi:10.3390/a11110173.

Reconfiguration of Colorings in Triangulations of the Sphere

Takehiro Ito  

Graduate School of Information Sciences, Tohoku University, Sendai, Japan

Yuni Iwamasa  

Graduate School of Informatics, Kyoto University, Japan

Yusuke Kobayashi  

Research Institute for Mathematical Sciences, Kyoto University, Japan

Shun-ichi Maezawa  

Department of Mathematics, Tokyo University of Science, Japan

Yuta Nozaki  

Faculty of Environment and Information Sciences, Yokohama National University, Japan
SKCM², Hiroshima University, Japan

Yoshio Okamoto  

Graduate School of Informatics and Engineering, The University of Electro-Communications,
Tokyo, Japan

Kenta Ozeki  

Faculty of Environment and Information Sciences, Yokohama National University, Japan

Abstract

In 1973, Fisk proved that any 4-coloring of a 3-colorable triangulation of the 2-sphere can be obtained from any 3-coloring by a sequence of Kempe-changes. On the other hand, in the case where we are only allowed to recolor a single vertex in each step, which is a special case of a Kempe-change, there exists a 4-coloring that cannot be obtained from any 3-coloring.

In this paper, we present a linear-time checkable characterization of a 4-coloring of a 3-colorable triangulation of the 2-sphere that can be obtained from a 3-coloring by a sequence of recoloring operations at single vertices. In addition, we develop a quadratic-time algorithm to find such a recoloring sequence if it exists; our proof implies that we can always obtain a quadratic length recoloring sequence. We also present a linear-time checkable criterion for a 3-colorable triangulation of the 2-sphere that all 4-colorings can be obtained from a 3-coloring by such a sequence. Moreover, we consider a high-dimensional setting. As a natural generalization of our first result, we obtain a polynomial-time checkable characterization of a k -coloring of a $(k - 1)$ -colorable triangulation of the $(k - 2)$ -sphere that can be obtained from a $(k - 1)$ -coloring by a sequence of recoloring operations at single vertices and the corresponding algorithmic result. Furthermore, we show that the problem of deciding whether, for given two $(k + 1)$ -colorings of a $(k - 1)$ -colorable triangulation of the $(k - 2)$ -sphere, one can be obtained from the other by such a sequence is PSPACE-complete for any fixed $k \geq 4$. Our results above can be rephrased as new results on the computational problems named k -RECOLORING and CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH, which are fundamental problems in the field of combinatorial reconfiguration.

2012 ACM Subject Classification Theory of computation \rightarrow Design and analysis of algorithms; Mathematics of computing \rightarrow Graph coloring

Keywords and phrases Graph coloring, Triangulation of the sphere, Combinatorial reconfiguration

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.43

Related Version Full preprint: [arXiv:2210.17105](https://arxiv.org/abs/2210.17105) [math.CO](https://arxiv.org/abs/2210.17105) [21]

Funding Takehiro Ito: JSPS KAKENHI Grant Numbers JP18H04091, JP19K11814, JP20H05793.

Yuni Iwamasa: JSPS KAKENHI Grant Numbers JP20H05795, JP20K23323, JP22K17854.

Yusuke Kobayashi: JSPS KAKENHI Grant Numbers JP20H05795.

Shun-ichi Maezawa: JSPS KAKENHI Grant Numbers JP20H05795, JP22K13956.



© Takehiro Ito, Yuni Iwamasa, Yusuke Kobayashi, Shun-ichi Maezawa, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki;

licensed under Creative Commons License CC-BY 4.0

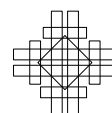
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 43; pp. 43:1–43:16



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Yuta Nozaki: JSPS KAKENHI Grant Numbers JP20H05795, JP20K14317.

Yoshio Okamoto: JSPS KAKENHI Grant Numbers JP20H05795, JP20K11670.

Kenta Ozeki: JSPS KAKENHI Grant Numbers JP20H05795, JP22K19773.

Acknowledgements We thank Naonori Kakimura and Naoyuki Kamiyama for related discussion, and anonymous reviewers for helpful suggestions.

1 Introduction

In 1973, Fisk [14] proved that all 4-colorings of a 3-colorable triangulation of the 2-sphere are *Kempe-equivalent*, that is, for any two 4-colorings of the graph, one is obtained from the other by a sequence of *Kempe-changes*. The method of Kempe-changes is known as a powerful tool for coloring of graphs (see e.g., [19, 10]), and has been intensively studied in graph theory (see e.g., [26, 24, 27, 28, 13, 1, 12, 2]). In particular, Mohar [27] proved that all 4-colorings of a 3-colorable planar graph are Kempe-equivalent using Fisk's result, and then Feghali [12] improved this for 4-critical planar graphs. Mohar and Salas [28] extended Fisk's result to toroidal triangulations.

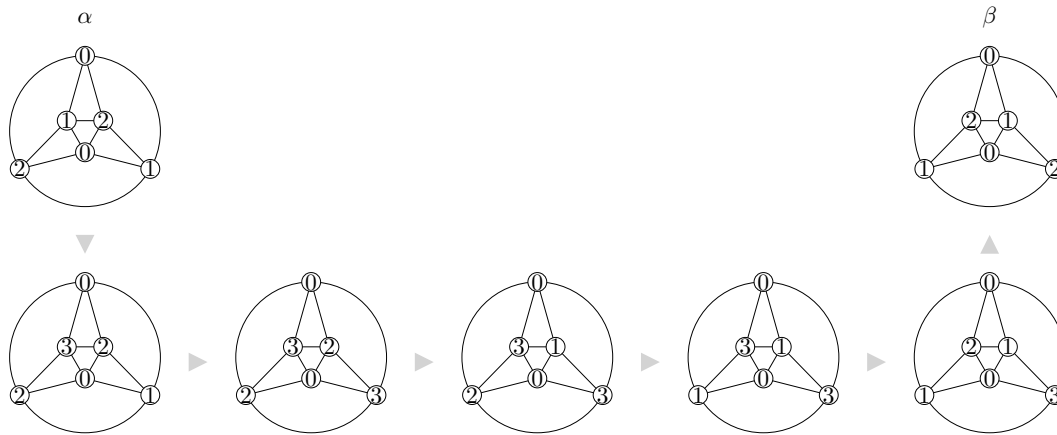
The formal definitions of Kempe-change and Kempe-equivalence are given as follows. Let $\alpha: V(G) \rightarrow \{0, 1, \dots, k-1\}$ be a k -coloring of a graph G , let a, b be two distinct colors in $\{0, 1, \dots, k-1\}$, and let C be a connected component of the subgraph of G induced by the vertices colored with either a or b . Then, a *Kempe-change* of α (at C) is an operation to give rise to a new k -coloring by exchanging the colors a and b on all vertices in C . In particular, if C consists of a single vertex, then we refer to such a Kempe-change at C as a *single-change*. Two k -colorings of G are *Kempe-equivalent* if one is obtained from the other by a sequence of Kempe-changes, and *single-equivalent* if one is obtained from the other by a sequence of single-changes.

Let us return to Fisk's result for the Kempe-equivalence. Let G be a 3-colorable triangulation of the 2-sphere. The proof consists of the following two statements: All 3-colorings of G are Kempe-equivalent under 4-colorings, and any 4-coloring of G is Kempe-equivalent to a 3-coloring. Here, a 3-coloring means that a coloring uses only three colors in $\{0, 1, 2, 3\}$. The first statement, which is folklore, can be easily obtained as follows. Since G is a 3-colorable triangulation, for any two 3-colorings α, β of G there uniquely exists a permutation π on $\{0, 1, 2, 3\}$ such that $\beta = \pi \circ \alpha$. Then, according to π , we can obtain β from α by a sequence of Kempe-changes (under 4-colorings) each of which changes a color at only one vertex, namely, a sequence of single-changes, by using the fourth color not appearing in α ; see Figure 1 for example. Therefore, the nontrivial and crucial part in Fisk's result is to show the second statement.

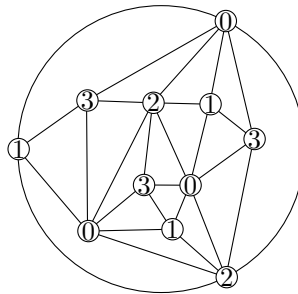
The above observation for the first statement says that all 3-colorings of G are single-equivalent under 4-colorings. On the other hand, in general, some 4-coloring is not single-equivalent to any of 3-colorings (Figure 2). Here natural questions arise: *What 4-colorings are single-equivalent to some 3-coloring?* and *which 3-colorable triangulations of the 2-sphere have the property that all 4-colorings are single-equivalent?*

In this paper, we resolve these questions in the following sense, where n denotes the number of vertices of G .

1. We present an $O(n)$ -time checkable characterization for a 4-coloring of G to be single-equivalent to some 3-coloring (Theorem 2). In addition, we show that, for any 4-colorings α, β of G single-equivalent to some 3-coloring, there exists a sequence of single-changes of length $O(n^2)$ from α to β and we can obtain it in $O(n^2)$ time (Theorem 6).



■ **Figure 1** Single-equivalence of two 3-colorings of a 3-colorable triangulation of the 2-sphere.



■ **Figure 2** A 4-coloring of a 3-colorable triangulation of the 2-sphere such that it is not single-equivalent to any 3-coloring; no vertex can be recolored by a single-change. This coloring is “frozen”.

2. We provide an $O(n)$ -time checkable criterion for a 3-colorable triangulation of the 2-sphere that all 4-colorings are single-equivalent (Theorem 8).

Furthermore, we consider a triangulation of a high-dimensional sphere. Let G be a $(k - 1)$ -colorable triangulation of the $(k - 2)$ -sphere for some positive integer $k \geq 4$. Then, by the same argument as in the case of $k = 4$ above, all $(k - 1)$ -colorings of G are single-equivalent under k -colorings. The following is a generalization of our first results (Theorem 2 and Theorem 6):

3. We present a characterization for a k -coloring of a $(k - 1)$ -colorable triangulation G of the $(k - 2)$ -sphere to be single-equivalent to some $(k - 1)$ -coloring. In addition, we show that, for any k -colorings α, β of G single-equivalent to some $(k - 1)$ -coloring, there exists a sequence of single-changes of length $O(n^{2\lfloor (k-1)/2 \rfloor})$ from α to β and we can obtain it in $O(n^{2\lfloor (k-1)/2 \rfloor})$ time.

In fact, the third result can be further generalized to $(k - 1)$ -colorable triangulations of connected closed $(k - 2)$ -manifolds satisfying a certain condition. This result is omitted in this paper, and given in the full version [21].

Our results are deeply related to the computational problems named k -RECOLORING and CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH, which ask the connectedness of a k -coloring reconfiguration graph. Here, the k -coloring reconfiguration graph of a k -colorable graph G , denoted by $\mathcal{R}_k(G)$, is a graph such that its vertex set consists of all k -colorings of G and there is an edge between two k -colorings α and β of G if and only if β is obtained from α by recoloring only a single vertex in G , i.e., by a single-change. Thus,

two k -colorings of G are single-equivalent if and only if they are connected in $\mathcal{R}_k(G)$. Then k -RECOLORING and CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH are defined as follows.

k -RECOLORING

Input: A k -colorable graph G and k -colorings α and β of G .

Output: YES if α and β are connected in $\mathcal{R}_k(G)$, and NO otherwise.

CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH

Input: A k -colorable graph G .

Output: YES if $\mathcal{R}_k(G)$ is connected, and NO otherwise.

The problems k -RECOLORING and CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH are fundamental in the recently emerging field of *combinatorial reconfiguration* (see [35, 30] for surveys and [22] for a general solver), which are extensively studied. It is shown that k -RECOLORING is polynomial-time solvable if $k \leq 3$ [6], while PSPACE-complete if $k \geq 4$ [3]. According to [35, Section 3.2], the situation is very different from that for Kempe-equivalence, whose complexity is widely open. Bonsma and Cereceda [3] considered k -RECOLORING for (bipartite) planar graphs; k -RECOLORING for planar graphs is PSPACE-complete if $4 \leq k \leq 6$ and that for bipartite planar graphs is PSPACE-complete if $k = 4$. Cereceda, van den Heuvel, and Johnson [4] showed that $\mathcal{R}_k(G)$ is connected for any $(k - 2)$ -degenerate graph. By combining it with the fact that any planar graph is 5-degenerate and any bipartite planar graph is 3-degenerate, we see that k -RECOLORING and CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH are in P (all instances are YES-instances) for any planar graph with $k \geq 7$ and for any bipartite planar graph with $k \geq 5$. In another paper [5], Cereceda, van den Heuvel, and Johnson also showed that CONNECTEDNESS OF 3-COLORING RECONFIGURATION GRAPH is coNP-complete in general and is in P for bipartite planar graphs.

The problem CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH is also fundamental in the studies of the Glauber dynamics (a class of Markov chains) for k -colorings of a graph, which are used for random sampling and approximate counting. In each step of the Glauber dynamics of k -colorings, we are given a k -coloring of a graph. Then, we pick a vertex v and a color c uniformly at random, and change the color of v to c when the neighbors of v are not colored by c . Hence, one step of this Markov chain is exactly a single-exchange as long as we move to another coloring, and the state space is identical to the k -coloring reconfiguration graph. The connectedness of the k -coloring reconfiguration graph ensures that the Markov chain is irreducible. For the Glauber dynamics, the mixing property is one of the main concerns. It is an open question whether the Glauber dynamics of k -colorings has polynomial mixing time when $k \geq \Delta + 2$, where Δ is the maximum degree of a graph [23]. From continuing work in the literature, we know that the Glauber dynamics mixes quickly when $k > 2\Delta$ [23], $k > \frac{6}{11}\Delta$ [36], and finally $k > (\frac{6}{11} - \varepsilon)\Delta$ for a small absolute constant $\varepsilon > 0$ [7]. Results on restricted classes of graphs have also been known. For example, Hayes, Vera and Vigoda [17] proved that the Glauber dynamics mixes fast for planar graphs when $k = \Omega(\Delta / \log \Delta)$.

Our proofs provide algorithms for special cases of k -RECOLORING and CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH. Here, we are supposed to be given a simplicial complex K whose geometric realization is homeomorphic to the $(k - 2)$ -sphere such that its 1-skeleton G is $(k - 1)$ -colorable. As we have seen, all $(k - 1)$ -colorings of G belong to the same connected component of $\mathcal{R}_k(G)$; we refer to it as the $(k - 1)$ -coloring component of $\mathcal{R}_k(G)$. Our third result (including the first) implies that, provided one of the input k -colorings α

and β belongs to the $(k - 1)$ -coloring component of $\mathcal{R}_k(G)$, the problem k -RECOLORING for G can be solved in linear time in the size $\#K$ of the input simplicial complex K . In particular, if k is fixed, then our result says that it can be solved in polynomial time in n . Our second result implies that CONNECTEDNESS OF 4-COLORING RECONFIGURATION GRAPH for a 3-colorable triangulation of the 2-sphere can be solved in linear time in n .

We further investigate the computational complexity of the recoloring problem for a $(k - 1)$ -colorable triangulation G of the $(k - 2)$ -sphere. It is still open whether k -RECOLORING for G can be solved in polynomial time, although we prove the polynomial-time solvability of the special case where one of the input k -colorings α and β belongs to the $(k - 1)$ -coloring component of $\mathcal{R}_k(G)$. In this paper, we additionally show that, if the number of colors which we can use increases by one, then it is difficult to check the single-equivalence between given two colorings:

4. For any fixed $k \geq 4$, the problem $(k + 1)$ -RECOLORING is PSPACE-complete for $(k - 1)$ -colorable triangulations of the $(k - 2)$ -sphere (Theorem 13).

In the case of $k = 4$, our result is stronger than the PSPACE-completeness of 5-RECOLORING for planar graphs, which is known in the literature [3].

We here emphasize that, for our algorithmic results, we are given a triangulation of a sphere, but not only its 1-skeleton. We need this assumption since our algorithm uses the triangulation and obtaining the triangulation from the 1-skeleton is hard. Indeed, for each fixed $d \geq 5$, the sphere recognition problem is undecidable [37, 8]: Namely it is undecidable whether a given simplicial complex is a triangulation of the d -sphere. This implies that it is also undecidable whether a given graph is the 1-skeleton of some triangulation of the d -sphere. When $d = 3$, the sphere recognition is decidable [31, 34], but not known to be solved in polynomial time (while it is known to be in NP [33]); the decidability is open when $d = 4$. Therefore, when $d \geq 3$, to filter out the intrinsic intractability of sphere recognition, we assume that a triangulation is also given along with a graph. On the other hand, when $d = 2$, we can decide whether a graph is the 1-skeleton of some triangulation in linear time [20]. In this case, the size of a triangulation is the same as the size of its 1-skeleton in the order of magnitude by Euler's formula, and therefore, the assumption that a triangulation is also given is not relevant.

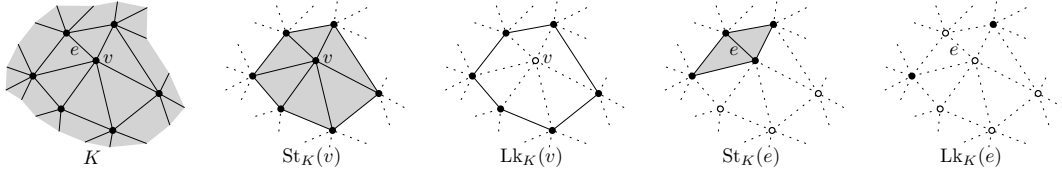
Organization

This paper is organized as follows. In Section 2, we introduce notation. We provide a linear-time checkable characterization of the 3-coloring component of a 3-colorable triangulation of the 2-sphere in Section 3, which answers the first question. Section 4 is devoted to resolving the second question: We present a linear-time checkable criterion for a 3-colorable triangulation of the 2-sphere that any two 4-colorings are single-equivalent. In Section 5, we show the PSPACE-completeness of $(k + 1)$ -RECOLORING for $(k - 1)$ -colorable triangulations of the $(k - 2)$ -sphere for $k \geq 4$. The arguments on our third result (a high-dimensional generalization of our first result) and several proofs of the statements marked with \star are omitted. They are given in the full version of this paper [21].

2 Preliminaries

For a set A , we denote by $\#A$ the cardinality of A .

Let $G = (V, E)$ be a graph. For $v \in V$, we denote by $N_G(v)$ the set of neighbors of v and by $\delta_G(v)$ the set of edges incident to v ; we simply write $N(v)$ and $\delta(v)$ if G is clear from the context. A map $\alpha: V \rightarrow \{0, 1, \dots, k - 1\}$ is called a k -coloring if $\alpha(u) \neq \alpha(v)$ for each edge



■ **Figure 3** An example of the star complexes and the link complexes of a 2-dimensional simplicial complex K .

$\{u, v\} \in E$. A vertex $v \in V$ is said to be *recolorable* with respect to a k -coloring α if there is a k -coloring α' such that $\alpha'(u) = \alpha(u)$ for $u \in V \setminus \{v\}$ and $\alpha'(v) \neq \alpha(v)$, i.e., we can change the color $\alpha(v)$ of v .

Let S^d denote the d -sphere. A *triangulation* of S^d is a pair of a simplicial complex K and a homeomorphism $h: |K| \rightarrow S^d$, where $|K|$ denotes the geometric realization of K . See, for instance, Munkres [29] for fundamental terminology in simplicial complexes. Throughout this paper, we identify $|K|$ with S^d and omit to write h . For a simplex $\sigma \in K$, its *star complex* $\text{St}_K(\sigma)$ and *link complex* $\text{Lk}_K(\sigma)$ are defined by

$$\begin{aligned} \text{St}_K(\sigma) &:= \{\tau \in K \mid \sigma \text{ and } \tau \text{ are faces of a common simplex in } K\}, \\ \text{Lk}_K(\sigma) &:= \{\tau \in K \mid \sigma \cap \tau = \emptyset, \sigma * \tau \in K\}, \end{aligned}$$

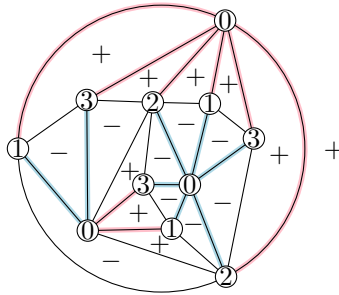
where $\sigma * \tau$ denotes the join of σ and τ (see [29, Section 62]). Figure 3 shows examples. Also, let $\text{St}_K^d(\sigma)$ denote the d -simplices in $\text{St}_K(\sigma)$. For a subset $K' \subseteq K$, we define $|K'| \subseteq S^d$ by $|K'| := \bigcup_{\sigma \in K'} \sigma$. For instance, if v is a vertex of a triangulation of a surface without boundary, then $|\text{St}_K(v)|$ and $|\text{Lk}_K(v)|$ are homeomorphic to a closed disk and a circle, respectively. In this paper, we specify a triangulation by an embedded graph G in S^d , which is actually the 1-skeleton of a triangulation K . Also, we suppose that the input of k -RECOLORING and CONNECTEDNESS OF k -COLORING RECONFIGURATION GRAPH is the simplicial complex K ; for example, we are given the set of faces of a triangulation of the 2-sphere. We use $\text{St}_G(\sigma)$ instead of $\text{St}_K(\sigma)$ by abuse of notation. For example, $\text{St}_G^0(v) \setminus \{v\} = N_G(v)$ and $\text{St}_G^1(v) \setminus \text{Lk}_G(v) = \delta_G(v)$. Also, we simply write $\text{St}(\sigma)$ and $\text{Lk}(\sigma)$ if G or K is clear from the context.

It is well-known that a triangulation of the 2-sphere is 3-colorable if and only if every vertex has an even degree (i.e., Eulerian). In this sense, a 3-colorable triangulation is said to be *even*. More generally, a triangulation K of a closed d -manifold is *even* if $\#\text{St}^d(\sigma^{d-2})$ is even for every $(d-2)$ -simplex $\sigma^{d-2} \in K$, where $d \geq 2$. If the 1-skeleton of K is $(d+1)$ -colorable, then K is even. By [16, Sections I.4 and VI.2], the converse is also true for S^d , more generally, for simply-connected manifolds. Hence, it is easy to check whether a given triangulation of S^d is $(d+1)$ -colorable.

3 A characterization of the $(k-1)$ -coloring component

In this section, we resolve the first question posed in Section 1: *In a 3-colorable triangulation G of the 2-sphere, what 4-colorings are single-equivalent to some 3-coloring?* A characterization for high-dimensional cases can be obtained by a similar argument, which is omitted and given in the full version of this paper [21].

Let $G = (V, E)$ be a 3-colorable triangulation of the 2-sphere. Recall that all 3-colorings of G belong to the same connected component of $\mathcal{R}_4(G)$; we refer to it as the 3-coloring component of $\mathcal{R}_4(G)$. Let n denote the number $\#V$ of vertices of G .



■ **Figure 4** An example of the signature assignment to the faces. Red edges depict the $+-$ -nonsingular edges and blue edges depict the $--$ -nonsingular edges.

Let F be the set of faces of G . We first define the signature on a face in F with respect to a 4-coloring of G and its related concepts, which were originally introduced in [18] (see also [32, Section 8 of Chapter 2]). These play an important role in our characterization.

Let $\alpha: V \rightarrow \{0, 1, 2, 3\}$ be a 4-coloring of G . We assign a signature $+1/-1$ to each face $f \in F$ so that, for every pair of adjacent faces f, f' with $f = \{u, v, w\}$ and $f' = \{u', v, w\}$, they have the same signature if and only if $\alpha(u) \neq \alpha(u')$, where two faces $f, f' \in F$ are said to be *adjacent* if f and f' share an edge, i.e., $\#(f \cap f') = 2$. Such an assignment can be obtained as follows. For each face $f = \{u, v, w\} \in F$, we denote by $[\alpha(f)]$ the cyclically ordered set $[\alpha(u)\alpha(v)\alpha(w)]$ on $\{\alpha(u), \alpha(v), \alpha(w)\}$, where u, v, w are arranged in counterclockwise order in G if we see it from the outside of the 2-sphere. We define $\varepsilon_\alpha: F \rightarrow \{+1, -1\}$ by

$$\varepsilon_\alpha(f) := \begin{cases} +1 & \text{if } [\alpha(f)] \in \{[123], -[023], [013], -[012]\}, \\ -1 & \text{if } [\alpha(f)] \in \{-[123], [023], -[013], [012]\}, \end{cases}$$

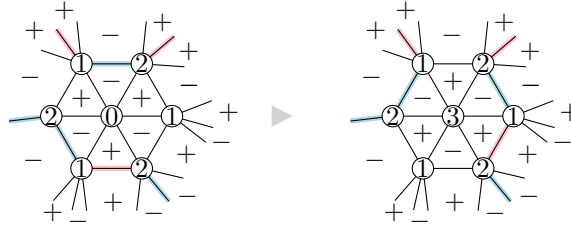
where the minus sign $-$ indicates the opposite order, that is, $-[ijk] = [jik]$. We note here that when we regard $[123], -[023], [013], -[012]$ as oriented 2-simplices, they appear in the boundary of an oriented 3-simplex $[0123]: \partial[0123] = [123] \cup -[023] \cup [013] \cup -[012]$. A face $f \in F$ with $\varepsilon_\alpha(f) = +1$ (resp. $\varepsilon_\alpha(f) = -1$) is called a $+face$ (resp. $-face$) with respect to α . Figure 4 shows an example. Recall that, for $v \in V$, the set of faces containing v is denoted as $St^2(v)$. For a 4-coloring α , let $F_\alpha^+(v)$ (resp. $F_\alpha^-(v)$) denote the set of $+faces$ (resp. $-faces$) in $St^2(v)$.

An edge $e \in E$ is said to be *singular* with respect to α if the two adjacent faces $f, f' \in F$ sharing e have different signatures, i.e., $\varepsilon_\alpha(f) \neq \varepsilon_\alpha(f')$, and to be *nonsingular* if it is not singular [14, 15]. A nonsingular edge is particularly said to be $+nonsingular$ (resp. $--nonsingular$) if $\varepsilon_\alpha(f) = \varepsilon_\alpha(f') = +1$ (resp. $\varepsilon_\alpha(f) = \varepsilon_\alpha(f') = -1$). Figure 4 also illustrates the $+-$ and $--$ -nonsingular edges. For $v \in V$, we denote by $NS_\alpha(v)$, $NS_\alpha^+(v)$, and $NS_\alpha^-(v)$ the set of nonsingular, $+nonsingular$, and $--nonsingular$ edges incident to v , respectively. Also, the set of nonsingular edges is denoted as NS_α . The following are obtained by direct observations.

► **Lemma 1.** *Let α be any 4-coloring of a 3-colorable triangulation G of the 2-sphere.*

- (1) *A vertex $v \in V$ is recolorable with respect to α if and only if all edges incident to v are singular, i.e., $NS_\alpha(v) = \emptyset$.*
- (2) *The coloring α is a 3-coloring if and only if all edges are singular, i.e., $NS_\alpha = \emptyset$.*

We can derive a necessary condition for a 4-coloring α of G to belong to the 3-coloring component of $\mathcal{R}_4(G)$ as follows. Let α' be a 4-coloring obtained from α by changing the color of v , i.e., $\alpha'(v) \neq \alpha(v)$ and $\alpha'(u) = \alpha(u)$ for all $u \in V \setminus \{v\}$. Then the signatures of all faces in $St^2(v)$ are inverted (see also Figure 5):



■ **Figure 5** An example of the change of the signatures by a single-change. As in Figure 4, red and blue edges depict the +- and --nonsingular edges, respectively.

$$\varepsilon_{\alpha'}(f) = \begin{cases} -\varepsilon_{\alpha}(f) & \text{if } f \in \text{St}^2(v), \\ \varepsilon_{\alpha}(f) & \text{if } f \notin \text{St}^2(v). \end{cases} \quad (1)$$

This implies that, if α and α' belong to the same connected component in $\mathcal{R}_4(G)$, then we have $\#F_{\alpha}^+(v) = \#F_{\alpha'}^+(v)$ and $\#F_{\alpha}^-(v) = \#F_{\alpha'}^-(v)$ for all $v \in V$. Furthermore, Lemma 1 (2) implies that, if α is a 3-coloring of G , then we have $\varepsilon_{\alpha}(f) \neq \varepsilon_{\alpha}(f')$ for all $v \in V$ and all adjacent $f, f' \in \text{St}^2(v)$. Therefore, it follows from the above equation and Lemma 1 (2) that the following *balanced condition* holds if α belongs to the 3-coloring component of $\mathcal{R}_4(G)$:

(B) For each $v \in V$,

$$\#F_{\alpha}^+(v) = \#F_{\alpha}^-(v). \quad (2)$$

Our main result in this subsection is showing that the balanced condition (B) is also sufficient, that is, condition (B) characterizes the 3-coloring component of $\mathcal{R}_4(G)$.

► **Theorem 2.** *Let $\alpha: V \rightarrow \{0, 1, 2, 3\}$ be a 4-coloring of a 3-colorable triangulation G of the 2-sphere. Then, α belongs to the 3-coloring component of $\mathcal{R}_4(G)$ if and only if it satisfies the balanced condition (B).*

For the proof of Theorem 2, we observe the behavior of NS_{α} when we recolor a vertex from a 4-coloring α . If we change the color $\alpha(v)$ of a vertex v , then it follows from the equation (1) that all singular edges in $\text{Lk}(v)$ will be nonsingular and vice versa (see Figure 5). Thus, the following holds, where, for sets A and B , let $A \Delta B$ denote the symmetric difference $(A \setminus B) \cup (B \setminus A)$ of A and B .

► **Lemma 3.** *Let α be a 4-coloring of a 3-colorable triangulation G of the 2-sphere and α' a 4-coloring obtained from α by changing the color of a vertex v . Then*

$$\text{NS}_{\alpha'}(u) = \begin{cases} \text{NS}_{\alpha}(u) & \text{if } u \notin N(v), \\ \text{NS}_{\alpha}(u) \Delta (\text{Lk}(v) \cap \delta(u)) & \text{if } u \in N(v). \end{cases}$$

In particular, $\text{NS}_{\alpha'} = \text{NS}_{\alpha} \Delta (\text{Lk}(v) \cap E)$.

In our proof, the set NS_{α} of nonsingular edges is viewed as the disjoint union of noncrossing closed trails in G . Here, a *closed trail* is a closed walk such that all edges are distinct. For a closed trail C of G and a vertex $v \in V$, we denote by C_v the set of subpaths of C obtained from the restriction of C to $\delta(v)$, i.e., $C_v := \{\{e, e'\} \mid \{e, e'\} \text{ is a subpath of } C \text{ such that } e, e' \in \delta(v)\}$. A closed trail C of G is said to be *noncrossing* if for any vertex v , no pair of subpaths $P, P' \in C_v$ crosses in S^2 , i.e., P' is contained in the closure of a connected component of $| \text{St}(v) | \setminus P$ in S^2 , where P is viewed as a curve in

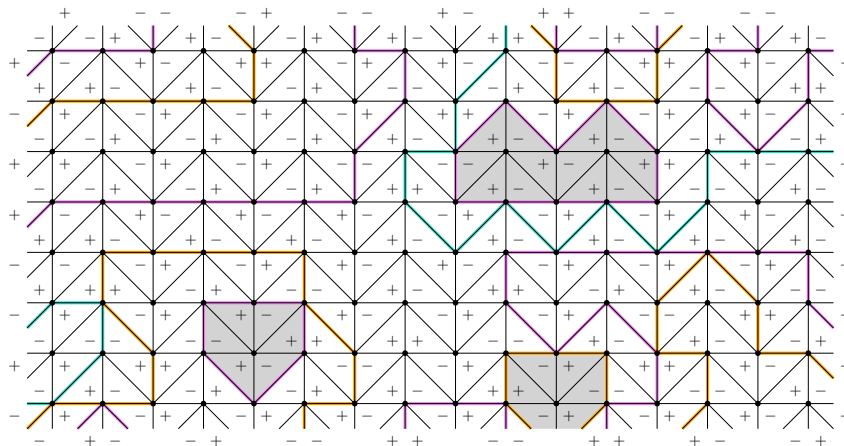


Figure 6 An example of NS-pairings. A part of a triangulation of a 2-sphere is depicted. Colors show closed trails. This NS-pairing is admissible. The gray areas show innermost closed trails.

S^2 . For a noncrossing closed trail C with a fixed orientation, we define L_C by the union of connected components of $S^2 \setminus C$ such that it lies on the left side of some edge in C . Similarly, we define R_C by the union of connected components of $S^2 \setminus C$ such that it lies on the right side of some edge in C . Since C is noncrossing, the family $\{L_C, R_C\}$ forms a bipartition of $S^2 \setminus C$.

By fixing a certain face $f_{\text{out}} \in F$ as the *outer face* of G , we can define the *volume* of a set of noncrossing closed trails in G as follows. We say that one of L_C and R_C is the *outside* of C if it contains the outer face f_{out} . The other is called the *inside* of C . Let $F_C \subseteq F$ denote the set of faces in the inside of C . Then, for a set \mathcal{C} of noncrossing closed trails in G , its *volume*, denoted as $\text{vol}(\mathcal{C})$, is the sum of the number of faces contained in the inside of C over all $C \in \mathcal{C}$, i.e., $\text{vol}(\mathcal{C}) := \sum_{C \in \mathcal{C}} \#F_C$. It is clear that $\text{vol}(\mathcal{C}) = 0$ if and only if $\mathcal{C} = \emptyset$. We will prove that any 4-coloring satisfying the balanced condition (B) has a recolorable vertex v such that, by changing the color of v , the volume of a set of noncrossing closed trails corresponding to the resulting 4-coloring strictly decreases from that of the original one. This implies that, by repeating this, we can obtain a 4-coloring such that its volume is zero, i.e., a 3-coloring.

We here see how NS_α corresponds to a set of noncrossing closed trails in G . It is known that, for any 4-coloring α of G and $v \in V$, the number $\#\text{NS}_\alpha(v)$ of nonsingular edges incident to v is even (see e.g., [14, Lemma 5]). For $v \in V$, let π_v be a partition of $\text{NS}_\alpha(v)$ such that each member of π_v is of size two (such a partition exists since $\#\text{NS}_\alpha(v)$ is even), and define $\pi := \bigcup_{v \in V} \pi_v$. We refer to π as an *NS-pairing* (with respect to α). An NS-pairing $\pi = \bigcup_{v \in V} \pi_v$ uniquely determines a family \mathcal{C}_π of closed trails in G satisfying that all closed trails in \mathcal{C}_π are disjoint and $\pi_v = \bigcup_{C \in \mathcal{C}_\pi} C_v$ for all $v \in V$. Note that NS_α equals the disjoint union of all closed trails $C \in \mathcal{C}_\pi$. Figure 6 provides an example of the set of closed trails induced by an NS-pairing.

An NS-pairing $\pi = \bigcup_{v \in V} \pi_v$ is said to be *admissible* if the following hold for any $v \in V$:

- (A1) All members of π_v consist of one $+$ -nonsingular edge and one $-$ -nonsingular edge;
- (A2) No two pairs $P, P' \in \pi_v$ cross in $|\text{St}(v)| \subseteq S^2$.

Let π be an admissible NS-pairing. Since each $C \in \mathcal{C}_\pi$ is noncrossing by (A2), the inside of C , and hence F_C , are well-defined. We define the face set family $\mathcal{L}_\pi \subseteq 2^F$ by $\mathcal{L}_\pi := \{F_C \mid C \in \mathcal{C}_\pi\}$.

43:10 Reconfiguration of Colorings in Triangulations of the Sphere

The admissibility of π induces interesting properties on \mathcal{C}_π and \mathcal{L}_π as follows. Here, a set family $\mathcal{F} \subseteq 2^A$ is said to be *laminar* if, for any $X, Y \in \mathcal{F}$, we have $X \subseteq Y$, $X \supseteq Y$, or $X \cap Y = \emptyset$.

► **Lemma 4.** *Let π be an admissible NS-pairing with respect to a 4-coloring α .*

- (1) *The restriction of α to C is a 2-coloring.*
- (2) *The family \mathcal{L}_π is laminar.*

Proof. (1). Take any member $\{\{u, v\}, \{v, w\}\}$ of π_v , which forms a subpath of some $C \in \mathcal{C}_\pi$. It suffices to show that $\alpha(u) = \alpha(w)$. We may assume that $\alpha(v) = 3$.

Let n_+ (resp. n_-) denote the number of $+$ -faces (resp. $-$ -faces) in $\text{St}^2(v) \cap F_C$. By the definition of the signature map ε_α , we have $\alpha(w) \equiv \alpha(u) + (n_+ - n_-) \pmod{3}$ or $\alpha(w) \equiv \alpha(u) - (n_+ - n_-) \pmod{3}$. Since π_v is noncrossing, the set of nonsingular edges incident to v in the inside of C is of the form of the union of a subset of π_v . Moreover, since all members of π_v consist of one $+$ -nonsingular edge and one $-$ -nonsingular edge, the number of $+$ -nonsingular edges incident to v in the inside of C equals that of $-$ -nonsingular edges. This implies that $n_+ = n_-$. Thus $\alpha(u) = \alpha(w)$ follows, as required.

(2). Take any two closed trails $C, C' \in \mathcal{C}_\pi$. Since π is admissible, in particular, no pair of members in π_v crosses in $|\text{St}(v)|$ for any $v \in V$, the closed trail C' is contained in either the inside or the outside of C . Thus, in the former case we have $F_{C'} \subseteq F_C$, and in the latter case we have $F_C \subseteq F_{C'}$ or $F_C \cap F_{C'} = \emptyset$, which implies that \mathcal{L}_π is laminar. ◀

Lemma 4 (2) implies that \mathcal{C}_π has an innermost closed trail in S^2 , which corresponds to a minimal set in \mathcal{L}_π .

We are ready to prove Theorem 2.

Proof of Theorem 2. We have already seen the only-if part. In the following, we show the if part. Let $\alpha: V \rightarrow \{0, 1, 2, 3\}$ be a 4-coloring of G satisfying the balanced condition (B) but not a 3-coloring, i.e., $\text{NS}_\alpha \neq \emptyset$ by Lemma 1 (2).

We first see that α has an admissible NS-pairing. Since $2 \cdot \#F_\alpha^+(v) = 2 \cdot \#\text{NS}_\alpha^+(v) + \#(\delta(v) \setminus \text{NS}_\alpha(v))$ and $2 \cdot \#F_\alpha^-(v) = 2 \cdot \#\text{NS}_\alpha^-(v) + \#(\delta(v) \setminus \text{NS}_\alpha(v))$, we have $\#\text{NS}_\alpha^+(v) = \#\text{NS}_\alpha^-(v)$ by (B). We construct an admissible NS-pairing as follows. For $v \in V$, let $\pi'_v := \emptyset$, $N_v^+ := \text{NS}_\alpha^+(v)$, and $N_v^- := \text{NS}_\alpha^-(v)$. While $N_v^+ \neq \emptyset$ and $N_v^- \neq \emptyset$, we take $e^+ \in N_v^+$ and $e^- \in N_v^-$ such that one of the connected components of $|\text{St}(v)| \setminus \{e^+, e^-\}$ contains no edges in $N_v^+ \cup N_v^-$ (such a pair (e^+, e^-) always exists) and update $\pi'_v \leftarrow \pi'_v \cup \{\{e^+, e^-\}\}$, $N_v^+ \leftarrow N_v^+ \setminus \{e^+\}$, and $N_v^- \leftarrow N_v^- \setminus \{e^-\}$. After the above procedure stops, we define π_v as the resulting π'_v . Then, we can see that π_v satisfies (A1) and (A2). Therefore, $\pi := \bigcup_{v \in V} \pi_v$ is an admissible NS-pairing.

The following claim is crucial for the proof of Theorem 2.

▷ **Claim 5.** There exists a recolorable vertex $v_0 \in V$ such that the 4-coloring α' obtained from α by recoloring v_0 has an admissible NS-pairing π' satisfying $\text{vol}(\mathcal{C}_{\pi'}) < \text{vol}(\mathcal{C}_\pi)$.

If this claim is true, then by recoloring such v_0 repeatedly, we finally obtain a 4-coloring α^* and an admissible NS-pairing π^* with respect to α^* such that $\text{vol}(\mathcal{C}_{\pi^*}) = 0$. The equality $\text{vol}(\mathcal{C}_{\pi^*}) = 0$ implies $\text{NS}_{\alpha^*} = \emptyset$, i.e., α^* is actually a 3-coloring by Lemma 1 (2). Therefore, α belongs to the 3-coloring component of $\mathcal{R}_4(G)$, as required.

In the following, we prove Claim 5. Take an arbitrary innermost closed trail $C \in \mathcal{C}_\pi$, the existence of which is guaranteed by Lemma 4 (2), and an edge $e = \{v_1, v_2\} \in C$. Let $\{v_0, v_1, v_2\}$ be the face in the inside of C , or in F_C . Since α is a 4-coloring, the color $\alpha(v_0)$ is different from both $\alpha(v_1)$ and $\alpha(v_2)$. Therefore v_0 does not belong to C by Lemma 4 (1),



■ **Figure 7** Reducing the volume. In this example, $\mathcal{C}' = \{C'_1, C'_2\}$.

implying that $\text{St}^2(v_0) \subseteq F_C$. Since C is an innermost closed trail, no edge incident to the vertex v_0 is nonsingular with respect to α . Thus, by Lemma 1 (1), we can change the color of v_0 .

Let α' be the 4-coloring obtained from α by changing the color of v_0 . For each $v \in N(v_0)$, we have $\#(\delta(v) \cap \text{Lk}(v_0)) = 2$, and denote $\delta(v) \cap \text{Lk}(v_0)$ by P_v . We define $\pi' = \bigcup_{v \in V} \pi'_v$ by

$$\pi'_v := \begin{cases} \pi_v & \text{if } v \notin N(v_0), \\ \pi_v \cup \{P_v\} & \text{if } v \in N(v_0) \text{ and } \text{NS}_\alpha(v) \cap \text{Lk}(v_0) = \emptyset, \\ (\pi_v \setminus \{P\}) \cup \{P \triangle P_v\} & \text{if } v \in N(v_0) \text{ and } \pi_v \text{ contains } P \text{ with } |P \cap P_v| = 1, \\ \pi_v \setminus \{P_v\} & \text{if } v \in N(v_0) \text{ and } \pi_v \text{ contains } P_v. \end{cases}$$

See also Figure 5. Then π' is an NS-pairing with respect to α' by Lemma 3.

Moreover, we can see that π' is admissible as follows. It is clear that $\pi'_v \cap \pi_v$ satisfies (A1) and (A2) for each $v \in V$, implying that π'_v satisfies (A1) and (A2) if $v \notin N(v_0)$, or $v \in N(v_0)$ and π_v contains P_v . Since the edge $\{v_0, v\}$ is singular with respect to α' , the path P_v (resp. $P \triangle P_v$) does not cross any $P' \in \pi_v$ (resp. $P' \in \pi_v \setminus \{P\}$); π'_v satisfies (A2) even for other v . Suppose that $P_v = \{\{u, v\}, \{v, w\}\}$. Then, we have $\varepsilon_{\alpha'}(\{u, v, v_0\}) = \varepsilon_\alpha(\{w, v, v_0\})$ and $\varepsilon_{\alpha'}(\{w, v, v_0\}) = \varepsilon_\alpha(\{u, v, v_0\})$. This implies that, if $v \in N(v_0)$ and π_v contains P with $|P \cap P_v| = 1$, then $P \triangle P_v$ consists of one $+$ -nonsingular edge and one $-$ -nonsingular edge with respect to α' , and if $v \in N(v_0)$ and $\text{NS}_\alpha(v) \cap \text{Lk}(v_0) = \emptyset$, then P_v consists of one $+$ -nonsingular edge and one $-$ -nonsingular edge with respect to α' . Thus π'_v satisfies (A1) for other v .

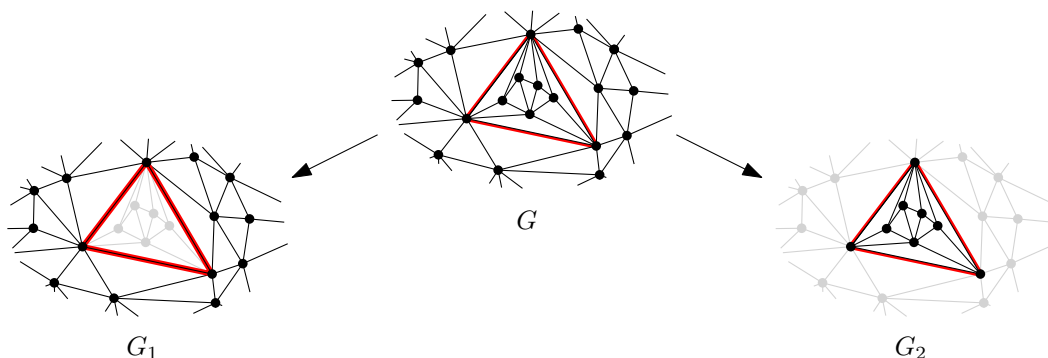
Let \mathcal{C}' be the set of the closed trails in $\mathcal{C}_{\pi'}$ containing some $e \in \text{Lk}(v_0) \cap \text{NS}_{\alpha'}$. Then, we have $F_C = \bigcup_{C' \in \mathcal{C}'} F_{C'} \cup \text{St}^2(v_0)$ and $\mathcal{C}_{\pi'} = \mathcal{C}_\pi \setminus \{C\} \cup \mathcal{C}'$. Therefore, we obtain $\text{vol}(\mathcal{C}_{\pi'}) = \text{vol}(\mathcal{C}_\pi) - \#\text{St}^2(v_0) < \text{vol}(\mathcal{C}_\pi)$; see also Figure 7

This completes the proof of the claim (and hence that of Theorem 2). ◀

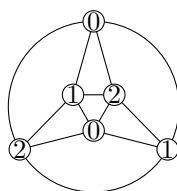
Our proof of Theorem 2 is constructive; for a 4-coloring α satisfying the balanced condition (B), we explicitly construct a sequence of single-changes from α to a certain 3-coloring α^* . This leads to the following.

► **Theorem 6** (★). *Let G be a 3-colorable triangulation of the 2-sphere. For any α and β belonging to the 3-coloring component of G , we can obtain in $O(n^2)$ time a sequence of single-changes of length $O(n^2)$ from α to β . In particular, the diameter of the 3-coloring component of G is $O(n^2)$.*

Theorems 2 and 6 immediately imply the polynomial-time solvability of 4-RECOLORING for G if one of the given α or β belongs to the 3-coloring component. We here note that, for a 4-coloring α of G , we can check if it satisfies the balanced condition (B) in $O(\#F) = O(n)$ time.



■ **Figure 8** A triangulation is split by the separating triangle C highlighted by red lines.



■ **Figure 9** The octahedral graph with a 3-coloring.

► **Corollary 7.** *Let G be a 3-colorable triangulation of the 2-sphere. 4-RECOLORING for G can be solved in $O(n)$ time, provided one of the input 4-colorings α and β belongs to the 3-coloring component of $\mathcal{R}_4(G)$. In addition, if both α and β belong to the 3-coloring component, then we can obtain a reconfiguration sequence from α to β in $O(n^2)$ time.*

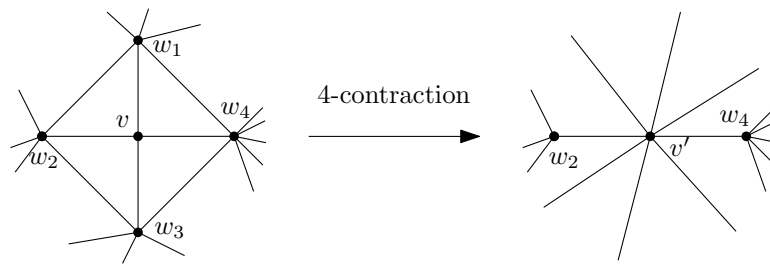
4 Connectedness of the 4-coloring reconfiguration graph

In this section, we solve the second question posed in Introduction: *In what 3-colorable triangulation of the 2-sphere all 4-colorings are single-equivalent?* To explain the answer, we introduce some notation. Since we deal with only the case of the 2-sphere in this section, we simply use the term a *triangulation* instead of a triangulation of the 2-sphere.

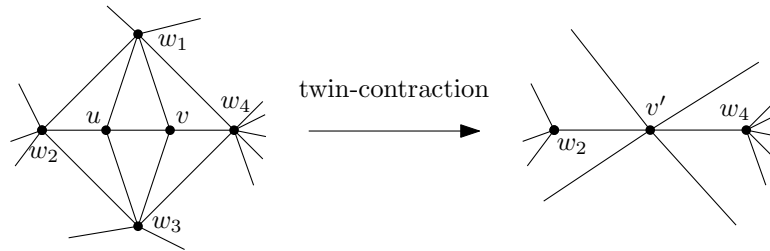
A *separating triangle* in a triangulation is a cycle of length 3 that does not bound a face. Note that a triangulation with at least five vertices is 4-connected if and only if it has no separating triangles. A triangulation with a separating triangle C can be split into two triangulations, the subgraph induced by the inside of C and that by the outside of C , respectively (Figure 8). Note that they share C . By iteratively applying this procedure to a triangulation G with k separating triangles, we obtain a collection of $k + 1$ triangulations without separating triangles. We call the $k + 1$ triangulations *4-connected pieces* of G . It is known [11] that the collection of the 4-connected pieces is uniquely determined. It is easy to see that G is 3-colorable if and only if every 4-connected piece of G is 3-colorable.

The *octahedral graph* is the 1-skeleton of the octahedron (Figure 9), which has six vertices, twelve edges, and eight faces, and is 3-colorable. A triangulation is said to be *octahedron-stacked* if every 4-connected piece of G is isomorphic to the octahedral graph. The following is the main theorem in this section.

► **Theorem 8 (*)**. *Let G be a 3-colorable triangulation. Then, $\mathcal{R}_4(G)$ is connected if and only if G is octahedron-stacked.*



■ **Figure 10** The 4-contraction of v at $\{w_1, w_3\}$.



■ **Figure 11** The twin-contraction of $\{u, v\}$ at $\{w_1, w_3\}$.

Since we can enumerate all separating triangles in linear time [9], the criterion in Theorem 8 can be used to obtain a linear-time algorithm for CONNECTEDNESS OF 4-COLORING RECONFIGURATION GRAPH for a 3-colorable triangulation of the 2-sphere, as follows.

► **Corollary 9.** CONNECTEDNESS OF 4-COLORING RECONFIGURATION GRAPH for a 3-colorable triangulation G of the 2-sphere is solvable in $O(n)$ time.

We prove Theorem 8 by combining some lemmas together with the so-called generating theorem. The following lemma deals with splitting a triangulation to obtain a 4-connected piece, and allows us to focus on 4-connected 3-colorable triangulations. Due to space limit, we leave a proof to the readers.

► **Lemma 10.** Let G be a 3-colorable triangulation with a separating triangle C , and let G_1 and G_2 be the two triangulations obtained by splitting along C . Then $\mathcal{R}_4(G)$ is connected if and only if both $\mathcal{R}_4(G_1)$ and $\mathcal{R}_4(G_2)$ are connected.

The if part of Theorem 8 is easily proven by Lemma 10 and the fact that $\mathcal{R}_4(G)$ is connected, where G is the octahedral graph. To prove the only if part, we now define two operations to reduce a 3-colorable triangulation G to a smaller triangulation G' as follows. Let v be a vertex of degree four in G and let $\{w_1, w_2, w_3, w_4, w_1\}$ be the cycle that forms the link of v . The 4-contraction of v at $\{w_1, w_3\}$, illustrated in Figure 10, is to remove v , identify the vertices w_1 and w_3 , and replace the two pairs of multiple edges obtained from $\{\{w_1, w_2\}, \{w_2, w_3\}\}$ and $\{\{w_1, w_4\}, \{w_3, w_4\}\}$ with two single edges, respectively. Let u and v be adjacent vertices of degree four, where $\{w_1, w_2, w_3, v, w_1\}$ and $\{w_1, u, w_3, w_4, w_1\}$ are the cycles that form the links of u and v , respectively. The twin-contraction of $\{u, v\}$ at $\{w_1, w_3\}$, illustrated in Figure 11, is to remove u and v , identify the vertices w_1 and w_3 , and replace the two pairs of multiple edges obtained from $\{\{w_1, w_2\}, \{w_2, w_3\}\}$ and $\{\{w_1, w_4\}, \{w_3, w_4\}\}$ with two single edges, respectively.

Notice that we do not perform these operations if they give rise to multiple edges. Matsumoto and Nakamoto proved the following generating theorem.

► **Theorem 11** ([25]). *For every 4-connected 3-colorable triangulation G , there exists a sequence G_0, G_1, \dots, G_ℓ from $G_0 := G$ such that G_ℓ is the octahedral graph, G_i is a 4-connected 3-colorable triangulation for $0 \leq i \leq \ell$, and G_i is obtained from G_{i-1} by either a 4-contraction or a twin-contraction for $1 \leq i \leq \ell$.*

For a 4-contraction and a twin-contraction, we need the following lemma.

► **Lemma 12.** *Let G be a 4-connected 3-colorable triangulation, and let G' be a 4-connected 3-colorable triangulation obtained from G by either a 4-contraction or a twin-contraction. If $\mathcal{R}_4(G')$ is disconnected, then so is $\mathcal{R}_4(G)$.*

It is not difficult to see that if the octahedral graph is obtained from a 3-colorable triangulation G by a 4-contraction, then $\mathcal{R}_4(G)$ is disconnected, and if the octahedral graph is obtained from a 3-colorable triangulation G by a twin-contraction, then G has a separating triangle, i.e. G is not 4-connected. Therefore, it follows from Theorem 11 and Lemma 12 that for a 4-connected 3-colorable triangulation G , $\mathcal{R}_4(G)$ is disconnected, unless G is the octahedral graph. By Lemma 10, this completes the proof of the only if part of Theorem 8.

The proof of Theorem 8 implies that if the answer to CONNECTEDNESS OF 4-COLORING RECONFIGURATION GRAPH is NO, then in a given 3-colorable triangulation G , we can find in polynomial time a 4-coloring that does not belong to the 3-coloring component of $\mathcal{R}_4(G)$. This would be a certificate for being a NO-instance.

5 PSPACE-completeness

As in Section 1, we show the following result in this section.

► **Theorem 13** (*). *For $k \geq 4$, the problem $(k + 1)$ -RECOLORING for $(k - 1)$ -colorable triangulations of the $(k - 2)$ -sphere is PSPACE-complete.*

When restricted to the case $k = 4$, Theorem 13 implies that 5-RECOLORING is PSPACE-complete even for planar 3-colorable triangulations (i.e., even triangulations).

In order to prove Theorem 13, we introduce a new recoloring problem. For a list coloring, we associate a *list assignment* $L = (L(v))_{v \in V(G)}$ with a graph G such that each $v \in V(G)$ is assigned a list $L(v)$ of colors. For a list assignment L of a graph G , a map α on $V(G)$ is an *L -coloring* if $\alpha(v) \in L(v)$ for every $v \in V(G)$ and $\alpha(u) \neq \alpha(v)$ for every $\{u, v\} \in E(G)$. For a graph G and a list assignment L of G , the *L -coloring reconfiguration graph*, denoted by $\mathcal{R}(G, L)$, is defined as follows: Its vertex set consists of all L -colorings of G and there is an edge between two L -colorings α and β of G if and only if β is obtained from α by recoloring only a single vertex in G . We consider the following reconfiguration problem named LIST-RECOLORING.

LIST-RECOLORING

Input: A graph G , a list assignment L of G , and two L -colorings α and β of G .

Output: YES if α and β are connected in $\mathcal{R}(G, L)$, and NO otherwise.

Bonsma and Cereceda [3] proved that LIST-RECOLORING is PSPACE-complete for particularly restricted graphs and list assignments.

We give a brief outline of the reduction from LIST-RECOLORING to 5-RECOLORING. In [3], restricted graphs are planar (not necessarily even triangulations) and a list of restricted list assignments is $\{0, 1, 2\}$ or $\{0, 1\}$. We construct an even triangulation graph from a restricted graph used in [3] by inserting some vertices and graphs into faces and consider a 5-coloring by using colors 0, 1, 2, 3, 4. Then, inserted graphs have a 5-coloring such that for each vertex v , all colors except for the color assigned to v appear in the neighbor of v . Such a 5-coloring

is called a frozen 5-coloring. We insert new graphs in such a way that their frozen 5-colorings do not conflict. Since the coloring in the inserted graphs are frozen, for each vertex v not contained in the original graph, all colors except for the color assigned to v appear in the neighbor of v , i.e., all vertices not contained in the original graph have the property being “frozen.” Therefore, the vertices contained in the original graph can only use colors in a restricted list assignment used in [3]. Consequently, 5-RECOLORING in our even triangulation is the same as LIST-RECOLORING in a restricted graph in [3].

References

- 1 Marthe Bonamy, Nicolas Bousquet, Carl Feghali, and Matthew Johnson. On a conjecture of Mohar concerning Kempe equivalence of regular graphs. *J. Combin. Theory, Ser. B*, 135:179–199, 2019. doi:10.1016/j.jctb.2018.08.002.
- 2 Marthe Bonamy, Vincent Delecroix, and Clement LeGrand-Duchesne. Kempe changes in degenerate graphs. *arXiv e-prints*, 2021. doi:10.48550/arXiv.2112.02313.
- 3 Paul Bonsma and Luis Cereceda. Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. *Theoret. Comput. Sci.*, 410(50):5215–5226, 2009. doi:10.1016/j.tcs.2009.08.023.
- 4 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Connectedness of the graph of vertex-colourings. *Discrete Math.*, 308:913–919, 2008. doi:10.1016/j.disc.2007.07.028.
- 5 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Mixing 3-colourings in bipartite graphs. *Europ. J. Combin.*, 30:1593–1606, 2009. doi:10.1016/j.ejc.2009.03.011.
- 6 Luis Cereceda, Jan van den Heuvel, and Matthew Johnson. Finding paths between 3-colourings. *J. Graph Theory*, 67:69–82, 2011. doi:10.1002/jgt.20514.
- 7 Sitan Chen, Michelle Delcourt, Ankur Moitra, Guillem Perarnau, and Luke Postle. Improved bounds for randomly sampling colorings via linear programming. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2216–2234. SIAM, 2019. doi:10.1137/1.9781611975482.134.
- 8 A.V. Chernavsky and V.P. Leksine. Unrecognizability of manifolds. *Annals of Pure and Applied Logic*, 141(3):325–335, 2006. doi:10.1016/j.apal.2005.12.011.
- 9 Norishige Chiba and Takao Nishizeki. Arboricity and subgraph listing algorithms. *SIAM J. Comput.*, 14(1):210–223, 1985. doi:10.1137/0214017.
- 10 Daniel W. Cranston and Landon Rabern. Brooks’ theorem and beyond. *J. Graph Theory*, 80:199–225, 2015. doi:10.1002/jgt.21847.
- 11 William H. Cunningham and Jack Edmonds. A combinatorial decomposition theory. *Canadian J. Math.*, 32(3):734–765, 1980. doi:10.4153/CJM-1980-057-7.
- 12 Carl Feghali. Kempe equivalence of 4-critical planar graphs. *arXiv e-prints*, 2021. doi:10.48550/arXiv.2101.04065.
- 13 Carl Feghali, Matthew Johnson, and Daniël Paulusma. Kempe equivalence of colourings of cubic graphs. *Europ. J. Combin.*, 59:1–10, 2017. doi:10.1016/j.ejc.2016.06.008.
- 14 Steve Fisk. Combinatorial structure on triangulations. I. The structure of four colorings. *Advances in Math.*, 11:326–338, 1973. doi:10.1016/0001-8708(73)90015-7.
- 15 Steve Fisk. Combinatorial structures on triangulations. II. Local colorings. *Advances in Math.*, 11:339–350, 1973. doi:10.1016/0001-8708(73)90016-9.
- 16 Steve Fisk. Geometric coloring theory. *Advances in Math.*, 24(3):298–340, 1977. doi:10.1016/0001-8708(77)90061-5.
- 17 Thomas P. Hayes, Juan Carlos Vera, and Eric Vigoda. Randomly coloring planar graphs with fewer colors than the maximum degree. *Random Struct. Algorithms*, 47(4):731–759, 2015. doi:10.1002/rsa.20560.
- 18 Percy J Heawood. On the four-colour map theorem. *Quart. J. Pure Appl. Math.*, 29:270–285, 1898.

- 19 Percy John Heawood. Map-colour theorems. *Quarterly J. Math.*, 24:332–338, 1890.
- 20 John E. Hopcroft and Robert Endre Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974. doi:10.1145/321850.321852.
- 21 Takehiro Ito, Yuni Iwamasa, Yusuke Kobayashi, Shun ichi Maezawa, Yuta Nozaki, Yoshio Okamoto, and Kenta Ozeki. Reconfiguration of colorings in triangulations of the sphere. *arXiv e-prints*, 2022. doi:10.48550/arXiv.2210.17105.
- 22 Takehiro Ito, Jun Kawahara, Yu Nakahata, Takehide Soh, Akira Suzuki, Junichi Teruyama, and Takahisa Toda. ZDD-based algorithmic framework for solving shortest reconfiguration problems. *arXiv e-prints*, 2022. doi:10.48550/arXiv.2207.13959.
- 23 Mark Jerrum. A very simple algorithm for estimating the number of k -colorings of a low-degree graph. *Random Struct. Algorithms*, 7(2):157–166, 1995. doi:10.1002/rsa.3240070205.
- 24 Michel Las Vergnas and Henri Meyniel. Kempe classes and the Hadwiger conjecture. *J. Combin. Theory, Ser. B*, 31:95–104, 1981. doi:10.1016/S0095-8956(81)80014-7.
- 25 Naoki Matsumoto and Atsuhiko Nakamoto. Generating 4-connected even triangulations on the sphere. *Discrete Math.*, 338:64–705, 2015. doi:10.1016/j.disc.2014.08.017.
- 26 Henri Meyniel. Les 5-colorations d’un graphe planaire forment une classe de commutation unique. *J. Combin. Theory, Ser. B*, 24:251–257, 1978. doi:10.1016/0095-8956(78)90042-4.
- 27 Bojan Mohar. Kempe equivalence of colorings. In *Graph theory in Paris*, Trends Math., pages 287–297. Birkhäuser, Basel, 2007. doi:10.1007/978-3-7643-7400-6_22.
- 28 Bojan Mohar and Jesús Salas. A new Kempe invariant and the (non)-ergodicity of the Wang–Swendsen–Kotecký algorithm. *J. Phys. A: Math. Theor.*, 42(22):225204, 2009. doi:10.1088/1751-8113/42/22/225204.
- 29 James R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley Publishing Company, Menlo Park, CA, 1984.
- 30 Naomi Nishimura. Introduction to reconfiguration. *Algorithms (Basel)*, 11(4):Paper No. 52, 25, 2018. doi:10.3390/a11040052.
- 31 Joachim H. Rubinstein. An algorithm to recognize the 3-sphere. In S. D. Chatterji, editor, *Proceedings of the International Congress of Mathematicians*, pages 601–611, Basel, 1995. Birkhäuser Basel.
- 32 Thomas L. Saaty. Thirteen colorful variations on Guthrie’s four-color conjecture. *American Math. Monthly*, 79:1, 1972. doi:10.2307/2978124.
- 33 Saul Schleimer. Sphere recognition lies in NP. In Michael Usher, editor, *Low-dimensional and Symplectic Topology*, volume 82 of *Proceedings of Symposia in Pure Mathematics*, pages 183–213, Providence, RI, 2011. American Mathematical Society.
- 34 Abigail Thompson. Thin position and the recognition problem for S^3 . *Mathematical Research Letters*, 1:613–630, 1994. doi:10.4310/MRL.1994.v1.n5.a9.
- 35 Jan van den Heuvel. The complexity of change. In *Surveys in combinatorics 2013*, volume 409 of *London Math. Soc. Lecture Note Ser.*, pages 127–160. Cambridge Univ. Press, Cambridge, 2013.
- 36 Eric Vigoda. Improved bounds for sampling colorings. *Journal of Mathematical Physics*, 41(3):1555–1569, 2000. doi:10.1063/1.533196.
- 37 I. A. Volodin, V. E. Kuznetsov, and A. T. Fomenko. The problem of discriminating algorithmically the standard three-dimensional sphere. *Russian Mathematical Surveys*, 29(5):71–172, 1974. doi:10.1070/RM1974v029n05ABEH001296.

On the Geometric Thickness of 2-Degenerate Graphs

Rahul Jain  

FernUniversität in Hagen, Germany

Marco Ricci  

FernUniversität in Hagen, Germany

Jonathan Rollin  

FernUniversität in Hagen, Germany

André Schulz  

FernUniversität in Hagen, Germany

Abstract

A graph is 2-degenerate if every subgraph contains a vertex of degree at most 2. We show that every 2-degenerate graph can be drawn with straight lines such that the drawing decomposes into 4 plane forests. Therefore, the geometric arboricity, and hence the geometric thickness, of 2-degenerate graphs is at most 4. On the other hand, we show that there are 2-degenerate graphs that do not admit any straight-line drawing with a decomposition of the edge set into 2 plane graphs. That is, there are 2-degenerate graphs with geometric thickness, and hence geometric arboricity, at least 3. This answers two questions posed by Eppstein [Separating thickness from geometric thickness. In *Towards a Theory of Geometric Graphs*, vol. 342 of *Contemp. Math.*, AMS, 2004].

2012 ACM Subject Classification Mathematics of computing → Graph theory; Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases Degeneracy, geometric thickness, geometric arboricity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.44

Related Version *Full Version*: <https://arxiv.org/abs/2302.14721> [16]

1 Introduction

A graph is planar if it can be drawn without crossings on a plane. Planar graphs exhibit many nice properties, which can be exploited to solve problems for this class more efficiently compared to general graphs. However, in many situations, graphs cannot be assumed to be planar even if they are sparse. It is therefore desirable to define graph classes that extend planar graphs. Several approaches for extending planar graphs have been established over the last years [4, 14]. Often these classes are defined via drawings, for which the types of crossings and/or the number of crossings are restricted. A natural way to describe how close a graph is to being a planar graph is provided by the graph parameter *thickness*. The thickness of a graph G is the smallest number $\theta(G)$ such that the edges of G can be partitioned into $\theta(G)$ planar subgraphs of G . Related graph parameters are *geometric thickness* and *book thickness*. Geometric thickness was introduced by Kainen under the name *real linear thickness* [17]. The geometric thickness $\bar{\theta}(G)$ of a graph G is the smallest number of colors that is needed to find an edge-colored geometric drawing (i.e., one with edges drawn as straight-line segments) of G with no monochromatic crossings. For the book thickness $\text{bt}(G)$, we additionally require that only geometric drawings with vertices in convex position are considered.

An immediate consequence from the definitions of thickness, geometric thickness and book thickness is that for every graph G we have $\theta(G) \leq \bar{\theta}(G) \leq \text{bt}(G)$. Eppstein shows that the three thickness parameters can be arbitrarily “separated”. Specifically, for any number k



© Rahul Jain, Marco Ricci, Jonathan Rollin, and André Schulz;
licensed under Creative Commons License CC-BY 4.0

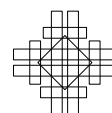
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 44; pp. 44:1–44:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



there exists a graph with geometric thickness 2 and book thickness at least k [10] as well as a graph with thickness 3 and geometric thickness at least k [11]. The latter result is particularly notable since any graph of thickness k admits a k -edge-colored drawing of G with no monochromatic crossings if edges are not required to be straight lines. This follows from a result by Pach and Wenger [22], stating that any planar graph can be drawn without crossings on arbitrary vertex positions with polylines.

Related to the geometric thickness is the *geometric arboricity* $\bar{a}(G)$ of a graph G , introduced by Dujmović and Wood [6]. It denotes the smallest number of colors that are needed to find an edge-colored geometric drawing of G without monochromatic crossings where every color class is acyclic. As every such plane forest is a plane graph, we have $\bar{\theta}(G) \leq \bar{a}(G)$. Moreover, every plane graph can be decomposed into three forests [24], and therefore $3\bar{\theta}(G) \geq \bar{a}(G)$.

Bounds on the geometric thickness are known for several graph classes. Due to Dillencourt et al. [5] we have $\frac{n}{5,646} + 0.342 \leq \bar{\theta}(K_n) \leq \frac{n}{4}$ for the complete graph K_n . Graphs with bounded degree can have arbitrarily high geometric thickness. In particular, as shown by Barárt et al. [2], there are d -regular graphs with n vertices and geometric thickness at least $c\sqrt{dn}^{1/2-4/d-\varepsilon}$ for every $\varepsilon > 0$ and some constant c . However, due to Duncan et al. [8], if the maximum degree of a graph is 4, its geometric thickness is at most 2. For graphs with treewidth t , Dujmović and Wood [6] showed that the maximum geometric thickness is $\lceil t/2 \rceil$. Hutchinson et al. [15] showed that graphs with n vertices and geometric thickness 2 can have at most $6n - 18$ edges. As shown by Durocher et al. [9], there are n -vertex graphs for any $n \geq 9$ with geometric thickness 2 and $2n - 19$ edges. In the same paper, it is proven that it is NP-hard to determine if the geometric thickness of a given graph is at most 2. Computing thickness [18] and book thickness [3] are also known to be NP-hard problems. For bounds on the thickness for several graph classes, we refer to the survey of Mutzel et al. [19]. A good overview on bounds for book thickness can be found on the webpage of Pupyrev [23].

A graph G is *d-degenerate* if every subgraph contains a vertex of degree at most d . So we can repeatedly find a vertex of degree at most d and remove it, until no vertices remain. The reversal of this vertex order (known as a *degeneracy order*) yields a construction sequence for G that adds vertex by vertex and each new vertex is connected to at most d previously added vertices (called its *predecessors*). Adding a vertex with exactly two predecessors is also known as a Henneberg 1 step [12]. In particular, any 2-degenerate graph is a subgraph of a Laman graph (i.e., a graph that is generically minimal rigid), however not every Laman graph is 2-degenerate. All d -degenerate graphs are (d, ℓ) -sparse, for any $\binom{d+1}{2} \geq \ell \geq 0$, that is, every subgraph on n vertices has at most $dn - \ell$ edges.

Our Results. In this paper, we study the geometric thickness of 2-degenerate graphs. Due to the Nash-Williams theorem [20, 21], every 2-degenerate graph can be decomposed into 2 forests and hence has arboricity at most 2 and therefore thickness at most 2. On the other hand, as observed by Eppstein [10], 2-degenerate graphs can have unbounded book thickness. Eppstein's examples of graphs with thickness 3 and arbitrarily high geometric thickness are 3-degenerate graphs [11]. Eppstein asks whether the geometric thickness of 2-degenerate graphs is bounded by a constant from above and whether there are 2-degenerate graphs with geometric thickness greater than 2. The currently best upper bound of $O(\log n)$ follows from a result by Duncan for graphs with arboricity 2 [7]. We improve this bound and answer both of Eppstein's questions with the following two theorems.

► **Theorem 1.** *For each 2-degenerate graph G we have $\bar{\theta}(G) \leq \bar{a}(G) \leq 4$.*

► **Theorem 2.** *There is a 2-degenerate graph G with $\bar{a}(G) \geq \bar{\theta}(G) \geq 3$.*

2 Proof of Theorem 1: The upper bound

In this section, we prove Theorem 1. To this end, we describe, for any 2-degenerate graph, a construction for a straight-line drawing such that the edges can be colored using four colors, avoiding monochromatic crossings and monochromatic cycles. This shows that 2-degenerate graphs have geometric arboricity, and hence geometric thickness, at most four.

Before we give a high-level description of the construction we introduce some definitions. For a graph G we denote its edge set with $E(G)$ and its vertex set with $V(G)$. Consider a 2-degenerate graph G with a given, fixed degeneracy order. We define the *height* of a vertex $\text{height}(v)$ as the length t of a longest path $u_0 \cdots u_t$ with $u_t = v$ such that for each $1 \leq i \leq t$ vertex u_{i-1} is a predecessor of u_i . The set of vertices of the same height is called a *level* of G . By definition, each vertex has at most two neighbors of smaller height.

Our construction process embeds G level by level with increasing height. The levels are placed alternately either strictly below or strictly to the right of the already embedded part of the graph. If a level is placed below, then we use specific colors v and vs (short for “vertical” and “vertical slanted”, respectively) for all edges between this level and levels of smaller height. Similarly, we use specific colors h and hs (short for “horizontal” and “horizontal slanted”, respectively) if a level is placed to the right. See Figure 1 (right).

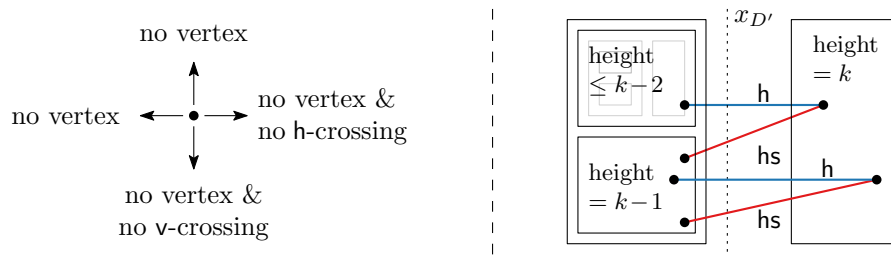
To make our construction work, we need several additional constraints to be satisfied in each step which we will describe next. For a point p in the plane, we use the notation $x(p)$ and $y(p)$ to refer to the x - and y -coordinates of p , respectively. Consider a drawing D of a 2-degenerate graph G of height k together with a coloring of the edges with colors $\{h, hs, v, vs\}$. For the remaining proof, we assume that each vertex of G has either 0 or exactly 2 predecessors. If not, we add a dummy vertex without predecessors to the graph and make it the second predecessor of all those vertices that originally only had 1 predecessor. We say that D is *feasible* if it satisfies the following constraints:

- (C1) For each vertex in G the edges to its predecessors are colored differently. If $k > 0$, then each vertex of height k in G is incident to one edge of color h and one edge of color hs .
- (C2) There exists some $x_D \in \mathbb{R}$ such that for each vertex $v \in V(G)$ we have $x(v) > x_D$ if and only if $\text{height}(v) = k$.
- (C3) There is no monochromatic crossing.
- (C4) No two vertices of G lie on the same horizontal or vertical line.
- (C5) Each $v \in V(G)$ is *h-open to the right*, that is, the horizontal ray emanating at v directed to the right avoids all h -edges.
- (C6) Each $v \in V(G)$ is *v-open to the bottom*, that is, the vertical ray emanating at v directed downwards avoids all v -edges.

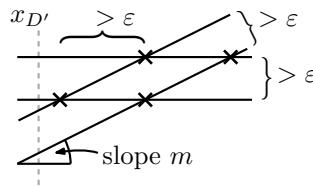
These constraints are schematized in Figure 1.

We now show how to construct a feasible drawing for G . We prove this using induction on the height of the graph. The base case $k = 0$ is trivial, as there are no edges in the graph. Assume that $k \geq 1$ and the theorem is true for all 2-degenerate graphs with height $k - 1$. Let H denote the subgraph of G induced by vertices with height less than k . By induction, there is a feasible drawing D of H .

As a first step, we reflect the drawing D at the straight line $x = -y$. Thus, a point (x, y) before transformation becomes $(-y, -x)$. Additionally, we swap the colors hs and vs as well as the colors h and v . Let D' denote the resulting drawing. From now on, all appearing coordinates of vertices refer to coordinates in D' . By construction, D' satisfies (C3–C6). Applying (C1) to D shows that in D' each vertex of height $k - 1$ is incident to one edge of color v and one edge of color vs . Applying (C2) to D shows that there exists $y_{D'} \in \mathbb{R}$ such that for each vertex $v \in V(H)$ we have $y(x) < y_{D'}$ in D' if and only if $\text{height}(v) = k - 1$.



■ **Figure 1** Left: For each vertex v in a feasible drawing, there are no other vertices on the vertical and the horizontal line through v . Moreover, v is h -open to the right and v -open to the bottom. Right: All vertices in the highest level (of height k) are placed to the right of all vertices of smaller height. Moreover, each vertex in that level is incident to one edge of color h and one edge of color hs .



■ **Figure 2** Horizontal lines intersecting straight lines of slope m . Conditions (ii–iv) are illustrated.

As the second (and last) step, we place the points of height k of G such that the resulting drawing is feasible. Let L_k denote the set of these vertices and let $x_{D'}$ denote the largest x -coordinate among all vertices in D' . Choose a sufficiently small slope m , with $m > 0$, and a sufficiently small ϵ , with $\epsilon > 0$, such that the following holds.

- (i) For any distinct $u, v \in V(H)$ with $y(u) < y(v)$, the horizontal line through v and the straight line through u with slope m intersect at a point p with $x(p) > x_{D'}$.
- (ii) For any distinct $u, v \in V(H)$ we have that $\epsilon < |y(u) - y(v)|$.
- (iii) For any distinct $u, u', v, v' \in V(H)$ let p be the intersection point of the straight line through u with slope m and the horizontal line through v and let p' be the intersection point of the straight line through u' with slope m and the horizontal line through v' . If $x(p) \neq x(p')$, then $\epsilon < |x(p) - x(p')|$.
- (iv) For any distinct $u, v \in V(H)$ we have that ϵ is smaller than the distance between the two straight lines of slope m through u and v , respectively.

The constraints are summarized in Figure 2. Such a choice of m and ϵ is possible, by choosing m according to Condition (i) first and then ϵ according to the Conditions (ii–iv).

For each vertex $w \in L_k$ let u and v be the two predecessors of w in H with $y(u) < y(v)$ and let p^w denote the intersection point of the straight line of slope m passing through u (called a slanted line) and the horizontal line passing through v . We will place w close to p^w and connect w to v using an edge of color h and we connect w to u using an edge of color hs . To determine the exact location of the vertices, we consider the horizontal lines through vertices $v \in V(H)$ from bottom to top (with increasing y -coordinate) and for each such line consider the intersections with slanted lines through vertices $u \in V(H)$ with $y(u) < y(v)$ from left to right (with increasing x -coordinate). Let p_1, \dots, p_t denote the intersection points in the order just described. For each intersection point p_i let ℓ_i denote the straight line through p_i with slope $-1/m$ (which is negative as $m > 0$), that is, ℓ_i is perpendicular to straight lines of slope m . Every vertex $w \in L_k$ with $p^w = p_i$ will be placed on ℓ_i at a certain distance from p^w (specified later). Note that there might be multiple points with the same

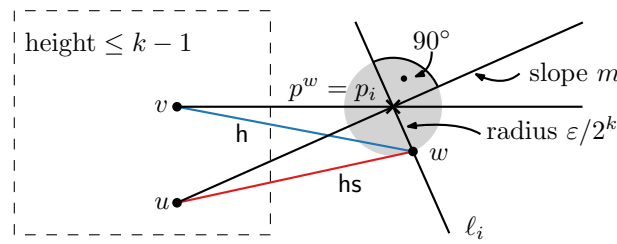


Figure 3 The placement of the k^{th} point w in order of vertices in L_k .

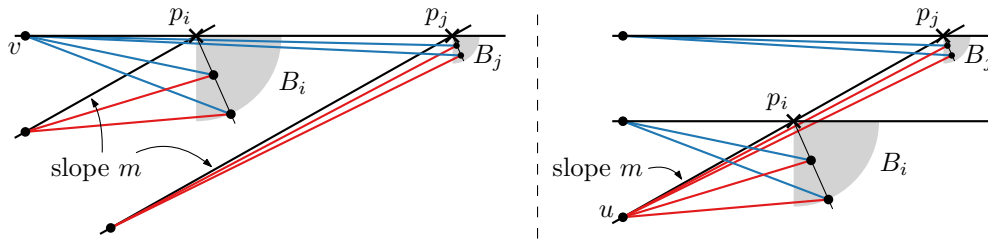


Figure 4 The placement of several points with a common “horizontal” predecessor v (left) or a common “slanted” predecessor u (right). Edges with color h are drawn blue, edges with color hs are drawn red.

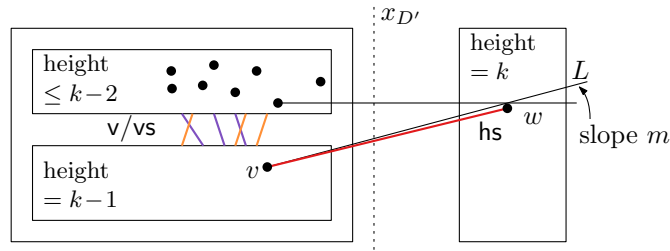
predecessors and hence multiple vertices $w \in L_k$ with $p^w = p_i$. For each p_i we order all such vertices arbitrarily. This gives an ordering of all vertices in L_k based on the ordering p_1, \dots, p_t . If w is the k^{th} vertex in this order, w is placed on ℓ_i to the bottom-right of p_i at distance $\epsilon/2^k$ from p_i ; see Figure 3. In this fashion, all vertices in L_k are placed with decreasing distance to their respective intersection point; see Figure 4.

We call the resulting drawing D_G . We claim that D_G demonstrates that the geometric arboricity of G is at most four.

Vertices on distinct points, edges intersect in at most one point in D_G . For each $i \leq t$ let δ_i denote the distance between p_i and the first vertex w placed close to p_i . Then $\delta_i \leq \delta_{i-1}/2$ for each i with $1 < i \leq t$. For each $i \leq t$ let B_i be the region formed by all points $q \in \mathbb{R}$ of distance at most δ_i to p_i with $x(q) > x(p_i)$ and $y(q) < y(p_i)$ (B_i is a quarter of a disk). Then all vertices $w \in L_k$ with $p^w = p_i$ are placed on distinct points along the intersection of the line ℓ_i with B_i ; see Figure 4.

Due to Conditions (ii) and (iv), all the regions B_i are disjoint. By construction, no two vertices are placed on the same point within a region B_i . This shows that no two vertices in G are placed on the same point in D_G . Moreover, for the same reasons, for each vertex $v \in V(H)$ the edges between v and vertices in L_k do not contain vertices in their interior and intersect in v only. This shows no edge in G contains vertices in its interior and any two edges in G intersect in at most one point.

- (C1) By construction, each vertex in L_k is incident to an edge of color h and an edge of color hs . Hence, D_G satisfies (C1).
- (C2) By Condition (i), any horizontal line through some vertex of H and a slanted straight line through a vertex of height $k - 1$ in H intersect in some point with x -coordinate larger than $x_{D'}$. Each vertex $w \in L_k$ is placed slightly to the right of such an intersection point. Hence, D_G satisfies (C2) with $x_{D_G} = x_{D'}$.
- (C3) The edges in the drawing D' of H were not changed, so there are still no monochromatic crossings of those edges. Consider an edge vw with $v \in V(H)$ and $w \in L_k$.



■ **Figure 5** Checking Constraint (C3) for hs -colored edges.

First, assume that its color is h . Then $x(w) > x(v)$ and $y(w) < y(v)$ by construction. Consider an edge e of color h in H . We shall prove that e does not cross vw . If both endpoints of e lie above v , then e does not cross vw . If e crosses the horizontal line through v in some point p , then $x(p) < x(v)$ since v is h -open to the right in D' . Moreover, one endpoint of e lies above v while the other endpoint lies below w due to Condition (ii). So e does not cross vw . If both endpoints of e lie below v , then their y -coordinates are smaller than $y(w)$ due to Condition (ii). Hence, e does not cross vw in either case.

Now consider an edge $v'w'$ of color h with $v' \in V(H)$, $y(v') < y(v)$ and $w' \in L_k$. As $y(w) > y(v')$ by Condition (ii) and $y(w') < y(v')$ by construction, these two edges do not cross. This shows that edges of color h do not cross in D_G .

Now assume that the color of vw is hs . By construction, v is the predecessor of w of the smallest y -coordinate. Since w has at least one predecessor of height $k - 1$ and, by induction, all vertices of this height are placed below the vertices of smaller height in D' , we have that $\text{height}(v) = k - 1$. Consider the slanted straight line L (of slope m) through v . By Condition (i), L does not intersect the convex hull of vertices of height less than $k - 1$ in D' ; see Figure 5. By induction, all vertices of height $k - 1$ in H are incident to edges of color v and vs only. Hence, L does not intersect any edge of color hs in D' . The edge vw has a positive slope slightly smaller than L and hence does not intersect any edge of color hs in D' either. It remains to show that vw does not intersect edges $v'w'$ of color hs with $v' \in V(H)$, $v' \neq v$, and $w' \in L_k$. Consider the slanted straight line L' (of slope m) through v' . Without loss of generality, assume that L is above L' (the case $L = L'$ produces no crossing since then $v = v'$). The edge $v'w'$ has a positive slope slightly smaller than L' . By Condition (iv), the distance between L and w is smaller than the distance between L and L' . Hence vw does not cross $v'w'$.

This shows that edges of color hs do not cross in D_G and hence D_G satisfies (C3).

(C4) No two vertices from H lie on a common vertical or horizontal line by induction. Consider $w \in L_k$ and the region B_i containing w . Due to Condition (ii) no horizontal line through B_i contains a vertex from H . Moreover, by (C2) no vertical line through B_i contains a vertex from H . Note that either two different regions B_i/B_j are separated by a horizontal line or $y(p_i) = y(p_j)$. In both cases, vertices placed in B_i/B_j cannot have the same y -coordinate. This is clear in the former case and in the latter it is true since we never select the same distance from p_i/p_j when placing the vertices. For the x -coordinates we can argue similarly. Hence, D_G satisfies (C4).

(C5) First, consider a vertex $v \in V(H)$ and the horizontal ray L emanating at v to the right. In the drawing D' , each vertex in H is h -open to the right, so L does not intersect any h -colored edge from H . It remains to consider h -colored edges $v'w'$ with $v' \in V(H)$ and $w' \in L_k$. Then $x(w) > x(v')$ and $y(v') > y(w) > y(v') - \varepsilon$ by construction. So if $y(v') < y(v)$, L does not intersect $v'w'$. If $y(v') > y(v)$, then observe

that $y(w) > y(v') - \varepsilon > y(v)$ by Condition (ii). Hence L does not intersect $v'w$ in either case and v is h-open to the right in G_D .

Now consider a vertex $w \in L_k$ and the horizontal ray L emanating at w to the right. By (C2), L does not intersect any edge from H . It remains to consider h-colored edges $v'w'$ with $w' \in L_k$. Let v be the neighbor of w in H with vw colored h.

If $v' = v$, consider the region B_i containing w . If w' is in B_i , then w' and w lie on the diagonal ℓ_i in B_i . If w' is in B_j with $j < i$, then w' is placed to the left of w , and if w' is on B_j with $j > i$, then w' is placed above w . In either case, L does not intersect $v'w'$.

Now suppose that $v' \neq v$. Assume that $y(v') < y(v)$ then by Condition (ii) and by construction $y(w) > y(v') > y(w')$. If on the other hand $y(v') > y(v)$ then $y(v') > y(w') > y(v) > y(w)$, again by Condition (ii) and by construction. In both cases, it follows that L does not intersect $v'w'$.

This shows that each vertex of G is h-open to the right in D_G .

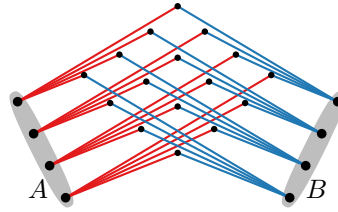
(C6) In the drawing D' , each vertex in H is v-open to the bottom. The vertices in L_k are not incident to any edges of color v. Hence, all vertices of G are v-open to the bottom in D_G . So (C6) is satisfied.

No monochromatic cycles. (C1–C6) are satisfied, thus D_G is feasible, and uses 4 colors. Consider any cycle in G and a vertex w of largest height in the cycle. Then its neighbors u and v in the cycle have to be its predecessors. Due to (C1), uw and vw do not have the same color. Hence there are no monochromatic cycles.

3 Proof of Theorem 2: The lower bound

In this section, we shall describe a 2-degenerate graph with geometric thickness at least 3. For a positive integer n let $G(n)$ denote the graph constructed as follows. Start with a vertex set Λ_0 of size n and for each pair of vertices from Λ_0 add one new vertex adjacent to both vertices from the pair. Let Λ_1 denote the set of vertices added in the last step. For each pair of vertices from Λ_1 add 89 new vertices, each adjacent to both vertices from the pair. Let Λ_2 denote the set of vertices added in the last step. For each pair of vertices from Λ_2 add one new vertex adjacent to both vertices from the pair. Let Λ_3 denote the set of vertices added in the last step. This concludes the construction. Observe that for each $i = 1, 2, 3$, each vertex in Λ_i has exactly two neighbors in Λ_{i-1} . Hence, $G(n)$ is 2-degenerate. We claim that for sufficiently large n the graph $G(n)$ has geometric thickness at least 3. To prove this result, we need several geometric and topological insights that are summarized in the following lemmas.

Let G_k denote the grid formed by k horizontal straight-line segments crossing k vertical straight-line segments. The grid G_k has four *sides*: the sets of left and right endpoints of the horizontal segments and the sets of lower and upper endpoints of the vertical segments form the four sides of G_k , respectively. The first and the last horizontal segment and the first and the last vertical segment form the *boundary* of G_k while all other segments are called the *inner edges* of G_k . We call an arrangement of straight-line segments combinatorially equivalent to G_k a *k-grid*. Here, we call two arrangements of straight lines or straight-line segments *combinatorially equivalent* if the embeddings given by the arrangement of their graphs (skeletons) are combinatorially equivalent. We point out that a *k-grid* sometimes refers to a set of disjoint red segments and a set of disjoint blue segments where every pair of red/blue segment intersects; e.g., [1]. Note that our definition is more restrictive. Among others, no two segments share an endpoint in our notion of a *k-grid*. The following lemma shows how both concepts are related. A proof is given in the full version [16, Section 3.1].



■ **Figure 6** A tidy drawing of H_4 , the full 1-subdivision of $K_{4,4}$. In particular, edges incident to A do not cross each other, edges incident to B do not cross each other, and, hence, there are no three pairwise crossing edges.

► **Lemma 3.** *Each arrangement of $k2^{k-1}$ disjoint red straight-line segments and k disjoint blue straight-line segments, where each red segment crosses each blue segment, contains a k -grid.*

In the following, we need a grid-structure with some additional properties summarized in the following definitions. For any point set Q in the plane, we call a straight-line segment in the plane a Q -edge if it has an endpoint in Q . We call two point sets A and B *separated* if $A \cup B$ is in convex position and the convex hull of A does not intersect the convex hull of B (that is, along the boundary of the convex hull of $A \cup B$ the sets do not interleave).

Consider a complete bipartite graph $K_{n,n}$ with bipartition classes A and B . Let H_n denote the graph obtained from $K_{n,n}$ by subdividing each edge exactly once. Let C denote the set of subdivision vertices of H_n . Observe that each edge of H_n has one endpoint in C and the other endpoint in $A \cup B$, and hence is either an A -edge or a B -edge. We call a geometric drawing of H_n *tidy*, if A and B are separated, there is no crossing between any two A -edges, and there is no crossing between any two B -edges. Figure 6 shows a tidy drawing of H_4 . Note that we make no (convexity) assumptions on the positions of subdivision vertices. Since A and B are separated, a tidy drawing induces an ordering of A and B by traversing these points along the convex hull of $A \cup B$ in the counterclockwise direction starting with the vertices in A . An edge of H_n is called an *inner edge* if it is not incident to the first or last vertex of A and not incident to the first or last vertex of B in the order given above. Similarly, we call an edge of the underlying copy of $K_{n,n}$ an *inner edge* if it corresponds to two inner edges of H_n .

Consider a k -grid T in H_n with one side in A and one side in B (and the respective opposite sides in C). We call the sides of T that are contained in A or B the A -side and B -side, respectively. Let a_1, \dots, a_k denote the vertices of the A -side of T in the order given by A and let b_1, \dots, b_k denote the vertices of the B -side of T in the order given by B . For each i let x_i^A denote the crossing point between the A -edge of T with endpoint a_i and the B -edge in T farthest away from a_i . For $i, j \leq k$, with $i < j$, the $A_{i,j}$ -corridor of T is the polygon enclosed by $x_i^A, a_i, a_{i+1}, \dots, a_j, x_j^A$. Crossing points x_1^B, \dots, x_k^B and $B_{i,j}$ -corridors are defined similarly. Figure 7 (right) shows examples of such corridors. A *tidy k -grid* is a topological subgraph T of a tidy drawing of H_n such that

- T is a k -grid with one side in A and one side in B (and the opposite sides in C),
- for each $i \leq k$, the segment $a_i x_i^A$ is contained in the $A_{1,k}$ -corridor of T ,
- for each $i \leq k$, the segment $b_i x_i^B$ is contained in the $B_{1,k}$ -corridor of T .

Figure 7 shows a tidy 3-grid and a 4-grid that is not tidy.

Our arguments require a tidy grid such that every cell contains a (subdivision) vertex from C . Such a grid is called *dotted*. The following lemma shows that we can always find a suitable dotted grid. A proof is given in the full version [16, Section 3.2].

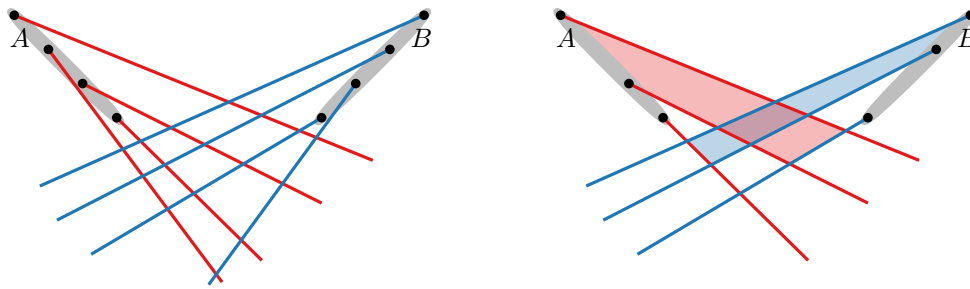


Figure 7 Left: A 4-grid with sides in A and B that is not tidy: there is a (red) A -edge not contained in the $A_{1,k}$ -corridor as well as a (blue) B -edge not contained in the $B_{1,k}$ -corridor. Right: A tidy (sub)grid. The $A_{1,2}$ -corridor and the $B_{2,3}$ -corridor are highlighted.

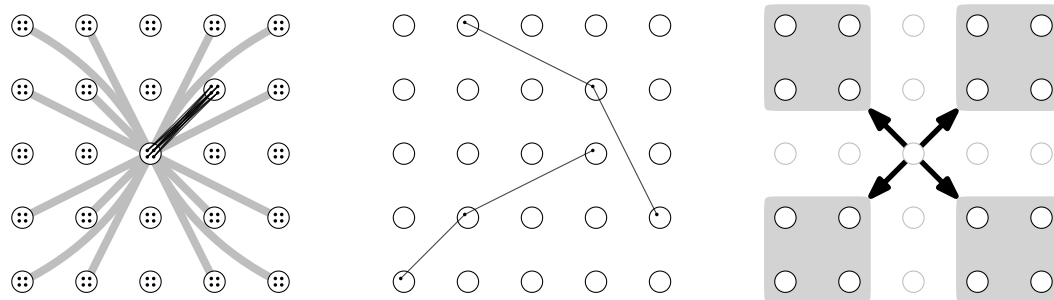


Figure 8 Left: An illustration of $\Gamma(5,4)$. Only edges incident to the central vertices are sketched. Middle: Two monotone paths in $\Gamma(5,4)$. Right: The four $(3,3)$ -quadrants.

► **Lemma 4.** *There is a constant c_2 such that for any integers n and k , with $n \geq 2^{c_2 k^4 2^{8k}}$ and $k \geq 3$, each tidy drawing of H_n contains a dotted tidy k -grid.*

Next, we will consider connections between the vertices inside of a dotted grid. To find such connections running in certain directions within the grid, we shall use a Ramsey type argument, summarized in the following Lemma 5. We will apply this lemma in such a way that the mentioned color r corresponds to connections within the grid. For positive integers k and t let $\Gamma(k, t)$ denote the graph whose vertex set consists of disjoint sets V_i^j , $i, j \leq k$, on t vertices each, such that $u \in V_i^j$ and $v \in V_p^q$ are adjacent if and only if $i \neq p$ and $j \neq q$. See Figure 8 (left) for an illustration of $\Gamma(5, 4)$. Let $r \geq 3$. We call an r -coloring of $E(\Gamma(k, t))$ *admissible* if each monochromatic copy of K_5 is of color r and any path uvw is not monochromatic in some color c with $3 \leq c < r$ in case $u \in V_i^j$, $v \in V_p^q$, and $w \in V_x^y$ with $1 \leq i < p < x \leq k$ and with $1 \leq j < q < y \leq k$ or $1 \leq y < q < j \leq k$. Loosely speaking, $\Gamma(k, t)$ is the t -blowup of the complement of a $k \times k$ -grid graph, and an r -coloring is admissible if any monochromatic copy of K_5 has color r and each monotone monochromatic path on at least two edges is colored with some color in $\{1, 2, r\}$. Given i and j , the (i, j) -quadrants of $\Gamma(k, t)$ are the four subgraphs induced by $\bigcup_{p < i, q < j} V_p^q$, $\bigcup_{p < i, q > j} V_p^q$, $\bigcup_{p > i, q < j} V_p^q$, and $\bigcup_{p > i, q > j} V_p^q$, respectively. See Figure 8 for an illustration. A proof of the following lemma is given in the full version [16, Section 3.3].

► **Lemma 5.** *Let r and t denote positive integers. There is a constant c_3 such that for each $k \geq c_3$ and each admissible r -coloring of $E(\Gamma(k, t))$ there are $i, j \leq k$ such that each vertex in V_i^j is incident to four edges of color r with endpoints in different (i, j) -quadrants.*

We also use the following bound on Erdős–Szekeres numbers.

44:10 On the Geometric Thickness of 2-Degenerate Graphs

► **Lemma 6** ([13]). *There is a constant c_4 such that for each positive integer k each set of $2^{k+c_4\sqrt{k\log k}}$ points in general position in the plane contains a subset of k points in convex position.*

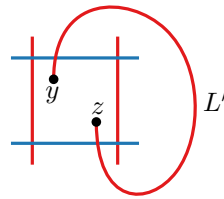
Finally, we prove that the graph $G(n)$ described in the beginning of this section has geometric thickness at least 3.

► **Theorem 7.** *Let k, m, n be integers with $k \geq c_3$ (with c_3 from Lemma 5 for $r = 11$ and $t = 5$), $n \geq 2^{c_2 2^{10k^2}}$ (with c_2 from Lemma 4) and $m \geq 12^n$. For each $N \geq 2^{2m+c_4\sqrt{2m\log(2m)}}$ (with c_4 from Lemma 6) the graph $G(N)$ has geometric thickness at least 3.*

Proof. Consider any geometric drawing of $G = G(N)$. We assume that the vertices are in general position, otherwise we can apply a small perturbation at the vertices to achieve this without introducing any new crossings. For the sake of a contradiction, suppose that there is a partition of G into two plane subgraphs \mathbb{A} and \mathbb{B} . We refer to the sets $\Lambda_0, \Lambda_1, \Lambda_2$, and Λ_3 as points sets like in the definition of G . Our proof proceeds as follows. We find a large tidy drawing of H_n with base points in Λ_0 and subdivision vertices in Λ_1 . Lemma 4 guarantees a dotted grid in this drawing. Then we consider the connections of the vertices in the grid cells via Λ_2 . We use Lemma 5 to show that many connections stay within the grid and hence many vertices of Λ_2 lie in the grid as well. Finally, we consider the connections of vertices from Λ_2 within the grid and use Lemma 5 again, to find a configuration of vertices from Λ_2 that leads to a contradiction.

Consider the point set Λ_0 . Lemma 6 yields a set $\Lambda'_0 \subseteq \Lambda_0$ of $2m$ points in convex position, since $N \geq 2^{2m+c_4\sqrt{2m\log(2m)}}$. We consider the points in Λ'_0 in counterclockwise order with an arbitrary first vertex. Consider the copy of H_m in G between the set A of the first m vertices of Λ'_0 and the set B of the last m vertices of Λ'_0 . The edges of the underlying copy of $K_{m,m}$ are of four different types: in H_m they correspond to two edges from \mathbb{A} , or to two edges from \mathbb{B} , or one edge from \mathbb{A} and one edge from \mathbb{B} (where either the edge from \mathbb{A} has an endpoint in A and the edge from \mathbb{B} has an endpoint in B or vice versa). Since $m \geq 12^n$ there is, due to the bipartite Ramsey theorem (precise statement given in the full version [16, Lemma 6]), a copy of $K_{n,n}$ with all edges of the same type, leading to a corresponding copy H of H_n . Since $n \geq 3$, this type cannot be one of the types with edges only from \mathbb{A} or only from \mathbb{B} as both \mathbb{A} and \mathbb{B} are planar but $K_{3,3}$ is not. Without loss of generality, assume that all edges in H incident to A are in \mathbb{A} and all edges incident to B are in \mathbb{B} . Observe that H is a tidy geometric drawing of H_n since \mathbb{A} and \mathbb{B} are crossing-free and the sets A and B are separated (their convex hulls do not intersect and $A \cup B = \Lambda'_0$ is in convex position). Further note that $2^{2k^2} \geq (k^2 + 1)^4 2^8$ for $k \geq 4$. Hence $n \geq 2^{c_2 2^{10k^2}} \geq 2^{c_2 (k^2+1)^4 2^{8(k^2+1)}}$, and there is, by Lemma 4, a dotted tidy $(k^2 + 1)$ -grid T in H with vertices from Λ_1 in the cells.

Let $\Lambda'_1 \subseteq \Lambda_1$ denote a set of vertices consisting of one vertex from each cell of T . Consider the graph Γ_1 with vertex set Λ'_1 where two vertices are adjacent if and only if they are in distinct rows and distinct columns of T . Then Γ_1 forms a copy of $\Gamma(k^2, 1)$. We will define an edge coloring Φ of Γ_1 based on the drawing of the edges between Λ'_1 and Λ_2 . Consider two vertices $x, x' \in \Lambda'_1$. There are 89 vertices in Λ_2 adjacent to both x and x' . We will distinguish 11 different cases how the edges between such $y \in \Lambda_2$ and x, x' are drawn. Then, by the pigeonhole principle, there will be nine vertices from Λ_2 with the same type of drawing of xy and $x'y$. The cases are not disjoint from each other and we break ties arbitrarily. If there are nine vertices $y \in \Lambda_2$ with $xy, x'y \in E(\mathbb{A})$, then $\Phi(xx') = 1$. If there are nine such vertices with $xy, x'y \in E(\mathbb{B})$, then $\Phi(xx') = 2$. Now assume that there are no such nine vertices, so there are 73 such vertices where one edge is from \mathbb{A} and the other edge is from

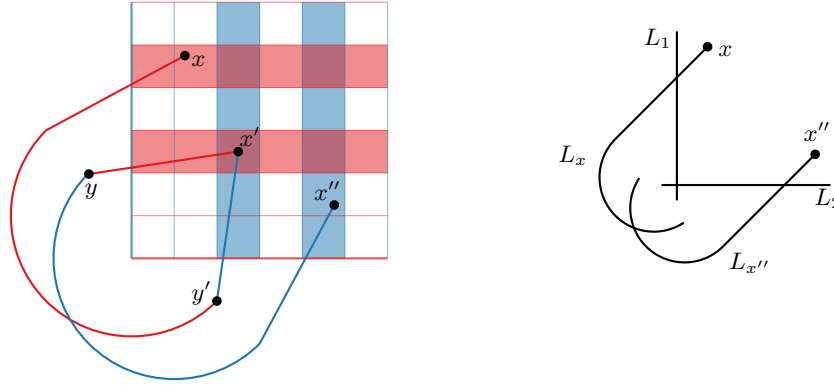


■ **Figure 9** This arrangement is not realizable by straight-line segments, since the straight line through L' does not intersect any of the other lines twice and does not intersect itself.

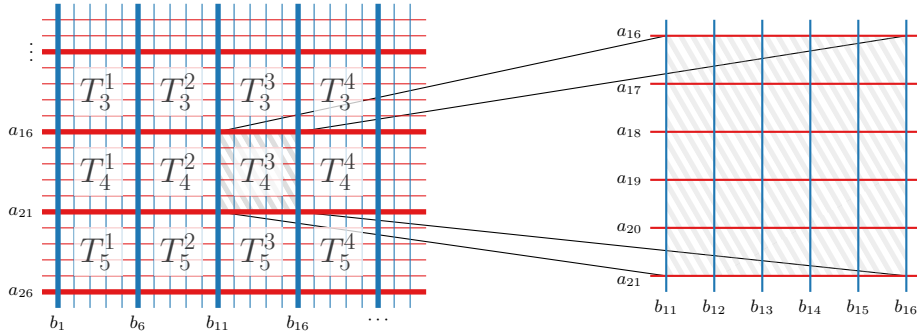
\mathbb{B} . These edges either leave T or stay within T . If we have at least nine vertices that stay within T , we pick $\Phi(xx') = 11$. Otherwise, we can assume that there are at least 65 vertices y , for which the bicolored path xyx' leaves T . The cell containing x is the intersection of an A -corridor and a B -corridor of T . So an edge xy intersects the boundary of T either at one of the two “ends” of the A -corridor (if $xy \in E(\mathbb{A})$) or at one of the two “ends” of the B -corridor (if $xy \in E(\mathbb{B})$). Similarly, an edge $x'y$ has four options to leave T . Also observe that each of xy and $x'y$ can intersect the boundary of T only once, see Figure 9. The figure shows the boundary edges of T and a supposedly straight-line segment L' intersecting the boundary twice. This arrangement can't be realized by straight lines as the straight line through L' intersects itself once or some other line twice otherwise. This gives 8 possibilities how the intersections can be located (under the assumption that xy and $x'y$ are not both in \mathbb{A} and not both in \mathbb{B}). We use colors $3, \dots, 10$ to encode these possibilities. Whenever there is a set \hat{Y} of nine vertices from Λ_2 such that the paths xyx' have the same locations of intersections for all $y \in \hat{Y}$, the edge xx' receives the corresponding color. If xx' is neither colored with 1, 2, or 11, we have at least 65 vertices connected via leaving T , and therefore at least one of the eight possibilities how to leave T occurs nine times. So Φ is well defined (up to breaking ties arbitrarily).

We claim that Φ is admissible. We first prove that colors $3, \dots, 10$ do not induce a monotone monochromatic path on two edges. For the sake of a contradiction, suppose that there is such a path $xx'x''$. By symmetry, we assume that there are vertices y, y' and edges $xy', x'y \in E(\mathbb{A})$, and $x'y', x''y \in E(\mathbb{B})$ such that xy' and $x'y$ leave T at the same sides of their respective A -corridors and $x'y'$ and $x''y$ leave T at the same sides of their respective B -corridors. The situation is depicted in Figure 10. We claim that this arrangement is not stretchable. To see this consider the 4-cycle between the intersections of $xy', x''y$ and the grid boundary as depicted in Figure 10 (right). This cycle needs to be embedded as a quadrilateral. For two opposing corners (the depicted crossings L_1/L_2 and $L_x/L_{x''}$) we have to embed the edges such that the “stubs” lie in the inside of the quadrilateral. To achieve this for one corner we need an incident concave angle in the quadrilateral and hence the realization of the quadrilateral would require at least two concave angles, which is not possible. Hence, such an arrangement is not stretchable. As a consequence, the colors $3, \dots, 10$ do not induce a monotone monochromatic path on two edges. This immediately shows that these colors also do not induce a monochromatic copy of K_5 . The color classes 1 and 2 correspond to subgraphs of the plane graphs \mathbb{A} and \mathbb{B} , respectively. Hence, they do not induce monochromatic copies of K_5 as well. This shows that all monochromatic copies of K_5 are of color $r = 11$. Therefore, Φ is admissible.

Now divide the $(k^2 + 1)$ -grid T into k^2 many $(k + 1)$ -grids T_i^j , with $i, j, \leq k$, where T_i^j consists of the A -edges on position $(i - 1)k + 1, \dots, ik + 1$ (in the ordering of A) and the B -edges with positions $(j - 1)k + 1, \dots, jk + 1$ (in the ordering of B). See Figure 11. Let Γ_i^j denote the subgraph of Γ_1 corresponding to T_i^j . Then Γ_i^j is a copy of $\Gamma(k, 1)$ and Φ is an



■ **Figure 10** Left: A monotone path that is monochromatic under Φ in some color in $\{3, \dots, 10\}$. Note that it is not possible that $x'y$ and $x'y'$ intersect. Right: The edges from the left part forming an arrangement that can't be realized by straight-line segments.



■ **Figure 11** A $(k^2 + 1)$ -grid T contains k^2 many $(k + 1)$ -subgrids. (Here $k = 5$.)

admissible 11-coloring of Γ_i^j . Consider some fixed $i, j \leq k$. Due to the choice of k there is, by Lemma 5, an edge xx' in Γ_i^j of color $r = 11$ (we do not need the stronger statement of the lemma here). Hence, there is a set $Y_i^j \in \Lambda_2$ of nine vertices such that for each $y \in Y_i^j$ the edges xy and $x'y$ stay within T . Let \mathcal{A}_x and \mathcal{B}_x denote the A -corridor and B -corridor whose intersection forms the cell containing x . Similarly, let $\mathcal{A}_{x'}$ and $\mathcal{B}_{x'}$ denote the respective corridors for the cell containing x' . As argued above, edges within T cannot leave their respective corridors. So each $y \in Y_i^j$ lies either in the cell $\mathcal{A}_x \cap \mathcal{B}_{x'}$ or in the cell $\mathcal{B}_x \cap \mathcal{A}_{x'}$. By the pigeonhole principle, there is a set $\tilde{Y}_i^j \subseteq Y_i^j$ of five vertices that lie in the same cell of T . Note that this cell is contained in T_i^j .

Consider the copy of $\Gamma(k, 5)$ whose vertex set consists of the union of all sets \tilde{Y}_i^j , with $i, j \leq k$, where two vertices $y \in \tilde{Y}_i^j$ and $y' \in \tilde{Y}_{i'}^{j'}$ are connected if and only if $i \neq i'$ and $j \neq j'$. For any two vertices $y, y' \in V(\Gamma(k, 5))$ there is a (unique) vertex in Λ_3 adjacent to both vertices. We define a coloring Ψ of the edges of $\Gamma(k, 5)$ similar to the coloring Φ above, except that the color of an edge yy' in $\Gamma(k, 5)$ is determined by the drawing of the unique edges yz and $y'z$, $z \in \Lambda_3$ (instead of a set of nine edge pairs behaving identically). Then Ψ is admissible by arguments similar to those applied for Φ . Due to the choice of k there are, by Lemma 5, indices $i, j \leq k$ such that each vertex in \tilde{Y}_i^j is incident to four edges of color 11 under Ψ with endpoints in different (i, j) -quadrants of $\Gamma(k, 5)$. See Figure 12 for illustrations.

Let $Y = \tilde{Y}_i^j$ for the specific indices i and j from above. Consider the A -corridor \mathcal{A} and B -corridor \mathcal{B} of T whose intersection forms the cell containing the set Y . For a vertex $y \in Y$ consider four vertices y_1, \dots, y_4 from different quadrants with $\Psi(yy_\ell) = 11$, $\ell = 1, \dots, 4$.

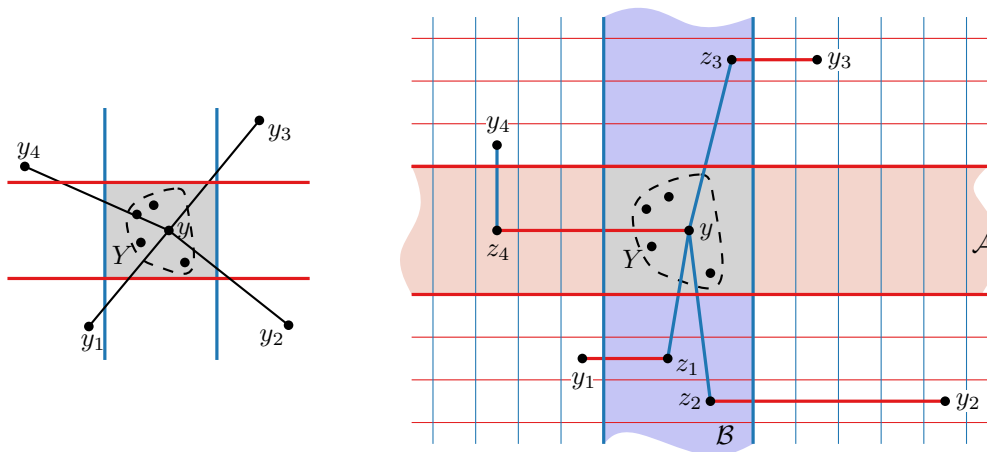


Figure 12 Left: Every vertex $y \in Y$ is incident to four edges in $\Gamma(k, 5)$ of color 11 with endpoints in different quadrants. Right: In G , each $y \in Y$ has two edges of the same type (\mathbb{A}/\mathbb{B}) that leave in the same direction relative to y ; here yz_1 and yz_2 .

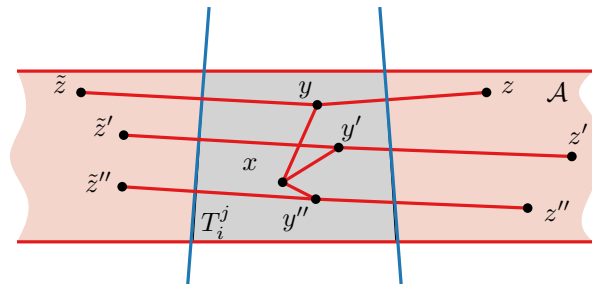
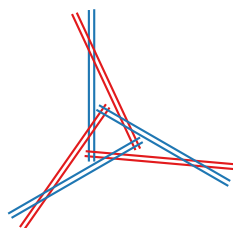


Figure 13 Construction in the proof of Theorem 7. Obtaining a monochromatic crossing at xy , xy' or xy'' is unavoidable.

Each edge $yy_\ell \in \Gamma(k, 5)$ corresponds to two edges (of G) yz_ℓ and $y_\ell z_\ell$ for some $z_\ell \in \Lambda_3$ such that z_ℓ lies within T . In particular, z_ℓ lies either in \mathcal{A} or in \mathcal{B} but not in the cell containing y . As y_1, \dots, y_4 are from four different quadrants, two of the vertices z_1, \dots, z_4 lie in \mathcal{A} or two lie in \mathcal{B} . Moreover, for either \mathcal{A} or \mathcal{B} two vertices lie on different “sides” of y within the corridor. If for y we have $|\mathcal{A} \cap \{z_1, \dots, z_4\}| \geq 2$ and at least two of these vertices lie on different sides in \mathcal{A} relative to y , we call y an \mathcal{A} -vertex, otherwise we call y a \mathcal{B} -vertex.

To get a contradiction we now show that Y contains at most two \mathcal{A} -vertices and at most two \mathcal{B} -vertices, which violates $|Y| = 5$. Due to the choice of $Y \subseteq Y_i^j$, there are vertices $x, x' \in V(T_i^j) = V(\Gamma_i^j)$ such that there are edges $xy \in E(\mathbb{A})$ and $x'y \in E(\mathbb{B})$ with $\Phi(xy) = \Phi(x'y) = 11$. That is, $xy \in \mathcal{A}$ and $x'y \in \mathcal{B}$. For the sake of a contradiction, suppose that there are three \mathcal{A} -vertices y, y', y'' in Y . Then there are three vertices $\tilde{y}, \tilde{y}', \tilde{y}'' \in V(\Gamma(k, 5)) \subseteq \Lambda_2$ and three vertices $z, z', z'' \in \Lambda_3$ such that $yz, y'z', y''z'' \in E(\mathbb{A})$, $\tilde{y}z, \tilde{y}'z', \tilde{y}''z'' \in E(\mathbb{B})$, and z, z', z'' lie in \mathcal{A} on the same side relative to y , but not in T_i^j . By the same reasoning we can find three vertices $\tilde{z}, \tilde{z}', \tilde{z}''$ such that $y\tilde{z}, y'\tilde{z}', y''\tilde{z}'' \in E(\mathbb{A})$, but now these vertices lie on the other side in \mathcal{A} relative to x (but also outside T_i^j). The edges $L = \{y\tilde{z}, yz, y'\tilde{z}', y'z', y''\tilde{z}'', y''z''\}$ split T_i^j in four zones. In one of these zones, x has to be located. No matter which zone we pick, there will always be a crossing of an edge from $\{xy, xy', xy''\} \subseteq E(\mathbb{A})$ with an edge in $L \subseteq E(\mathbb{A})$ (see Figure 13), a contradiction. Consequently, there are no three \mathcal{A} -vertices in Y .



■ **Figure 14** This arrangement of $3k$ red segments and $3k$ blue segments contains no copy of G_{k+1} . For each color and each slope there are k parallel segments (here $k = 2$ is depicted).

Similarly, there are no three \mathcal{B} -vertices in Y . This contradicts $|Y| \geq 5$. Hence, the geometric thickness of G is at least 3. ◀

Theorem 2 is a direct consequence of Theorem 7.

4 Conclusions

We proved that the largest geometric thickness among 2-degenerate graphs is either 3 or 4, answering two questions posed by Eppstein [11]. It remains open to decide whether there is a 2-degenerate graph of geometric thickness or geometric arboricity 4.

Our proof of the lower bound shows a geometric thickness of at least 3 for a tremendously large 2-degenerate graph. This is mainly due to using several rounds of Ramsey type arguments. We make little attempts to reduce this size and there are several places in the proof where a smaller size could be attained easily, for instance by using better or more specific Ramsey numbers (Lemma 5). In one step in the proof (Lemma 3) we are given a collection of red and blue straight-line segments in the plane and we need to find k red segments and k blue segments forming a grid combinatorially equivalent to G_k (which is formed by k horizontal segments crossing k vertical lines). We need exponentially many segments to be given, however it seems that a linear number suffices. An arrangement of $3k$ red segments and $3k$ blue segments without copy of G_{k+1} is given in Figure 14.

► **Question 1.** *Given an arrangement of $3k$ disjoint red straight-line segments and $3k$ disjoint blue straight-line segments, where each red segment crosses each blue segment, are there always k red segments and k blue segments forming a grid combinatorially equivalent to G_k ?*

The 2-degenerate graphs form a subclass of Laman graphs, which in turn form a subclass of all graphs of arboricity 2. Our lower bound gives a graph of geometric thickness 3 in either of these classes. However, for both larger classes it is unknown whether the geometric thickness is bounded by a constant from above.

References

- 1 Eyal Ackerman, Jacob Fox, János Pach, and Andrew Suk. On grids in topological graphs. *Computational Geometry*, 47(7):710–723, 2014. doi:10.1016/j.comgeo.2014.02.003.
- 2 János Barát, Jiří Matoušek, and David R. Wood. Bounded-degree graphs have arbitrarily large geometric thickness. *The Electronic Journal of Combinatorics*, 13:R3, 2006. doi:10.37236/1029.
- 3 Fan R. K. Chung, Frank T. Leighton, and Arnold L. Rosenberg. Embedding graphs in books: A layout problem with applications to VLSI design. *SIAM Journal on Algebraic Discrete Methods*, 8(1):33–58, 1987. doi:10.1137/0608002.
- 4 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Computing Surveys*, 52:1–37, January 2020. doi:10.1145/3301281.

- 5 Michael B. Dillencourt, David Eppstein, and Daniel S. Hirschberg. Geometric thickness of complete graphs. *Journal of Graph Algorithms & Applications*, 4(3):5–17, 2000. [arXiv:math/9910185](#), [doi:10.7155/jgaa.00023](#).
- 6 Vida Dujmović and David R. Wood. Graph treewidth and geometric thickness parameters. *Discrete & Computational Geometry*, 37:641–670, 2007. [arXiv:math/0503553](#), [doi:10.1007/s00454-007-1318-7](#).
- 7 Christian A. Duncan. On graph thickness, geometric thickness, and separator theorems. *Computational Geometry*, 44:95–99, February 2011. [doi:10.1016/j.comgeo.2010.09.005](#).
- 8 Christian A. Duncan, David Eppstein, and Stephen G. Kobourov. The geometric thickness of low degree graphs. In *SCG '04: Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pages 340–346, New York, NY, USA, June 2004. Association for Computing Machinery. [doi:10.1145/997817.997868](#).
- 9 Stephane Durocher, Ellen Gethner, and Debajyoti Mondal. Thickness and colorability of geometric graphs. *Computational Geometry*, 56:1–18, 2016. [doi:10.1016/j.comgeo.2016.03.003](#).
- 10 David Eppstein. Separating geometric thickness from book thickness. Preprint, 2001. [arXiv:math/0109195](#).
- 11 David Eppstein. Separating thickness from geometric thickness. In János Pach, editor, *Towards a Theory of Geometric Graphs*, volume 342 of *Contemporary Mathematics*. American Mathematical Society, 2004. [arXiv:math/0204252](#).
- 12 Lebrecht Henneberg. Die Graphische Statik der Starren Körper. In Felix Klein and Conrad Müller, editors, *Encyklopädie der Mathematischen Wissenschaften mit Einschluss ihrer Anwendungen: Vierter Band: Mechanik*, pages 345–434, Wiesbaden, Germany, 1908. B. G. Teubner Verlag. [doi:10.1007/978-3-663-16021-2_5](#).
- 13 Andreas F. Holmsen, Hossein Nassajian Mojarrad, János Pach, and Gábor Tardos. Two extensions of the Erdős–Szekeres problem. *Journal of the European Mathematical Society*, 22(12):3981–3995, 2020. [doi:10.4171/JEMS/1000](#).
- 14 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs. Communications of NII Shonan Meetings*. Springer, 2020. [doi:10.1007/978-981-15-6533-5](#).
- 15 Joan P. Hutchinson, Thomas C. Shermer, and Andrew Vince. On representations of some thickness-two graphs. *Computational Geometry*, 13(3):161–171, 1999. [doi:10.1016/S0925-7721\(99\)00018-8](#).
- 16 Rahul Jain, Marco Ricci, Jonathan Rollin, and André Schulz. On the geometric thickness of 2-degenerate graphs. Preprint, 2023. [arXiv:2302.14721](#).
- 17 Paul C. Kainen. Thickness and coarseness of graphs. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 39:88–95, 1973. [doi:10.1007/BF02992822](#).
- 18 Anthony Mansfield. Determining the thickness of graphs is NP-hard. *Mathematical Proceedings of the Cambridge Philosophical Society*, 93(1):9–23, 1983. [doi:10.1017/S030500410006028X](#).
- 19 Petra Mutzel, Thomas Odenthal, and Mark Scharbodt. The thickness of graphs: A survey. *Graphs and Combinatorics*, 14:59–73, 1998. [doi:10.1007/PL00007219](#).
- 20 Crispin St.J. A. Nash-Williams. Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society*, 36:445–450, 1961. [doi:10.1112/jlms/s1-36.1.445](#).
- 21 Crispin St.J. A. Nash-Williams. Decomposition of finite graphs into forests. *Journal of the London Mathematical Society*, 39:12, 1964. [doi:10.1112/jlms/s1-39.1.12](#).
- 22 János Pach and Rephael Wenger. Embedding planar graphs at fixed vertex locations. *Graphs and Combinatorics*, 17(4):717–728, 2001. [doi:10.1007/PL00007258](#).
- 23 Sergey Pupyrev. Linear Graph Layouts. Accessed: 2022-11-30. URL: <https://spupyrev.github.io/linearlayouts.html>.
- 24 Walter Schnyder. Embedding planar graphs on the grid. In *SODA '90: Proceedings of the first annual ACM-SIAM Symposium on Discrete Algorithms*, pages 138–148. Society for Industrial and Applied Mathematics, January 1990. [doi:10.5555/320176.320191](#).

The Localized Union-Of-Balls Bifiltration

Michael Kerber  

Institute of Geometry, Technische Universität Graz, Austria

Matthias Söls 

Institute of Geometry, Technische Universität Graz, Austria

Abstract

We propose an extension of the classical union-of-balls filtration of persistent homology: fixing a point q , we focus our attention to a ball centered at q whose radius is controlled by a second scale parameter. We discuss an absolute variant, where the union is just restricted to the q -ball, and a relative variant where the homology of the q -ball relative to its boundary is considered. Interestingly, these natural constructions lead to bifiltered simplicial complexes which are not k -critical for any finite k . Nevertheless, we demonstrate that these bifiltrations can be computed exactly and efficiently, and we provide a prototypical implementation using the CGAL library. We also argue that some of the recent algorithmic advances for 2-parameter persistence (which usually assume k -criticality for some finite k) carry over to the ∞ -critical case.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry; Mathematics of computing \rightarrow Algebraic topology

Keywords and phrases Topological Data Analysis, Multi-Parameter Persistence, Persistent Local Homology

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.45

Related Version *Full Version*: <https://arxiv.org/abs/2303.07002>

Supplementary Material A prototypical implementation can be found here: https://bitbucket.org/mkerber/demo_absolute_2d/src/master/.

Funding The authors acknowledge the support of the Austrian Science Fund (FWF).

Michael Kerber: Austrian Science Fund (FWF) grant P 33765-N.

Matthias Söls: Austrian Science Fund (FWF): grant W1230.

Acknowledgements The authors thank Anton Gfrerrer and Thomas Pock for helpful discussions.

1 Introduction

In the past years, the theory of *multi-parameter persistent homology* has gained increasing popularity. This theory extends the theory of (single-parameter) persistent homology by filtering a data set with several scale parameters and observing how the topological properties change when altering the ensemble of parameters. Most standard examples define two scale parameters, where the first one is based on the distance within the data set and the second one on its local density. The motivation for that choice is an increased robustness against outliers in the data set.

Localized bifiltrations. We suggest a different type of bifiltration where the second parameter controls the *locality* of the data. Let P be a finite set of data points in Euclidean space \mathbb{R}^d and $q \in \mathbb{R}^d$ a further point that we call the *center*. For two real values $s, r \geq 0$, we define

$$L_{s,r} = \left(\bigcup_{p \in P} B_s(p) \right) \cap B_r(q)$$



© Michael Kerber and Matthias Söls;

licensed under Creative Commons License CC-BY 4.0

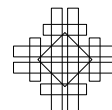
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 45; pp. 45:1–45:19

Leibniz International Proceedings in Informatics

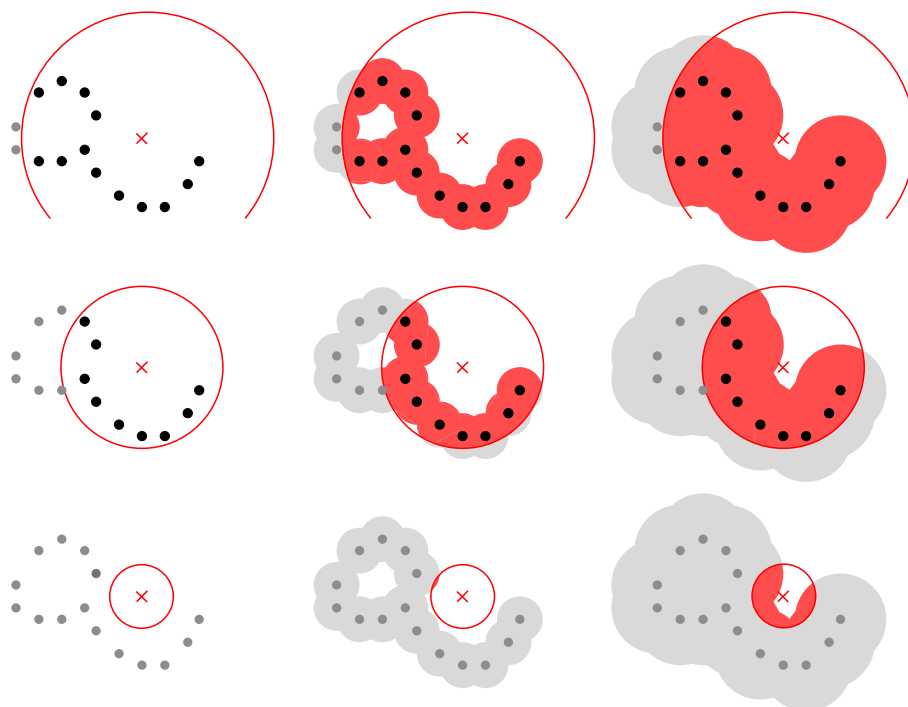


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



where $B_\alpha(x)$ is the ball of radius $\sqrt{\alpha}$ centered at x (taking the square root is not standard, but will be convenient later). In other words, we consider the union of balls around the data points (as in many applications of persistent homology), but we limit attention to a neighborhood around the center. It is immediate that $L_{s,r} \subseteq L_{s',r'}$ for $s \leq s'$ and $r \leq r'$ so $L := (L_{s,r})_{s,r \geq 0}$ is a nested sequence of spaces. We define the collection of spaces L to be the *absolute localized bifiltration*, see Figure 1 for an illustration. The goal of this paper is to compute a combinatorial representation of this bifiltration: a bifiltration of simplicial complexes which is homotopy equivalent to the absolute localized bifiltration at every choice (s, r) of parameters.

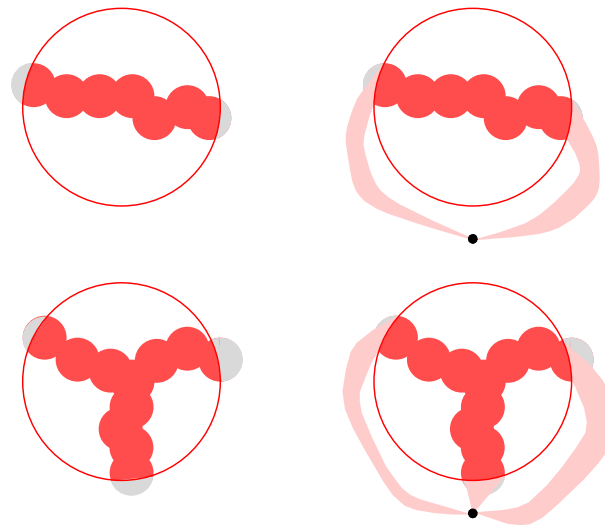
Alternatively, we consider the variant where all points of $L_{s,r}$ on the boundary of $B_r(q)$ are identified (see Figure 2). This version gives rise to a bifiltration that we call the *relative localized bifiltration*. This sometimes reveals more local information around q (as in Figure 2) and is more frequently used in applications (see related work below). Again, we are asking for an equivalent simplicial description.



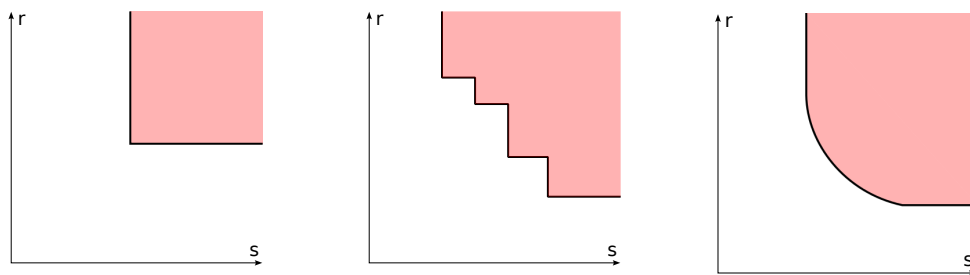
■ **Figure 1** Illustration of $L_{s,r}$. The radius s controls the radius around the points in P (black dots) and grows in the horizontal direction. The radius r of the center point (red cross) grows in vertical direction. The sets $L_{s,r}$ are marked in dark, red color.

An interesting feature of these localized bifiltrations is that topological changes arise along curves in the two-dimensional parameter space spanned by s and r . The perhaps simplest example is obtained by setting $d = 1$, $q = 0$ and $P = \{1\}$. Then, $L_{s,r} \neq \emptyset$ if and only if $r + s \geq 1$. Hence, the empty and non-empty regions are separated by a line in the parameter space. This implies that any equivalent simplicial bifiltration is ∞ -critical, meaning that there is no integer k for which it is k -critical. See Figure 3 for an illustration of k -criticality.

Given that ∞ -critical simplicial bifiltrations are obtained from such a simple construction, we pose the question whether such bifiltrations allow for an efficient algorithmic treatment. In this paper, we focus on the first step, the *generation* of such bifiltrations. This requires



■ **Figure 2** Left: The two examples of $L_{s,r}$ are both contractible and therefore cannot be distinguished further by homological methods. Right: The quotient space of $L_{s,r}$ relative to $L_{s,r} \cap \partial B_r(q)$ can be visualized by coning all points on the boundary of the ball with a (virtual) vertex, drawn as a black dot here. In the upper example, the resulting space has one hole, whereas the lower example has two. Therefore the homology of the spaces changes and allows for distinction.



■ **Figure 3** The active region and entry curve of a fixed simplex in the parameter space of a simplicial bifiltration, if it is 1-critical (left), 4-critical (middle) and ∞ -critical (right). For a more formal definition of k -criticality, see, for instance, [19].

to compute, for every simplex its *entry curve*, that is, the boundary between the region of the parameter space where the simplex is present and where it is not present (see Figure 3). A natural idea might be to reduce to the k -critical case, approximating the entry curve by a staircase with k steps (a sequence of horizontal and vertical segments). However, to ensure an accurate approximation, the value of k might be quite high which complicates the algorithmic treatment and introduces another parameter to the problem. Also, resorting to an approximation is unsatisfying, especially for the generation step which is only the first step in the computational pipeline: while a discretization might suffice for many tasks, it restricts the possibilities of subsequent steps. We therefore advocate the computation of an *exact* representation instead.

Contributions. We give algorithms to compute absolute and relative localized bifiltrations exactly and efficiently. The entry time of every simplex into the bifiltration is described by a curve in the 2-dimensional parameter space that consists of line segments and parabolic arcs.

In the absolute case, a simplicial bifiltration is obtained using alpha complexes (also known as Delaunay complexes) and the Persistent Nerve Theorem. To determine the entry curve of a (Delaunay) simplex, we solve a convex minimization problem on the dual Voronoi polytope parameterized in s . We show that the solutions yield a polygonal chain within the Voronoi polytope, and every line segment translates to one arc of the entry curve in the parameter space. We also describe an efficient algorithm to compute the entry curves of all simplices in amortized constant time per simplex.

In the relative case, we use a variant of the Persistent Nerve Theorem for pairs. However, the Voronoi partition does not satisfy the prerequisites of this Nerve theorem; we show how to subdivide the Voronoi cells for planar inputs to overcome this problem. The entry curves of simplices are determined by the same convex optimization problem as in the absolute case, but now asking for a maximal solution, and can be treated with similar methods.

We provide a prototypical implementation¹ of the absolute case in the plane, based on the CGAL library. This software allows us to visualize the entry curves of localized bifiltrations and serves as a starting point for subsequent algorithmic studies of ∞ -critical bifiltrations. We argue that an algorithmic treatment is in reach by showing that barcode templates of such bifiltrations are computable with the same strategy as in the 1-critical case.

Related work. Applying the homology functor with field coefficients to absolute (relative) localized bifiltrations leads to persistence modules which we call absolute (relative) localized persistence modules. Horizontal and vertical slices of the relative localized persistence module are known as *persistent local homology (PLH) modules* in literature, see the survey [25].

A persistent version of local homology was first considered by Bendich et al. [5] to infer the local homology of a stratified space given by a point cloud. Their PLH modules are defined via extended persistence diagrams [16] and the inherent two parameters (s and r in our notation) are taken into account through vineyards [17]. The study of stratified spaces with the help of PLH is continued in [7], where points of stratified spaces are clustered into same strata. In [5] and [7], PLH modules are computed with modified alpha complexes. Skraba and Wang [38] define two variants of PLH and show how both of them can be computed via approximations by Vietoris-Rips complexes. A further application is given by Ahmed, Fasy and Wenk who define a PLH based distance on graphs used for road network comparison [1].

The work mentioned above relies on relative versions of PLH. A persistence module similar to a horizontal slice of an absolute localized persistence module is used by Stolz [39] for outlier robust landmark selection. Von Rohrscheidt and Rieck [40] consider (samples of) tri-persistence modules to measure how far a given neighborhood of a point is from being Euclidean to obtain the “manifoldness” of point clouds. These tri-persistence modules are obtained by removing the open ball $B_t^o(q)$ from $L_{s,r}$. PLH modules of a filtration $(L_{s,r} \cap \partial B_r(q))_{s \geq 0}$ in combination with multi-scale local principal component analysis are used in [6] to extract relevant features for machine learning from a data set. Other applications of variants of persistent local (co)homology include [21], [41] and [42].

Our work suggests bifiltrations for multi-parameter persistence. Other natural constructions of bifiltrations resulting from point-set inputs are the density-Rips bifiltration [13], the degree-Rips bifiltration [32, 36, 37] and the multi-cover filtration [18, 24]; see [10] for a comparison of these approaches in terms of stability properties. In all these approaches, the second scale parameter models the density of a point, which is different from our approach where it rather models the locality with respect to a point q .

¹ https://bitbucket.org/mkerber/demo_absolute_2d/src/master/

Computational questions in multi-parameter persistence have received a lot of attention recently, including algorithms for visualization [32], decomposition [11, 22], compression [2, 26, 33] and distances [9, 29, 30]. In all aforementioned approaches, the input is assumed to be a simplicial bifiltration that is at least k -critical (and usually even 1-critical), so none of these algorithms is readily applicable to the filtrations computed in this work.

The ∞ -critical bifiltrations appearing in our work yield persistence modules that fulfill the tameness conditions of Miller [35]. In [35] the foundations to vastly generalize the aforementioned k -critical setting are laid. However, it is of rather theoretical nature and in contrast, we propose an initial setup for a concrete algorithmic treatment which might be followed along the suggested route in [34] (see Section 20).

2 The absolute case

We need the following concepts to formally state our problem: For $s, r \in \mathbb{R}$, we write $(s, r) \leq (s', r')$ if $s \leq s'$ and $r \leq r'$. A *bifiltration* is a collection of topological spaces $X := (X_{s,r})_{s,r \geq 0}$ such that $X_{s,r} \subseteq X_{s',r'}$ whenever $(s, r) \leq (s', r')$. A bifiltration is *finite simplicial* if each $X_{s,r}$ is a subcomplex of some simplicial complex K .

A *map of bifiltrations* $\phi : X \rightarrow Y$ is a collection of continuous maps $(\phi_{s,r} : X_{s,r} \rightarrow Y_{s,r})_{s,r \geq 0}$ that commute with the inclusion maps of X and Y . Two bifiltrations are *equivalent* if there is a map $\phi : X \rightarrow Y$ such that each $\phi_{s,r}$ gives a homotopy equivalence (i.e., there exists a map $\psi : Y \rightarrow X$ with each $\psi_{s,r}$ being a homotopy inverse to $\phi_{s,r}$).

For a finite point set P in \mathbb{R}^d , called *sites* from now on, and a center point $q \in \mathbb{R}^d$ (not necessarily a site), we consider the bifiltration L defined by

$$L_{s,r} = \left(\bigcup_{p \in P} B_s(p) \right) \cap B_r(q)$$

with $B_s(p)$ the set of points in distance at most \sqrt{s} from p . Our goal is to compute a finite simplicial bifiltration that is equivalent to L .

Localized alpha complexes. The filtration $L_{s,\infty}$ is the union-of-balls filtration, one of the standard filtration types in persistent homology (with one parameter). It is also well-known that *alpha complexes* [23] provide a practically feasible way of computing an equivalent simplicial representation (at least if d is small). We summarize this technique next; the only difference is that we localize the alpha complexes with respect to $B_r(q)$, which introduces a second parameter but results in no theoretical problem:

For a site p , its *Voronoi region* is the set of points in \mathbb{R}^d for which p is a closest site:

$$\text{Vor}(p) := \{x \in \mathbb{R}^d \mid \|x - p\| \leq \|x - p'\| \forall p' \in P\}.$$

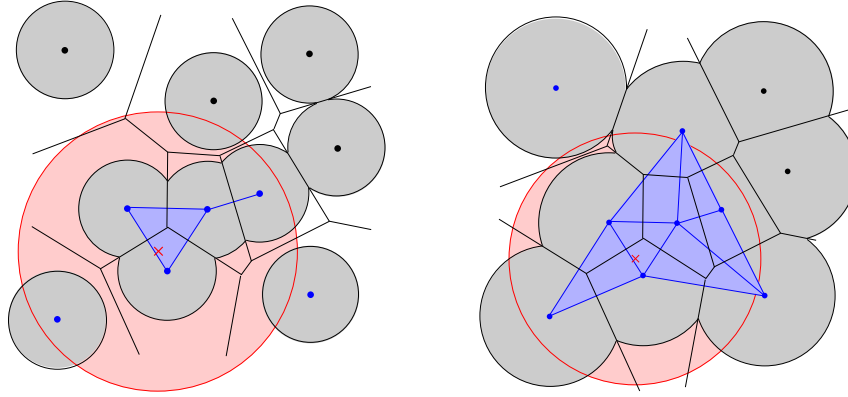
Every $\text{Vor}(p)$ is closed and convex. The *restricted cover* $\mathcal{U}_{s,r} := \{U_{s,r}(p) \mid p \in P\}$ is given by

$$U_{s,r}(p) := B_s(p) \cap B_r(q) \cap \text{Vor}(p).$$

For every $s, r \geq 0$, we have that $\bigcup_{p \in P} U_{s,r}(p) = L_{s,r}$. The *localized alpha complex* is the *nerve* of $\mathcal{U}_{s,r}$, that is, the abstract simplicial complex that encodes the intersection pattern of the restricted cover elements:

$$A_{s,r} := \text{Nrv} \mathcal{U}_{s,r} = \{\{p_0, \dots, p_k\} \subseteq P \mid U_{s,r}(p_0) \cap \dots \cap U_{s,r}(p_k) \neq \emptyset\}.$$

See Figure 4 for an illustration. All $U_{s,r}(p)$ are closed and convex, and $U_{s,r}(p) \subseteq U_{s',r'}(p)$ for $(s,r) \leq (s',r')$. With these conditions, the *Persistent Nerve Theorem* [4, 15] ensures that there is a homotopy equivalence between $L_{s,r}$ and $A_{s,r}$ for every $s,r \geq 0$, and moreover, these homotopy equivalences commute with the inclusion maps of $L_{s,r}$ and $A_{s,r}$. Hence, the bifiltrations are equivalent.



■ **Figure 4** The Voronoi regions, the restricted cover $U_{s,r}$ (in gray) and the localized alpha complex $A_{s,r}$ for two different choices of (s,r) . On the right, the upper blue vertices are indeed not connected although the gray balls intersect, because they do not intersect within the (red) center circle.

Assuming that the sites are in generic position, that is, not more than $k + 2$ sites lie on a common k -dimensional sphere for $1 \leq k \leq d - 1$, the complexes $A_{s,r}$ are subcomplexes of the *Delaunay triangulation* of P , whose size is known to be at most $O(n^{\lceil d/2 \rceil})$ [3] and which can be computed efficiently, especially in dimensions 2 and 3 [28, 43].

Entry curves. We further specify what it means to “compute” the finite simplicial bifiltration A : computing the Delaunay triangulation of P yields all simplices belonging to A . For each simplex $\sigma \in A$, we want to compute an explicit representation of its *active region*, defined as

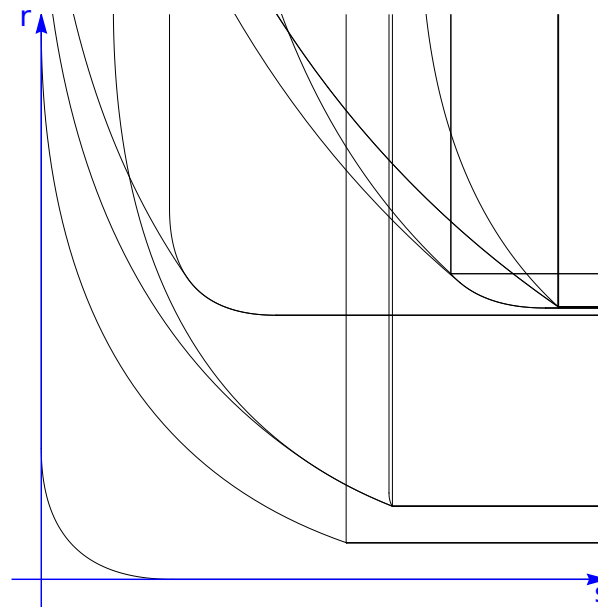
$$R_\sigma := \{(s,t) \in \mathbb{R}^2 \mid \sigma \in A_{s,r}\}.$$

The active region is closed under \leq in \mathbb{R}^2 , meaning that if $(x,y) \in R_\sigma$, the whole upper-right quadrant anchored at (x,y) also belongs to R_σ . We call the boundary of the active region the *entry curve* of σ . See Figure 5 for an illustration of a family of entry curves.

To understand the structure of the entry curve, we define for $\sigma = (p_0, \dots, p_k)$ the polytope $V_\sigma := \text{Vor}(p_0) \cap \dots \cap \text{Vor}(p_k)$. Then, letting p be some point of σ , we have that $\sigma \in A_{s,r}$ if and only if $V_\sigma \cap B_s(p) \cap B_r(q) \neq \emptyset$. Since V_σ is convex, there is a unique point $\hat{p} \in V_\sigma$ as well as $\hat{q} \in V_\sigma$ with minimal distance to p as well as q . We write $s_0 := \|p - \hat{p}\|^2$, $s_1 := \|p - \hat{q}\|^2$ and $r_1 := \|q - \hat{q}\|^2$.

► **Lemma 1.** *The active region R_σ lies in the half-plane $s \geq s_0$. Moreover, restricted to the half-plane $s \geq s_1$, R_σ is bounded by the line $r = r_1$, that is, the area above that line is in R_σ and the area below the line is not.*

Proof. The first part follows because by definition, for $s < s_0$, the intersection $V_\sigma \cap B_s(p)$ is empty. Likewise, for $r < r_1$, the intersection $V_\sigma \cap B_r(q)$ is empty, so R_σ is contained in the half-plane $r \geq r_1$. Moreover, for $s \geq s_1$, the point \hat{q} lies in $V_\sigma \cap B_s(p)$, so the intersection $V_\sigma \cap B_s(p) \cap B_r(q)$ is non-empty if and only if $\hat{q} \in B_r(q)$, which is equivalent to $r \geq r_1$. ◀



■ **Figure 5** Section of entry curves of a planar absolute localized bifiltration on 25 input points.

It remains to compute the entry curve in the s -range $[s_0, s_1]$. For that, we want to compute for each such s , what is the minimal r -value for which $V_\sigma \cap B_s(p) \cap B_r(q) \neq \emptyset$. This minimal r -value, in turn, is simply the distance of q to the set $V_\sigma \cap B_s(p)$. We will study this geometric problem in the next subsection; the solution will give us a parameterization of the boundary of the active region R_σ by line segments and parabolic arcs.

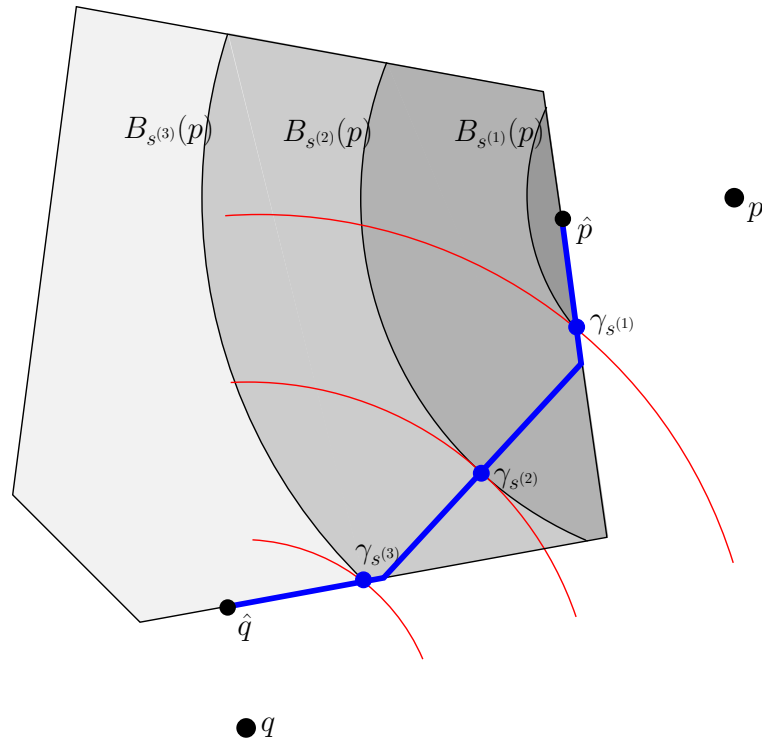
Minimizing paths. Slightly generalizing the setup of the previous paragraph, let $p, q \in \mathbb{R}^d$, and let V be a closed convex polytope in \mathbb{R}^d , that is, the intersection of finitely many closed half-spaces in \mathbb{R}^d . Let \hat{p}, \hat{q} be the points in V with minimal distance to p and q , respectively. Note that $p = \hat{p}$ is possible if $p \in V$, and $\hat{p} \in \partial V$ otherwise. This holds likewise for q . We set $s_0 := \|p - \hat{p}\|^2$ and $s_1 := \|p - \hat{q}\|^2$. For any $s \in [s_0, s_1]$, the intersection $V \cap B_s(p)$ is not empty, and we let γ_s denote the point in that intersection that is closest to q . See Figure 6 for an illustration of the case $d = 2$. The proofs of the next two statements are elementary and only exploit convexity of V and that consequently, the distance function to q restricted to V has only one local minimum. The proofs are in Appendix A of the full version.

► **Lemma 2.** *We have that $\|p - \gamma_s\|^2 = s$. In particular, the function $\|p - \gamma_s\|^2$ is strictly increasing for $s \in [s_0, s_1]$.*

► **Lemma 3.** *The function $\gamma : [s_0, s_1] \rightarrow \mathbb{R}^d, s \mapsto \gamma_s$ is continuous and injective.*

It follows that γ defines a path in \mathbb{R}^d which we call the *minimizing path* for (V, p, q) . A minimizing path in the plane is displayed in Figure 6 (in blue). Because of the following lemma, we henceforth assume wlog that we only consider instances (V, p, q) where V is full-dimensional. The statement follows easily by the Pythagorean Theorem – see full version.

► **Lemma 4 (Dimension reduction).** *Let V be a polytope in \mathbb{R}^d contained in an affine subspace W . Let $p, q \in \mathbb{R}^d$ and p', q' be the orthogonal projections of p and q to W , respectively. Then, the minimizing path of (V, p, q) equals the minimizing path of (V, p', q') up to a shift in the parameterization.*



■ **Figure 6** For three radii $s^{(1)} < s^{(2)} < s^{(3)}$, the sets $V \cap B_{s_i}(p)$ and the corresponding points $\gamma_{s_1}, \gamma_{s_2}, \gamma_{s_3}$ are illustrated. In fact, all points γ_s lie on the blue curve from \hat{p} to \hat{q} . The red arcs are arcs of a circle centered at q and indicate that γ_{s_i} is indeed the minimizing point for s_i .

A *face* of a convex polytope V is the intersection of V with a hyperplane H such that all of V lies in one of the closed half-spaces induced by H .

► **Lemma 5** (Face lemma). *Let γ be the minimizing path of (V, p, q) and let F be a face of V . Then, $\gamma \cap F$, the part of γ that runs along F , is a subset of the minimizing path of (F, p, q) .*

Proof. Let x be a point on $\gamma \cap F$. By definition, $x = \gamma_s$ for some s , that is, x is the closest point to q in $V \cap B_s(p)$. Since $F \subseteq V$, x is also the closest point to q in $F \cap B_s(p)$, so x lies on the minimizing path of F . ◀

By the Face Lemma, we know that the part of the minimizing path of (V, p, q) that runs along ∂V coincides with the minimizing paths of its faces. It remains to understand the minimizing path in the interior of V . The central concept to understand this sub-path is the following simple definition:

► **Definition 6.** *The bridge of (V, p, q) is the (possibly empty) line segment $V \cap \overline{pq}$, where \overline{pq} is the line segment of p and q .*

► **Lemma 7** (Bridge lemma). *Let γ denote the minimizing path of (V, p, q) . Then, the bridge is a subset of γ . Moreover, every point on γ that does not belong to the bridge lies on ∂V .*

Proof. Fix a point x on the bridge and let $s := \|x - p\|^2$. It is simple to verify that $s \in [s_0, s_1]$. We argue that $x = \gamma_s$: indeed, the point x is the closest point to q in $B_s(p)$ (as it lies on \overline{pq}), and since it also lies in V , it minimizes the distance to q for the subset $V \cap B_s(p)$. That proves the first part.

For the second part, assume for a contradiction the existence of a point $y = \gamma_s$ for some $s \in [s_0, s_1]$ that is in the interior of V , but not on the bridge. Since y must lie on the boundary of $V \cap B_s(p)$, and is not on ∂V , it must lie in the interior of some spherical patch of $\partial B_s(p)$. The distance function to q , restricted to the $(d-1)$ -dimensional sphere $\partial B_s(p)$ has no local minimum except at the intersection of \overline{pq} with the boundary, but since y does not lie on the bridge, it is not that minimizing point. Hence, moving in some direction along the spherical patch decreases the distance to q , contradicting the assumption that $y = \gamma_s$. ◀

The Bridge Lemma tells us that the minimizing path runs through the interior of V at most along a single line segment, the bridge; before and after the bridge, it might have sub-paths on the boundary. Figure 6 gives an example where all three sub-paths are present.

► **Theorem 8** (Structure Theorem). *The minimizing path of (V, p, q) is a simple path starting at \hat{p} and ending at \hat{q} , and every point on the path lies on the bridge of some face of V . In particular, the path is a polygonal chain.*

Proof. It is clear that the path goes from \hat{p} to \hat{q} and is simple because γ is injective. Since every point x on the path lies in the relative interior of some face F , the Face Lemma implies that x lies on the minimizing path of (F, p', q') (with p', q' the projections in the subspace of F), and the Bridge lemma implies that x lies on the bridge of F . Hence, the path is contained in a union of finitely many line segments, and therefore is a polygonal path. ◀

Algorithm. Let σ be a Delaunay simplex, V its Voronoi polytope, and p one of the closest sites. We follow the natural approach to compute the minimizing path for (V, p, q) first. Then, for every line segment \overline{ab} on the minimizing path, we use the parameterization

$$s = \|p - ((1-t)a + tb)\|^2 \quad r = \|q - ((1-t)a + tb)\|^2, \quad (1)$$

which for $t \in [0, 1]$ yields a branch of the entry curve. Both s and r are quadratic polynomials in t , therefore the resulting curve is a parabola or line – see Appendix C of the full version for a simple proof.

To compute the minimizing path, we outline the construction and defer details to Appendix B of the full version: we compute the points \hat{p} and \hat{q} which are the start- and endpoint of that path. Then we compute the bridges of V and of all its faces. This yields a collection of line segments and we compute the induced graph whose vertices are endpoints of bridges (in this graph, bridges can be split into sub-segments if the endpoint of another bridge lies in the interior). This graph can be directed such that the distance to p increases along every edge. In this graph, we walk from \hat{p} to \hat{q} to compute the minimizing path; the only required predicate is to determine the next edge to follow at a vertex x . This is the edge along which the distance to q drops the most which can be easily determined by evaluating the gradient of the (squared) distance function to q restricted to the outgoing edges.

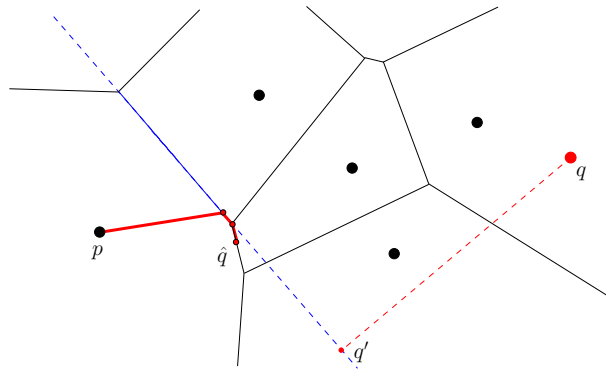
We sketch the complexity analysis of this algorithm, again deferring to Appendix B of the full version for details: Let N be the size of the Delaunay triangulation. Computing all bridges over all Delaunay simplices is linear in N . Writing f for the number of faces of V , the graph constructed for V consists of f bridges. Since bridges do not (properly) cross, the constructed graph for V has still $O(f)$ edges, and the traversal to find the minimizing path is done in $O(f)$ as well. This immediately yields the complexity bound.

► **Theorem 9.** *Let P be n points in general position in \mathbb{R}^d where d is constant. Let N be the size of the Delaunay triangulation of P . We can compute the entry curves of all Delaunay simplices in time $O(N)$.*

3 Implementation

We have implemented the case of absolute localized bifiltrations in the plane using the CGAL library. Our code computes the Delaunay triangulation [43] and computes the minimizing path for each simplex, using a simplified algorithm for the plane. For the minimizing path of a Delaunay vertex, we refer to Figure 7 for an illustration: The path is obtained by starting in p and following the bridge until the boundary is hit. The path must continue in one of the two directions along the boundary and follows the boundary until \hat{q} is met (see the red path in Fig. 7). The decision on which direction to follow can be answered by projecting q to the supporting line of the boundary segment that is hit by the bridge and going towards that projection point q' (the bridge might also hit the boundary in a vertex; we ignore this degenerate case in our prototype).

All geometric predicates required for this implementation are readily available in the geometric kernel of CGAL [12]. For the absolute, planar problem, it was possible to work exclusively with the Delaunay triangulation, avoiding the explicit construction of the Voronoi polytope, but this will probably not be possible for other variants. In either case, the exact geometric computation paradigm of CGAL guarantees that the obtained parameterization is an exact representation of the simplicial bifiltration for the input data.



■ **Figure 7** Illustration of the minimizing path in the plane for a Voronoi region.

4 The relative case

Relative localized bifiltrations. To better fit the situation studied in this section, we re-define $(s, r) \leq (s', r')$ if $s \leq s'$ and $r \geq r'$. With this poset, we can define bifiltrations, finite simplicial bifiltrations, and equivalence in an analogous way as in Section 2. We extend these notions to pairs: A *bifiltration of pairs* is a collection of pairs of topological spaces (or simplicial complexes) $(X, A) := (X_{s,r}, A_{s,r})_{s,r \geq 0}$ such that X and A are bifiltrations and $A_{s,r} \subseteq X_{s,r}$. We have a *finite simplicial bifiltration of pairs* if X is a finite simplicial bifiltration, and $A_{s,r}$ is a subcomplex of $X_{s,r}$ (which implies that also A is finite simplicial). We call two bifiltrations of pairs (X, A) and (Y, B) *equivalent* if there is an equivalence $\phi : X \rightarrow Y$ such that the restriction maps $(\phi|_{A_{s,r}})_{s,r \geq 0}$ yield an equivalence of A and B .

For a finite point set P , a center q and $s, r \geq 0$, set $L_s = \bigcup_{p \in P} B_s(p)$ for the union of balls and $B_r^o(q)$ for the open ball around the center. Then the *relative localized bifiltration* is the following collection of pairs of spaces

$$(L, L^c q) := (L_s, L_s \setminus B_r^o(q))_{s,r \geq 0}.$$

Indeed, it can be checked easily that this construction yields a bifiltration of spaces (notice the contravariance in the r -parameter). Our computational task is to find a finite simplicial bifiltration of pairs equivalent to it.

What is the significance of this bifiltration? Note first that in general, a pair of bifiltrations (X, A) induces a (relative) persistence module $(H_n(X_{s,r}, A_{s,r}))_{s,r \geq 0}$, leading to

$$H_n((L, L^c q)) = (H_n(L_s, L_s \setminus B_r^o(q)))_{s,r \geq 0},$$

which we call the *localized relative persistence module*. By excision and the fact that all pairs are “good pairs” [27], we have isomorphisms [38]

$$H_n(L_s, L_s \setminus B_r^o(q)) \cong H_n(L_s \cap B_r(q), L_s \cap \partial B_r(q)) \cong \tilde{H}_n(L_s \cap B_r(q) / (L_s \cap \partial B_r(q)))$$

with \tilde{H} denoting reduced homology. These isomorphisms imply that the relative localized persistence module corresponds point-wise to the space considered in Figure 2 (right). Moreover, the excision isomorphism commutes with the inclusion maps $L_s \subseteq L_{s'}$ for $s \leq s'$ which implies that the rows of the localized relative persistence module (with fixed r) are isomorphic to the module $(H_n(L_s \cap B_r(q), L_s \cap \partial B_r(q)))_{s \geq 0}$ which was studied in [5], [7], [38]. Also the columns of the bifiltration (with s fixed) have been studied earlier in [38]. Hence, the localized relative bifiltration encodes both types of modules of local persistent homology studied in previous work (see Appendix D of the full version for a summary of basic notions).

We also remark that the computation of the relative localized persistence module can easily be reduced to the case of absolute homology via the well-known coning construction [27, p.125], yielding isomorphisms $H_n(X, A) \cong \tilde{H}_n(X \cup \omega * A)$ for pairs of topological spaces (X, A) with ω denoting a new vertex. These isomorphisms are functorial, yielding an isomorphism between the relative persistence module of a pair of bifiltrations and the absolute persistence module of a bifiltration (using reduced homology). Moreover, if the pair (X, A) is finite simplicial, so is $X \cup \omega * A$.

Nerves of pairs. To obtain an equivalent pair of finite simplicial complexes, we will define suitable covers for L_s and $L_s \setminus B_r^o(q)$ and construct the corresponding nerve complexes. For a (compact) topological space X , a cover \mathcal{U}_X of closed sets is *good* if the intersection of any subset of cover elements of \mathcal{U}_X is empty or contractible. In particular, a closed convex cover (as used in Section 2) is good because convex sets are contractible, and a non-empty intersection of convex sets is convex.

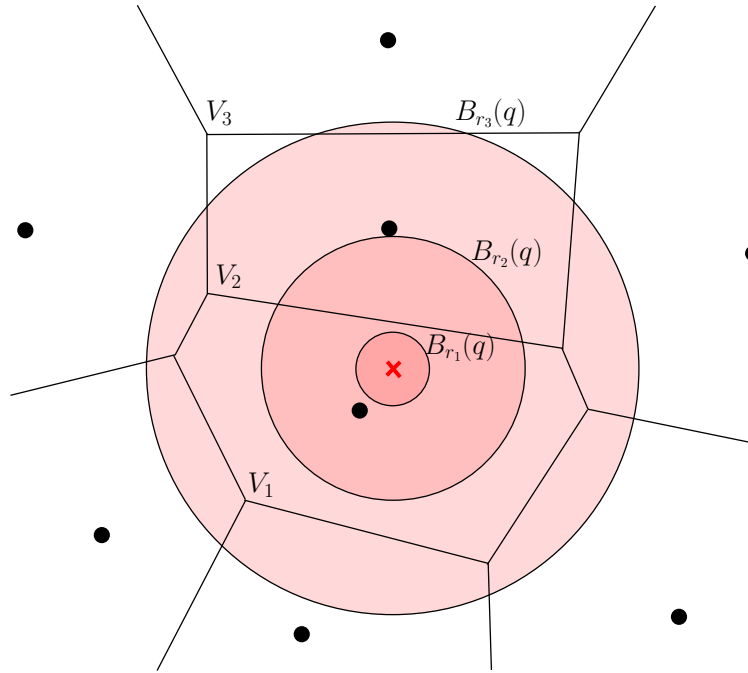
For a pair (X, A) of spaces with $A \subseteq X$ closed, a closed cover \mathcal{U}_X of X induces a cover \mathcal{U}_A of A by restricting every cover element to A . We say that the cover \mathcal{U}_X is *good for the pair* (X, A) if \mathcal{U}_X and \mathcal{U}_A are both good covers. With this definition, we obtain the following version of the Persistent Nerve Theorem; it follows directly from the results of [4] and we summarize the argument in Appendix A of the full version.

► **Theorem 10** (Functorial nerve theorem of pairs). *Let $A \subseteq X \subseteq \mathbb{R}^d$ and \mathcal{U}_X be a good cover for the pair (X, A) . Then the spaces X and $\text{Nrv}\mathcal{U}_X$ are equivalent via a map $\phi : X \rightarrow \text{Nrv}\mathcal{U}_X$, and the restriction map of ϕ to A yields a homotopy equivalence of A and $\text{Nrv}\mathcal{U}_A$. Moreover, the map ϕ is functorial which means that if $X \subseteq X'$ and $A \subseteq A'$, the cover $\mathcal{U}_{X'}$ is obtained by enlarging each element of \mathcal{U}_X (or leaving it unchanged) and is a good cover for the pair (X', A') , then the maps ϕ and ϕ' commute with the inclusion maps $X \rightarrow X'$ and $\text{Nrv}\mathcal{U}_X \rightarrow \text{Nrv}\mathcal{U}_{X'}$, and the restricted maps to A and A' commute with the inclusion maps $A \rightarrow A'$ and $\text{Nrv}\mathcal{U}_A \rightarrow \text{Nrv}\mathcal{U}_{A'}$.*

45:12 The Localized Union-Of-Balls Bifiltration

This (rather bulky) theorem implies the following simple corollary in our situation: If we can find a filtration of (good) covers \mathcal{U}_s for the union of balls L_s , such that for every $r \geq 0$, the induced cover $\mathcal{U}_{s,r}$ on $L_s \setminus B_q^o(r)$ is a good cover, then the pair $(\text{Nrv}\mathcal{U}_s, \text{Nrv}\mathcal{U}_{s,r})_{s,r \geq 0}$ is a finite simplicial bifiltration of pairs equivalent to the localized relative bifiltration.

A good cover in the plane. A natural attempt to construct a good cover for $(L_s, L_s \setminus B_r^o(q))$ would be to consider the cover induced by the Voronoi regions of P , namely $\{\text{Vor}(p) \cap B_s(p) \mid p \in P\}$, similar as in Section 2. While this cover is good for L_s , it is not good for the pair, because the removal of an (open) ball can lead to non-contractible cover elements and intersections; Figure 8 illustrates several problems.



■ **Figure 8** Problems that arise when removing a ball $B_r(q)$ from the Voronoi cover: for radius r_1 , the Voronoi region V_1 becomes non-simply connected. For r_2 , the Voronoi regions V_1 and V_2 intersect in two connected components. For r_3 , the Voronoi region V_2 is disconnected.

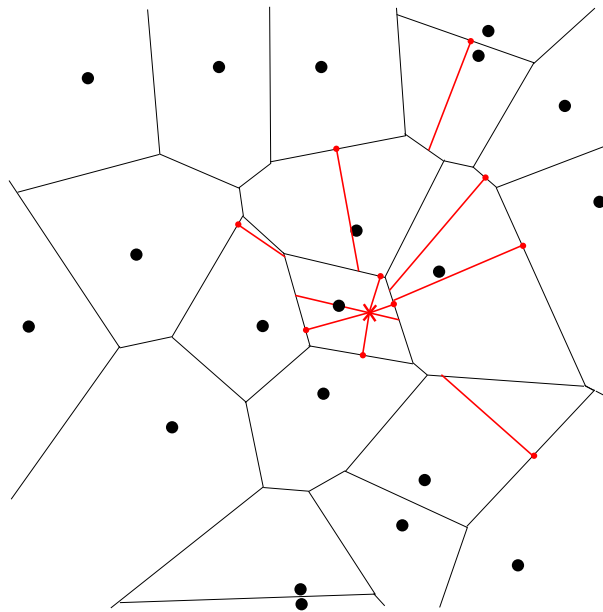
While the Voronoi cover itself does not work, we show that after suitably subdividing the Voronoi cells into convex pieces, the induced cover is good for the pair $(L_s, L_s \setminus B_r^o(q))$. We will give the construction and will informally explain en passant why it removes all obstructions for being good that are visible in Figure 8.

The construction has two parts: first, we split the Voronoi cell $\text{Vor}(p)$ that contains q into two pieces, by cutting along the line \overline{pq} (we assume for simplicity that q lies in the interior of some Voronoi cell). This initial cut ensures in particular that q does not lie in the interior of a cover set anymore; this is a necessary condition because otherwise, removing a sufficiently small open ball $B_r^o(q)$ yields a non-simply connected set in the induced cover for $L_s \setminus B_r^o(q)$ (compare radius r_1 in Figure 8). This cut also avoids configurations where $\text{Vor}(p) \cap B_s(p)$ gets disconnected because $B_r(q)$ is touching $B_s(p)$ from the inside; we refer to the proof of Lemma 16 in Appendix A of the full version for details.

To understand the second part of the construction, we call a line segment or ray e *problematic* if the distance function to q , restricted to e , has a local minimum in the interior of e . This is equivalent to the property that the orthogonal projection of q on the supporting

line of the segment lies on e . Observe in Figure 8 that both for radius r_2 and r_3 , the issue comes from a problematic edge (for r_2 , the edge between V_1 and V_2 is problematic, for r_3 , the edge between V_2 and V_3 is problematic).

In the second part of the construction, we go over all problematic edges in the Voronoi diagram of P . For every such edge e , let \hat{q} denote the orthogonal projection of q on that edge. The line segment $\overline{q\hat{q}}$ cuts through one of the polygons P incident to e , and we cut P using this line segment into two parts. This ends the description of the subdivision. We denote the set of 2-dimensional polygons obtained as $S_2(P, q)$. See Figure 9 for an illustration. Note that the cuts introduced here cut every problematic edge into two non-problematic sub-edges and also avoids that a polygon gets disconnected when removing $B_r(q)$.



■ **Figure 9** Illustration of the subdivision $S_2(P, q)$. The subdivision lines are in red.

Importantly, all cuts introduced are line segments on lines through q . This implies that no two of the cuts can cross. Moreover, no edge arising from a cut can be problematic. This leads to the following theorem, whose proof is given in Appendix A of the full version.

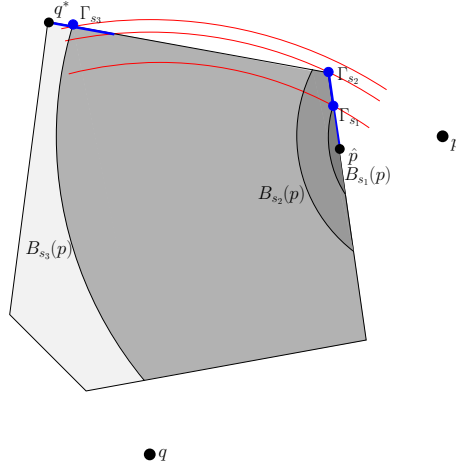
► **Theorem 11.** *The cover $\{L_s \cap V \mid V \in S_2(P, q)\}$ is good for $(L_s, L_s \setminus B_r^o(q))_{s, r \geq 0}$.*

Every cut increases the number of polygons by one, and the number of edges by at most 3. Since we introduce at most one cut per Voronoi edge, plus one initial cut, the complexity of the subdivision is $O(n)$.

Entry curves. It is left to compute the entry curves of every simplex in the nerve induced by $S_2(P, q)$. The described method generalizes also to higher dimensions assuming a subdivision inducing a good cover is available, but we keep the description planar for simplicity. In this case, every simplex σ represents a polygon, edge, or vertex of the planar subdivision defined by $S_2(P, q)$, which we denote by V . Note that σ has two entry curves: one for L_s and one for $L_s \setminus B_r^o(q)$. The former is simple to describe: since L_s does not depend on r , the entry curve is determined by the line of the form $s = \|p - \hat{p}\|^2$, with p a closest site to V and \hat{p} the closest point to p in V .

45:14 The Localized Union-Of-Balls Bifiltration

For $L_s \setminus B_r^o(q)$, we fix s and search for the largest r such that $B_s(p) \cap V \setminus B_r^o(q)$ is not empty. This is equivalent to finding the point on $V \cap B_s(p)$ with *maximal* distance to q , which is the opposite problem to what was considered in Section 2. For brevity, we restrict the discussion to the case of bounded polytopes V , postponing the (straight-forward) extension to unbounded polytopes to Appendix B of the full version. We set q^* as a point in V with maximal distance to q , $s_0 := \|p - \hat{p}\|^2$ and $s_1 := \|p - q^*\|^2$. For $s \in [s_0, s_1]$, we define Γ_s to be a point with maximal distance to q in the set $V \cap B_s(p)$.



■ **Figure 10** Illustration of Γ_s for the same (V, p, q) as in Figure 6. The point Γ_{s_2} remains the maximal point for a range of s -values, and the maximizing “path” (blue) jumps at some value of s .

The maximizing problem is not as well-behaved as the minimization problem: for general polytopes V , the point Γ_s might not be unique, and there can be discontinuities in the image of Γ ; see Figure 10 for an example. However, such problems are caused by local extrema on the boundary of $B_s(p) \cap V$ which can be excluded for the polytopes of $S_2(P, q)$. Hence, we can define the *maximizing curve* $\Gamma : [s_0, s_1] \rightarrow \Gamma_s$ and infer (see Appendix B of full version):

► **Lemma 12.** *For $V \in S_2(P, q)$, the point Γ_s is unique and the curve Γ is continuous.*

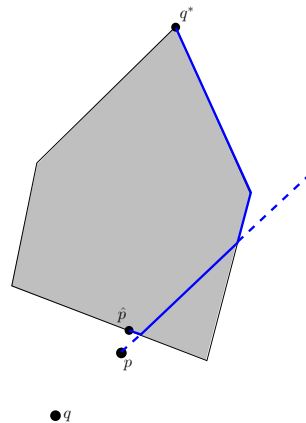
The structure of the curve Γ is similar to its minimizing counterpart γ . The Dimension Reduction Lemma and the Face Lemma also hold for Γ , with identical proofs. We define the *anti-bridge* as the intersection of V with the ray emanating from p in the direction $-pq$ (see Figure 11). With that, we obtain the statement that the anti-bridge is part of Γ , and the rest of Γ lies on ∂V , with an analogue proof as for the bridge lemma. This leads to

► **Theorem 13** (Structure Theorem, maximal version). *The maximizing path Γ is a subset of the union of all anti-bridges over all faces of V . In particular, it is a polygonal chain.*

The theorem also infers a way to compute the entry curve of σ constructing and traversing a graph obtained by anti-bridges. We omit further details which are analogous to the minimization case, and yield the same bound as in Theorem 9.

5 Barcode templates for ∞ -critical bifiltrations

We demonstrate in this section that representing a bifiltration via non-linear entry curves does not prevent an efficient algorithmic treatment. We focus on the case of the computation of *barcode templates* as introduced by Lesnick and Wright [32]. The idea is as follows: restricting



■ **Figure 11** Illustration of the maximizing path Γ for a polygon with only one local minimum and one local maximum on the boundary. The path goes through the interior along the anti-bridge.

a bifiltration to a line of positive slope (called a *slice*) gives a filtration in one parameter, and hence a persistence barcode. While we cannot associate a persistence barcode to the bifiltration [13], the collection of barcodes over all slices yields a wealth of information about the bifiltration. The software library RIVET² provides a visualization of these sliced barcodes for finite simplicial bifiltrations. To speed up the visualization step, RIVET precomputes all *combinatorial barcodes*, that is, it clusters slices together on which the simplices enter the filtration in the same order – it is well-known that the barcode combinatorially only depends on this order, not the concrete critical values. The barcode template is then, roughly speaking, the collection of all these combinatorial barcodes. Barcode templates have also been used for exact computations of the matching distance of bifiltrations [9, 29].

All aforementioned approaches assume the bifiltration to be 1-critical, which means in our notation that the active region of every simplex σ is the upper-right quadrant of a point $v_\sigma \in \mathbb{R}^2$. We argue that this restriction is unnecessary: indeed, the combinatorial barcode is determined by the order in which the slice intersects the entry curves of all simplices. For a finite set M of points in \mathbb{R}^2 , a non-vertical line partitions M into $(M_\uparrow, M_\downarrow, M_{on})$, denoting the points above the line, below the line, and on the line, respectively.

► **Lemma 14.** *Let I denote the set of intersection points of pairs of entry curves. If two slices have the same partition of I , they have the same combinatorial barcode.*

Proof. Let ℓ_1, ℓ_2 be slices with different barcodes. This means, that there is at least one pair of simplices (σ, τ) for which the entry curves are intersected in a different order. When continuously transforming ℓ_1 into ℓ_2 , we therefore have to cross an intersection point of these entry curves, and hence the partition changes. ◀

Using standard point-line duality [20, Ch. 8] which is known to preserve above/below orders of points and lines, we obtain at once:

► **Corollary 15.** *Let $\text{dual}(I)$ be the line arrangement obtained by the dual of all points in I . Every region of this arrangement is dual to a set of slices with the same combinatorial barcode.*

² <https://rivet.readthedocs.io/en/latest/index.html#>

We mention that this result generalizes the well-studied 1-critical case: the entry curve of every σ consists of a vertical and a horizontal ray emanating from a point v_σ in this case, and the curves of σ and τ intersect in the join of v_σ and v_τ . Hence I is the union of all pairwise joins of the critical values of all vertices.

Computationally, the only complication consists in computing the intersection points of the entry curves, which have a more complicated structure than in the 1-critical case. However, if the curves are semi-algebraic (and of small degree), as in the case studied in this paper, computing these intersection points is feasible and efficient software is available [8].

6 Conclusion

Localized persistence modules are, on the one hand, a suitable set of examples for further algorithmic work in the pipeline of ∞ -critical multi-parameter persistence. On the other hand, because one-dimensional sub-modules of the module have been considered in the context of applications, we hope that the module itself can be of use in applied contexts.

There are many natural follow-up questions for our work: most directly, the only obstacle to extend our relative approach in higher dimensions is the subdivision scheme which is currently only proved for \mathbb{R}^2 . Another future task is the implementation of our approaches beyond the absolute case in the plane. Such implementations are well in-reach, given that all underlying geometric primitives are provided by the CGAL library.

One aspect we have not touched upon is stability. A natural assumption is that if P and P' as well as the centers q and q' are ε -close in the L_∞ -distance, then the localized bifiltrations for (P, q) and (P', q') are ε -interleaved [14, 31]. This claim is true (and straight-forward to prove) if we re-define the notion $B_\alpha(x)$ to be ball of radius α around x (instead of $\sqrt{\alpha}$ as in this work). While our non-standard notion for balls prevents us from making this claim, we point out that it simplifies the description of the entry curves. In the standard notion, the parameterization in (1) in Section 2 would involve square roots and therefore does not yield parabolic arcs. We decided to favor the simplicity of the entry curves in this work.

We hope that our initial efforts will result in an algorithmic treatment of multi-parameter persistence that can cope with ∞ -critical filtrations with comparable efficiency as in the 1-parameter case. A natural next question is whether minimal presentations of ∞ -critical filtrations can be computed efficiently, generalizing the recent approaches from [26, 33].

Localizing the union-of-balls filtration in a center yields more fine-grained information about the point set in the vicinity of the center. We pose the question whether and how this information can be leveraged in the numerous application domains of topological data analysis. We speculate that even in situations where the data set does not contain a canonical center location, considering a sample of centers and analyzing the ensemble of localized bifiltrations yields a more discriminative topological proxy than the union-of-balls filtration.

References

- 1 Mahmuda Ahmed, Brittany Terese Fasy, and Carola Wenk. Local persistent homology based distance between maps. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL GIS 2014)*, pages 43–52, 2014. doi:10.1145/2666310.2666390.
- 2 Ángel Javier Alonso, Michael Kerber, and Siddharth Pritam. Filtration-domination in bifiltered graphs. In *2023 Proceedings of the Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 27–38, 2023. doi:10.1137/1.9781611977561.ch3.
- 3 Franz Aurenhammer, Rolf Klein, and Der-Tsai Lee. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013. doi:10.1142/8685.

- 4 Ulrich Bauer, Michael Kerber, Fabian Roll, and Alexander Rolle. A Unified View on the Functorial Nerve Theorem and its Variations, 2022. doi:10.48550/ARXIV.2203.03571.
- 5 Paul Bendich, David Cohen-Steiner, Herbert Edelsbrunner, John Harer, and Dmitriy Morozov. Inferring Local Homology from Sampled Stratified Spaces. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 536–546, 2007. doi:10.1109/FOCS.2007.45.
- 6 Paul Bendich, Ellen Gasparovic, John Harer, Rauf Izmailov, and Linda Ness. Multi-scale local shape analysis and feature selection in machine learning applications. In *2015 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2015. doi:10.1109/IJCNN.2015.7280428.
- 7 Paul Bendich, Bei Wang, and Sayan Mukherjee. Local Homology Transfer and Stratification Learning. In *Proceedings of the 2012 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1355–1370, 2012. doi:10.1137/1.9781611973099.107.
- 8 Eric Berberich, Michael Hemmer, and Michael Kerber. A generic algebraic kernel for non-linear geometric applications. In *Proceedings of the Twenty-Seventh Annual Symposium on Computational Geometry (SoCG 2011)*, pages 179–186, 2011. doi:10.1145/1998196.1998224.
- 9 Håvard Bakke Bjerkevik and Michael Kerber. Asymptotic Improvements on the Exact Matching Distance for 2-parameter Persistence, 2021. doi:10.48550/ARXIV.2111.10303.
- 10 Andrew J. Blumberg and Michael Lesnick. Stability of 2-Parameter Persistent Homology. *Foundations of Computational Mathematics*, 2022. doi:10.1007/s10208-022-09576-6.
- 11 Magnus Bakke Botnan, Steffen Oppermann, and Steve Oudot. Signed Barcodes for Multi-Parameter Persistence via Rank Decompositions. In *38th International Symposium on Computational Geometry (SoCG 2022)*, pages 19:1–19:18, 2022. doi:10.4230/LIPIcs.SoCG.2022.19.
- 12 Hervé Brönnimann, Andreas Fabri, Geert-Jan Giezeman, Susan Hert, Michael Hoffmann, Lutz Kettner, Sylvain Pion, and Stefan Schirra. 2D and 3D linear geometry kernel. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.1 edition, 2022. URL: <https://doc.cgal.org/5.5.1/Manual/packages.html#PkgKernel23>.
- 13 Gunnar E. Carlsson and Afra Zomorodian. The Theory of Multidimensional Persistence. *Discrete & Computational Geometry*, 42:71–93, 2009. doi:10.1007/s00454-009-9176-0.
- 14 Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Oudot. Proximity of Persistence Modules and Their Diagrams. In *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry (SoCG 2009)*, pages 237–246, 2009. doi:10.1145/1542362.1542407.
- 15 Frédéric Chazal and Steve Oudot. Towards Persistence-Based Reconstruction in Euclidean Spaces. In *Proceedings of the Twenty-Fourth Annual Symposium on Computational Geometry (SoCG 2008)*, pages 232–241, 2008. doi:10.1145/1377676.1377719.
- 16 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending Persistence Using Poincaré and Lefschetz Duality. *Foundations of Computational Mathematics*, 9:79–103, 2009. doi:10.1007/s10208-008-9027-z.
- 17 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and Vineyards by Updating Persistence in Linear Time. In *Proceedings of the Twenty-Second Annual Symposium on Computational Geometry (SoCG 2006)*, pages 119–126, 2006. doi:10.1145/1137856.1137877.
- 18 René Corbet, Michael Kerber, Michael Lesnick, and Georg Osang. Computing the Multicover Bifiltration. *Discrete & Computational Geometry*, 2023. doi:10.1007/s00454-022-00476-8.
- 19 René Corbet, Ulderico Fugacci, Michael Kerber, Claudia Landi, and Bei Wang. A kernel for multi-parameter persistent homology. *Computers & Graphics: X*, 2, 2019. doi:10.1016/j.cagx.2019.100005.
- 20 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational Geometry: Algorithms and Applications, 3rd Edition*. Springer, 2008.

- 21 Tamal Dey, Fengtao Fan, and Yusu Wang. Dimension Detection with Local Homology. In *Proceedings of the 26th Canadian Conference on Computational Geometry (CCCG 2014)*, 2014. URL: <http://www.cccg.ca/proceedings/2014/papers/paper40.pdf>.
- 22 Tamal K. Dey and Cheng Xin. Generalized persistence algorithm for decomposing multiparameter persistence modules. *Journal of Applied and Computational Topology*, 6:271–322, 2022. doi:10.1007/s41468-022-00087-5.
- 23 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, 2010.
- 24 Herbert Edelsbrunner and Georg Osang. The Multi-Cover Persistence of Euclidean Balls. *Discrete & Computational Geometry*, 65:1296–1313, 2021. doi:10.1007/s00454-021-00281-9.
- 25 Brittany Terese Fasy and Bei Wang. Exploring persistent local homology in topological data analysis. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6430–6434, 2016. doi:10.1109/ICASSP.2016.7472915.
- 26 Ulderico Fugacci, Michael Kerber, and Alexander Rolle. Compression for 2-parameter persistent homology. *Computational Geometry*, 109, 2023. doi:10.1016/j.comgeo.2022.101940.
- 27 Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2005.
- 28 Clément Jamin, Sylvain Pion, and Monique Teillaud. 3D triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.1 edition, 2022. URL: <https://doc.cgal.org/5.5.1/Manual/packages.html#PkgTriangulation3>.
- 29 Michael Kerber, Michael Lesnick, and Steve Oudot. Exact Computation of the Matching Distance on 2-Parameter Persistence Modules. In *35th International Symposium on Computational Geometry (SoCG 2019)*, pages 46:1–46:15, 2019. doi:10.4230/LIPIcs.SoCG.2019.46.
- 30 Michael Kerber and Arnur Nigmatov. Efficient Approximation of the Matching Distance for 2-Parameter Persistence. In *36th International Symposium on Computational Geometry (SoCG 2020)*, pages 53:1–53:16, 2020. doi:10.4230/LIPIcs.SoCG.2020.53.
- 31 Michael Lesnick. The Theory of the Interleaving Distance on Multidimensional Persistence Modules. *Foundations of Computational Mathematics*, 15:613–650, 2015. doi:10.1007/s10208-015-9255-y.
- 32 Michael Lesnick and Matthew Wright. Interactive Visualization of 2-D Persistence Modules, 2015. doi:10.48550/ARXIV.1512.00180.
- 33 Michael Lesnick and Matthew Wright. Computing Minimal Presentations and Bigraded Betti Numbers of 2-Parameter Persistent Homology. *SIAM Journal on Applied Algebra and Geometry*, 6:267–298, 2022. doi:10.1137/20M1388425.
- 34 Ezra Miller. Data structures for real multiparameter persistence modules, 2017. doi:10.48550/ARXIV.1709.08155.
- 35 Ezra Miller. Homological algebra of modules over posets, 2020. doi:10.48550/ARXIV.2008.00063.
- 36 Alexander Rolle. The Degree-Rips Complexes of an Annulus with Outliers. In *38th International Symposium on Computational Geometry (SoCG 2022)*, pages 58:1–58:14, 2022. doi:10.4230/LIPIcs.SoCG.2022.58.
- 37 Alexander Rolle and Luis Scoccola. Stable and consistent density-based clustering, 2020. doi:10.48550/ARXIV.2005.09048.
- 38 Primoz Skraba and Bei Wang. Approximating Local Homology from Samples. In *Proceedings of the 2014 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 174–192, 2014. doi:10.1137/1.9781611973402.13.
- 39 Bernadette J. Stolz. Outlier-robust subsampling techniques for persistent homology, 2021. doi:10.48550/ARXIV.2103.14743.
- 40 Julius von Rohrscheidt and Bastian Rieck. TOAST: Topological Algorithm for Singularity Tracking, 2022. doi:10.48550/ARXIV.2210.00069.
- 41 Bei Wang, Brian Summa, Valerio Pascucci, and Mikael Vejdemo-Johansson. Branching and Circular Features in High Dimensional Data. *IEEE Transactions on Visualization and Computer Graphics*, 17:1902–11, 2011. doi:10.1109/TVCG.2011.177.

- 42 Matthew Wheeler, Jose Bouza, and Peter Bubenik. Activation Landscapes as a Topological Summary of Neural Network Performance. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3865–3870, 2021. doi:10.1109/BigData52589.2021.9671368.
- 43 Mariette Yvinec. 2D triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.1 edition, 2022. URL: <https://doc.cgal.org/5.5.1/Manual/packages.html#PkgTriangulation2>.

Online and Dynamic Algorithms for Geometric Set Cover and Hitting Set

Arindam Khan ✉ 🏠 

Indian Institute of Science, Bengaluru, India

Aditya Lonkar ✉

Indian Institute of Science, Bengaluru, India

Saladi Rahul ✉ 🏠 

Indian Institute of Science, Bengaluru, India

Aditya Subramanian ✉ 🏠

Indian Institute of Science, Bengaluru, India

Andreas Wiese ✉ 🏠 

Technische Universität München, Germany

Abstract

Set cover and hitting set are fundamental problems in combinatorial optimization which are well-studied in the offline, online, and dynamic settings. We study the geometric versions of these problems and present new online and dynamic algorithms for them. In the online version of set cover (resp. hitting set), m sets (resp. n points) are given and n points (resp. m sets) arrive online, one-by-one. In the dynamic versions, points (resp. sets) can arrive as well as depart. Our goal is to maintain a set cover (resp. hitting set), minimizing the size of the computed solution.

For online set cover for (axis-parallel) squares of arbitrary sizes, we present a tight $O(\log n)$ -competitive algorithm. In the same setting for hitting set, we provide a tight $O(\log N)$ -competitive algorithm, assuming that all points have integral coordinates in $[0, N]^2$. No online algorithm had been known for either of these settings, not even for unit squares (apart from the known online algorithms for arbitrary set systems).

For both dynamic set cover and hitting set with d -dimensional hyperrectangles, we obtain $(\log m)^{O(d)}$ -approximation algorithms with $(\log m)^{O(d)}$ worst-case update time. This partially answers an open question posed by Chan et al. [SODA'22]. Previously, no dynamic algorithms with polylogarithmic update time were known even in the setting of squares (for either of these problems). Our main technical contributions are an *extended quad-tree* approach and a *frequency reduction* technique that reduces geometric set cover instances to instances of general set cover with bounded frequency.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Online algorithms

Keywords and phrases Geometric Set Cover, Hitting Set, Rectangles, Squares, Hyperrectangles, Online Algorithms, Dynamic Data Structures

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.46

Related Version *Full Version:* <https://arxiv.org/abs/2303.09524>

Funding *Arindam Khan:* gratefully acknowledges the generous support due to IUSSTF virtual center on “Polynomials as an Algorithmic Paradigm”, Pratiksha Trust Young Investigator Award, Google India Research Award, Google ExploreCS Award, and SERB Core Research Grant (CRG/2022/001176) on “Optimization under Intractability and Uncertainty”.

Aditya Subramanian: supported in part by Kotak IISc AI-ML Centre (KIAC) PhD Fellowship.



© Arindam Khan, Aditya Lonkar, Saladi Rahul, Aditya Subramanian, and Andreas Wiese; licensed under Creative Commons License CC-BY 4.0

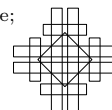
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 46; pp. 46:1–46:17

Leibniz International Proceedings in Informatics



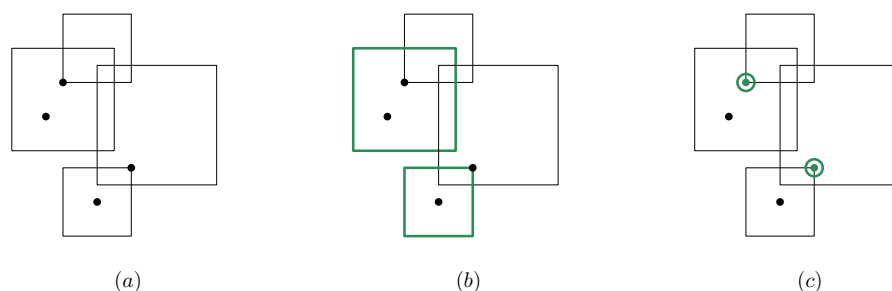
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Geometric set cover is a fundamental and well-studied problem in computational geometry [19, 16, 31, 26, 29]. Here, we are given a universe P of n points in \mathbb{R}^d , and a family \mathcal{S} of m sets, where each set $S \in \mathcal{S}$ is a geometric object (we assume S to be a *closed* set in \mathbb{R}^d and S *covers* all points in $P \cap S$), e.g., a hyperrectangle. Our goal is to select a collection $\mathcal{S}' \subseteq \mathcal{S}$ of these sets that contain (i.e., cover) all elements in P , minimizing the cardinality of \mathcal{S}' (see Figure 1 for an illustration). The *frequency* f of the set system (P, \mathcal{S}) is defined as the maximum number of sets that contain an element in P .

In the offline setting of some cases of geometric set cover, better approximation ratios are known than those for the general set cover, e.g., there is a polynomial-time approximation scheme (PTAS) for (axis-parallel) squares [30]. However, much less is understood in the online and in the dynamic variants of geometric set cover. In the online setting, the sets are given offline and the points arrive one-by-one, and for an uncovered point, we have to select a (covering) set in an immediate and irrevocable manner. To the best of our knowledge, even for 2-D unit squares, there is no known online algorithm with an asymptotically improved competitive ratio compared to the $O(\log n \log m)$ -competitive algorithm for general online set cover [3, 14]. In the dynamic case, the sets are again given offline and at each time step a point is inserted or deleted. Here, we are interested in algorithms that update the current solution quickly when the input changes. In particular, it is desirable to have algorithms whose update times are polylogarithmic. Unfortunately, hardly any such algorithm is known for geometric set cover. Agarwal et al. [2] initiated the study of dynamic geometric set cover for intervals and 2-D unit squares and presented $(1 + \varepsilon)$ - and $O(1)$ -approximation algorithms with polylogarithmic update times, respectively. To the best of our knowledge, for more general objects, e.g., rectangles, three-dimensional cubes, or hyperrectangles in higher dimensions, no such dynamic algorithms are known. Note that in dynamic geometric set cover, the inserted points are represented by their coordinates, which is more compact than for general (dynamic) set cover (where for each new point p we are given a list of the sets that contain p , and hence, already to read this input we might need $\Omega(f)$ time).



■ **Figure 1** (a) A set of squares \mathcal{S} and a set of points P , (b) A set cover (in green) $\mathcal{S}' \subseteq \mathcal{S}$ covering P , (c) A hitting set (green points) $P' \subseteq P$ for \mathcal{S} .

Related to set cover is the hitting set problem (see Figure 1 for an illustration) where, given a set of points P and a collection of sets \mathcal{S} , we seek to select the minimum number of points $P' \subseteq P$ that hit each set $S \in \mathcal{S}$, i.e., such that $P' \cap S \neq \emptyset$. Again, in the offline geometric case, there are better approximation ratios known than for the general case, e.g., a PTAS for squares [30], and an $O(\log \log \text{OPT})$ -approximation for rectangles [4]. However, in the online and the dynamic cases, only few results are known that improve on the results for the general case. In the online setting, there is an $O(\log n)$ -competitive algorithm for $d = 1$,

i.e., intervals, and an $O(\log n)$ -competitive algorithm for unit disks [21]. In the dynamic case, the only known algorithms are for intervals and unit squares (and thus, also for quadrants), yielding approximation ratios of $(1 + \varepsilon)$ and $O(1)$, respectively [2].

1.1 Our results

In Section 2 we study online set cover for axis-parallel squares of arbitrary sizes and provide an online $O(\log n)$ -competitive algorithm. We also match (asymptotically) the lower bound of $\Omega(\log n)$, and hence, our competitive ratio is tight. In our online model (as in [3]), we assume that the sets (squares) are given initially and the elements (points) arrive online.

Our online algorithm is based on a new offline algorithm that is *monotone*, i.e., it has the property that if we add a new point p to P , the algorithm outputs a superset of the squares that it outputs given only P without p . The algorithm is based on a quad-tree decomposition. It traverses the tree from the root to the leaves, and for each cell C in which points are still uncovered, it considers each edge e of C and selects the “most useful” squares containing e , i.e., the squares with the largest intersection with C . We assume (throughout this paper) that all points and all corners of the squares have integral coordinates in $[0, N]^2$ for a given N , and we obtain a competitive ratio of $O(\log N)$. If we know that all the inserted points come from an initially given set of n candidate points P_0 (as in, e.g., Alon et al. [3]), we improve our competitive ratio to $O(\log n)$. For this case, we use the BBD-tree data structure due to Arya et al. [6] which uses a more intricate decomposition into cells than a standard quad-tree, and adapt our algorithm to it in a non-trivial manner. Due to the monotonicity of our offline algorithm, we immediately obtain an $O(\log n)$ -competitive online algorithm.

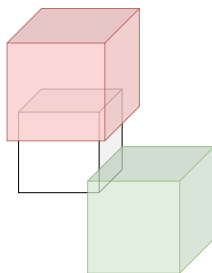
In Section 3 we present an $O(\log N)$ -competitive algorithm for online hitting set for squares of arbitrary sizes, where the points are given initially and the squares arrive online. This matches the best-known $O(\log N)$ -competitive algorithm for the much simpler case of intervals [21]. Also, there is a matching lower bound of $\Omega(\log N)$, even for intervals.

In a nutshell, if a new square S is inserted by the adversary, we identify $O(\log N)$ quad-tree cells for which S contains one of its edges. Then, we pick the most useful points in these cells to hit such squares: those are the points closest to the four edges of the cell. We say that this *activates* the cell. In our analysis, we turn this around: we show that for each point $p \in \text{OPT}$ there are only $O(\log N)$ cells that can possibly get activated if a square S is inserted that is hit by p . This yields a competitive ratio of $O(\log N)$.

Then, in Section 4 and 5 we present our dynamic algorithms for set cover and hitting set for hyperrectangles in d dimensions. Note that no dynamic algorithm with polylogarithmic update time and polylogarithmic approximation ratio is known even for set cover for rectangles and it was asked explicitly by Chan et al. [18] whether such an algorithm exists. Thus, we answer this question in the affirmative for the case when only points are inserted and deleted. Note that this is the relevant case when we seek to store our solution explicitly, as discussed above. Even though our considered objects are very general, our algorithms need only polylogarithmic worst-case update time. In contrast, Abboud et al. [1] showed that under Strong Exponential Time Hypothesis any general (dynamic) set cover algorithm with an amortized update time of $O(f^{1-\varepsilon})$ must have an approximation ratio of $\Omega(n^\alpha)$ for some constant $\alpha > 0$, and f can be as large as $\Theta(m)$.

We first discuss our algorithm for set cover. We start with reducing the case of hyperrectangles in d dimensions to $2d$ -dimensional hypercubes with integral corners in $[0, 4m]^{2d}$. Then, a natural approach would be to adapt our algorithm for squares from above to these hypercubes. A canonical generalization would be to build a quad-tree, traverse it from the root to the leaves, and to select for each cell C and for each facet F of C the most

useful hypercube S containing F , i.e., the hypercube S with maximal intersection with C . Unfortunately, this is no longer sufficient, not even in 3-D: it might be that there is a cell C for which it is necessary that we select cubes that contain only an edge of C but not a facet of C (see Figure 2). Here, we introduce a crucial new idea: for each cell C of the (standard) quad-tree and for each dimension $i \in [2d]$, consider the hypercubes which are “edge-covering” C along dimension i . Based on these hypercubes a $(2d-1)$ -dimensional recursive secondary structure is built on all the dimensions except the i -th dimension (see Figure 11).



■ **Figure 2** The red cube is the only cube that covers a facet of the (uncolored) cell. The green cube (from OPT) only covers an edge of the cell. Note that there is no corner of a cube from OPT in the cell. Picking the red cube does not cover the the intersection of the green cube with the cell.

We call the resulting tree the *extended quad-tree*. Even though it is much larger than the standard quad-tree, we show that each point is contained in only $(\log m)^{O(d)}$ cells. Furthermore, we use it for our second crucial idea to *reduce the frequency* of the set cover instance: we build an auxiliary instance of general set cover with bounded frequency. It has the same points as the given instance of geometric set cover, but different sets: for each node corresponding to a one-dimensional cell C of the extended quadtree, we consider each of its endpoints p and introduce a set that corresponds to the “most useful” hypercube covering p , i.e., the hypercube covering p with maximal intersection with C . Since each point is contained in only $(\log m)^{O(d)}$ cells, the resulting frequency is bounded by $(\log m)^{O(d)}$. Also, we show that our auxiliary set cover instance admits a solution with at most $\text{OPT} \cdot (\log m)^{O(d)}$ sets. Then we use a dynamic algorithm from [12] for *general* set cover to maintain an approximate solution for our auxiliary instance, which yields a dynamic $(\log m)^{O(d)}$ -approximation algorithm.

We further adapt our dynamic set cover algorithm mentioned above to hitting set for d -dimensional hyperrectangles with an approximation ratio of $(\log n)^{O(d)}$. Finally, we extend our algorithms for set cover and hitting set for d -dimensional hyperrectangles even to the weighted case, at the expense of only an extra factor of $(\log W)^{O(1)}$ in the update time and approximation ratio, assuming that all sets/points in the input have weights in $[1, W]$.

Due to space limitations, many proofs are omitted and we refer the readers to the full version [28] for the details. See the following tables for a summary of our results.

■ **Table 1** Online algorithms for geometric set cover and hitting set.

Problem	Objects	Competitive ratio	Lower bound
Set cover	intervals	2	2
	2-D squares	$O(\log n)$	$\Omega(\log n)$
Hitting set	intervals	$O(\log N)$ [21]	$\Omega(\log N)$ [21]
	2-D squares	$O(\log N)$	$\Omega(\log N)$ [21]

■ **Table 2** Dynamic algorithms for geometric set cover and hitting set. Update times in [2] are amortized and for the unweighted case. Our results are for worst-case update times.

Problem	Objects	Approximation ratio	Update time
Set cover	2-D unit squares	$O(1)$ [2]	$(\log n)^{O(1)}$
	d -D hyperrectangles	$O(\log^{4d-1} m) \log W$	$O(\log^{2d} m) \log^3(Wn)$
Hitting set	unit squares	$O(1)$ [2]	$(\log n)^{O(1)}$
	d -D hyperrectangles	$O(\log^{4d-1} n) \log W$	$O(\log^{2d-1} n) \log^3(Wm)$

1.2 Other related work

The general set cover is well-studied in both online and dynamic settings. Several variants and generalizations of online set cover have been considered, e.g., online submodular cover [25], online set cover under random-order arrival [22], online set cover with recourse [23], etc.

For dynamic setting, Gupta et al. [23] initiated the study and provided $O(\log n)$ -approximation algorithm with $O(f \log n)$ -amortized update time, even in the weighted setting. Similar to our model, in their model sets are given offline and only elements can appear or depart. After this, there has been a series of works [1, 9, 11, 10, 12, 23, 24, 7].

Bhattacharya et al. [12] have given deterministic $(1 + \epsilon)f$ -approximation in $O(f \log^2(Wn)/\epsilon^3)$ -worst-case update time, and $O((f^2/\epsilon^3) + (f/\epsilon^2) \log(W))$ -amortized update time, where W denotes the ratio of the weights of the highest and lowest weight sets. Assadi and Solomon [7] have given a randomized f -approximation algorithm with $O(f^2)$ -amortized update time.

Agarwal et al. [2] studied another dynamic setting for geometric set cover, where both points and sets can arrive or depart, and presented $(1 + \epsilon)$ - and $O(1)$ -approximation with sublinear update time for intervals and unit squares, respectively. Chan and He [17] extended it to set cover with arbitrary squares. Recently, Chan et al. [18] gave $(1 + \epsilon)$ -approximation for the special case of intervals in $O(\log^3 n/\epsilon^3)$ -amortized update time. They also gave $O(1)$ -approximation for dynamic set cover for unit squares, arbitrary squares, and weighted intervals in amortized update time of $2^{O(\sqrt{\log n})}$, $n^{1/2+\epsilon}$, and $2^{O(\sqrt{\log n \log \log n})}$, respectively.

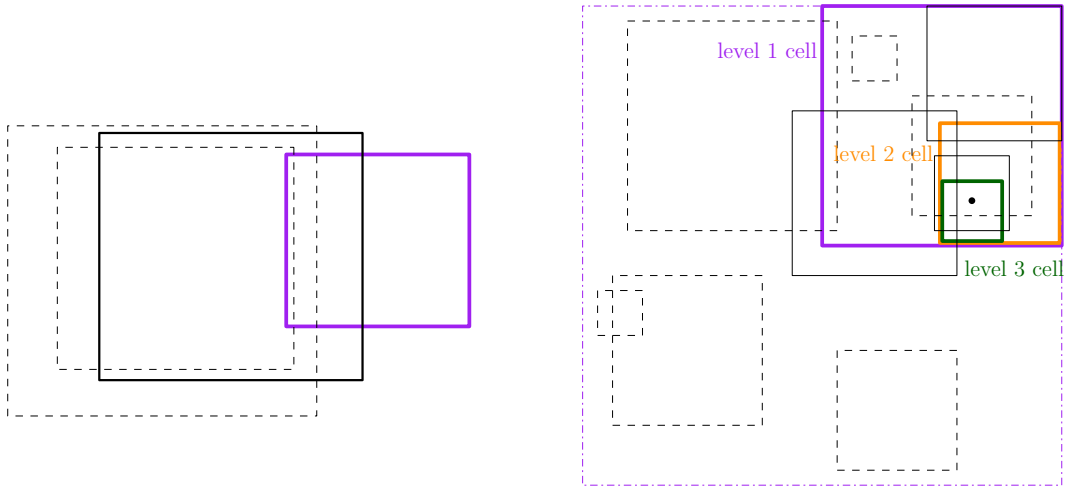
Dynamic algorithms are also well-studied for other geometric problems such as maximum independent set of intervals and hyperrectangles [27, 13, 15], and geometric measure [20].

2 Set cover for squares

In this section we present our online and dynamic algorithms for set cover for squares. We are given a set of m squares \mathcal{S} such that each square $S \in \mathcal{S}$ has integral corners in $[0, N]^2$. W.l.o.g. assume that N is a power of 2. We first describe an offline $O(\log N)$ -approximate algorithm. Then we construct an online algorithm based on it, such that it has an approximation ratio of $O(\log N)$ as well. For our offline algorithm, we assume that in addition to \mathcal{S} and N , we are given a set of points P that we need to cover, such that $P \subseteq [0, N]^2$, and each point $p \in P$ has integral coordinates.

Quad-tree

We start with the definition of a quad-tree $T = (V, E)$, similarly as in, e.g., [5, 8]. In T each node $v \in V$ corresponds to a square cell $C_v \subseteq [0, N]^2$ whose corners have integral coordinates. The root $r \in V$ of T corresponds to the cell $C_r := [0, N]^2$. Recursively, consider a node $v \in V$, corresponding to a cell C_v and assume that $C_v = [x_1^{(1)}, x_2^{(1)}] \times [x_1^{(2)}, x_2^{(2)}]$. If C_v is a unit



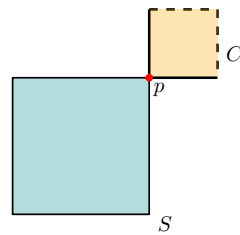
■ **Figure 3** Left figure shows a quad-tree cell in purple. The maximum area-covering square (solid black) is picked, while the other edge-covering squares (dashed) are not. Right figure shows the quad-tree cells (level-wise color-coded) containing an uncovered point. In increasing order of depth of these cells, at most 4 maximum-area covering squares (solid black) are picked together per cell.

square, i.e., $|x_2^{(1)} - x_1^{(1)}| = |x_2^{(2)} - x_1^{(2)}| = 1$, then we define that v is a leaf. Otherwise, we define that v has four children v_1, v_2, v_3, v_4 that correspond to the four cells that we obtain if we *partition* C_v into four equal sized smaller cells, i.e., define $x_{\text{mid}}^{(1)} := (x_2^{(1)} - x_1^{(1)})/2$ and $x_{\text{mid}}^{(2)} := (x_2^{(2)} - x_1^{(2)})/2$ and $C_{v_1} = [x_1^{(1)}, x_{\text{mid}}^{(1)}) \times [x_1^{(2)}, x_{\text{mid}}^{(2)})$, $C_{v_2} = [x_1^{(1)}, x_{\text{mid}}^{(1)}) \times [x_{\text{mid}}^{(2)}, x_2^{(2)})$, $C_{v_3} = [x_{\text{mid}}^{(1)}, x_2^{(1)}) \times [x_1^{(2)}, x_{\text{mid}}^{(2)})$, and $C_{v_4} = [x_{\text{mid}}^{(1)}, x_2^{(1)}) \times [x_{\text{mid}}^{(2)}, x_2^{(2)})$. Note that the depth of this tree is $\log N$, where depth of a node in the tree is its distance from the root of T , and depth of T is the maximum depth of any node in T . By the construction, each leaf node contains at most one point and it will lie on the bottom-left corner of the corresponding cell.

Offline algorithm

In the offline algorithm \mathcal{A}_{off} , we traverse T in a breadth-first-order, i.e., we order the nodes in V by their distances to the root r and consider them in this order (breaking ties arbitrarily but in a fixed manner). Suppose that in one iteration we consider a node $v \in V$, corresponding to a cell C_v . We check whether the squares selected in the ancestors of v cover all points in $P \cap C_v$. If this is the case, we do not select any squares from \mathcal{S} in this iteration (corresponding to v). Observe that hence we also do not select any squares in the iterations corresponding to the descendants of v in T (so we might as well skip the whole subtree rooted at v).

Suppose now that the squares selected in the ancestors of v do *not* cover all points in $P \cap C_v$. We call such a node to be *explored* by our algorithm. Let e be an edge of C_v . We say that a square containing e is *edge-covering* for e . We select a square from \mathcal{S} that is edge-covering for e and that has the largest intersection with C_v among all such squares in \mathcal{S} (we call such a square *maximum area-covering* for C_v for edge e). We break ties in an arbitrary but fixed way. If there is no square in \mathcal{S} that is edge-covering for e then we do not select a square corresponding to e . We do this for each of the four edges of C_v . See Figure 3. If we reach a leaf node, and if there is an uncovered point (note that it must be on the bottom-left corner of the cell), then we select any arbitrary square that covers the point (the existence of such a square is guaranteed as some square in OPT covers it). See Figure 4. This guarantees the feasibility of the solution.



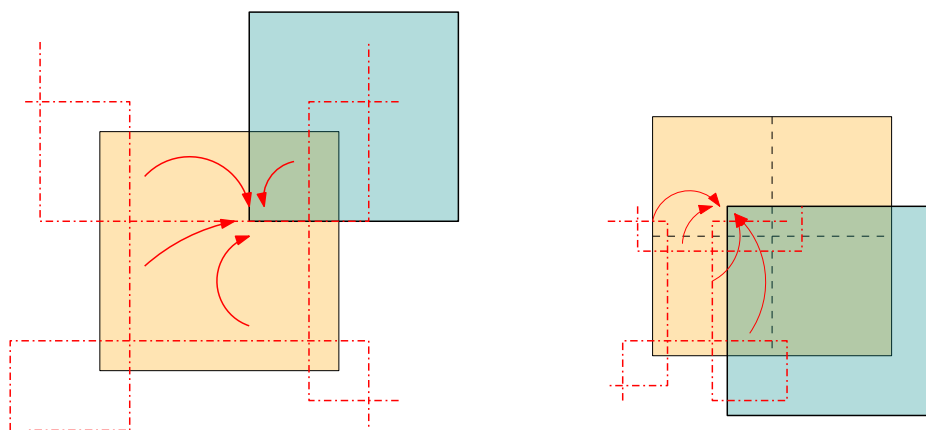
■ **Figure 4** Point p lies in a leaf cell C (which may not even have any edge-covering squares). In this case, we pick an arbitrary square S to cover the point (since one such square always exists).

► **Lemma 1.** \mathcal{A}_{off} outputs a feasible set cover for the points in P .

Approximation ratio

Let $\text{ALG} \subseteq \mathcal{S}$ denote the selected set of squares and let OPT denote the optimal solution. To prove $O(\log N)$ -approximation guarantee, the main idea is the following: consider a node $v \in V$ and suppose that we selected at least one square in the iteration corresponding to v . If C_v contains a corner of a square $S \in \text{OPT}$, then we charge the (at most four) squares selected for v to S . Otherwise, we argue that the squares selected for v cover at least as much of C_v as the squares in OPT , and that they cover all the remaining uncovered points in $P \cap C_v$. Thus we do not select any further squares in the descendants of v . The squares selected for v are charged to the parent of v (which contains a corner of a square $S \in \text{OPT}$). See Figure 5. Since each corner of each square $S \in \text{OPT}$ is contained in $O(\log N)$ cells, we show that each square $S \in \text{OPT}$ receives a total charge of $O(\log N)$. Thus, we obtain the following lemma.

► **Lemma 2.** We have that $|\text{ALG}| = O(\log N) \cdot |\text{OPT}|$.



■ **Figure 5** Charging picked (red) edge-covering squares to the corner of a (cyan) square in OPT . In the image on the left, the (yellow) cell contains a corner of the square from OPT , and in the image on the right, the parent of the cell contains such a corner.

2.1 Online set cover for squares

In the following, we first present an $O(\log N)$ -competitive online algorithm for online set cover for squares. Then we improve its competitive ratio to $O(\log n)$ in the setting where we are given a set of n points at the beginning, and the adversary can introduce only points from this set.

2.1.1 $O(\log N)$ -competitive online algorithm

We want to turn our offline algorithm \mathcal{A}_{off} into an online algorithm \mathcal{A}_{on} , assuming that in each *round* a new point is introduced by the adversary. The key insight for this is that the algorithm above is *monotone*, i.e., if we add a point to P , then it outputs a superset of the squares from \mathcal{S} that it had output before (when running it on P only). For a given set of points P , let $\text{ALG}(P) \subseteq \mathcal{S}$ denote the set of squares that our (offline) algorithm outputs.

► **Lemma 3.** *Consider a set of points P and a point p . Then $\text{ALG}(P) \subseteq \text{ALG}(P \cup \{p\})$.*

Initially, $P = \emptyset$. If a point p is introduced by the adversary, then we compute $\text{ALG}(P)$ (where P denotes the set of previous points, i.e., *without* p) and $\text{ALG}(P \cup \{p\})$ and we add the squares in $\text{ALG}(P \cup \{p\}) \setminus \text{ALG}(P)$ to our solution. Therefore, due to Lemma 2 and Lemma 3 we obtain an $O(\log N)$ -competitive online algorithm.

2.1.2 $O(\log n)$ -competitive online set cover for squares

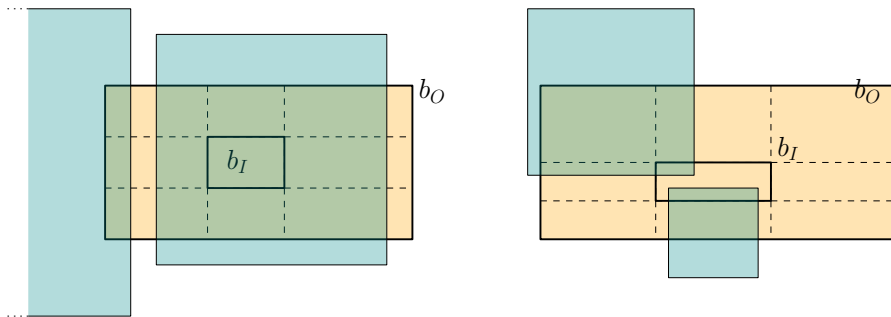
We assume now that we are given a set $\tilde{P} \subseteq \mathbb{R}^2$ with $|\tilde{P}| = n$ such that in each round a point from \tilde{P} is inserted to P , i.e., $P \subseteq \tilde{P}$ after each round. We want to get a competitive ratio of $O(\log n)$ in this case. If $N = n^{O(1)}$ then this is immediate. Otherwise, we extend our algorithm such that it uses the balanced box-decomposition tree (or BBD-tree) data structure due to Arya et al. [6], instead of the quad-tree. Before the first round, $P = \emptyset$ and we initialize the BBD-tree which yields a tree $\tilde{T} = (\tilde{V}, \tilde{E})$ with the following properties:

- each node $v \in \tilde{V}$ corresponds to a cell $\tilde{C}_v \subseteq [0, N]^2$ which is described by an outer box $b_O \subseteq [0, N]^2$ and an inner box $b_I \subseteq b_O$; both of them are axis-parallel rectangles and $\tilde{C}_v = b_O \setminus b_I$ (Note that b_I could be the empty set).
- the aspect ratio of b_O , i.e., the ratio between the length of the longest edge to the length of the shortest edge of b_O , is bounded by 3.
- if $b_I \neq \emptyset$, then b_I is *sticky* which intuitively means that in each dimension, the distance of b_I to the boundary of b_O is either 0 or at least the width of b_I . Formally, assume that $b_O = [x_O^{(1)}, x_O^{(2)}] \times [y_O^{(1)}, y_O^{(2)}]$ and $b_I = [x_I^{(1)}, x_I^{(2)}] \times [y_I^{(1)}, y_I^{(2)}]$. Then $x_O^{(1)} = x_I^{(1)}$ or $x_I^{(1)} - x_O^{(1)} \geq x_I^{(2)} - x_I^{(1)}$. Also $x_O^{(2)} = x_I^{(2)}$ or $x_O^{(2)} - x_I^{(2)} \geq x_I^{(2)} - x_I^{(1)}$. Analogous conditions also hold for the y -coordinates.
- each node $v \in \tilde{V}$ is a leaf or it has two children $v_1, v_2 \in \tilde{V}$; in the latter case $\tilde{C}_v = \tilde{C}_{v_1} \cup \tilde{C}_{v_2}$.
- the depth of \tilde{T} is $O(\log n)$ and each point $q \in [0, N]^2$ is contained in $O(\log n)$ cells.
- each leaf node $v \in \tilde{V}$ contains at most one point in \tilde{P} .

In the construction of the BBD-tree, we make the cells at the same depth disjoint so that a point p may be contained in exactly one cell at a certain depth. Hence, for a cell $\tilde{C}_v = b_O \setminus b_I$ we assume both b_O and b_I to be *closed* set. We now describe an adjustment of our offline algorithm from Section 2, working with \tilde{T} instead of T . Similarly, as before, we traverse \tilde{T} in a breadth-first-order. Suppose that in one iteration we consider a node $v \in \tilde{V}$

corresponding to a cell \tilde{C}_v . We check whether the squares selected in the ancestors of v cover all points in $P \cap \tilde{C}_v$. If this is the case, we do not select any squares from \mathcal{S} in this iteration corresponding to v .

Suppose now that the squares selected in the ancestors of v do *not* cover all points in $P \cap \tilde{C}_v$. Similar to Section 2, we want to select $O(1)$ squares for \tilde{C}_v such that if \tilde{C}_v contains no corner of a square $S \in \text{OPT}$, then the squares we selected for \tilde{C}_v should cover all points in $P \cap \tilde{C}_v$. Similarly as before, for each edge e of b_O we select a square from \mathcal{S} that contains e and that has the largest intersection with b_O among all such squares in \mathcal{S} . We break ties in an arbitrary but fixed way. However, as \tilde{C}_v may not be a square and can have holes (due to b_I), apart from the edge-covering squares, we need to consider two additional types of squares in OPT with nonempty overlap with \tilde{C}_v : (a) crossing \tilde{C}_v , i.e., squares that intersect two parallel edges of b_O ; (b) has one or two corners inside b_I . See Figure 6.



■ **Figure 6** Possible intersections of a (cyan) square from OPT with a cell, such that no corner of the square is in the cell. The left image shows edge-covering, and crossing squares. The right image shows squares with one of two corners inside b_I .

The following *greedy subroutine* \mathcal{G} will be useful in our algorithm to handle such problematic cases. Let R be a box of width w and height h such that $w/h \leq B$, for some constant $B \in \mathbb{N}$; and P_R be a set of points inside R that can be covered by a collection of vertically-crossing (i.e., they intersect both horizontal edges of R) squares \mathcal{S}' . Then, the set of squares picked according to \mathcal{G} covers P_R in the following way:

- While there is an uncovered point $p' \in P_R$:
 - Consider the leftmost such uncovered point $p \in P_R$.
 - Select the vertically-crossing square intersecting p (by assumption, such a square exists) with the rightmost edge.

(The above subroutine is for finding vertically-crossing squares. For finding horizontally-crossing squares, we can appropriately rotate the input 90° anti-clockwise, and apply the same subroutine.) Then, we have the following claim about the aforementioned subroutine.

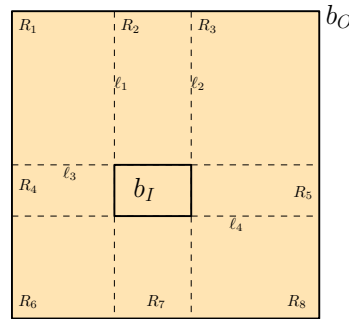
▷ **Claim 4.** Let R be a box of width w and height h such that $w/h \leq B$, for some constant $B \in \mathbb{N}$; and P_R be a set of points inside R that can be covered by a collection of vertically-crossing (i.e., they intersect both horizontal edges of R) squares \mathcal{S}' . Then we can find at most $B + 1$ squares from \mathcal{S}' that can cover all points inside R .

We have an analogous claim for horizontally-crossing squares when $h/w \leq B$.

Now we describe our algorithm. First, we take care of the squares that can cross b_O . So, we apply the greedy subroutine \mathcal{G} on b_O . As b_O has bounded aspect ratio of 3, from Claim 4, we obtain at most $(3 + 1) + (1 + 1) = 6$ squares that can cross C_v vertically or horizontally. If $b_I = \emptyset$, we do not select any more squares. Otherwise, we need to take care of the squares

that can have one or two corners inside b_I . Let $\ell_1, \ell_2, \ell_3, \ell_4$ denote the four lines that contain the four edges of b_I . Observe that $\ell_1, \ell_2, \ell_3, \ell_4$ partition b_O into up to nine rectangular regions, one being identical to b_I . See Figure 7. For each such rectangular region R , if it is sharing a horizontal edge with b_I , we again use \mathcal{G} to select vertically-crossing squares. Otherwise, if R is sharing a vertical edge with b_I , we use the subroutine \mathcal{G} appropriately to select horizontally-crossing squares. This takes care of squares having two corners inside b_I . Otherwise, if the rectangular region R does not share an edge with b_I , then we check if there is a square $S \in \mathcal{S}$ with a corner within b_I that completely contains R . We add S to our solution too. This finally takes care of the case when a square has a single corner inside b_I .

Finally, to complete our algorithm, before its execution, we do the following: for every leaf v for which C_v contains at most one point $p \in \tilde{P}$, we associate a fixed square which covers p . Then, if our algorithm reaches a leaf v while traversing that has an uncovered point p , we pick the associated square with this leaf that covers it. This condition in our algorithm guarantees feasibility.



■ **Figure 7** Outer box b_O being partitioned into at most 9 rectangles due to inner box b_I .

Then using the following lemma, we can establish a similar charging scheme as in Section 2.

► **Lemma 5.** *Let \tilde{C}_v be a cell such that the squares selected in the ancestors of v do not cover all points in $P \cap \tilde{C}_v$. Then*

- (a) *we select at most $O(1)$ squares for \tilde{C}_v and*
- (b) *if \tilde{C}_v contains no corner of a square $S \in \text{OPT}$, then the squares we selected for \tilde{C}_v cover all points in $P \cap \tilde{C}_v$.*

To pay for our solution, we charge each corner q of a square $S \in \text{OPT}$ at most $O(\log n)$ times. Hence, our approximation ratio is $O(\log n)$. Similarly as in Section 2, we can modify the above offline algorithm to an online algorithm with an approximation ratio of $O(\log n)$.

► **Theorem 6.** *There is a deterministic $O(\log n)$ -competitive online algorithm for set cover for axis-parallel squares of arbitrary sizes.*

It is a natural question whether algorithms having a competitive factor better than $O(\log n)$ are possible for online set cover for squares. We answer this question in the negative.

► **Theorem 7.** *Any deterministic or randomized online algorithm for set cover for unit squares has a competitive ratio of $\Omega(\log n)$, even if all squares contain the origin and all points are contained in the same quadrant.*

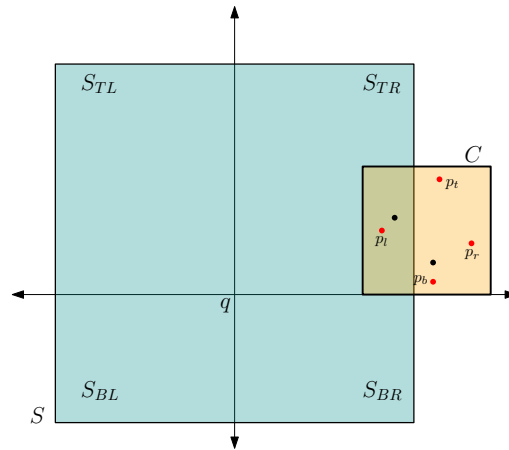
3 Online hitting set for squares

Now we present our online algorithm for hitting set for squares. We are given a fixed set of points $P \subseteq [0, N]^2$ with integral coordinates. We maintain a set P' of selected points such that initially $P' := \emptyset$. In each round, we are given a square $S \subseteq [0, N]^2$ whose corners

have integral coordinates. We assume w.l.o.g. that N is a power of 2. Let Q be all (grid) points with integral coordinates in $[0, N]^2$, i.e., $P \subseteq Q$. For each point $q \in Q$ we say that $q = (q_x, q_y)$ is of level ℓ if both q_x and q_y are integral multiples of $N/2^\ell$, but not both are integral multiples of $N/2^{(\ell-1)}$. We build the same quad-tree as in Section 2. We say that a cell C_v is of level ℓ if its side length equals $N/2^\ell$.

We present our algorithm now. Suppose that in some round a new square S is given. If $S \cap P' \neq \emptyset$ then we do not add any point to P' . Suppose now that $S \cap P' = \emptyset$. Let q be a point of smallest level among all points in $Q \cap S$ (if there are many such points, then we select an arbitrary point in $Q \cap S$ of smallest level). Intuitively, we interpret q as if it were the origin and partition the plane into four quadrants. We define $O_{TR} := \{(p_x, p_y) \mid p_x \geq q_x, p_y \geq q_y\}$, and $S_{TR} := O_{TR} \cap S$, and define similarly O_{TL}, O_{BR}, O_{BL} , and S_{TL}, S_{BR}, S_{BL} . Consider O_{TR} and S_{TR} . For each level $\ell = 0, 1, \dots, \log N$, we do the following. Consider each cell C of level ℓ in some fixed order such that $C \subseteq O_{TR}$ and S_{TR} is edge-covering for some edge e of C . Then, for each edge identify the point p_b (p_t, p_l, p_r , resp.) in $P \cap C$ that is closest to its bottom (top, left, and right, resp.) edge. We add these (at most 4) points to our solution if at least one of p_b, p_t, p_l, p_r is contained in S_{TR} (see Figure 8). If we add at least one such point p of the cell C to P' in this way, we say that C gets activated. Note that we add possibly all of the points p_b, p_t, p_l, p_r to P' even though only one may be contained in S_{TR} . This is to ensure that C gets activated at most once during a run of the online algorithm. If for the current level ℓ we activate at least one cell C of level ℓ , then we stop the loop and do not consider the other levels $\ell + 1, \dots, \log N$. Otherwise, we continue with level $\ell + 1$. We do a symmetric operation for the pairs $(O_{TL}, S_{TL}), (O_{BR}, S_{BR})$, and (O_{BL}, S_{BL}) .

For the analysis of the algorithm, we show that for a point $p \in \text{OPT}$, the number of rounds for which the adversary can possibly introduce a square S such that $p \in S$ and $S \cap P' = \emptyset$ is $O(\log N)$. More specifically, we identify a set of cells \mathcal{C}_p such that $|\mathcal{C}_p| = O(\log N)$ and in any such round where $p \in S$, one of the cells in \mathcal{C}_p is activated. The competitive ratio of the algorithm follows from the fact that any cell of the quad-tree is activated at most once.



■ **Figure 8** In the cell C (contained in O_{TR}) the red points are chosen by the algorithm.

► **Theorem 8.** *There is an $O(\log N)$ -competitive deterministic online algorithm for hitting set for axis-parallel squares of arbitrary sizes.*

This is tight, as even for intervals, Even et al. [21] have shown an $\Omega(\log N)$ lower bound.

4 Dynamic set cover for d -dimensional hyperrectangles

In this section, we will design an algorithm to dynamically maintain an approximate set cover for d -dimensional hyperrectangles. The main result we prove in this section is the following.

► **Theorem 9.** *After performing a pre-processing step which takes $O(m \log^{2d} m)$ time, there is an algorithm for dynamic set cover for d -dimensional hyperrectangles with an approximation factor of $O(\log^{4d-1} m)$ and an update time of $O(\log^{2d+2} m)$.*

Our goal is to adapt the quad-tree based algorithms designed in the previous sections of the paper. As a first step towards that, we transform the problem such that the points and hyperrectangles in \mathbb{R}^d get transformed to points and *hypercubes* in \mathbb{R}^{2d} , and the new problem is to cover the points in \mathbb{R}^{2d} with these hypercubes. As discussed in the introduction, a simple $2d$ -dimensional quad-tree on the hypercubes does not suffice for our purpose. We augment the quad-tree in two ways: (a) at each node, we collect the hypercubes which are edge-covering w.r.t. that node and “ignore” that dimension in which they are edge-covering, and (b) recursively construct a $(2d-1)$ -dimensional quad-tree on these hypercubes based on the remaining $2d-1$ dimensions. We call this new structure an *extended quad-tree*. This yields the important property that any point in \mathbb{R}^{2d} will belong to only $O(\log^{2d} m)$ cells in the extended quad-tree. Furthermore, at the 1-dimensional cells of the extended quad-tree, for each cell we will identify $O(1)$ “most useful” hypercubes. This ensures that any point belongs to only $O(\log^{2d} m)$ of these most useful hypercubes. As a result, a “bounded frequency” set system can be constructed with the most useful hypercubes. The dynamic algorithm from Bhattacharya et al. [12] (for general set cover) works efficiently on bounded frequency set systems and applying it in our setting leads to an $O(\log^{4d-1} m)$ -approximation algorithm.

4.1 Transformation to hypercubes in \mathbb{R}^{2d}

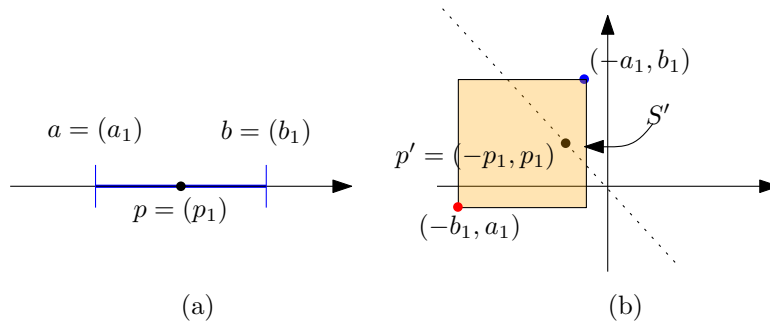
Recall that the input is a set of points P and \mathcal{S} is a collection of hyperrectangles in \mathbb{R}^d .

By a standard rank-space reduction, we can assume that each corner of each hyperrectangle in \mathcal{S} is contained in $\{0, 1, \dots, 2m\}^d$ and that the intersection of any two input hyperrectangles in \mathcal{S} is either d -dimensional or empty. Also, we perturb each input point $p \in P$ slightly so that p is not contained in the face of any hyperrectangle in \mathcal{S} , without changing the collection of hyperrectangles that cover p .

The first step of the algorithm is to transform the hyperrectangles in \mathcal{S} to hypercubes in \mathbb{R}^{2d} . Consider a hyperrectangle $S \in \mathcal{S}$ with $a = (a_1, \dots, a_d)$ and $b = (b_1, \dots, b_d)$ being the “lower-left” and the “upper-right” corners of S , respectively. Let $\Delta = \max_{j=1}^d (b_j - a_j)$. Then S is transformed to a hypercube S' in \mathbb{R}^{2d} with side-length Δ and “top-right” corner $(-a_1, -a_2, \dots, -a_d, b_1, b_2, \dots, b_d)$. Let \mathcal{S}' be the collection of these m transformed hypercubes. Let P' be the set of n points in \mathbb{R}^{2d} obtained by transforming each point $p = (p_1, \dots, p_d) \in P$ to $p' = (-p_1, \dots, -p_d, p_1, \dots, p_d)$. See Figure 9 for an example.

► **Observation 10.** *A point $p = (p_1, \dots, p_d)$ lies inside S if and only if the point $p' = (-p_1, \dots, -p_d, p_1, \dots, p_d)$ lies inside S' .*

After applying the above transformation, we note that the coordinates of each corner of each hypercube in \mathcal{S}' will be contained in $\{-4m, \dots, 0\}^d \times \{-2m, \dots, 2m\}^d$. We perform a suitable shifting so that all the corners of the hypercubes in \mathcal{S}' will be contained in $\{0, \dots, 4m\}^{2d}$. Then, our assumption on the input set of hyperrectangles \mathcal{S} and the input points P implies that for any point $p' \in P'$, it does not lie on a face of any hypercube in \mathcal{S}' .



■ **Figure 9** (a) A point p in 1-D lying inside an interval $S = [a_1, b_1]$, and (b) the transformation of p into a point $p' = (-p_1, p_1)$, and the transformation of S into a square S' in 2-D.

4.2 Constructing a bounded frequency set system

We will now present a technique to select a set $\hat{S} \subseteq S'$ with the following properties:

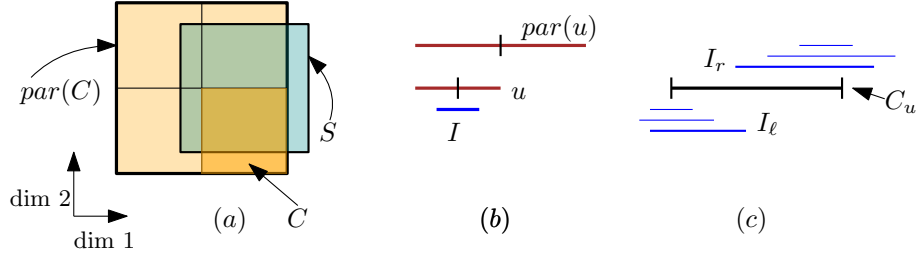
1. (*Bounded frequency*) Any point in P' lies inside $O(\log^{2d} m)$ hypercubes in \hat{S} .
2. An α -approximation dynamic set cover algorithm for (P', \hat{S}) implies an $O(\alpha \log^{2d-1} m)$ -approximation dynamic set cover algorithm for (P', S') .
3. The time taken to update the solution for the set system (P', \hat{S}) is $O(\log^{2d} m \cdot \log^2 n)$.
4. The time taken to construct the set \hat{S} is $O(m \log^{2d} m)$.

4.2.1 Extended quad-tree for 2-dimensional squares

Given a set of squares S' , we construct a 2-dimensional quad-tree \mathbb{T} (as defined in Section 2) such that its root cell contains all the squares in S' . Consider a node $v \in \mathbb{T}$ and a square $S \in S'$. Let C and $par(C)$ be the cell corresponding to node v and the parent node of v , respectively. Let $proj_i(C)$, $proj_i(par(C))$ and $proj_i(S)$ be the projection of C , $par(C)$ and S , respectively, on to the i -th dimension. Then S is i -long at v if and only if $proj_i(C) \subseteq proj_i(S)$ but $proj_i(par(C)) \not\subseteq proj_i(S)$. See Figure 10(a). For all $u \in \mathbb{T}$, let $\mathcal{S}(u, i) \subseteq S'$ be the squares which are i -long at node u . Intuitively, these are squares that cover the edge of C in the i -th dimension but do not cover any edge of $par(C)$ in the i -th dimension. Now, at each node of \mathbb{T} we will construct two *secondary structures* as follows: the first structure is a 1-dimensional quad-tree built on the projection of the squares in $\mathcal{S}(u, 1)$ on to the second dimension, and the second structure is a 1-dimensional quad-tree built on the projection of the squares in $\mathcal{S}(u, 2)$ on to the first dimension.

In each secondary structure, an interval I (corresponding to a square $S \in S'$) is *assigned* to a node u if and only if u is the node with the smallest depth (the root is at depth zero) where I intersects either the left endpoint or the right endpoint of the cell C_u . See Figure 10(b). By this definition, any interval will be assigned to at most two nodes in the secondary structure.

Now we will use \mathbb{T} to construct the geometric collection \hat{S} . Let V_{sec} be the set of nodes in all the secondary structures of \mathbb{T} . For any node $u \in V_{sec}$, among its assigned intervals which intersect the left (resp., right) endpoint of the cell C_u , identify the *maximal* interval I_ℓ (resp., I_r), i.e., the interval which has maximum overlap with C_u . See Figure 10(c). We then do the following set of operations over all the nodes in V_{sec} : For a node $u \in V_{sec}$, denote by S' and S'' the corresponding squares for the assigned intervals I_ℓ and I_r , respectively. Further, let w be the node in \mathbb{T} , on which the secondary structure of u was constructed. Then, we include in \hat{S} the rectangles $S_1 \cap C_w$ and $S_2 \cap C_w$.



■ **Figure 10** (a) A square S which is 1-long at node v (corrs. cell C is highlighted in darker orange), (b) I is assigned to the two children of v , and (c) the maximal intervals I_ℓ and I_r at C_v .

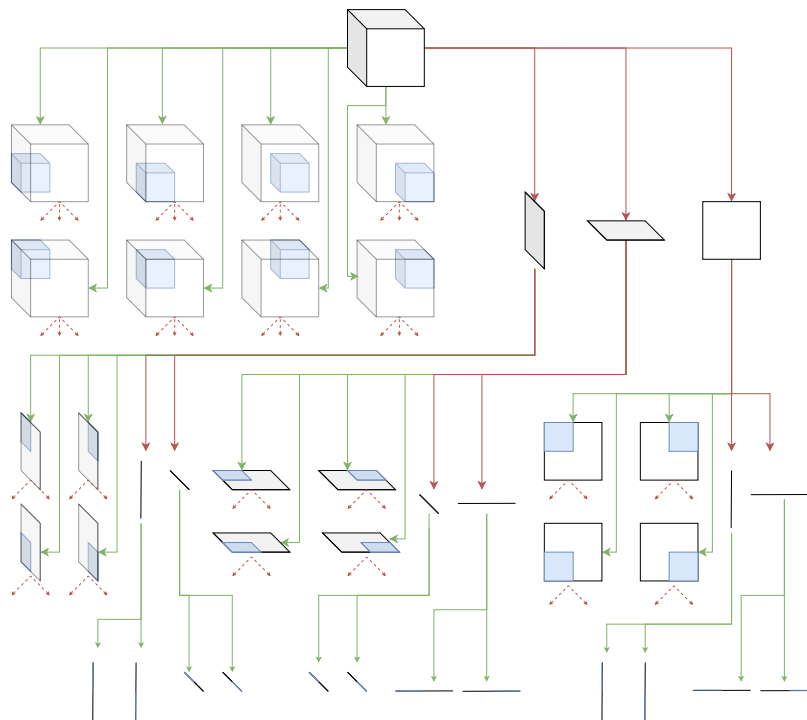
4.2.2 Extended quad-tree for $2d$ -dimensional hypercubes

For $2d$ -dimensions, we need a generalization of the quad-tree defined in Section 2. For $d' > 2$, a d' -dimensional quad-tree is defined analogously to the the quad-tree defined in Section 2, where instead of four, each internal node will now have $2^{d'}$ children. Assume by induction that we have defined how to construct the extended quad-tree for all dimensions less than or equal to $2d-1$. (The base case is the extended quad-tree built for 2-dimensional squares.) We define now how to construct the structure for $2d$ -dimensional hypercubes. First construct the regular $2d$ -dimensional quad-tree \mathbb{T} for the set of hypercubes \mathcal{S}' . Consider any node $v \in \mathbb{T}$. Generalizing the previous definition, for any $1 \leq i \leq 2d$, a hypercube $S \in \mathcal{S}'$ is defined to be i -long at node v if and only if $\text{proj}_i(C) \subseteq \text{proj}_i(S)$, but $\text{proj}_i(\text{par}(C)) \not\subseteq \text{proj}_i(S)$. For all $v \in \mathbb{T}$, let $\mathcal{S}(v, i) \subseteq \mathcal{S}'$ be the hypercubes which are i -long at node v . Now, at each node of \mathbb{T} we will construct $2d$ secondary structures as follows: for all $1 \leq i \leq 2d$, the i -th secondary structure is a $(2d-1)$ -dimensional extended quad-tree built on $\mathcal{S}(v, i)$ and all its $2d$ dimensions except the i -th dimension. Specifically, any hypercube $S \in \mathcal{S}(v, i)$ of the form $\ell_1 \times \cdots \times \ell_i \times \cdots \times \ell_{2d}$ is projected to a $(2d-1)$ -dimensional hypercube $\ell_1 \times \cdots \times \ell_{i-1} \times \ell_{i+1} \times \cdots \times \ell_{2d}$. Let $\hat{\mathcal{S}}_v$ be the collection of the $(2d-1)$ -dimensional hyperrectangles that are inductively picked for the secondary structure constructed at $v \in \mathbb{T}$ using the routine. Define the function g which maps a $(2d-1)$ -dimensional hyperrectangle picked as part of the collection $\hat{\mathcal{S}}_v$ (for a $v \in \mathbb{T}$) to its corresponding $2d$ -dimensional hypercube $S \in \mathcal{S}'$. We now define the collection of sets $\hat{\mathcal{S}}$ consisting of $2d$ -dimensional hyperrectangles: $\hat{\mathcal{S}} \leftarrow \bigcup_{v \in \mathbb{T}} (\bigcup_{S' \in \hat{\mathcal{S}}_v} (g(S') \cap C_v))$. Then we prove the following three key properties of $\hat{\mathcal{S}}$.

- ▶ **Lemma 11.** (*Feasibility*) Any point $p \in P'$ is covered by at least one set in $\hat{\mathcal{S}}$.
- ▶ **Lemma 12.** (*Bounded frequency*) Any point in P' lies inside $O(\log^{2d} m)$ sets in $\hat{\mathcal{S}}$.
- ▶ **Lemma 13.** If there is an α -approximation dynamic set cover algorithm for $(P', \hat{\mathcal{S}})$ then there is an $O(\alpha \log^{2d-1} m)$ -approximation dynamic set cover algorithm for (P', \mathcal{S}') .

4.3 The final algorithm

We run the $O(f)$ -approximate algorithm by Bhattacharya et al. [12] for the dynamic set cover problem as a black box on the instance $(P', \hat{\mathcal{S}})$. If ALG is the reported solution, we also report ALG as our solution for the instance (P', \mathcal{S}') . One can prove that this yields Theorem 9.



■ **Figure 11** Extended quad-tree with a $2 \times 2 \times 2$ cube as the root.

4.4 Weighted setting

We present an extension of our algorithm to the setting where each hyperrectangle $S \in \mathcal{S}$ has a weight $w_S \in [1, W]$. First, we *round* the weight of each set S to the smallest power of two greater than or equal to w_S , leading to $O(\log W)$ weight classes. Next, for each weight class, we build an extended quad-tree as in the previous section. Finally, let $\hat{\mathcal{S}}$ be the collection of (maximal) hypercubes obtained from all the $O(\log W)$ extended quad-trees. We run the dynamic set cover algorithm of Bhattacharya et al. [12] on $(P', \hat{\mathcal{S}})$.

► **Theorem 14.** *There is an algorithm for weighted dynamic set cover for d -dimensional hyperrectangles with an approximation factor of $O(\log^{4d-1} m \cdot \log W)$ and an update time of $O(\log^{2d} m \cdot \log^3(Wm))$.*

5 Dynamic hitting set for d -dimensional hyperrectangles

In this section we claim a dynamic algorithm for hitting set for d -dimensional hyperrectangles. We obtain this by reducing the problem to an instance of dynamic set cover for $2d$ -dimensional hypercubes and use the algorithm designed in the previous section to solve the instance.

► **Theorem 15.** *After performing a pre-processing step which takes $O(n \log^{2d} n)$ time, there is an algorithm for hitting set for d -dimensional hyperrectangles with an approximation factor of $O(\log^{4d-1} n)$ and an update time of $O(\log^{2d+2} n)$. In the weighted setting, the approximation factor is $O(\log^{4d-1} n \cdot \log W)$ and the update time is $O(\log^{2d} n \log^3(Wn))$.*

References

- 1 Amir Abboud, Raghavendra Addanki, Fabrizio Grandoni, Debmalya Panigrahi, and Barna Saha. Dynamic set cover: improved algorithms and lower bounds. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 114–125, 2019.
- 2 Pankaj K. Agarwal, Hsien-Chih Chang, Subhash Suri, Allen Xiao, and Jie Xue. Dynamic geometric set cover and hitting set. In *36th International Symposium on Computational Geometry, SoCG 2020*, volume 164 of *LIPICs*, pages 2:1–2:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- 3 Noga Alon, Baruch Awerbuch, and Yossi Azar. The online set cover problem. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing (STOC)*, pages 100–105, 2003.
- 4 Boris Aronov, Esther Ezra, and Micha Sharir. Small-size ϵ -nets for axis-parallel rectangles and boxes. *SIAM Journal on Computing*, 39(7):3248–3282, 2010.
- 5 Sanjeev Arora. Polynomial time approximation schemes for euclidean traveling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.
- 6 Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM (JACM)*, 45(6):891–923, 1998.
- 7 Sepehr Assadi and Shay Solomon. Fully dynamic set cover via hypergraph maximal matching: An optimal approximation through a local approach. In *29th Annual European Symposium on Algorithms, ESA 2021*, page 8. Schloss Dagstuhl-Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, 2021.
- 8 Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf. Computational geometry. In *Computational geometry*, pages 1–17. Springer, 1997.
- 9 Sayan Bhattacharya, Monika Henzinger, and Giuseppe F. Italiano. Deterministic fully dynamic data structures for vertex cover and matching. *SIAM Journal on Computing*, 47(3):859–887, 2018.
- 10 Sayan Bhattacharya, Monika Henzinger, and Giuseppe F. Italiano. Dynamic algorithms via the primal-dual method. *Information and Computation*, 261:219–239, 2018.
- 11 Sayan Bhattacharya, Monika Henzinger, and Danupon Nanongkai. A new deterministic algorithm for dynamic set cover. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 406–423. IEEE, 2019.
- 12 Sayan Bhattacharya, Monika Henzinger, Danupon Nanongkai, and Xiaowei Wu. Dynamic set cover: Improved amortized and worst-case update time. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2537–2549. SIAM, 2021.
- 13 Sujoy Bhore, Jean Cardinal, John Iacono, and Grigorios Koumoutsos. Dynamic geometric independent set. In *Japan conference on Discrete and Computational Geometry, Graphs, and Games*, 2021.
- 14 Niv Buchbinder and Joseph Naor. The design of competitive online algorithms via a primal-dual approach. *Found. Trends Theor. Comput. Sci.*, 3(2-3):93–263, 2009.
- 15 Jean Cardinal, John Iacono, and Grigorios Koumoutsos. Worst-case efficient dynamic geometric independent set. In *29th Annual European Symposium on Algorithms (ESA 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.
- 16 Timothy M. Chan, Elyot Grant, Jochen Könemann, and Malcolm Sharpe. Weighted capacitated, priority, and geometric set cover via improved quasi-uniform sampling. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pages 1576–1585. SIAM, 2012.
- 17 Timothy M. Chan and Qizheng He. More dynamic data structures for geometric set cover with sublinear update time. In *37th International Symposium on Computational Geometry (SoCG 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

- 18 Timothy M. Chan, Qizheng He, Subhash Suri, and Jie Xue. Dynamic geometric set cover, revisited. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3496–3528. SIAM, 2022.
- 19 Kenneth L. Clarkson and Kasturi Varadarajan. Improved approximation algorithms for geometric set cover. In *Proceedings of the twenty-first annual symposium on Computational geometry (SoCG)*, pages 135–141, 2005.
- 20 Justin Dallant and John Iacono. Conditional lower bounds for dynamic geometric measure problems. *arXiv preprint arXiv:2112.10095*, 2021.
- 21 Guy Even and Shakhar Smorodinsky. Hitting sets online and vertex ranking. In *Algorithms - ESA 2011 - 19th Annual European Symposium*, volume 6942 of *Lecture Notes in Computer Science*, pages 347–357. Springer, 2011.
- 22 Anupam Gupta, Gregory Kehne, and Roie Levin. Random order online set cover is as easy as offline. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1253–1264. IEEE, 2022.
- 23 Anupam Gupta, Ravishankar Krishnaswamy, Amit Kumar, and Debmalya Panigrahi. Online and dynamic algorithms for set cover. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 537–550, 2017.
- 24 Anupam Gupta and Roie Levin. Fully-dynamic submodular cover with bounded recourse. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1147–1157. IEEE, 2020.
- 25 Anupam Gupta and Roie Levin. The online submodular cover problem. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1525–1537. SIAM, 2020.
- 26 Sarel Har-Peled and Mira Lee. Weighted geometric set cover problems revisited. *Journal of Computational Geometry*, 3(1):65–85, 2012.
- 27 Monika Henzinger, Stefan Neumann, and Andreas Wiese. Dynamic approximate maximum independent set of intervals, hypercubes and hyperrectangles. In *36th International Symposium on Computational Geometry, SoCG 2020*, volume 164 of *LIPICs*, pages 51:1–51:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- 28 Arindam Khan, Aditya Lonkar, Saladi Rahul, Aditya Subramanian, and Andreas Wiese. Online and dynamic algorithms for geometric set cover and hitting set. *CoRR*, abs/2303.09524, 2023. [arXiv:2303.09524](https://arxiv.org/abs/2303.09524).
- 29 Nabil H. Mustafa, Rajiv Raman, and Saurabh Ray. Settling the apx-hardness status for geometric set cover. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 541–550. IEEE, 2014.
- 30 Nabil H. Mustafa and Saurabh Ray. PTAS for geometric hitting set problems via local search. In *Proceedings of the 25th ACM Symposium on Computational Geometry (SoCG)*, pages 17–22. ACM, 2009.
- 31 Kasturi Varadarajan. Weighted geometric set cover via quasi-uniform sampling. In *Proceedings of the forty-second ACM symposium on Theory of computing (STOC)*, pages 641–648, 2010.

Sparse Euclidean Spanners with Optimal Diameter: A General and Robust Lower Bound via a Concave Inverse-Ackermann Function

Hung Le ✉

University of Massachusetts, Amherst, MA, USA

Lazar Milenković ✉

Tel Aviv University, Israel

Shay Solomon ✉

Tel Aviv University, Israel

Abstract

In STOC'95 [6] Arya et al. showed that any set of n points in \mathbb{R}^d admits a $(1 + \epsilon)$ -spanner with hop-diameter at most 2 (respectively, 3) and $O(n \log n)$ edges (resp., $O(n \log \log n)$ edges). They also gave a general upper bound tradeoff of hop-diameter k with $O(n\alpha_k(n))$ edges, for any $k \geq 2$. The function α_k is the inverse of a certain Ackermann-style function, where $\alpha_0(n) = \lceil n/2 \rceil$, $\alpha_1(n) = \lceil \sqrt{n} \rceil$, $\alpha_2(n) = \lceil \log n \rceil$, $\alpha_3(n) = \lceil \log \log n \rceil$, $\alpha_4(n) = \log^* n$, $\alpha_5(n) = \lfloor \frac{1}{2} \log^* n \rfloor$, \dots . Roughly speaking, for $k \geq 2$ the function α_k is close to $\lfloor \frac{k-2}{2} \rfloor$ -iterated log-star function, i.e., log with $\lfloor \frac{k-2}{2} \rfloor$ stars.

Despite a large body of work on spanners of bounded hop-diameter, the fundamental question of whether this tradeoff between size and hop-diameter of Euclidean $(1 + \epsilon)$ -spanners is optimal has remained open, even in one-dimensional spaces. Three lower bound tradeoffs are known:

- An optimal k versus $\Omega(n\alpha_k(n))$ by Alon and Schieber [4], but it applies to stretch 1 (not $1 + \epsilon$).
- A suboptimal k versus $\Omega(n\alpha_{2k+6}(n))$ by Chan and Gupta [13].
- A suboptimal k versus $\Omega(\frac{n}{2^{6\lfloor k/2 \rfloor}} \alpha_k(n))$ by Le et al. [38].

This paper establishes the optimal k versus $\Omega(n\alpha_k(n))$ lower bound tradeoff for stretch $1 + \epsilon$, for any $\epsilon > 0$, and for any k . An important conceptual contribution of this work is in achieving optimality by shaving off an extremely slowly growing term, namely $2^{6\lfloor k/2 \rfloor}$ for $k \leq O(\alpha(n))$; such a fine-grained optimization (that achieves optimality) is very rare in the literature.

To shave off the $2^{6\lfloor k/2 \rfloor}$ term from the previous bound of Le et al., our argument has to drill much deeper. In particular, we propose a new way of analyzing recurrences that involve inverse-Ackermann style functions, and our key technical contribution is in presenting the **first explicit construction of concave versions** of these functions. An important advantage of our approach over previous ones is its *robustness*: While all previous lower bounds are applicable only to restricted 1-dimensional point sets, ours applies even to *random* point sets in constant-dimensional spaces.

2012 ACM Subject Classification Theory of computation → Sparsification and spanners

Keywords and phrases Euclidean spanners, Ackermann functions, convex functions, hop-diameter

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.47

Funding Hung Le is supported by the National Science Foundation under Grants No. CCF-2121952 and No. CCF-2237288. Lazar Milenković and Shay Solomon are co-funded by the European Union (ERC, DynOpt, 101043159). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. This research was also supported by the Israel Science Foundation (ISF) grant No.1991/1, and by a grant from the United States-Israel Binational Science Foundation (BSF), Jerusalem, Israel, and the United States National Science Foundation (NSF).



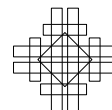
© Hung Le, Lazar Milenković, and Shay Solomon;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 47; pp. 47:1–47:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Let P be a set of n points in \mathbb{R}^d and let $G_P = (P, \binom{P}{2}, \|\cdot\|)$ be the complete weighted graph induced by P , which contains an edge (p, q) of weight $w(p, q) = \|p - q\|$, for every $p, q \in P$. A subgraph $G = (P, E, \|\cdot\|)$ of G_P , $E \subseteq \binom{P}{2}$, is called a *geometric graph*. For a parameter $t \geq 1$, a geometric graph G is called a *t -spanner* for P if, for all $p, q \in P$, G contains a *t -spanner path* between p and q (i.e., a path of *weight* at most $t\|p - q\|$).

Euclidean spanners have been studied extensively [17, 35, 5, 12, 19, 6, 20, 8, 47, 2, 13, 21, 51, 53, 23, 39, 32, 38]. They are important in theory and practice, having found many applications, e.g., in geometric approximation algorithms, network topology design, and distributed computing [19, 40, 47, 26, 28, 27, 31, 41]; see also the book [42].

The most basic requirement of a spanner is to be *sparse*, while achieving small stretch. Cornerstone results settle the stretch-size tradeoff: for any d -dimensional n -point Euclidean space and for any $\epsilon > 0$, there exists a $(1 + \epsilon)$ -spanner with $O_{\epsilon, d}(n)$ edges [58, 18, 34, 48, 35, 5], where the $O_{\epsilon, d}$ suppresses the dependence on ϵ and d . (More precisely, the size upper bound is $n \cdot O(\epsilon^{-d+1})$, and it was shown to be tight [39].) In many applications, however, the spanner should have additional useful properties of the underlying metric. One such property is the (*hop*-)diameter: a t -spanner for P has (*hop*-)diameter of k if, for any $p, q \in P$, there is a t -spanner path between p and q with at most k edges (or hops). Having a small diameter is important for various applications (e.g., routing protocols) [7, 1, 2, 13, 21, 32].

While the stretch-size tradeoff is fully understood including the dependence on ϵ and d , the extended tradeoff of stretch-size-diameter is not fully understood yet even for fixed ϵ and d . Our goal is to achieve a full understanding of this tradeoff for fixed ϵ and d .

If the points are in general position, a 1-spanner must include basically all $\binom{n}{2}$ edges of the underlying metric. For points lying on a line, the simple path connecting them provides 1-spanner, but its diameter is worst-possible, $n - 1$. Surprisingly perhaps, *all* previous lower bounds for the stretch-size-diameter tradeoff apply to line metrics. Understanding line metrics, and more generally tree metrics, is also important from the upper bounds front. In particular, the problem of constructing sparse 1-spanners with bounded diameter for line and tree metrics is closely related to several other fundamental problems. As an example, consider the extremely well-studied problem of partial sums, where we are given an array A of semigroup elements $A[1], \dots, A[n]$ and are asked to construct a small-sized data structure, so that given a query i, j for $1 \leq i < j \leq n$, the partial sum $\sum_{i \leq k \leq j} A[k]$ can be computed efficiently. A 1-spanner for the corresponding set $A[1], \dots, A[n]$ with bounded diameter is basically what we are looking for: A 1-spanner path between $A[i]$ and $A[j]$ that consists of at most k edges can be used for answering a query i, j within time $O(k)$. Other closely related problems include the tree product queries in semigroup problem (a generalization of partial sums) and its variants (see [55, 58, 4, 16, 46, 2], and the references therein), the MST verification problem [37, 36, 44], and the problem of shortcutting digraphs [56, 57, 10].

1.1 Previous Work on Spanners with Tiny diameter

1.1.1 Upper bounds

1-spanners for line and tree metrics. Let $T = (T, rt)$ be a (possibly weighted) n -vertex rooted tree, and let M_T be the tree metric induced by T . A spanning subgraph G of M_T is said to be a *1-spanner* for T , if for every pair of vertices, their distance in G is equal to their distance in T . One can define t -spanners for T , with $t \geq 1$, but essentially all previous work here concerned stretch 1. Alon and Schieber [4] showed that for any

n -point tree metric, a 1-spanner with diameter 2 (respectively, 3) and $O(n \log n)$ edges (resp., $O(n \log \log n)$ edges) can be built within time linear in its size; for $k \geq 4$, they showed that 1-spanners with diameter at most $2k$ and $O(n\alpha_k(n))$ edges can be built in $O(n\alpha_k(n))$ time. The function α_k is the inverse of a certain Ackermann-style function at the $\lfloor k/2 \rfloor$ th level of the primitive recursive hierarchy, where $\alpha_0(n) = \lceil n/2 \rceil$, $\alpha_1(n) = \lceil \sqrt{n} \rceil$, $\alpha_2(n) = \lceil \log n \rceil$, $\alpha_3(n) = \lceil \log \log n \rceil$, $\alpha_4(n) = \log^* n$, $\alpha_5(n) = \lfloor \frac{1}{2} \log^* n \rfloor$, etc. Roughly speaking, for $k \geq 2$ the function α_k is close to $\lfloor \frac{k-2}{2} \rfloor$ -iterated log-star function, i.e., log with $\lfloor \frac{k-2}{2} \rfloor$ stars. Also, $\alpha_{2\alpha(n)+2}(n) \leq 4$, where $\alpha(n)$ is the one-parameter inverse Ackermann function, which is an extremely slowly growing function. (See [38] for a formal definition.) Bodlaender et al. [11] constructed 1-spanners with diameter at most k and $O(n\alpha_k(n))$ edges, but for $k \geq 4$ their construction time is rather high ($\Omega(n^2)$). Solomon [52] gave a linear-time construction with the same diameter-size tradeoff k versus $O(n\alpha_k(n))$ as [11].

Alternative constructions, by Yao [58] for line metrics and by Chazelle [14] for general tree metrics, achieve a tradeoff of m edges versus diameter $\Theta(\alpha(m, n))$, where $\alpha(m, n)$ is the two-parameter inverse-Ackermann function (defined in [38]). However, these constructions provide 1-spanners with diameter $\Gamma' \cdot k$, only for constant $\Gamma' > 30$.

1.1.1.1 $(1 + \epsilon)$ -spanners

The seminal STOC'95 of Arya et al. [6] established the ‘‘Dumbbell Theorem’’: For any d -dimensional Euclidean space, a $(1 + \epsilon, O(\frac{\log(1/\epsilon)}{\epsilon^d}))$ -tree cover can be constructed in $O(\frac{\log(1/\epsilon)}{\epsilon^d} \cdot n \log n + \frac{1}{\epsilon^{2d}} \cdot n) = O_{\epsilon,d}(n \log n)$ time. (For the definition of *tree cover*, see e.g. [32].) The consequence of the Dumbbell Theorem is that any construction of 1-spanners for tree metrics can be transformed into a construction of Euclidean $(1 + \epsilon)$ -spanners, and the running time of the transformation is $O_{\epsilon,d}(n \log n)$ (plus a linear term in the size bound of the 1-spanner construction). The construction of 1-spanners for tree metrics from [52] thus yields an $O(n \log n)$ -time construction of Euclidean $(1 + \epsilon)$ -spanners with diameter k and $O(n\alpha_k(n))$ edges. Moreover, this result of [52] generalizes for the wider family of *doubling metrics* via the recent tree cover theorem of Bartal et al. [9].

1.1.2 Lower bounds

The celebrated work of Yao [58] provided the first lower bound on 1-spanners for tree metrics, where a tradeoff of m edges versus diameter of $\Omega(\alpha(m, n))$ was proved for the uniform line metric. A stronger lower bound on 1-spanners, still for the uniform line metric, was given in [4]: diameter k versus $\Omega(n\alpha_k(n))$ edges, for any k ; as shown in [38], the lower bound of [4] implies that of [58], but the converse isn't true. These lower bounds apply only to 1-spanners.

Chan and Gupta [13] extended the lower bound of [58] to $(1 + \epsilon)$ -spanners, still for line metrics, proving a lower bound tradeoff of m edges versus diameter of $\Omega(\alpha(m, n))$. This tradeoff only provides a meaningful lower bound for sufficiently large values of diameter (above say 30). Specifically, the result of [13] can be used to show that any $(1 + \epsilon)$ -spanner for a certain line metric with diameter at most k must have $\Omega(n\alpha_{2k+6}(n))$ edges. When $k = 2$ (resp. $k = 3$), this gives $\Omega(n \log^{****} n)$ (resp. $\Omega(n \log^{*****} n)$) edges, which is far from the upper bound of $O(n \log n)$ (resp., $O(n \log \log n)$).

In SoCG'22 Le et al. [38] gave the following suboptimal lower bound tradeoff, for $(1 + \epsilon)$ -spanners of the uniform line metric: k versus $\Omega(\frac{n}{2^{6\lfloor k/2 \rfloor}} \alpha_k(n))$. While the result of [38] is tight for constant k , the following question remains open for more than three decades:

► **Question 1.1.** *Is there a lower bound of k versus $\Omega(n\alpha_k(n))$ between the diameter and the number of edges, for all k , for Euclidean $(1 + \epsilon)$ -spanners?*

1.1.2.1 Putting Question 1.1 into perspective.

Question 1.1 has been answered affirmatively by [38] for constant values of k . Recall that $\alpha_{2\alpha(n)+4}(n) \leq 4$, where $\alpha(n)$ is the one-parameter inverse Ackermann function. In other words, the gap underlying Question 1.1 holds only for $k = \omega(1), \dots, O(\alpha(n))$, which is admittedly a very small regime. The gap itself is exponential in k , which is at most exponential in $\alpha(n)$, hence it is a very small gap.

One might wonder – why is Question 1.1 of any interest? Indeed, from a quantitative perspective, $\alpha(n)$ grows asymptotically even more slowly than $\log^* n$, which, in turn, is at most 5 for $n < 2^{65536}$. Thus a gap of $\exp(\alpha(n))$ is a constant factor gap for all practical purposes. However, we argue that Question 1.1 is important from a qualitative perspective. Indeed, there are numerous breakthrough works whose “only goal” was to shave off factors that grow as slowly as inverse-Ackermann type functions. For example, for the Union-Find data structure, efforts to achieve a linear time algorithm led to a lower bound showing that inverse-Ackermann function dependence is necessary [25], matching Tarjan’s cornerstone upper bound [54]. Another prime example is in the context of the MST problem, where the inverse-Ackermann function dependence was shaved off from the upper bound of [15] to achieve a linear time algorithm by means of randomization [33] or under certain assumptions [24]; and it remains a major question whether there exists a linear time deterministic comparison-based MST algorithm. Yet another example is in the context of Davenport-Schinzel sequences, whose study involves optimizing inverse-Ackermann style functions – including the functions $\alpha(n)$ and $\alpha_k(n)$ – has led to important advances in discrete and computational Geometry. Indeed, Davenport and Schinzel [29] gave sharp bounds on sequences of order 1 and 2, namely $\lambda_1(n) = n$ and $\lambda_2(n) = 2n - 1$, and since then numerous applications of the sequences have been found, such as to geometric containment problems, computing shortest paths, and convex hulls. Achieving a tight bound for order-3 sequences spanned a long line of work [29, 22, 30, 43], and it is now understood that $\lambda_3(s) = 2n\alpha(n) + O(n\sqrt{\alpha(n)})$, i.e., the asymptotic behavior is known up to the leading constant. The case for $k \geq 4$ also spanned much work [22, 30, 49, 50, 3, 43, 45] and was settled up to leading constants in front of $\alpha(n)$ in the exponent, i.e., $\lambda_4(n) = \Theta(n2^{\alpha(n)})$, $\lambda_5(n) = \Theta(n\alpha(n)2^{\alpha(n)})$, $\lambda_6(n) = 2^{(1+o(1))\alpha^t(n)/t!}$.

We stress that in this work we are not merely shaving off an inverse-Ackermann function dependence slack from a previous upper bound (that of [38]) – we shave off such a slack *to achieve a tight bound*. This is a rare example where such a tiny slack is shaved to achieve optimality, and we believe that it is a significant evidence for the importance of our result, especially in light of our technical contribution.

1.1.2.2 A robust lower bound?

All previous lower bounds [58, 4, 13, 38] apply to very specific line metrics: either to the uniform line metric [58, 4, 38] or to one that is derived from hierarchically well-separated trees (HSTs) and is very far from being uniform [13].

A natural question is whether one can improve the longstanding construction of Euclidean $(1 + \epsilon)$ -spanners by Arya et al. [6] for “typical” point sets, which arise in real-life applications – such as *random points in low-dimensional spaces*. While random point sets are important from a practical perspective, none of the previous lower bounds [58, 4, 13, 38] precludes the existence of improved spanner constructions for such point sets.

► **Question 1.2.** *Can one improve the k versus $O(n\alpha_k(n))$ longstanding upper bound by Arya et al. [6] for random point sets in constant-dimensional Euclidean space?*

1.2 Our Contribution

1.2.1 The basic lower bound (settling Question 1.1 in the affirmative)

We prove that any $(1 + \epsilon)$ -spanner for the uniform line metric with diameter k has $\Omega(n\alpha_k(n))$ edges, for any k . We first prove the following general statement, which applies to subspaces of the uniform line metrics of any *density*.

► **Theorem 1.** *Let P be a set of p points in the interval $[0, L]$ such that every unit sub-interval $[i, i + 1]$ for integer i , $1 \leq i \leq L - 1$ contains at most 1 point of P . For any $\epsilon \in [0, 1/4]$ and integer $k \geq 1$, any $(1 + \epsilon)$ -spanner with diameter k for P contains $\Omega((p^2/L)\alpha_k(p))$ edges.*

For technical reasons we prove a more general lower bound, stated in Lemmas 12, 14, and 16, which applies to *Steiner spanners*, namely, spanners that may contain additional Steiner points. The following direct corollary of Theorem 1 improves the previous lower bound by Le et al. [38] by a factor of $2^{\Omega(k)}$, and it settles Question 1.1 in the affirmative.

► **Corollary 2** (The longstanding upper bound is tight for all k). *Let $P = \{0, 1, \dots, n - 1\}$ be the set of n points on the uniform line metric contained on interval $[0, n]$. For any $\epsilon \in [0, 1/4]$ and integer $k \geq 1$, any $(1 + \epsilon)$ -spanner with diameter k for P contains $\Omega(n\alpha_k(n))$ edges.*

1.2.2 A robust lower bound (settling Question 1.2 in the negative)

Our lower bound of Theorem 1 applies to subspaces of the uniform line metric. We first demonstrate that this lower bound can be naturally extended to obtain analogs for *constant* dimensions. Second, we show that this lower bound carries over for random point sets in spaces of constant dimension, thereby settling Question 1.2 in the negative. We note that our approach seamlessly extends to higher constant dimensions.

The constant-dimensional hypercube and grid

The proof of the following theorem is omitted from this version due to space constraints.

► **Theorem 3.** *Let P be a set of p points in the hypercube $[0, L]^d$ for a constant $d \geq 2$ and some integer $L \geq 0$ such that every unit hypercube with integer vertices in $[0, L]^d$ contains at most one point of P . For any $\epsilon \in [0, 1/4]$ and any integer $k \geq 1$, any $(1 + \epsilon)$ -spanner with diameter k for P contains $\Omega((p^d/L^d)\alpha_k(p^d))$ edges.*

Thus for $d = 2$ and $d = 3$, we get lower bounds $\Omega((p^2/L^2)\alpha_k(p^2))$ and $\Omega((p^3/L^3)\alpha_k(p^3))$.

► **Corollary 4.** *Let P be the set of n^d points on the d -dimensional grid $[0, n]^d$, for a constant $d \geq 2$. Then, for any $\epsilon \in [0, 1/4]$ and any integer $k \geq 1$, any $(1 + \epsilon)$ -spanner with diameter k for P contains $\Omega(n^d\alpha_k(n^d))$ edges.*

Random point sets in the d -dimensional hypercube

We omit the proof of the following theorem from this version due to space constraints.

► **Theorem 5.** *Let P be a set of n points sampled uniformly at random on the hypercube $[0, 1]^d$ for any constant $d \geq 1$. For any $\epsilon \in [0, 1/4]$, and any integer $k \geq 1$, any $(1 + \epsilon)$ -spanner with diameter k for P contains $\Omega(n\alpha_k(n))$ edges.*

Remark. Theorem 5 applies to $d = 1$ as well, i.e., random points on the unit interval $[0, 1]$.

1.2.3 A concave inverse-Ackermann function

Our technique for proving Theorem 1 requires a significantly deeper understanding of inverse-Ackermann style functions than used in previous works [58, 4, 13, 38]. A key technical contribution in our work is an explicit construction of continuous versions of these functions. To our knowledge, this work is the first to introduce such functions for $\alpha_k(n)$ for $k > 4$. We then show that these functions are *concave*, which allows us to apply Jensen's inequality in our inductive proof, leading to a lower bound that is not only optimal for all values of k , but is also more robust, and in particular precludes the existence of better constructions for random point sets.

► **Theorem 6.** Fix an arbitrary constant $\frac{1}{10000} \leq \Delta \leq \frac{1}{256}$. There exists a family of functions $\{f_k(x) : k \geq 2, k \in \mathbb{Z}\}$ such that each $f_k : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ is twice differentiable in $(0, +\infty)$ and:

1. For $x > 1$, $f_2(x) = \log x$; $f_3(x) = \log \log x$; and $f_k(x) = \Delta + f_k(f_{k-2}(x))$ for every $k \geq 4$.
2. For all $x \in \mathbb{R}^{\geq 1}$ and $k \geq 4$, function $x^2 f_k(x)$ is convex.
3. For all $x \in \mathbb{R}^{\geq 0}$ and $k \geq 4$, it holds that $f_k(x) \geq \frac{\Delta}{5} \alpha_k(\lceil x \rceil) - 1$.
4. For all $x \in \mathbb{R}^{\geq 0}$ and $k \geq 2$, it holds that $f_k(\lceil x \rceil) \leq \alpha_k(\lceil x \rceil)$.
5. For all $k \geq 2$, $k \in \mathbb{Z}$ and $x \geq 200$, $x \in \mathbb{R}$, it holds that $2 \left\lfloor \frac{x}{f_k(x)} \right\rfloor f_k \left(\frac{2 \lfloor x/f_k(x) \rfloor}{4} \right) \geq x/2$.

Items 3 and 4 of Theorem 6 imply that $f_k(n) = \Theta(\alpha_k(n))$. Item 2 is a key property of our function $f_k(x)$, which does not hold for its discrete counterpart $\alpha_k(n)$.

2 One-dimensional instances

This section is dedicated to proving Theorem 1. The proof is by double induction on the number of points and the diameter of the spanner. There are two base cases in the proof: $k = 2$ and $k = 3$ presented in Section 2.2 and Section 2.3, respectively. The proof for $k \geq 4$ is given in Section 2.4. Together, they imply Theorem 1. We choose $\Delta = 1/256$.

For a constant d and given set of points P on the d -dimensional hypercube $[0, L]^d$, we require that every unit hypercube with integer vertices in $[0, L]^d$ contains at most one point in P . We call the condition *unit interval condition*.

2.1 Classification of cross edges

Given a point set P contained on an interval $[0, n]$ and given an $\epsilon \in [0, 1/4]$, let H be any $(1 + \epsilon)$ -spanner for P . Consider Algorithm 1 with parameter $\ell = 0$ being the *recursion level*, k being the diameter, and I being the interval containing P . This algorithm is used to classify the edges of H only. It divides I into a smaller set of b subintervals and defines a set of separators, which are the endpoints of the subintervals excluding the two endpoints of I . A cross edge of the interval I at level ℓ is an edge (1) needed to preserve the distance between two points in P and (2) crossing a separator.

Next we study properties of cross edges and classify them.

► **Lemma 7.** Let e be a cross edge of some interval $I = [c, d]$ and let $L := |d - c|$ denote the interval length. Then, both endpoints of e are within $[c - L/4, d + L/4]$.

Proof. Suppose toward the contradiction that there is an edge containing an endpoint outside of $[c - L/4, d + L/4]$. Without loss of generality we take the case where the right endpoint of e has coordinate larger than $d + L/4$. Let $x < y$ be two points in I for which e is on their

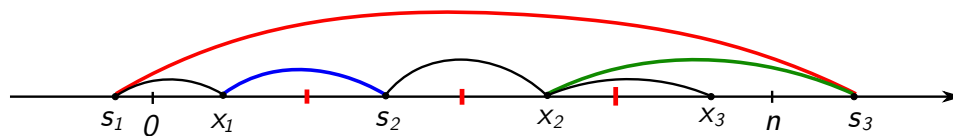
■ **Algorithm 1** Procedure describing the terms used in the proof. It is initially invoked with a given set of p points P on interval I , and $\ell = 0$. Here, H is a $(1 + \epsilon)$ -spanner for P .

```

procedure CROSS_EDGES( $P, I = [c, d], k, H, \ell$ )
  if ( $k \leq 3$  and  $p \leq 1$ ) or ( $k \geq 4$  and  $f_k(p) < 1$ ) then return
  Let  $b \leftarrow 2$  if  $k = 2$ ,  $b \leftarrow \lceil \sqrt{p} \rceil$  if  $k = 3$ , and  $b \leftarrow 2 \cdot \lfloor p/f_{k-2}(p) \rfloor$  otherwise.
   $M \leftarrow (d - c)/b$  ▷ dividing  $I$  into  $b$  subintervals
  for  $1 \leq j \leq b - 1$  do
     $I_j \leftarrow [c + (j - 1)M, c + jM]$ 
     $P_j \leftarrow P \cap [c + (j - 1)M, c + jM]$ 
   $P_b \leftarrow P \cap [c + (b - 1)M, c + bM]$ 
  Let  $\{c + jM \mid 1 \leq j \leq b - 1\}$  be the set of separators of  $I$ .
  A cross edge of interval  $I$  is every edge  $e = (x, y)$  of  $H$  such that: (i)  $e$  is on some  $(1 + \epsilon)$ -spanner path between two points in  $P$  and (ii) there exists a separator  $s$  such that  $x \leq s \leq y$ .
  for  $1 \leq j \leq b$  do CROSS_EDGES( $P_j, I_j, k, H, \ell + 1$ )
  
```

$(1 + \epsilon)$ spanner path, say $\pi_{x,y}$, in H . Since $\pi_{x,y}$ is a $(1 + \epsilon)$ -spanner path, its length $|\pi_{x,y}|$ must be at most $(1 + \epsilon)|y - x| \leq 5|y - x|/4 \leq |y - x| + L/4$. However, the length of $\pi_{x,y}$ is strictly greater than $|x - y| + L/4$ since the right endpoint of e is larger than $d + L/4$, a contradiction. ◀

We classify the cross edges as follows. We call a cross edge of some interval *interior* if it contains both endpoints inside of the interval. If both of its endpoints are outside of the interval, we call it *exterior*. Otherwise, we call it *mixed*. See Figure 1 for an illustration.



■ **Figure 1** The separators are marked by short red lines. Here $P = \{x_1, x_2, x_3\}$. The spanner could use Steiner points which are points not in P ; they are s_1, s_2, s_3 in this figure. The red edge (s_1, s_3) is an exterior cross edge of $[0, n]$; the blue edge (x_1, s_2) is an interior cross edge of $[0, n]$; and the green edge (x_2, s_3) is a mixed cross edge of $[0, n]$. Edge (s_1, x_1) is not a mixed edge since it does not cross any separator.

► **Lemma 8.** *Let e be an interior cross edge of some interval. Then, it cannot be an interior cross edge of any other interval.*

Proof. Let ℓ be the level at which e is an interior cross edge of some interval $I = [c, d]$. By definition, e cannot be an interior cross edge of any other interval at level ℓ , since the intervals at the same level are disjoint. Since the intervals of levels lower than ℓ contain no separators inside $[c, d]$, e cannot be a cross edge at these levels. Finally, after level ℓ , I is split at the separators into smaller intervals, and hence e cannot have two endpoints in the same interval at levels higher than ℓ . ◀

► **Lemma 9.** *Let e be an exterior cross edge of some interval. Then, it cannot be an exterior cross edge of any other interval.*

Proof. Suppose that $e = (u, v)$ is an exterior cross edge of more than one interval. Among such intervals, let $[c, d]$ be of the highest level, say ℓ . We have that $u < c$ and $d < v$ since e is exterior. Let $L = |d - c|$. The length of intervals at levels lower than ℓ are at least $2L$. From Lemma 7, we know that $c - L/4 \leq u$ and $v \leq d + L/2$, so the length of e is at most $3L/2$. This means that e cannot be an exterior edge at levels lower than ℓ . ◀

► **Lemma 10.** *Let e be a mixed cross edge of some interval. Then, it can be a mixed cross edge for at most one other interval, an exterior cross edge for at most one other interval and an interior cross edge for at most one other interval.*

Proof. Let ℓ be the level at which $e = (u, v)$ is a mixed cross edge of some interval $I = [c, d]$ of length $L := |d - c|$. Without loss of generality, we assume that $u \in [c, d]$ and $v \geq d$. By Lemma 7, $v < d + L/4$. Let $I' = [c', d']$ be another interval such that $I' \neq I$ and e is a cross edge of I' . We consider three cases.

If the level of I' is strictly smaller than ℓ . If d is not a separator of I' , then by definition e cannot be a cross edge of I' . If d is a separator of I' , then $d' \geq d + L$. On the other hand, $v < d + L/4$, so $e = (u, v)$ must be an interior cross edge of I' . By Lemma 8, it cannot be an interior cross edge of any other interval.

If the level of I' is exactly ℓ . I. Since $v < d + L/4$, the only case where e is a cross edge of I' is that d is the left endpoint of I' and e is a mixed cross edge of I' . Thus, e could not be a mixed cross edge of any other interval at level ℓ .

If the level of I' is strictly larger than ℓ . Then the length of I' is at most L/b . Since u is on the left of at least one separator, say s , of I and $v > d$, the distance between s and d is at least L/b . It follows that the length of e is at least L/b . Hence, the only possible way for e to be a cross edge of I' is that it is an exterior cross edge. By Lemma 9, e' will not be an exterior cross edge of any other interval. ◀

► **Corollary 11.** *Every cross edge considered in the process above is counted at most 4 times.*

In the sequel, we will be proving the lower bound on the number of cross edges. We say that a point of P in an interval I is *global* if it is incident on at least one cross edge of I . Otherwise, we say that it is *non-global*.

2.2 Hop-diameter 2

In this section, we show one of the two base cases of our inductive proof: a lower bound for diameter $k = 2$.

► **Lemma 12.** *Let P be a set of $p \geq 2$ points in the interval $[0, L]$ satisfying the unit interval condition. For any $\epsilon \in [0, 1/4]$, any Steiner $(1 + \epsilon)$ -spanner for P with diameter 2 contains at least $T_2(p, L) \geq \frac{p^2 \log p}{16L}$ edges.*

Proof. Our proof is by induction on the number of points in P . Let H be any $(1 + \epsilon)$ -spanner for P with diameter 2. We split the interval $[0, L]$ into two disjoint intervals $[0, L/2]$ and $[L/2, L]$. Let the number of points in the intervals be $p_1 := |P \cap [0, L/2]|$ and $p_2 := |P \cap (L/2, L]|$. We claim that the number of edges of H can be lower bounded by $T_2(p, L)$ which satisfies:

$$T_2(p, L) \geq T_2(p_1, L/2) + T_2(p_2, L/2) + \min(p_1, p_2)/4 \quad (1)$$

The base cases are $T_2(0, L_0) = T_2(1, L_0) = 0$, for any $L_0 > 0$. The terms $T_2(p_1, L/2)$ (resp., $T_2(p_2, L/2)$) come from the cross edges contributed by the intervals in $[0, L/2]$ (resp., $[L/2, L]$)

and their recursive divisions in Algorithm 1. We will show in Claim 13 that the number of cross edges of $[0, L]$ is at least $\min(p_1, p_2)$. By Corollary 11, each cross edge is counted at most 4 times. Thus, we use $\min(p_1, p_2)/4$ in Equation (1). This implies that the number of edges of H is bounded by $T_2(p, L)$.

▷ **Claim 13.** H contains at least $\min(p_1, p_2)$ cross edges of the interval $[0, L]$.

Proof. Without loss of generality, assume $p_1 \leq p_2$. For contradiction, assume that the number of cross edges is less than p_1 . This means that there is a non-global point a in $[0, L/2]$. (Recall that we call a point non-global if it is not incident on any cross edge of the interval $[0, L]$.) A path from a to any point b in $P \cap [L/2, L]$ is of the form (a, a_b, b) , where a_b is a point on the left of $L/2$. Then (a_b, b) is a cross edge by definition. That is, for each point in $P \cap [L/2, L]$, there is a corresponding cross edge in the path to a . Thus, $[0, L]$ contains $p_2 \geq p_1$ different cross edges, which is a contradiction. ◀

We now solve the recurrence in Equation (1). We prove by induction that $T_2(p, L) \geq \frac{p^2 \log p}{16L}$. Note that $L \geq p$ by the unit interval condition in Lemma 12. Assume without loss of generality that $p_1 \leq p_2$. First, we assume that $p_1 \geq p/4$.

$$\begin{aligned} T_2(p, L) &\geq T_2(p_1, L/2) + T_2(p_2, L/2) + \frac{p_1}{4} \geq \frac{p_1^2 \log p_1}{8L} + \frac{p_2^2 \log p_2}{8L} + \frac{p_1}{4} \geq \frac{p^2 \log(p/2)}{16L} + \frac{p_1}{4} \\ &= \frac{p^2(\log(p) - 1) + 4Lp_1}{16L} \geq \frac{p^2(\log(p) - 1) + 4pp_1}{16L} \geq \frac{p^2 \log p}{16L} \text{ (since } p_1 \geq p/4) \end{aligned}$$

The second inequality follows by induction hypothesis, third by Jensen’s inequality, fourth by the unit interval condition, and the fifth since $p_1 \geq p/4$. When $p_1 < p/4$, we have the following.

$$T_2(p, L) \geq T_2(p_1, L/2) + T_2(p_2, L/2) + \frac{p_1}{4} \geq T_2(p_2, L/2) \geq \frac{(3p/4)^2 \log(3p/4)}{8L} \geq \frac{p^2 \log p}{16L}$$

The penultimate inequality follows by using $p_2 \geq 3p/4$ and the induction hypothesis, whereas the last one holds for all $p \geq 14$. When $2 \leq p \leq 13$, we use $T_2(p_1, L/2) + T_2(p_2, L/2) + p_1/4 \geq \frac{p_1^2 \log p_1}{8L} + \frac{p_2^2 \log p_2}{8L} + \frac{p_1}{4} \geq \frac{p^2 \log p}{16L}$, where the last inequality can be manually verified. The lemma now follows. ◀

2.3 Hop-diameter 3

In this section, we show the remaining base case of our inductive proof: a lower bound for diameter $k = 3$.

► **Lemma 14.** *Let P be a set of $p \geq 2$ points in the interval $[0, L]$ satisfying the unit interval condition. For any $\epsilon \in [0, 1/4]$, any Steiner $(1 + \epsilon)$ -spanner for P with diameter 3 contains at least $T_3(p, L) \geq \frac{p^2 \log \log p}{800L}$ edges.*

Proof. Let H be any $(1 + \epsilon)$ -spanner for P with diameter 3. We split the interval $[0, L]$ into $b := \lceil \sqrt{p} \rceil$ disjoint intervals of length L/b : $[0, L/b], [L/b, 2(L/b)], \dots, [(b - 1)(L/b), L]$. Let $P_i = P \cap [(i - 1)(L/b), i(L/b))$ for $1 \leq i < b$ and $P_b = P \cap [L - L/b, L]$. In other words, we divide the interval as in Algorithm 1. Let the number of points in the i -th interval be denoted by $p_i := |P_i|$. We claim that the number of edges of H can be lower bounded by $T_3(p, L)$ which satisfies:

$$T_3(p, L) \geq \sum_{i=1}^b T_3(p_i, L/b) + |E_C|/4 \tag{2}$$

47:10 Sparse Euclidean Spanners with Optimal Diameter

Here E_C denotes the set of cross edges for the interval $[0, L]$ and the term $T_3(p_i, l_i)$, where $1 \leq i \leq b$, is the lower bound on the number of cross edges of H at higher levels restricted to preserving distances in P_i . By Corollary 11, each cross edge is counted at most 4 times. Thus, we use $|E_C|/4$ in Equation (2). Thus, $|E(H)| \geq T_3(p, L)$. The base cases are $T_3(0, L_0) = T_3(1, L_0) = 0$, for any $L_0 > 0$.

We now inductively show that $T_3(p, L) \geq \frac{p^2 \log \log p}{800L}$. Suppose first that there is a collection of $c \leq \sqrt{p}/2$ intervals which in total contain at least $9p/10$ points. Without loss of generality, assume that these are the first c intervals; that is, $\sum_{i=1}^c b_i = 9p/10$. In this case, we show that the inequality holds even without the contribution of the cross edges.

$$\begin{aligned} \sum_{i=1}^c T_3(p_i, L/b) &\geq \sum_{i=1}^c \frac{p_i^2 \log \log p_i}{800L/b} \geq c \cdot \frac{\left(\frac{9p}{10c}\right)^2 \log \log \left(\frac{9p}{10c}\right)}{800L/b} \\ &\geq \frac{81}{50} \cdot \frac{p^2 \log \log \left(\frac{9\sqrt{p}}{5}\right)}{800L} \geq \frac{p^2 \log \log p}{800L} \end{aligned}$$

The first inequality follows from the induction hypothesis, second by Jensen's inequality, and third using $b \geq \sqrt{p}$ and $c \leq \sqrt{p}/2$. We next bound the number of cross edges in the complementary case.

▷ **Claim 15.** Assume that there is no collection of $c \leq \sqrt{p}/2$ intervals that in total contain at least $9p/10$ points. Then, $|E_c| \geq p/100$.

Proof. Suppose first there are at least $p/10$ global points. The number of cross edges they contribute is at least $p/20$, since each edge can be counted at most twice. In the complementary regime, there are at least $9p/10$ non-global points. By the assumption of the claim, we know that they are contained in at least $\sqrt{p}/2$ intervals. Consider two non-global points x and y contained in two different intervals, X and Y , respectively. Since x and y are non-global, i.e., they are not incident on any cross edge, every 3-hop path between x and y must be of the form $\langle x, x', y', y \rangle$, where $x' \in X$ and $y' \in Y$. We conclude that every pair of different intervals containing non-global points induces a different cross edge. Hence, the number of cross edges can be lower bounded by $\binom{\sqrt{p}/2}{2} \geq \frac{p}{100}$ for $p \geq 5$. When $2 \leq p \leq 4$, there is at least one cross edge, and the bound holds as well. ◀

We now solve Equation (2) by induction. By Claim 15, we have:

$$\begin{aligned} T_3(p, L) &\geq \sum_{i=1}^b T_3(p_i, L/b) + \frac{p}{400} \geq \sum_{i=1}^b \frac{p_i^2 \log \log p_i}{800L/b} + \frac{p}{400} \geq b \cdot \frac{(p/b)^2 \log \log (p/b)}{800L/b} + \frac{p}{400} \\ &= \frac{p^2 \log \log (p/b)}{800L} + \frac{p}{400} = \frac{p^2 \log \log (p/b) + 2pL}{800L} \geq \frac{p^2 \log \log p}{800L} \end{aligned}$$

The second inequality follows from the induction hypothesis, third by Jensen's inequality, and the last from the unit interval condition and the choice $b = \lceil \sqrt{p} \rceil$. The lemma now follows. ◀

2.4 Hop-diameter $k \geq 4$

In this section, we show a lower bound for $k \geq 4$, concluding the proof of Theorem 1. Our proof will use function $f_k(x)$ in Theorem 6 with $\Delta = 1/256$. In particular, we will show the lower bound $\Omega\left(\frac{p^2 f_k(p)}{L}\right)$ on the number of edges. Since $f_k(p) = \Omega(\alpha_k(b))$ by Item 3 of Theorem 6, the number of edges of the spanner is $\Omega\left(\frac{p^2 \alpha_k(p)}{L}\right)$ as claimed in Theorem 1.

► **Lemma 16.** *Let P be a set of $p \geq 2$ points in the interval $[0, L]$ satisfying the unit interval condition. For any $\epsilon \in [0, 1/4]$, any Steiner $(1 + \epsilon)$ -spanner for P with hop-diameter $k \geq 2$ contains at least $T_k(p, n) \geq \frac{p^2 f_k(p)}{800L}$ edges.*

Proof. The base cases $k = 2$ and $k = 3$ follow from the definition of $f_2(x) = \log x$ and $f_3(x) = \log \log x$ and Lemmas 12 and 14. The base case for p happens when $f_k(p) < 1$. Here, we use the fact that any spanner on p points must have at least $p - 1$ edges and $p - 1 \geq \frac{p^2 f_k(p)}{800L}$ so the claim follows.

Let H be any $(1 + \epsilon)$ -spanner for P with hop-diameter k . We split the interval $[0, L]$ into $b := 2 \cdot \lfloor p/f_{k-2}(p) \rfloor$ disjoint intervals of length L/b : $I_1 = [0, L/b), I_2 = [L/b, 2(L/b)), \dots, I_{b-1} = [(b-2)(L/b), (b-1)(L/b)), I_b = [(b-1)(L/b), L]$. Let the number of points in the i -th interval be denoted by $p_i := |P \cap I_i|$. By the same proof of Lemma 14, the number of edges of H can be lower bounded by $T_k(p, n)$ which satisfies:

$$T_k(p, L) \geq \sum_{i=1}^b T_k(p_i, L/b) + |E_C|/4 \tag{3}$$

Here E_C denotes the set of cross edges for the interval $[0, L]$ and the term $T_k(p_i, L/b)$, where $1 \leq i \leq b$, come from the cross edges contributed by the i -th interval and its recursive subdivisions.

We now inductively show that $T_k(p, L) \geq \frac{p^2 f_k(p)}{800L}$ for $k \geq 4$. Suppose first that there is a collection of $c \leq b/4$ intervals that in total contain at least $3p/4$ points. Then the inequality holds even without considering $|E_C|$. Recall that by Item 2 in Theorem 6, $x^2 f_k(x)$ is convex and hence we can apply the Jensen's inequality.

$$\begin{aligned} T_k(p, L) &\geq \sum_{i=1}^c T_k(p_i, L/b) \geq \sum_{i=1}^c \frac{p_i^2 f_k(p_i)}{800L/b} \\ &\geq c \cdot \frac{\left(\frac{3p}{4c}\right)^2 f_k\left(\frac{3p}{4c}\right)}{800L/b} && \text{(Jensen's inequality)} \\ &\geq \frac{9}{4} \cdot \frac{p^2 f_k\left(\frac{3p}{b}\right)}{800L} \geq \frac{9}{4} \cdot \frac{p^2 f_k(f_{k-2}(p))}{800L} && \text{(using } c \leq b/4 \text{ and } b := 2 \cdot \lfloor p/f_{k-2}(p) \rfloor) \\ &= \frac{9}{4} \cdot \frac{p^2 (f_k(p) - \Delta)}{800L} && \text{(by Item 1 in Theorem 6)} \\ &\geq \frac{p^2 f_k(p)}{800L} && \text{using that } f_k(p) \geq 1 \end{aligned}$$

Now we consider the complementary case where there is no collection of $c \leq b/4$ intervals that in total contain at least $3p/4$ points. For this case, we need to take the number of cross edges into account.

► **Claim 17.** Assume that there is no collection of $c \leq b/4$ intervals that in total contain at least $3p/4$ points. Then, $|E_C| \geq p/25600$.

Proof. If there is at least $p/4$ global points, then we have at least $p/8$ cross edges. In the complementary regime, there are at least $3p/4$ non-global points. By assumption, they are contained in at least $b/4$ non-global blocks. From each interval that contains non-global points we take exactly one non-global point and let the resulting set of points be denoted P' . We use the induction hypothesis with $k - 2$ on P' . Note that $|P'| \geq b/4$. The following observation allows us to use the scaled version of the induction hypothesis.

47:12 Sparse Euclidean Spanners with Optimal Diameter

► **Observation 18.** *Suppose that a set of points P' on interval $[0, L]$ satisfies that when we divide $[0, L]$ into consecutive intervals of length M , every such interval contains at most one point from P' and H' be any $(1 + \epsilon)$ spanner of P' with hop-diameter k . Let Q' be a set of points in P' scaled down by a factor of L . Such a set of points is contained on interval $[0, L/M]$ and it satisfies the unit interval condition. Let H'' be the scaled version of H' . Then, H'' is a $(1 + \epsilon)$ -spanner for Q' with hop-diameter k .*

We proceed to lower bound the number of cross edges, using the observation.

$$T_{k-2}(|P'|, b) \geq T_{k-2}\left(\frac{b}{4}, b\right) \geq \frac{\frac{b^2}{16} f_{k-2}\left(\frac{b}{4}\right)}{800b} = \frac{2\lfloor p/f_{k-2}(p) \rfloor f_{k-2}\left(\frac{2\lfloor p/f_{k-2}(p) \rfloor}{4}\right)}{12800} \geq \frac{p}{25600}$$

The second inequality follows by the induction hypothesis for $k - 2$, and the last by Item 5 in Theorem 6. This concludes the proof of Claim 17. ◀

We now solve Equation (3) by induction. Recall that we choose $\Delta = 1/256$. By Claim 17, we have:

$$\begin{aligned} T_k(p, L) &\geq \sum_{i=1}^b T_k(p_i, L/b) + \frac{p}{102400} \\ &\geq \sum_{i=1}^b \frac{p_i^2 f_k(p_i)}{800L/b} + \frac{p}{102400} && \text{(induction hypothesis)} \\ &\geq b \cdot \frac{\left(\frac{p}{b}\right)^2 f_k\left(\frac{p}{b}\right)}{800L/b} + \frac{p}{102400} && \text{(Jensen's inequality)} \\ &= \frac{p^2 f_k\left(\frac{p}{2\lfloor p/f_{k-2}(p) \rfloor}\right)}{800L} + \frac{p}{102400} && \text{(replacing } b := 2\lfloor p/f_{k-2}(p) \rfloor) \\ &\geq \frac{p^2 (f_k(p) - 3\Delta)}{800L} + \frac{p}{102400} \\ &\geq \frac{p^2 f_k(p)}{800L} && \text{(using } p \leq L \text{ and } \Delta = 1/256) \end{aligned}$$

The lemma now follows. ◀

3 Concave Ackermann-type functions

In this section, we introduce the concave inverse-Ackermann function $f_k(x)$. We omit the details from this extended abstract due to space constraints. We fix a constant $\Delta < 1/256$.

► **Definition 19** ($f_k(n)$ for even k). *For all $x \in \mathbb{R}^{\geq 0}$ and even $k \geq 2$, we let $f_k(x)$ be:*

$$\begin{aligned} f_2(x) &= \log x \\ f_k(x) &= a_k x^3 + b_k x^2 + c_k x - \Delta && \text{for } 0 \leq x \leq 1, k \geq 4 \\ f_k(x) &= \Delta + f_k(f_{k-2}(x)) && \text{for } x > 1, k \geq 4 \end{aligned}$$

Constants a_k , b_k , and c_k are chosen so that they satisfy the following relations.

$$a_k + b_k + c_k = \Delta \quad \forall k \geq 4 \tag{4}$$

$$3a_4 + 2b_4 + c_4 = \frac{c_4}{\ln 2} \tag{5}$$

$$6a_4 + 2b_4 = \frac{2b_4 - c_4 \ln 2}{\ln^2 2} \tag{6}$$

$$3a_k + 2b_k + c_k = c_k \cdot (3a_{k-2} + 2b_{k-2} + c_{k-2}) \tag{7}$$

$$6a_k + 2b_k = 2b_k \cdot (3a_{k-2} + 2b_{k-2} + c_{k-2})^2 + c_k \cdot (6a_{k-2} + 2b_{k-2}) \tag{8}$$

In this section, we solve the recurrence in Definition 19 for even k by giving estimates on the values of a_k, b_k and c_k . We will use these estimates in the proof of Theorem 6, which is omitted from this extended abstract due to space constraints.

For $k = 4$, by solving a linear system of equations defined by Equations (4), (5), and (8) we obtain the following estimates.

► **Lemma 20.** a_4, b_4 and c_4 satisfy the following equation:

$$\begin{aligned} -0.0819\Delta &\leq a_4 \leq -0.0818\Delta \\ 0.2966\Delta &\leq b_4 \leq 0.2967\Delta \\ 0.7852\Delta &\leq c_4 \leq 0.7853\Delta \end{aligned} \tag{9}$$

In estimating the values of a_k, b_k and c_k , we will use the following sequences:

$$\begin{aligned} \lambda_4 &= 1.1328\Delta, & \lambda_k &= \frac{3\Delta\lambda_{k-2}}{1 + 4\lambda_{k-2}} \\ r_4 &= 11.0439, & r_k &= \frac{\Lambda_{k-2}^3 + \Lambda_{k-2}}{2\Lambda_{k-2}^3 - 2\Lambda_{k-2}^2 + \frac{\Lambda_{k-2}}{r_{k-2}}}, \text{ where } \Lambda_k = 0.3777 \cdot (3\Delta)^{(k-2)/2} \end{aligned} \tag{10}$$

► **Lemma 21.** $\lambda_k \geq 0.3265(3\Delta)^{\frac{k-2}{2}}$ and $r_k < 25$ for all $k \geq 4$.

Proof. Solving the recurrence we get

$$\begin{aligned} \lambda_k &= \frac{236 \cdot (1 - 3\Delta) \cdot 3^{(k-2)/2}}{(625 + 957\Delta) \left(\frac{1}{\Delta}\right)^{(k-2)/2} - 944 \cdot 3^{(k-2)/2}} \\ &\geq \frac{236 \cdot (1 - 3\Delta) \cdot (3\Delta)^{(k-2)/2}}{625 + 957\Delta} \geq 0.3265 \cdot (3\Delta)^{\frac{k-2}{2}}. \end{aligned}$$

The last inequality holds whenever $\Delta \leq 1/32$.

We use induction to show that $r_k < 25$; the base case holds by definition of r_4 . Observe that $0 \leq \Lambda_k \leq \Lambda_4 \leq 0.3777 \cdot (3\Delta) \leq 0.3777 \cdot \frac{3}{256}$. By induction, $r_{k-2} < 25$. Thus, we have $r_k = \frac{\Lambda_{k-2}^3 + \Lambda_{k-2}}{2\Lambda_{k-2}^3 - 2\Lambda_{k-2}^2 + \frac{\Lambda_{k-2}}{r_{k-2}}} \leq r_{k-2} + 3000\Lambda_{k-2}$, where the last inequality follows since the left-hand side grows with Λ_{k-2} for all $0 \leq \Lambda_k \leq 0.3777 \cdot \frac{3}{256}$, when $11.0439 \leq r_{k-2} < 25$. It follows that: $r_k \leq r_4 + 3000 \sum_{i=1}^{(k-2)/2} \Lambda_k \leq r_4 + 3000 \sum_{i=1}^{\infty} \Lambda_k \leq 11.0439 + 3000 \cdot 0.3777 \cdot \frac{3\Delta}{1-3\Delta} < 25$, as desired. ◀

► **Lemma 22.** Let $X_k = 2a_k + b_k + \Delta$ and $Y_k = 6a_k + 2b_k$. Then

$$0.3265 \cdot (3\Delta)^{(k-2)/2} \leq \lambda_k \leq X_k \leq 0.3777 \cdot (3\Delta)^{(k-2)/2} \tag{11}$$

$$11.041 \leq \frac{X_k}{Y_k} \leq r_k < 25 \tag{12}$$

$$\Delta - X_k \leq a_k \leq \Delta - X_k + \frac{X_k}{22} \tag{13}$$

$$-3\Delta + 3X_k - \frac{X_k}{11} \leq b_k \leq -3\Delta + 3X_k \tag{14}$$

$$3\Delta - 2X_k \leq c_k \leq 3\Delta - 2X_k + \frac{X_k}{22} \tag{15}$$

Proof. Observe by Equation (4) that $X_k = 2a_k + b_k + \Delta = 3a_k + 2b_k + c_k$ and that $c_k = \frac{Y_k}{2} - 2X_k + 3\Delta$. Thus, the system from Definition 19 for $k \geq 6$ can be written as follows.

$$\begin{aligned} X_k &= X_{k-2} \cdot \left(\frac{Y_k}{2} - 2X_k + 3\Delta \right) \\ Y_k &= X_{k-2}^2 \cdot (6X_k - 2Y_k - 6\Delta) + Y_{k-2} \cdot \left(\frac{Y_k}{2} - 2X_k + 3\Delta \right) \end{aligned}$$

Solving the above system of equations for X_k and Y_k , we get:

$$\begin{aligned} X_k &= \frac{\begin{vmatrix} 6X_{k-2}\Delta & -X_{k-2} \\ (6Y_{k-2} - 12X_{k-2}^2)\Delta & (4X_{k-2}^2 + 2 - Y_{k-2}) \end{vmatrix}}{\begin{vmatrix} (4X_{k-2} + 2) & -X_{k-2} \\ (4Y_{k-2} - 12X_{k-2}^2) & (4X_{k-2}^2 + 2 - Y_{k-2}) \end{vmatrix}} = \frac{6\Delta(X_{k-2}^3 + X_{k-2})}{2X_{k-2}^3 + 4X_{k-2}^2 + 4X_{k-2} - Y_{k-2} + 2} \\ Y_k &= \frac{\begin{vmatrix} (4X_{k-2} + 2) & 6X_{k-2}\Delta \\ (4Y_{k-2} - 12X_{k-2}^2) & (6Y_{k-2} - 12X_{k-2}^2)\Delta \end{vmatrix}}{\begin{vmatrix} (4X_{k-2} + 2) & -X_{k-2} \\ (4Y_{k-2} - 12X_{k-2}^2) & (4X_{k-2}^2 + 2 - Y_{k-2}) \end{vmatrix}} = \frac{6\Delta(2X_{k-2}^3 - 2X_{k-2}^2 + Y_{k-2})}{2X_{k-2}^3 + 4X_{k-2}^2 + 4X_{k-2} - Y_{k-2} + 2} \end{aligned} \quad (16)$$

For the base case, $X_4 = 2a_4 + b_4 + \Delta$ and $Y_4 = 6a_4 + 2b_4$. By Lemma 20, we have:

$$1.1328\Delta \leq X_4 \leq 1.1331\Delta \quad \text{and} \quad 0.1018\Delta \leq Y_4 \leq 0.1026\Delta \quad (17)$$

Next, we show both Equation (11) and Equation (12) by induction; the base case ($k = 4$) holds by Equation (17). By Equation (16), we have: $X_k \leq \frac{6\Delta(X_{k-2}^3 + X_{k-2})}{2X_{k-2}^3 + 4X_{k-2}^2 + 4X_{k-2} - \frac{X_{k-2}}{11.041} + 2} \leq 3\Delta X_{k-2} \leq 3\Delta \cdot 0.3777 \cdot (3\Delta)^{(k-4)/2} = 0.3777 \cdot (3\Delta)^{(k-2)/2}$. The lower bound on X_k follows also by induction: $X_k = \frac{6\Delta(X_{k-2}^3 + X_{k-2})}{2X_{k-2}^3 + 4X_{k-2}^2 + 4X_{k-2} - Y_{k-2} + 2} \geq \frac{3\Delta X_{k-2}}{1 + 4X_{k-2}} \geq \frac{3\Delta\lambda_{k-2}}{1 + 4\lambda_{k-2}} = \lambda_k$, by Equation (10). For the lower bound on $\frac{X_k}{Y_k}$, by Equation (16), we have: $\frac{X_k}{Y_k} = \frac{X_{k-2}^3 + X_{k-2}}{2X_{k-2}^3 - 2X_{k-2}^2 + Y_{k-2}} \geq \frac{X_{k-2}^3 + X_{k-2}}{2X_{k-2}^3 - 2X_{k-2}^2 + \frac{X_{k-2}}{11.041}} \geq 11.041$, where the last inequality holds since $X_{k-2} \leq 1.1331\Delta \leq \frac{1.1331}{256}$. Finally, we show an upper bound on $\frac{X_k}{Y_k} = \frac{X_{k-2}^3 + X_{k-2}}{2X_{k-2}^3 - 2X_{k-2}^2 + Y_{k-2}} \leq \frac{X_{k-2}^3 + X_{k-2}}{2X_{k-2}^3 - 2X_{k-2}^2 + \frac{X_{k-2}}{r_{k-2}}} \leq \frac{\Lambda_{k-2}^3 + \Lambda_{k-2}}{2\Lambda_{k-2}^3 - 2\Lambda_{k-2}^2 + \frac{\Lambda_{k-2}}{r_{k-2}}} = r_k < 25$, by Lemma 21. This concludes the inductive proof of Equation (11) and Equation (12). For Equations (13)–(15), we express a_k , b_k , and c_k in terms of X_k and Y_k as follows: $a_k = \Delta + \frac{Y_k}{2} - X_k$, $b_k = -3\Delta + 3X_k - Y_k$, and $c_k = 3\Delta + \frac{Y_k}{2} - 2X_k$. Eq. (13)–(15) follow. \blacktriangleleft

References

- 1 I. Abraham and D. Malkhi. Compact routing on Euclidean metrics. In *Proc. of 23rd PODC*, pages 141–149, 2004.
- 2 P. K. Agarwal, Y. Wang, and P. Yin. Lower bound for sparse Euclidean spanners. In *Proc. of 16th SODA*, pages 670–671, 2005.
- 3 Pankaj K. Agarwal, Micha Sharir, and Peter W. Shor. Sharp upper and lower bounds on the length of general davenport-schinzel sequences. *J. Comb. Theory, Ser. A*, 52(2):228–274, 1989.
- 4 N. Alon and B. Schieber. Optimal preprocessing for answering on-line product queries. *Manuscript*, 1987.

- 5 I. Althöfer, G. Das, D. P. Dobkin, D. Joseph, and J. Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993.
- 6 S. Arya, G. Das, D. M. Mount, J. S. Salowe, and M. H. M. Smid. Euclidean spanners: short, thin, and lanky. In *Proc. of 27th STOC*, pages 489–498, 1995.
- 7 S. Arya, D. M. Mount, and M. H. M. Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *Proc. of 35th FOCS*, pages 703–712, 1994.
- 8 S. Arya and M. H. M. Smid. Efficient construction of a bounded degree spanner with low weight. *Algorithmica*, 17(1):33–54, 1997.
- 9 Yair Bartal, Nova Fandina, and Ofer Neiman. Covering metric spaces by few trees. In *ICALP*, volume 132 of *LIPICs*, pages 20:1–20:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.
- 10 Arnab Bhattacharyya, Elena Grigorescu, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. Transitive-closure spanners. *SIAM J. Comput.*, 41(6):1380–1425, 2012.
- 11 H. L. Bodlaender, G. Tel, and N. Santoro. Trade-offs in non-reversing diameter. *Nord. J. Comput.*, 1(1):111–134, 1994.
- 12 P. B. Callahan and S. R. Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *Proc. of 4th SODA*, pages 291–300, 1993.
- 13 H. T.-H. Chan and A. Gupta. Small hop-diameter sparse spanners for doubling metrics. In *Proc. of 17th SODA*, pages 70–78, 2006.
- 14 B. Chazelle. Computing on a free tree via complexity-preserving mappings. *Algorithmica*, 2:337–361, 1987.
- 15 B. Chazelle. A minimum spanning tree algorithm with inverse-ackermann type complexity. *Journal of the ACM*, 47(6):1028–1047, 2000. doi:10.1145/355541.355562.
- 16 B. Chazelle and B. Rosenberg. The complexity of computing partial sums off-line. *Int. J. Comput. Geom. Appl.*, 1:33–45, 1991.
- 17 L. P. Chew. There is a planar graph almost as good as the complete graph. In *Proc. of 2nd SOCG*, pages 169–177, 1986.
- 18 K. Clarkson. Approximation algorithms for shortest path motion planning. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 56–65, 1987.
- 19 G. Das and G. Narasimhan. A fast algorithm for constructing sparse Euclidean spanners. In *Proc. of 10th SOCG*, pages 132–139, 1994.
- 20 G. Das, G. Narasimhan, and J. S. Salowe. A new way to weigh malnourished Euclidean graphs. In *Proc. of 6th SODA*, pages 215–222, 1995.
- 21 Y. Dinitz, M. Elkin, and S. Solomon. Low-light trees, and tight lower bounds for Euclidean spanners. *Discrete & Computational Geometry*, 43(4):736–783, 2010.
- 22 E. Szemerédi. On a problem of Davenport and Schinzel. *Acta Arith.*, 25, 1973.
- 23 Michael Elkin and Shay Solomon. Optimal euclidean spanners: Really short, thin, and lanky. *J. ACM*, 62(5):35:1–35:45, 2015.
- 24 M. L. Fredman and D. E. Willard. Trans-dichotomous algorithms for minimum spanning trees and shortest paths. *Journal of Computer and System Sciences*, 48(3):533–551, 1994. Announced at FOCS'90. doi:10.1016/s0022-0000(05)80064-9.
- 25 Michael L. Fredman and Michael E. Saks. The cell probe complexity of dynamic data structures. In *STOC*, pages 345–354. ACM, 1989.
- 26 J. Gudmundsson, C. Levkopoulos, G. Narasimhan, and M. H. M. Smid. Approximate distance oracles for geometric graphs. In *Proc. of 13th SODA*, pages 828–837, 2002.
- 27 J. Gudmundsson, C. Levkopoulos, G. Narasimhan, and M. H. M. Smid. Approximate distance oracles for geometric spanners. *ACM Transactions on Algorithms*, 4(1), 2008.
- 28 J. Gudmundsson, G. Narasimhan, and M. H. M. Smid. Fast pruning of geometric spanners. In *Proc. of 22nd STACS*, pages 508–520, 2005.
- 29 H. Davenport and A. Schinzel. A combinatorial problem connected with differential equations. *Am. J. Math*, 1965.


- 30 Sergiu Hart and Micha Sharir. Nonlinearity of davenport - schinzel sequences and of generalized path compression schemes. *Comb.*, 6(2):151–178, 1986.
- 31 Y. Hassin and D. Peleg. Sparse communication networks and efficient routing in the plane. In *Proc. of 19th PODC*, pages 41–50, 2000.
- 32 Omri Kahalon, Hung Le, Lazar Milenkovic, and Shay Solomon. Can’t see the forest for the trees: Navigating metric spaces by bounded hop-diameter spanners. In *PODC*, pages 151–162. ACM, 2022.
- 33 D. R. Karger, P. N. Klein, and R. E. Tarjan. A randomized linear-time algorithm to find minimum spanning trees. *Journal of the ACM*, 42(2):321–328, 1995. doi:10.1145/201019.201022.
- 34 J. M. Keil. Approximating the complete Euclidean graph. In *Proc. of 1st SWAT*, pages 208–213, 1988.
- 35 J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete & Computational Geometry*, 7:13–28, 1992.
- 36 Valerie King. A simpler minimum spanning tree verification algorithm. *Algorithmica*, 18(2):263–270, 1997.
- 37 János Komlós. Linear verification for spanning trees. *Comb.*, 5(1):57–65, 1985.
- 38 Hung Le, Lazar Milenkovic, and Shay Solomon. Sparse euclidean spanners with tiny diameter: A tight lower bound. In *SoCG*, volume 224 of *LIPICs*, pages 54:1–54:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
- 39 Hung Le and Shay Solomon. Truly optimal euclidean spanners. In *FOCS*, pages 1078–1100. IEEE Computer Society, 2019.
- 40 C. Levkopoulos, G. Narasimhan, and M. H. M. Smid. Efficient algorithms for constructing fault-tolerant geometric spanners. In *Proc. of 30th STOC*, pages 186–195, 1998.
- 41 Y. Mansour and D. Peleg. An approximation algorithm for min-cost network design. *DIMACS Series in Discr. Math and TCS*, 53:97–106, 2000.
- 42 G. Narasimhan and M. Smid. *Geometric Spanner Networks*. Cambridge University Press, 2007.
- 43 Gabriel Nivasch. Improved bounds and new techniques for davenport-schinzel sequences and their generalizations. *J. ACM*, 57(3):17:1–17:44, 2010.
- 44 Seth Pettie. An inverse-ackermann type lower bound for online minimum spanning tree verification. *Comb.*, 26(2):207–230, 2006.
- 45 Seth Pettie. Sharp bounds on davenport-schinzel sequences of every order. *J. ACM*, 62(5):36:1–36:40, 2015.
- 46 M. Pătrașcu and E. D. Demaine. Tight bounds for the partial-sums problem. In *Proc. of 15th SODA*, pages 20–29, 2004.
- 47 S. Rao and W. D. Smith. Approximating geometrical graphs via “spanners” and “banyans”. In *Proc. of 30th STOC*, pages 540–550, 1998.
- 48 J. Ruppert and R. Seidel. Approximating the d -dimensional complete Euclidean graph. In *Proc. of 3rd CCCG*, pages 207–210, 1991.
- 49 Micha Sharir. Almost linear upper bounds on the length of general davenport-schinzel sequences. *Comb.*, 7(1):131–143, 1987.
- 50 Micha Sharir. Improved lower bounds on the length of davenport - schinzel sequences. *Comb.*, 8(1):117–124, 1988.
- 51 S. Solomon and M. Elkin. Balancing degree, diameter and weight in Euclidean spanners. In *Proc. of 18th ESA, Part 1*, pages 48–59, 2010.
- 52 Shay Solomon. Sparse euclidean spanners with tiny diameter. *ACM Trans. Algorithms*, 9(3):28:1–28:33, 2013.
- 53 Shay Solomon. From hierarchical partitions to hierarchical covers: optimal fault-tolerant spanners for doubling metrics. In *STOC*, pages 363–372. ACM, 2014.
- 54 R. E. Tarjan. Efficiency of a good but not linear set union algorithm. *J. ACM*, 22(2):215–225, 1975.

- 55 R. E. Tarjan. Applications of path compression on balanced trees. *J. ACM*, 26(4):690–715, 1979.
- 56 Mikkel Thorup. Shortcutting planar digraphs. *Comb. Probab. Comput.*, 4:287–315, 1995.
- 57 Mikkel Thorup. Parallel shortcutting of rooted trees. *J. Algorithms*, 23(1):139–159, 1997.
- 58 A. C. Yao. Space-time tradeoff for answering range queries. In *Proc. of 14th STOC*, pages 128–136, 1982.



Shortest Paths in Portalgons

Maarten Löffler  

Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Computer Science, Tulane University, New Orleans, LA, USA

Tim Ophelders 

Department of Information and Computing Sciences, Utrecht University, The Netherlands
Department of Mathematics and Computer Science, TU Eindhoven, The Netherlands

Rodrigo I. Silveira  

Department de Matemàtiques, Universitat Politècnica de Catalunya, Barcelona, Spain

Frank Staals 

Department of Information and Computing Sciences, Utrecht University, The Netherlands

Abstract

Any surface that is intrinsically polyhedral can be represented by a collection of simple polygons (*fragments*), glued along pairs of equally long oriented edges, where each fragment is endowed with the geodesic metric arising from its Euclidean metric. We refer to such a representation as a *portalgon*, and we call two portalgons equivalent if the surfaces they represent are isometric.

We analyze the complexity of shortest paths. We call a fragment *happy* if any shortest path on the portalgon visits it at most a constant number of times. A portalgon is happy if all of its fragments are happy. We present an efficient algorithm to compute shortest paths on happy portalgons.

The number of times that a shortest path visits a fragment is unbounded in general. We contrast this by showing that the intrinsic Delaunay triangulation of any polyhedral surface corresponds to a happy portalgon. Since computing the intrinsic Delaunay triangulation may be inefficient, we provide an efficient algorithm to compute happy portalgons for a restricted class of portalgons.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Polyhedral surfaces, shortest paths, geodesic distance, Delaunay triangulation

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.48

Related Version *Full Version*: <https://arxiv.org/abs/2303.08937> [18]

Funding *Tim Ophelders*: Partially supported by the Dutch Research Council (NWO) under project no. VI.Veni.212.260.

Rodrigo I. Silveira: Partially funded by MICINN through project PID2019-104129GB-I00/ MCIN/AEI/ 10.13039/501100011033.

Acknowledgements We are grateful to the anonymous reviewers for the detailed and valuable feedback provided, which helped us to improve the paper considerably.

1 Introduction

We define a *portalgon* \mathcal{P} to be a collection of simple polygons (*fragments*) with some pairs of edges identified, see Figure 1. When we stitch together all fragments of a portalgon we obtain a two-dimensional surface Σ , whose *intrinsic metric is polyhedral* [7, 19]. Note that Σ is not necessarily embeddable in \mathbb{R}^3 with flat faces or without self-intersections, and not necessarily orientable. We say that \mathcal{P} is a *representation* of Σ ; crucially, the same surface may in principle have many different representations. Portalgons can be seen as a generalization of simple polygons, polygons with holes, polyhedral surfaces, and even developable surfaces.

We are interested in the *computational complexity* of computing *shortest paths* on portalgons. In particular, we analyze the *shortest path map* $\text{SPM}(s)$; a representation of all shortest



© Maarten Löffler, Tim Ophelders, Rodrigo I. Silveira, and Frank Staals;
licensed under Creative Commons License CC-BY 4.0

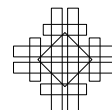
39th International Symposium on Computational Geometry (SoCG 2023).

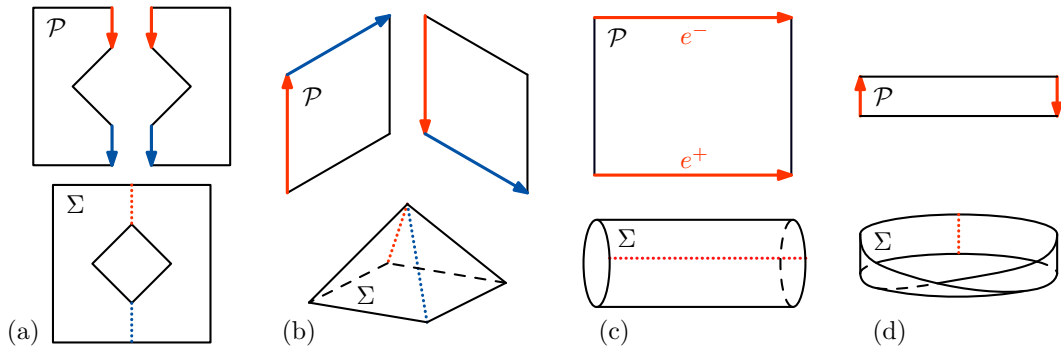
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 48; pp. 48:1–48:16



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





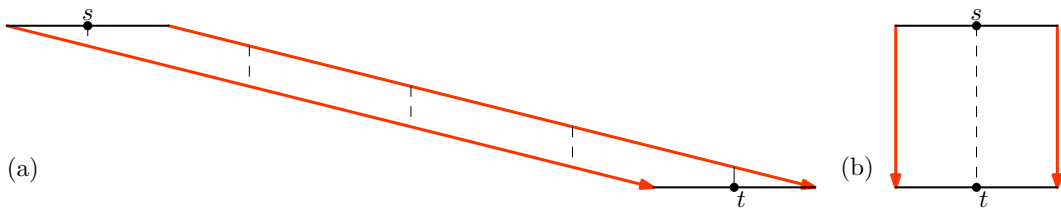
■ **Figure 1** Examples of portalgons. Arrows of the same color represent portals that are identified with each other. (a) A portalgon that represents a polygon with a hole. (b) A portalgon that represents the surface of a bottomless pyramid. (c) A portalgon that represents the surface of a cylinder. (d) A portalgon that represents a Möbius strip, a non-orientable surface.

paths from a source point s to all other points in the portalgon. Our main insights are:

- The complexity of a shortest path on a surface Σ , represented by a given portalgon \mathcal{P} , may be *unbounded* in terms of the combinatorial properties of Σ and \mathcal{P} .
- The complexity of a shortest path depends on a parameter of the particular portalgon \mathcal{P} representing the surface that we refer to as its *happiness* h . In particular, we show that the maximum complexity of a shortest path is $\Theta(n + hm)$, where n is the total number of vertices in the portalgon, $m \leq n$ is the number of portals.
- Given a source point in \mathcal{P} , the complexity of its shortest path map is $O(n^2h)$. Moreover, if \mathcal{P} is triangulated, it can be computed in $O(\lambda_4(k) \log^2 k)$ time, where k is the output complexity, and $\lambda_4(k)$ the length of an order-4 Davenport-Schinzel sequence on k symbols.
- Every surface with a polyhedral intrinsic metric admits a representation as a portalgon where the happiness h is constant. Specifically, one such representation is given by its intrinsic Delaunay triangulation. In such portalgons shortest paths have complexity $O(n)$.
- Since the intrinsic Delaunay triangulation is not easy to compute, we investigate the problem of transforming a given portalgon of happiness h into one with constant happiness. We present an algorithm to do so in $O(n + \log h)$ time, for a restricted class of portalgons. The question of how to compute such a good representation, in general, remains open.

1.1 Comparison to related work

Shortest paths have been studied in many different geometric settings, such as simple polygons, polygonal domains, terrains, surfaces, and polyhedra (see e.g. [5, 11, 13, 24]; refer to [21] for a comprehensive survey). The efficient *computation* of shortest paths is a fundamental



■ **Figure 2** (a) A portalgon where a shortest path between s and t (dashed) has unbounded complexity. (b) A different representation of the same surface where the path has constant complexity.

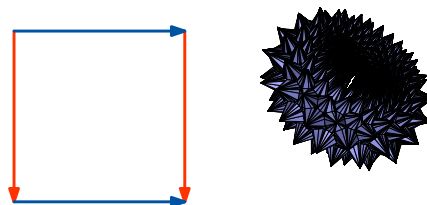
problem in computational geometry [20, 23, 12, 5, 27, 26]. When the environment is a simple polygon the situation is well understood [12, 11]. For polygons with holes, efficient solutions have also been known for quite a while [13], recently culminating in an optimal $O(n + k \log k)$ time algorithm, where n is the total number of vertices of the polygon, and k the number of holes [26]. For more complex environments, like the surface of a convex polyhedron, several algorithms have been developed [5, 22, 28], and even implemented [14, 15]. However, for more general surfaces the situation is less well understood.

Portalgons generalize many of the geometric settings studied before. Hence, our goal is to unify existing shortest paths results. To the best of our knowledge, this has not been attempted before, even though several questions closely related to the ones addressed in this work were posed as open problems in a blog post almost two decades ago [9]. Instead, portalgons are a rather unexplored concept which, though it has been long adopted into popular culture [6, 8, 25], have only been studied from a computational point of view in the context of annular ray shooting by Erickson and Nayyeri [10].

When measuring the complexity of a shortest path, we can distinguish between its intrinsic complexity, and complexity caused by the representation of the underlying surface. For example, a shortest path π on a convex polyhedron Σ in \mathbb{R}^3 with n vertices, may cross (and thus bend) at $O(n)$ edges. Hence, a description of π on Σ has complexity $O(n)$. However, it is known that any convex polyhedron in \mathbb{R}^3 can be unfolded into a simple planar polygon P_Σ , and in such a way that a shortest path π corresponds to a line segment in P_Σ [1, 2, 5]. Hence, π actually has a constant complexity description in P_Σ . It is easy to see that some portalgons may have shortest paths of *unbounded* complexity; as illustrated in Figure 2(a). This unbounded complexity, however, is completely caused by the representation, and indeed there is another equivalent portalgon without this behavior (Figure 2(b)). We introduce a parameter that explicitly measures the potential for shortest paths to have high complexity, which we call *happiness* – refer to Section 2 for a formal definition.

In Section 3, we analyze the complexity of shortest paths in terms of the happiness. Our first main result is that, if we have a portalgon with n vertices and happiness h , then the complexity of its shortest path map from a given source point is $O(n^2 h)$. Moreover, we show that, for triangulated portalgons, it can be computed in an output-sensitive fashion: if k is the complexity of the shortest path map, then it can be computed in $O(\lambda_4(k) \log^2 k)$ time.

It is worth noting that our analysis of the shortest path map has similarities with techniques used to compute shortest paths on polyhedral surfaces, most notably [5, 22]. However, the fact that portalgons are more general implies important differences with previous methods. We need to handle surfaces of non-zero genus (e.g., see Figure 3), while



■ **Figure 3** (a) A portalgon representing a flat torus. (b) Interestingly, a flat torus *can* be embedded isometrically in \mathbb{R}^3 ; however, the resulting embedding has very high complexity [4, 16] (image from <https://www.imaginary.org/es/node/2375>).

Chen and Han [5] require genus zero to compute the map in the interior of triangles (see the proof of their Theorem 4). Moreover, Σ may be non-embeddable in Euclidean space with flat triangles. Another difference is that shortest paths in portalgons can cross the same portal edge multiple times, something that is often (explicitly or implicitly) assumed to be impossible in algorithms for polyhedral surfaces (e.g., in [22]).

The fact that the complexity of the shortest path map can be upper bounded by a function of the happiness, leads to the following natural question: for a given surface Σ , can it always be represented by a portalgon \mathcal{P} with *bounded happiness*? In Section 4, we prove that the answer to this question is “yes”. In particular, our second main result is that for any portalgon, its *intrinsic Delaunay triangulation* [3] has constant happiness.

In turn, this then leads to another natural question: given a surface Σ , can we actually efficiently *compute* a portalgon representing it that has bounded, preferably constant, happiness? Clearly, the answer to this question depends on how Σ is represented. When Σ is given as a portalgon \mathcal{P} , possibly with unbounded happiness, this question then corresponds to the problem of transforming \mathcal{P} into an equivalent portalgon with constant happiness. This is the problem we study in Section 5. Given the above result, the natural approach is to try to construct the intrinsic Delaunay triangulation of \mathcal{P} . Unfortunately, it is unknown if it is possible to do that efficiently (that is, with guarantees in terms of n and h). Our third main result shows that for a restricted class of portalgons we can give such guarantees. In particular, if the input portalgon has only one portal, n vertices, and happiness h , we can construct an equivalent portalgon that has constant happiness in $O(n + \log h)$ time.

2 Definitions and observations

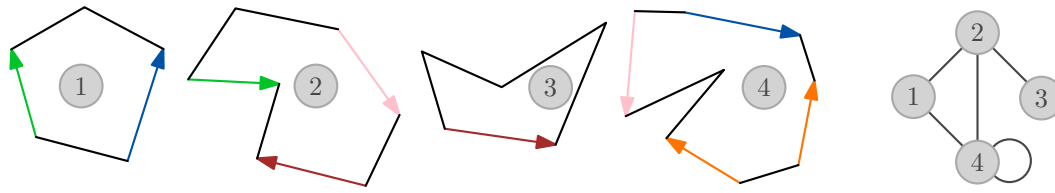
Portalgons. We define a *portalgon* \mathcal{P} to be a pair (\mathcal{F}, P) , where \mathcal{F} is a collection of simple polygons, called *fragments*, and P is a collection of portals. A *portal* is an unordered pair $e = (e^-, e^+)$ of directed, distinct, equal length, edges from some fragment(s) of \mathcal{F} . We refer to e^- and e^+ as *portal edges*, see Figure 1(c), and require that each portal edge appears in one portal. If p^- is a point on e^- , then p^+ will denote the corresponding point on e^+ .

Let n be the total number of vertices in the fragments of \mathcal{P} , and let m be the number of portal edges; the number of portals is $m/2$. Note that $m \leq n$. We denote the number of vertices and the number of portal edges in a fragment $F \in \mathcal{F}$ by n_F and m_F , respectively.

If we “glue” the edges of the fragments along their common portal edges, then the portalgon \mathcal{P} describes a surface (2-manifold with boundary) $\Sigma = \Sigma(\mathcal{P})$, see Figure 1. Specifically, Σ is the space obtained from \mathcal{P} by taking the collection \mathcal{F} and identifying corresponding pairs of points on portal edges. We can write Σ as a quotient space $(\bigcup_{F \in \mathcal{F}} F) / \sim$, where \sim is an equivalence relation that glues corresponding portal edges e^- and e^+ .

Fragment Graph. A portalgon $\mathcal{P} = (\mathcal{F}, P)$ induces a (multi)graph that we refer to as the *fragment graph* G of \mathcal{P} (see Figure 4). Each fragment $F \in \mathcal{F}$ is a node in G , and there is a link between F_1 and F_2 in G if and only if there is a portal e with e^- in F_1 and e^+ in F_2 , or vice versa. Note there may be multiple portals connecting F_1 and F_2 .

Paths and shortest paths. A path π from $s \in \Sigma$ to $t \in \Sigma$ is a continuous function mapping the interval $[0, 1]$ to Σ , where $s = \pi(0)$ and $t = \pi(1)$. For two points $p = \pi(a)$ and $q = \pi(b)$, we use $\pi[p, q]$ to denote the restriction of π to the interval $[a, b]$. The fragments of \mathcal{P} split π into a set Π of maximal, non-empty, subpaths π_1, \dots, π_z , where (the image of) each π_i is contained in a single fragment. To be precise, for each fragment F , the intersection of (the



■ **Figure 4** The fragment graph of a portalgon with four fragments and five portals.

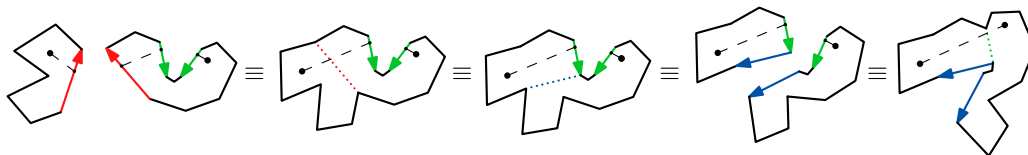
image of) π with F is a set of maximal subpaths, and Π is the union of those sets over all fragments F^1 . We define the length of π as the sum of the lengths of its subpaths, and the distance $d(s, t)$ between $s \in \Sigma$ and $t \in \Sigma$ as the infimum of length over all paths between s and t . We inherit the property that $d(s, t) = 0$ if and only if $s = t$ from the metric in each fragment. It then follows that d is also a metric. Moreover, (Σ, d) is a geodesic space.

Observe that if π is a shortest path between s and t , each subpath π_i is a polygonal path whose vertices are either endpoints of π_i or vertices of the fragment containing π_i . Furthermore, when π crosses a portal the path does not bend (otherwise we can again locally shortcut it). It then follows that a shortest path π is also polygonal and can be uniquely described by an alternating sequence $s = v_1, E_1, v_2, \dots, E_k, v_{k+1} = t$ of vertices (s, t , or portalgon vertices) and sequences of portal edges crossed by the path. We refer to such a description as the *combinatorial representation* of the path, and to the total length of these sequences as the *complexity* of π . In the remainder of the paper, we will use $\pi(s, t)$ to denote an arbitrary minimum complexity shortest path between s and t . As we observed before (see Figure 2(b)), a shortest path may still intersect a single portal edge many times, and hence the complexity of a shortest path may be unbounded in terms of n and m .

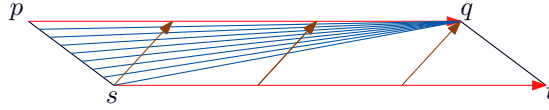
Isometry. A map $f: X \rightarrow Y$ between metric spaces X and Y is an *isometry* if $d_X(x, x') = d_Y(f(x), f(x'))$ for all $x, x' \in X$. We say that f is a *local isometry* at a point x if there exists an open neighborhood U_x of x such that the restriction of f to U_x is an isometry.

Equivalent portalgons. Given a portalgon \mathcal{P} , there are many other portalgons that describe the same surface Σ . For instance, we can always cut a fragment into two smaller fragments by transforming a chord of the fragment into a portal, or, assuming this does not cause any overlap, we can glue two fragments along a portal, see Figure 5. More formally, two portalgons \mathcal{P} and \mathcal{Q} are *equivalent*, denoted $\mathcal{P} \equiv \mathcal{Q}$, if there is a bijective isometry between them (i.e., if for any pair of points $s, t \in \mathcal{P}$, their distance in \mathcal{P} and \mathcal{Q} is the same).

¹ Note that if π passes through a vertex of \mathcal{P} that appears in multiple portals, Π contains subpaths for which the image consists of only a single point; the vertex itself. We later restrict our attention to minimum complexity paths, which allows us to get rid of such singleton paths.



■ **Figure 5** Five equivalent portalgons, with the shortest path between (the same) two points.



■ **Figure 6** A simple h -happy portalgon in which a shortest path has complexity $\Omega(hm)$ as it crosses $\Omega(m)$ portal edges (the blue portal edges) $\Omega(h)$ times each.

Happiness. Our ultimate goal will be to find, for a given input portalgon, an equivalent portalgon such that all its shortest paths have bounded complexity. To this end, we introduce the notion of a *happy portalgon*, and more specifically, a *happy fragment* of a portalgon.

Let $\mathcal{P} = (\mathcal{F}, P)$ be a portalgon, let $F \in \mathcal{F}$ be a fragment in \mathcal{F} , and let $\Pi(p, q)$ denote the set of all shortest paths between $p, q \in \Sigma$. We define $c(X)$ as the number of connected components in X . The *happiness* $\mathcal{H}(F) = \max_{p, q \in \Sigma} \max_{\pi \in \Pi(p, q)} c(F \cap \pi)$ of fragment F is defined as the maximum number of times a shortest path π between any pair of points $p, q \in \Sigma$ can go through the fragment. The happiness of \mathcal{P} is then defined as $\mathcal{H}(\mathcal{P}) = \max_{F \in \mathcal{F}} \mathcal{H}(F)$ the maximum happiness over all fragments. We call a portalgon h -happy when $\mathcal{H}(\mathcal{P}) \leq h$. Further, we sometimes refer to an $O(1)$ -happy portalgon as *happy* (without an h value).

► **Lemma 1.** *Let $\mathcal{P} = (\mathcal{F}, P)$ be an h -happy portalgon, and let $\mathcal{P}' = (\mathcal{F}', P')$ be a triangulation of \mathcal{P} . The happiness of \mathcal{P}' is at most h ; that is, \mathcal{P}' is an h -happy portalgon.*

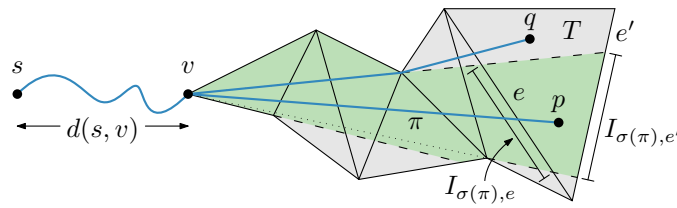
Note that if \mathcal{P} has n vertices and m portals, any triangulation \mathcal{P}' of \mathcal{P} consists of n vertices and $O(m + n)$ portals. The fact that in an h -happy portalgon a shortest path crosses every portal at most $h - 1$ times implies the following.

► **Lemma 2.** *Let \mathcal{P} be an h -happy portalgon and let s and t be two points in \mathcal{P} . A shortest path $\pi(s, t)$ between s and t has complexity $O(n + hm)$. This bound is tight in the worst case.*

Proof. The vertices of $\pi(s, t)$ are either: (i) s or t itself, (ii) vertices of \mathcal{P} , or (iii) points in which $\pi(s, t)$ crosses a portal edge. There are only two vertices of type (i) on $\pi(s, t)$. A shortest path can visit any vertex of \mathcal{P} at most once; hence, there are at most n vertices of type (ii). Finally, since \mathcal{P} is h -happy, $\pi(s, t)$ crosses every fragment $F \in \mathcal{F}$ at most h times. A shortest path of complexity $\Omega(n)$ is easy to attain in a fragment without portals and a chain with $\Omega(n)$ reflex vertices. We get the $\Omega(hm)$ term using a portalgon like in Figure 6. For any h , we can choose the length of the red portal so that the portalgon is h -happy. ◀

Shortest path map. Given a portalgon \mathcal{P} , a source point $s \in \Sigma$, and a region $\mathcal{X} \subseteq \Sigma$ the shortest path map $\text{SPM}_{\mathcal{X}}(s)$ of s is a subdivision of \mathcal{X} into maximally connected regions, such that for all points q in the interior of a region $R \in \text{SPM}_{\mathcal{X}}(s)$ the shortest path from s to q is unique, and has the same combinatorial structure, i.e., visits the same sequence of vertices and portal edges of \mathcal{P} . Note that the complexity of $\text{SPM}_{\mathcal{X}}(s)$ depends on the representation of the surface $\Sigma = \Sigma(\mathcal{P})$, that is, the portalgon \mathcal{P} . So changes to \mathcal{P} may affect the complexity of $\text{SPM}_{\mathcal{X}}(s)$. For example, splitting faces of \mathcal{P} increases the complexity of $\text{SPM}_{\mathcal{X}}(s)$. Hence, when Σ is fixed, an important problem is to find a good portalgon (i.e. one for which $\text{SPM}(s) = \text{SPM}_{\Sigma}(s)$ has low complexity) representing it.

Intrinsic Delaunay triangulation. A triangulation of a portalgon is an equivalent portalgon whose vertex set is the same, and all of whose fragments are triangles. In particular, among all such triangulations, the intrinsic Delaunay triangulation is such that for any interior edge of the triangulation, for the two triangles t and t' incident to that edge, the corners of t and t' not incident to that edge sum up to at most 180 degrees [3].



■ **Figure 7** The length of the path π from s to point p in triangle T is $d(s, v) + \|p - v\|$. The signature of π defines intervals $I_{\sigma(\pi), e}$ and $I_{\sigma(\pi), e'}$ on edges e and e' of T , as well as a distance function $d_{\sigma(\pi)}$ illustrated by the path to point q in T .

3 Shortest paths in portalgons

In this section we sketch how to compute the shortest path map $\text{SPM}_{\mathcal{P}} := \text{SPM}_{\mathcal{P}}(s)$ of a vertex s of a triangulated h -happy portalgon \mathcal{P} (a full description is in the full version [18]).

For a path π starting from the source s , define its signature, denoted $\sigma(\pi)$, to be the sequence of vertices and portals it passes through. Note that paths may simultaneously pass through a vertex and a portal (or multiple portals incident to that vertex). In this case, we break ties in the sequence by placing vertices before portals, and portals in the order that the path would pass through them if it were perturbed away from vertices (in a consistent way), where we always perturb the start of the path into a fixed triangle T_s incident to s .

If π is a shortest path from the source s to a point p in a triangle T , and v is the last vertex on $\sigma(\pi)$, then the length of π is $d(s, v) + d(v, p)$, where $d(v, p)$ can be expressed as the length of a line segment. We think of T as being embedded locally isometrically in the Euclidean plane, and by unfolding the fragments that π passes through after v , we can compute a copy of v in the plane, as well as copies of the portals and triangles that π passes through after v ; see Figure 7. The locations of these copies in the plane depend only on $\sigma(\pi)$ (but not on p or π itself). Then, $d(v, p)$ is the Euclidean distance between the copies of v and p in the plane. Let e be an edge of T , and define $I_{\sigma(\pi), e}$ to be the interval of points p on e for which the segment \overline{vp} passes through all the unfolded copies of portals of $\sigma(\pi)$ after v . For a point q in T , define

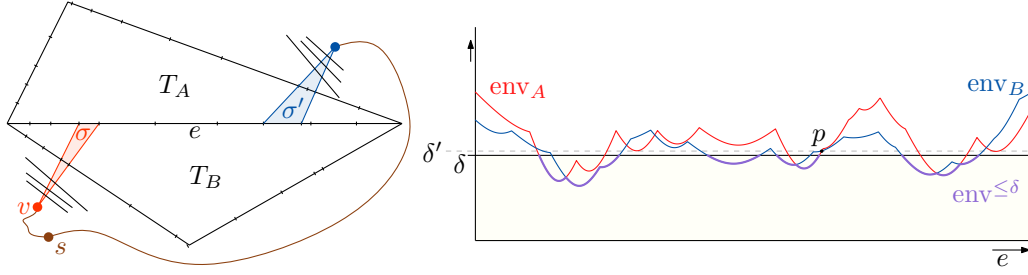
$$f_{\sigma(\pi)}(q) = \begin{cases} d(s, v) + \|\overline{vq}\| & \text{if the segment } \overline{vq} \text{ passes through } I_{\sigma(\pi), e}, \\ \infty & \text{otherwise} \end{cases}$$

Define $d_{\sigma(\pi)}(q)$ to be the infimum length over paths from s to q with signature $\sigma(\pi)$. This infimum is not necessarily realized by a path with the same signature, but is realized by a path that potentially bends around additional vertices, which are therefore inserted in its signature. We say that such a signature *reduces* to $\sigma(\pi)$. Note that if $f_{\sigma(\pi)}(q)$ is finite, then $f_{\sigma(\pi)}(q) = d_{\sigma(\pi)}(q)$. If π is a shortest path from s to p , then π has length $f_{\sigma(\pi)}(p)$. For a portal e of T , let $f_{\sigma(\pi)|e} : e \rightarrow \mathbb{R} \cup \{\infty\}$ be the restriction of $f_{\sigma(\pi)}$ to points on e .

3.1 A data structure for maintaining a lower envelope

Let F be a set of m partial functions, each pair of which can intersect at most twice. We describe a data structure storing the lower envelope env_F of F , that supports the operations:

- NextLocalMinimum(δ):** report the smallest local minimum of env_F that is larger than δ .
- NextVertex(f, q):** given a function $f \in F$ that realizes env_F at point q , find (if it exists) the lowest endpoint (v, δ') of the segment of env_F containing $(q, f(q))$ for which $\delta' > f(q)$.



■ **Figure 8** The information that we maintain while computing the SPM of the edges.

Insert(f): insert a new function into F .

► **Lemma 3.** *We can maintain env_F of a set F of partial functions, each pair of which intersects at most twice, in a data structure so that any sequence of m **Insert** operations and k **NextLocalMinimum** and **NextVertex** queries take $O(k \log^2 m + \lambda_4(m) \log m)$ time.*

3.2 Computing a shortest path map

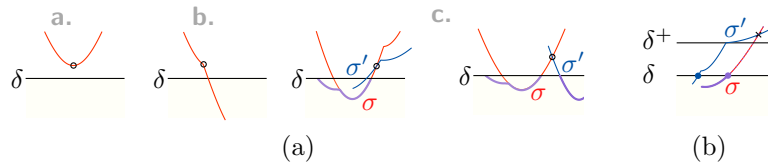
We first compute the shortest path map of s , restricted to the edges of the portalgon. As in earlier work [22, 23] we propagate a wavefront of points at distance δ from s from 0 to ∞ . However, it will be more convenient to view this as a collection of simultaneous sweepline algorithms. For each portal edge e , we sweep a horizontal line at height δ upward through the “position along $e \times$ distance from s ” space (see Figure 8), while we construct the part of SPM_e below the sweep line. The main challenge is computing the next event—the first vertex in the lower envelope of the distance functions above the sweep line—in time.

For each portal e connecting two fragments T_A and T_B , we maintain the following:

1. Let $S_A(e, \delta)$ (resp. $S_B(e, \delta)$) be the set of signatures of shortest paths from s to points on the boundary of T_A (resp. T_B), where the last element of the signature is not e , and the length of the path is at most δ . We represent each signature $\sigma \in S_A(e, \delta) \cup S_B(e, \delta)$ implicitly by storing the interval $I_{\sigma, e}$, the position of (the unfolded copy of) the last vertex v on σ , and $d(s, v)$, so that we can compute $f_{\sigma|e}$ in constant time.
2. We store the lower envelope $\text{env}_A(e, \delta)$ (resp. $\text{env}_B(e, \delta)$) of the functions $f_{\sigma|e}$, where σ ranges over the signatures $S_A(e, \delta)$ (resp. $S_B(e, \delta)$), in the data structure of Lemma 3.
3. Let $\text{env}(e, \delta)$ be the lower envelope of the functions $\text{env}_A(e, \delta)$ and $\text{env}_B(e, \delta)$. We maintain only the part of $\text{env}(e, \delta)$ that lies below the sweep line, denoted $\text{env}^{\leq \delta}(e, \delta)$.
4. We maintain a binary search tree $\text{env}^{=\delta}(e, \delta)$ storing the intersection points of $\text{env}_A(e, \delta)$ and $\text{env}_B(e, \delta)$ with the sweep line, in order along the sweep line. For each intersection point we store the function(s) from $\text{env}_A(e, \delta)$ or $\text{env}_B(e, \delta)$ realizing this intersection.
5. Finally, we maintain a set of events pertaining to the edge e . We aggregate the events of all edges in a global priority queue, and use it to advance the sweep line algorithm to the next relevant value. The events that we store for an edge e are the values $\delta' > \delta$ such that
 - a. δ' corresponds to a minimum of a function $f_{\sigma|e}$ with $\sigma \in S_A(e, \delta) \cup S_B(e, \delta)$,
 - b. δ' corresponds to a vertex of $\text{env}_A(e, \delta)$ or $\text{env}_B(e, \delta)$, or
 - c. δ' corresponds to an intersection between functions $f_{\sigma|e}$ on $\text{env}_A(e, \delta)$ and $f_{\sigma'|e}$ on $\text{env}_B(e, \delta)$, where $f_{\sigma|e}$ and $f_{\sigma'|e}$ are neighbors in $\text{env}^{=\delta}(e, \delta)$.

For each event, we also keep track of the type and corresponding functions.

► **Lemma 4.** $\text{env}^{\leq \delta}(e, \delta)$ encodes the shortest paths of length $\leq \delta$ to points on the edge e .



■ **Figure 9** (a) The different event types: **a.** local minima in env_A or env_B , **b.** breakpoints of env_A or env_B , and two events of type **c.** first intersections of env_A and env_B . (b) We will detect the intersection between env_A and env_B once $f_{\sigma|e}$ and $f_{\sigma'|e}$ become neighbors in $\text{env}^{-\delta}$ at time δ^+ .

Event handling. At an event (of any type), the order in which the functions of $\text{env}_A(e, \delta)$ and $\text{env}_B(e, \delta)$ intersect the sweep line changes. We therefore update $\text{env}^{-\delta}(e, \delta)$ by removing and inserting the appropriate functions associated with this event, and additionally make sure that we discover any additional events. To this end, we can use the **NextVertex** and **NextLocalMinimum** queries on the data structures storing $\text{env}_A(e, \delta)$ and $\text{env}_B(e, \delta)$.

As a result of the event, a new function, say $f_{\sigma'|e} \in \text{env}_B(e, \delta)$, may have appeared on $\text{env}^{\leq \delta}(e, \delta)$. We insert it into $\text{env}^{\leq \delta}(e, \delta)$, and propagate σ' , extended by edge e , into the sets $S_B(e', \delta)$ of the other two edges e' incident to T_A . We therefore call **Insert** to insert a new function into $\text{env}_A(e', \delta)$, and **NextLocalMinimum**(δ) to update the next local minimum. ²

Analysis. Let $k = |\text{SPM}_{\partial \mathcal{P}}|$ denote the complexity of the shortest path restricted to the edges. Let k_A and k_B be the number of signatures in $S_A(e, \infty)$ and $S_B(e, \infty)$, respectively. We argue that the total number of events on e is $O(\lambda_4(k_A) + \lambda_4(k_B)) + |\text{SPM}_e|$ (essentially by charging them to $\text{env}_A(e, \infty)$ and $\text{env}_B(e, \infty)$). A signature σ appears in $S_B(e, \delta)$ only when there is a shortest path to another edge e' of T_B . Since every such edge e' propagates to the edges of the two triangles incident to e' it then follows that $\sum_{e \in \partial \mathcal{P}} (|S_A(e, \delta)| + |S_B(e, \delta)|) = O(|\text{SPM}_{\partial \mathcal{P}}|)$. Thus, the total number of events, over all edges e , is $O(\lambda_4(k))$. Since we can handle every event in $O(\log^2 k)$ time, we can then compute $\text{SPM}_{\partial \mathcal{P}}$ in $O(\lambda_4(k) \log^2 k)$ time.

► **Lemma 5.** For an h -happy portalgon, the complexity of $\text{SPM}_{\partial \mathcal{P}}$ is $O(n^2 h)$.

Extension into the Interior. For each triangle T , we can now compute SPM_T from $\text{SPM}_{\partial T}$ reusing the algorithm from Mitchell et al. [22]. We therefore obtain the following result.

► **Theorem 6.** Let \mathcal{P} be a triangulated h -happy portalgon with n vertices and m portals, and let s be a given source point. The shortest path map $\text{SPM}_{\mathcal{P}}(s)$ of s has complexity $k = O(n^2 h)$ and can be computed in $O(\lambda_4(k) \log^2 k)$ time.

4 Existence of happy portalgons

In this section, we show that for every portalgon there exists an equivalent $O(1)$ -happy portalgon; specifically, the *intrinsic Delaunay triangulation* [3] \mathcal{T} of Σ induces a happy

² Mitchell, Mount, and Papadimitriou [22] use a similar overall algorithm. They define a notion of T_A -free paths that arrive at edge e from T_B . They prove that these paths act sufficiently like “real” shortest paths so that they can explicitly maintain the set of shortest T_A -free and T_B -free paths. Unfortunately, some of the arguments used hold only when a shortest path may cross a portal edge at most once (i.e., when \mathcal{P} is 1-happy). In case of the weighted region problem, Mitchell and Papadimitriou show how to deal with this by extending this notion of T_A -free paths to *locally T_A -free* paths [23]. However, it is unclear how to bound the number of such paths when the genus may be non-zero.

portalgon, whose fragments correspond to the triangles of \mathcal{T} . The resulting portalgon may have more fragments than the original, but its total complexity is still linear.

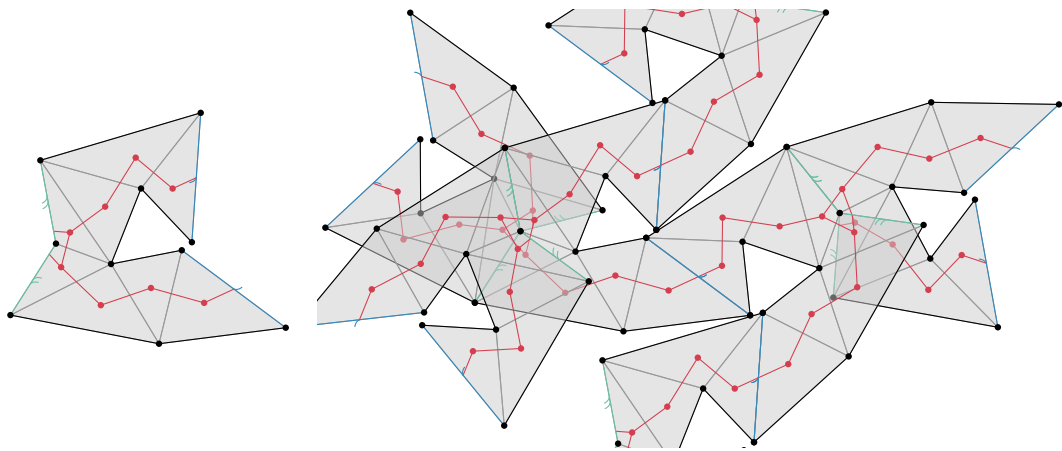
4.1 Intrinsic Delaunay triangulation

Let \mathcal{T} be a triangulated portalgon, and let Σ be its surface; let V be the set of vertices of Σ . Intuitively, the intrinsic Delaunay triangulation of Σ has a (straight) edge between two vertices $u, v \in V$ when there exists a circle with u and v on its boundary that contains no other vertices of V when we “unfold” Σ , for example by identifying edges of \mathcal{T} as in Section 3.

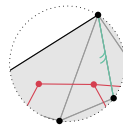
To simplify geometric arguments, we derive a simply-connected space $\hat{\Sigma}$ from Σ . We can think of $\hat{\Sigma}$ as the universal cover of $\Sigma \setminus V$, with vertices reinserted at the corresponding locations. More formally, we can define $\hat{\Sigma}$ by considering the directed fragment graph G of \mathcal{T} . For any portal e of \mathcal{T} , we denote by $T_{e^-}, T_{e^+} \in \mathcal{F}$ the triangles (nodes of G) that contain the respective portal edge; G has a link \dot{e} from T_{e^-} to T_{e^+} and a link \dot{e}^{-1} from T_{e^+} to T_{e^-} . We say that \dot{e} and \dot{e}^{-1} are inverses of each other. A *walk* in G from a triangle T to a triangle T' is a (possibly empty, if $T = T'$) sequence of links of G , such that T is the source of the first link, and the source of the $(i + 1)$ -st is the target of the i -th link, and the target of the last link is T' . For a walk w from T to T' , we write $w: T \rightarrow_G T'$ and say that a walk is *backtracking* if it contains two consecutive links that are inverses of each other. Fix an arbitrary root triangle T_0 , and for a walk $w: T_0 \rightarrow_G T$, let T_w be a copy of the target triangle T placed in the Euclidean plane by unfolding T_0, T_1, \dots, T_w along their common portals. Let $\hat{\mathcal{F}} = \bigsqcup \{T_w \mid T \in \mathcal{F}, w: T_0 \rightarrow T, w \text{ is not backtracking}\}$ be the disjoint union of Euclidean triangles of target triangles of walks starting at T_0 . We can now define $\hat{\Sigma}$ as $\hat{\mathcal{F}}/\sim$, where for any two walks w and w' , where w' is obtained from w by removing its final link, \sim glues $T_{w'}$ to T_w along the sides corresponding to the last link of w . Let $q: \hat{\Sigma} \rightarrow \Sigma$ be the map that sends points of $\hat{\Sigma}$ to their corresponding point in Σ . A map $\hat{g}: X \rightarrow \hat{\Sigma}$ is a *lift* of $g: X \rightarrow \Sigma$ if $g = q \circ \hat{g}$. Any path $\pi: [0, 1] \rightarrow \Sigma$ has a lift $\hat{\pi}$ in $\hat{\Sigma}$.

There is a map $f: \hat{\Sigma} \rightarrow \mathbb{R}^2$ whose restriction to any triangle of $\hat{\Sigma}$ is an isometry, and whose restriction to any pair of adjacent triangles is injective. We can think of f as unfolding $\hat{\Sigma}$ so that it lies flat in the plane and is locally isometric everywhere except at the vertices (vertices are the only source of curvature), see Figure 10.

► **Observation 7.** Let $B_r(x, y) = \{p \in \mathbb{R}^2 \mid \|p - (x, y)\| < r\}$ be the open disk in the plane of



■ **Figure 10** Σ (left) and a local region of $\hat{\Sigma}$ (right). The dual graph of $\hat{\Sigma}$ is an infinite tree (red).



■ **Figure 11** Although f is not injective in general, for a triangle T of a Delaunay triangulation, the restriction of f to $C(T)$ is injective and contains no vertices of the triangulation in the interior.

radius r centered at (x, y) . If a component U of $f^{-1}(B_r(x, y))$ contains no vertices, then the restriction of f to the closure of U is an isometry.

For a triangle T of $\hat{\Sigma}$, let $D(T)$ be the open disk bounded by the circumcircle of $f(T)$, and let $C(T)$ be the closure of the component of $f^{-1}(D(T))$ that contains the interior of T . A triangle T of $\hat{\Sigma}$ is *Delaunay* if $C(T)$ does not contain any vertices of triangles adjacent to T in its interior. \mathcal{T} is an *intrinsic Delaunay triangulation* of Σ if and only if all triangles of $\hat{\mathcal{T}}$ are Delaunay. Lemma 8 generalizes a well-known property of Delaunay triangulations in \mathbb{R}^2 .

► **Lemma 8.** [Bobenko et al. [3]] *If $\hat{\mathcal{T}}$ is an intrinsic Delaunay triangulation of $\hat{\Sigma}$, then for any $T \in \hat{\mathcal{T}}$, $C(T)$ contains no vertices in its interior.*

Together with Observation 7 this implies the following corollaries, see also Figure 11.

► **Corollary 9.** *For any triangle T of an intrinsic Delaunay triangulation of $\hat{\Sigma}$, the restriction of f to $C(T)$ is injective, and shortest paths intersect $C(T)$ in straight segments.*

► **Corollary 10.** *For any triangle T of an intrinsic Delaunay triangulation of $\hat{\Sigma}$, $\partial f(C(T))$ is a union of chords and circular arcs of $\partial D(T)$, where each chord is a boundary edge.*

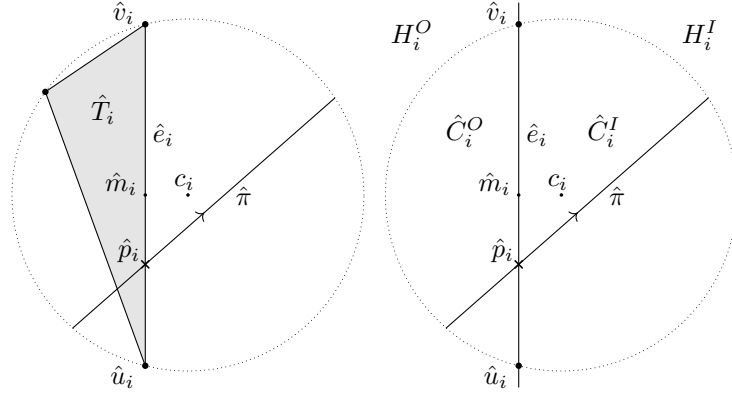
4.2 Intrinsic Delaunay triangulations are happy

We are now ready to prove the main result of this section: the intrinsic Delaunay triangulation of any portalgon has happiness less than 7.

Let the portalgon \mathcal{T} be an intrinsic Delaunay triangulation of Σ . We want to bound the number of intersections between a shortest path π on Σ and portals (edges of the triangulation). For this, let π be a shortest path between two given points on Σ , and among all such paths, assume that π has a minimum number of crossings with portals of \mathcal{T} .

Now consider an arbitrary edge e between two vertices u and v in V and let m be its midpoint. We will show that π intersects e only constantly often. For a contradiction, suppose that π intersects e at least 7 times. To bound the total number of intersections with e we analyze the geometry of $\hat{\pi}$ in a local neighborhood of a lift \hat{e} of e ; in particular, we will argue that there cannot be many other copies of e in the neighbourhood of \hat{e} that are all crossed by π . Since clearly there can also not be any copies of e far away that are crossed by π (since then it would be better to walk along e), the result will follow. Here, we give a sketch of the arguments, the full proof can be found in the full version [18].

Arbitrarily fix one of the triangles T incident to e and consider the neighborhood $C(\hat{T})$ of \hat{e} for the corresponding lift \hat{T} incident to \hat{e} . By Corollary 9, the restriction of f to $C(\hat{T})$ is an isometry, and $\hat{\pi}$ is locally a straight line segment. Let p_i be the i -th point of intersection of π with e , and define λ_i such that $\pi(\lambda_i) = p_i$. Let \hat{e}_i be a lift of e containing $\hat{\pi}(\lambda_i)$, and let $\hat{p}_i, \hat{T}_i, \hat{u}_i, \hat{m}_i, \hat{v}_i$ be the respective lifts of p_i, T, u, m, v incident to \hat{e}_i , see Figure 12 (left). Let $D_i := D(\hat{T}_i)$, and let f_i be the restriction of f to $C_i := C(\hat{T}_i)$. Let c_i be the center of



■ **Figure 12** The crossing p_i of π with e is inward.

D_i , i.e., the circumcenter of $f(\hat{T}_i)$. Define H_i^I and H_i^O to be the two half-planes bounded by the line through $f(\hat{e}_i)$, such that H_i^I contains c_i (if $c_i = f(\hat{m}_i)$, label the half-planes by H_i^I and H_i^O arbitrarily). We respectively call H_i^I and H_i^O the *inner* and *outer* half-plane of \hat{e}_i , and define $C_i^I := f_i^{-1}(H_i^I)$ to be the *inner* component of \hat{e}_i , and $C_i^O := f_i^{-1}(H_i^O)$ to be the *outer* component of \hat{e}_i . We call the crossing p_i *inward* if p_i is a non-transversal crossing or $\hat{\pi}$ crosses \hat{p}_i from C_i^O to C_i^I , see Figure 12 (right).

Let s be the segment of e from u to m . Assume without loss of generality that at least four of the (at least seven) crossings of π with e lie on s (otherwise relabel u and v).

► **Lemma 11.** *For any inward crossing p_i of π with s , none of the crossings p_j with $j > i$ lie on the segment of s between u and p_i .*

If p_i of π is not inward, then it is inward on the reverse of π . Corollary 12 follows.

► **Corollary 12.** *If some crossing p_i of π with s is not inward, then none of the crossings p_h with $h < i$ lie on the segment of s between u and p_i .*

Lemma 11 implies that for the sequence of intersections p_i of π with s , the distance function from u to p_i along s has only one local minimum. So there exists a subsequence of at least three crossings p_h, p_i and p_j ($h < i < j$) of π with s such that p_i lies between p_h and p_j . Assume without loss of generality that the distances (along s) from u to p_h, p_i , and p_j are increasing (the other case follows by considering the reverse of π). We now observe:

► **Observation 13.** $f(\hat{m}_i) \neq c_i$

By Corollary 12, p_i is an inward crossing. We show how the possible locations of the subsequent crossings inside C_i are constrained. Define D_i' to be the disk concentric with D_i , whose boundary passes through $f(\hat{m}_i)$. Any chord of D_i that passes through D_i' is at least as long as e . Let $\bar{e}_h := f(\hat{e}_h) \cap D_i$. We observe:

► **Lemma 14.** \bar{e}_h cannot intersect the closure of D_i' .

We will arrive at a contradiction to our initial assumption by showing that:

► **Lemma 15.** *If π crosses e at least 7 times then \bar{e}_h must intersect the closure of D_i' .*

We conclude that π intersects e at most 6 times. We thus obtain:

► **Theorem 16.** *Let \mathcal{P} be a portalgon with n vertices. There exists a portalgon $\mathcal{P}' \equiv \mathcal{P}$ with $O(n)$ vertices that is $O(1)$ -happy.*

5 Making portalgons happy

While the approach in Section 4 is, in principle, constructive, its running time depends on the number of edge flips required to reconfigure an initial triangulation of \mathcal{P} into the intrinsic Delaunay triangulation. This number does not (only) depend on the input complexity, except for the case where there is a bound on the minimum angle on the input triangles [17].

In this section, we analyze what can be done when there is no such bound. We study conditions for a portalgon \mathcal{P} to be happy, and present a method to rearrange an unhappy portalgon into an equivalent one that is happy. In the full version [18], we show we can reduce any portalgon whose fragment graph has at most one cycle to a portalgon with just one fragment and one portal. Thus, here we only consider one portal, and give an efficient method to compute an equivalent happy portalgon. For brevity, we present the case where the portal edges are parallel. We also note that if the angle between the two edges is at least a constant, the fragment is already happy. Hence, the interesting case is that of nearly-parallel edges.

Single parallel portal analysis. The happiness of a fragment with a single portal depends on the *shift* of the two (parallel) portal edges. The shift of two portal edges e^-, e^+ is the distance between the two perpendiculars to the portal edges that go through the start vertices of e^- and e^+ , respectively. In the following, we assume without loss of generality that the portals are horizontal, thus the shift Δ is simply the difference in x -coordinate between the start vertices of the portal edges. Let v denote the vertical distance between e^- and e^+ .

► **Lemma 17.** *Let F be a fragment with exactly two portal edges e^-, e^+ , which are parallel and belong to the same portal. If the shift Δ of e^-, e^+ is 0, then the fragment is 2-happy.*

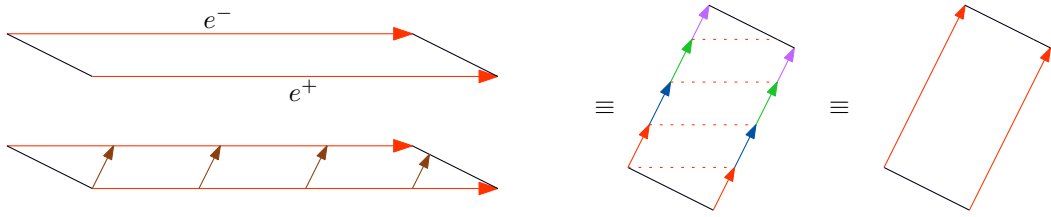
If $\Delta \neq 0$, a fragment might be happy or not. Next we present a method to transform a fragment that is not happy into an equivalent portalgon that is happy. The idea is to create a new portal by cutting F through a line in the direction orthogonal to the line through the two start vertices of the portals. First we present the idea for the case where F is a parallelogram with two parallel portal edges e^- and e^+ . Assume without loss of generality that e^- is above e^+ , and that e^- starts to the left of e^+ . In this case, the slope of the line in the direction orthogonal to the portal start points is $z = -\Delta/v$. We begin at the leftmost vertex of e^+ , and shoot a ray with slope z in the interior of F until we hit the boundary. Every time the ray crosses the portal, we “cut” along this ray, creating a new portal along it. This creates several smaller fragments, which we glue together along the pieces of the original portals, into one fragment. The resulting fragment is a rectangle F' . See Figure 13 for an illustration. Note that, by definition of z , this new fragment F' now has shift zero. Hence:

► **Lemma 18.** *For any parallelogram with two parallel portal edges, there is an equivalent parallelogram that has $\Delta = 0$ and therefore is 2-happy.*

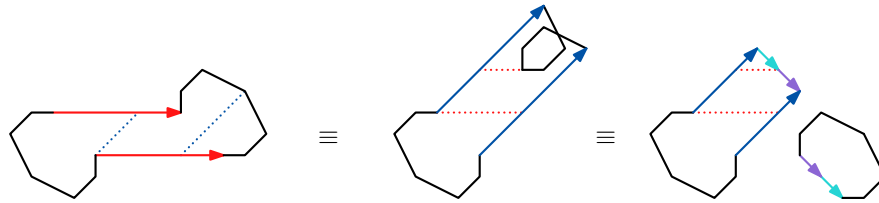
For a fragment with two parallel portals and non-zero shift, there is a unique equivalent fragment with zero shift, which is the one obtained by cutting along the perpendicular ray. However, if F is not a parallelogram, it may occur that when gluing together the new smaller fragments we obtain a non-simple polygon. Therefore, we may not be able to transform F into a single equivalent happy fragment, as shown in Figure 14. In that case, cutting along the perpendicular ray produces a non-simple fragment.

Fortunately, we can transform any fragment with two parallel portal edges into a small constant number of happy fragments.

48:14 Shortest Paths in Portalgons



■ **Figure 13** Left: a fragment with two parallel edges and non-zero shift. Center: result of cutting the fragment along a ray orthogonal to the line through the two start vertices of the portals, resulting in a new fragment with several portals. The latter is equivalent to a fragment with one portal and zero shift (right).



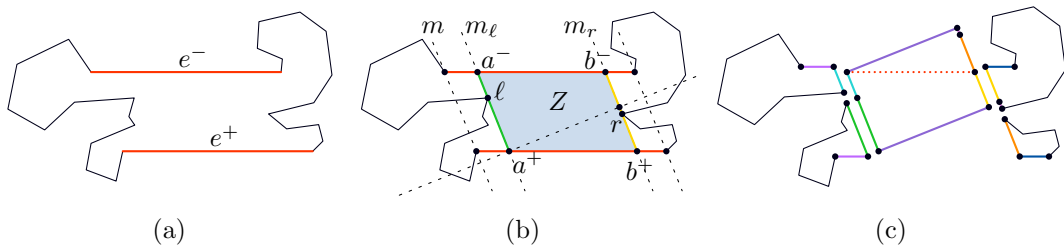
■ **Figure 14** Example of a fragment (left) where the technique of cutting along the perpendicular ray produces a non-simple polygon (center). This can be fixed by using one more fragment (right).

► **Lemma 19.** *Let \mathcal{P} be a portalgon with one fragment F with n vertices, and one portal whose edges are parallel. There exists a 5-happy portalgon \mathcal{P}' equivalent to \mathcal{P} consisting of at most three fragments and total complexity $O(n)$.*

Proof sketch. Assume w.l.o.g. that e^- and e^+ are horizontal and oriented left-to-right. We argue that when no three vertices of F are colinear, and F is not already 5-happy, we can split F into at most seven 4-happy fragments of complexity $O(n)$. See Figure 15.

Let m be the line through the start points of e^- and e^+ . If there is no translate of m whose intersection with F contains a segment connecting e^- to e^+ , F is already 5-happy. Let m_ℓ be the leftmost such translate of m and m_r the rightmost such translate; m_ℓ contains a vertex ℓ of F and m_r contains a vertex r of F (possibly, ℓ or m is an endpoint of e^- or e^+). Let a^- and a^+ be the intersection points of m_ℓ with e^- and e^+ , and let b^- and b^+ be the intersection points of m_r with e^- and e^+ . We cut the parallelogram $Z = a^-b^-a^+b^+$ from F , which splits F into at most seven fragments.

We now transform Z into a 2-happy fragment using Lemma 18. Let T and B be the fragments containing the starting points of e^- and e^+ , respectively. We argue that T is 4-happy (other arguments are symmetric): consider the maximal connected components of a



■ **Figure 15** (a) Fragment with two parallel portals e^- and e^+ . (b) Lines m_ℓ and m_r define a parallelogram Z , which splits F in at most seven fragments. (c) The resulting 5-happy fragments.

shortest path $\pi = \pi(s, t)$ with F . Such a component either: (i) contains s , (ii) contains t , or (iii) connects a point p_i on e^- to a point q_i on e^+ . Each such component can intersect a^-a^+ at most once, so each such component can intersect T at most once.

We can further classify the type (iii) components into three types, depending on whether p_i lies on the part of e^- in T and whether q_i lies on the part of e^+ in B . A case-by-case analysis shows that there are up to four components of $\pi \cap F$ that intersect T , and each of them intersects T in one consecutive subpath. Hence, T is 4-happy. ◀

► **Lemma 20.** *Let F be a parallelogram with two parallel portal edges. We can compute an equivalent 2-happy parallelogram F' in $O(1)$ time.*

Proof sketch. To compute the new, equivalent fragment F' , we could explicitly generate the ray with slope z , and compute its intersection points with the portal edges, tracing the ray until it hits a non-portal edge of the fragment. However, this would result in a running time linear in the number of such intersections. Instead, we show in the full version that after cutting along the ray and gluing the pieces together, the result is always a rectangle (as in Figure 13, right) that can be computed directly in $O(1)$ time. ◀

Single non-parallel portal analysis. In the full version [18], we address the case where the fragment is not a parallelogram, and, more interestingly, where the two portal edges are not parallel. The proof of the equivalent of Lemma 19 proceeds analogously, but is more technical. Lemma 20, however, does not have a direct analogue. In particular, the time required to compute an equivalent happy portalgon depends also on the bit complexity of the input coordinates. However, we show that we can achieve $O(\log h)$ time for h the initial happiness of the portalgon. We do this by encoding a single pass through the portal in a transformation matrix M , and computing M^h by performing an exponential search in h . This result is summarized as follows.

► **Theorem 21.** *Let \mathcal{P} be an h -happy portalgon with n vertices and fragment graph G , such that G has at most one simple cycle. We can transform \mathcal{P} into an equivalent 5-happy portalgon \mathcal{P}' of total complexity $O(n)$ in $O(n + \log h)$ time.*

References


- 1 Pankaj K. Agarwal, Boris Aronov, Joseph O'Rourke, and Catherine A. Schevon. Star unfolding of a polytope with applications. *SIAM Journal on Computing*, 26(6):1689–1713, 1997. doi:10.1137/S0097539793253371.
- 2 Boris Aronov and Joseph O'Rourke. Nonoverlap of the star unfolding. *Discrete & Computational Geometry*, 8(3):219–250, 1992.
- 3 Alexander I Bobenko and Boris A Springborn. A discrete Laplace–Beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4):740–756, 2007.
- 4 Vincent Borrelli, Saïd Jabrane, Francis Lazarus, and Boris Thibert. Flat tori in three-dimensional space and convex integration. *Proceedings of the National Academy of Sciences*, 109(19):7218–7223, 2012. doi:10.1073/pnas.1118478109.
- 5 Jindong Chen and Yijie Han. Shortest paths on a polyhedron. *Int. J. Comput. Geom. Appl.*, 6(2):127–144, 1996. doi:10.1142/S0218195996000095.
- 6 Valve Corporation. Portal, 2007. Video game.
- 7 Yu. D. Burago V. A. Zalgaller (Eds.). *Geometry III: Theory of Surfaces*. Springer Verlag, 1993.
- 8 H Ellison. The city on the edge of forever, 1967. Star Trek, season 1, episode 28.

- 9 Jeff Erickson. Ernie's 3d pancakes: Shortest paths on pl surfaces, 2006. March 14, 2023. URL: https://3dpancakes.typepad.com/ernie/2006/03/shortest_paths_.html.
- 10 Jeff Erickson and Amir Nayyeri. Tracing compressed curves in triangulated surfaces. *Discret. Comput. Geom.*, 49(4):823–863, 2013. doi:10.1007/s00454-013-9515-z.
- 11 Leonidas Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989. doi:10.1016/0022-0000(89)90041-X.
- 12 Leonidas Guibas, John Hershberger, Daniel Leven, Micha Sharir, and Robert E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1):209–233, 1987. doi:10.1007/BF01840360.
- 13 John Hershberger and Subhash Suri. An Optimal Algorithm for Euclidean Shortest Paths in the Plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.
- 14 Biliana Kaneva and Joseph O'Rourke. An implementation of Chen & Han's shortest paths algorithm. URL: <http://cs.smith.edu/~jorourke/Papers/shortest.ps.gz>.
- 15 Stephen Kiazzyk, Sébastien Loriot, and Éric Colin de Verdière. Triangulated surface mesh shortest paths. URL: https://doc.cgal.org/latest/Surface_mesh_shortest_path/index.html.
- 16 Francis Lazarus and Florent Talerie. A Universal Triangulation for Flat Tori. *CoRR*, March 2022. URL: <https://arxiv.org/abs/2203.05496>.
- 17 Yong-Jin Liu, Chun-Xu Xu, Dian Fan, and Ying He. Efficient Construction and Simplification of Delaunay Meshes. *ACM Transactions on Graphics*, 34(6):1–13, 2015. doi:10.1145/2816795.2818076.
- 18 Maarten Löffler, Tim Ophelders, Rodrigo I. Silveira, and Frank Staals. Shortest paths in portalgons. *CoRR*, 2023. URL: <https://arxiv.org/abs/2303.08937>.
- 19 A.D. Milka. Multidimensional spaces with polyhedral metric of nonnegative curvature I. *Ukrain. Geom. Sb.*, 5(6):103–114, 1968. In Russian.
- 20 Joseph S. B. Mitchell. A new algorithm for shortest paths among obstacles in the plane. *Annals of Mathematics and Artificial Intelligence*, 3(1):83–105, March 1991. doi:10.1007/BF01530888.
- 21 Joseph S. B. Mitchell. Shortest paths and networks. In Jacob E. Goodman, Joseph O'Rourke, and Csaba D. Toth, editors, *Handbook of Discrete and Computational Geometry, Third Edition*, pages 811–848. Chapman and Hall/CRC, 2017.
- 22 Joseph S. B. Mitchell, David M. Mount, and Christos H. Papadimitriou. The discrete geodesic problem. *SIAM J. Comput.*, 16(4):647–668, 1987. doi:10.1137/0216045.
- 23 Joseph S. B. Mitchell and Christos H. Papadimitriou. The weighted region problem: Finding shortest paths through a weighted planar subdivision. *J. ACM*, 38(1):18–73, 1991. doi:10.1145/102782.102784.
- 24 Yevgeny Schreiber. An optimal-time algorithm for shortest paths on realistic polyhedra. *Discret. Comput. Geom.*, 43(1):21–53, 2010. doi:10.1007/s00454-009-9136-8.
- 25 A. Wachowski and L. Wachowski. Matrix revolutions, 2003. Warner Bros. Motion picture.
- 26 Haitao Wang. A new algorithm for Euclidean shortest paths in the plane. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2021*, pages 975–988, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3406325.3451037.
- 27 Haitao Wang. Shortest paths among obstacles in the plane revisited. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 810–821. SIAM, 2021. doi:10.1137/1.9781611976465.51.
- 28 Shi-Qing Xin and Guo-Jin Wang. Improving Chen and Han's algorithm on the discrete geodesic problem. *ACM Trans. Graph.*, 28(4), September 2009. doi:10.1145/1559755.1559761.

The Geodesic Edge Center of a Simple Polygon

Anna Lubiw  

David R. Cheriton School of Computer Science, University of Waterloo, Canada

Anurag Murty Naredla 

David R. Cheriton School of Computer Science, University of Waterloo, Canada

Institut für Informatik, University of Bonn, Bonn, Germany

Abstract

The *geodesic edge center* of a polygon is a point c inside the polygon that minimizes the maximum geodesic distance from c to any edge of the polygon, where *geodesic distance* is the shortest path distance inside the polygon. We give a linear-time algorithm to find a geodesic edge center of a simple polygon. This improves on the previous $O(n \log n)$ time algorithm by Lubiw and Naredla [European Symposium on Algorithms, 2021]. The algorithm builds on an algorithm to find the geodesic *vertex* center of a simple polygon due to Pollack, Sharir, and Rote [Discrete & Computational Geometry, 1989] and an improvement to linear time by Ahn, Barba, Bose, De Carufel, Korman, and Oh [Discrete & Computational Geometry, 2016].

The geodesic edge center can easily be found from the geodesic farthest-edge Voronoi diagram of the polygon. Finding that Voronoi diagram in linear time is an open question, although the geodesic *nearest* edge Voronoi diagram (the medial axis) can be found in linear time. As a first step of our geodesic edge center algorithm, we give a linear-time algorithm to find the geodesic farthest-edge Voronoi diagram restricted to the polygon boundary.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases geodesic center of polygon, farthest edges, farthest-segment Voronoi diagram

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.49

Related Version *Full Version:* <https://arxiv.org/abs/2303.09702>

Funding Research supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

Acknowledgements For helpful comments we thank Boris Aronov, Therese Biedl, John Hershberger, Joseph Mitchell, and the SoCG reviewers.

1 Introduction

The most basic “center” problem is Sylvester’s problem: given n points in the plane, find the smallest disc that encloses the points. The center of this disc is a point that minimizes the maximum distance to any of the given points. We consider a center problem that differs in two ways from Sylvester’s problem. First, the domain is a simple polygon and the distance measure is not Euclidean distance, but rather the shortest path, or “geodesic” distance inside the polygon. Second, the sites are not points but rather the edges of the polygon. More precisely, the problem is to find, given a simple polygon in the plane, the **geodesic edge center**, which is a point in the polygon that minimizes the maximum geodesic distance to a polygon edge. See Figure 1. More formally, let E be the set of edges of the polygon P , and for point $p \in P$ and edge $e \in E$, define $d(p, e)$ to be the geodesic distance from p to e . Define the **geodesic radius** of a point $p \in P$ to be $r(p) := \max\{d(p, e) : e \in E\}$. Then the **geodesic edge center** is a point $p \in P$ that minimizes $r(p)$.

Our main result is a linear-time algorithm to find the geodesic edge center of a simple n -vertex polygon. This improves our previous $O(n \log n)$ time algorithm [15]. The algorithm follows the strategies used to find the geodesic *vertex* center, which is a point in the polygon



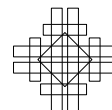
© Anna Lubiw and Anurag Murty Naredla;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

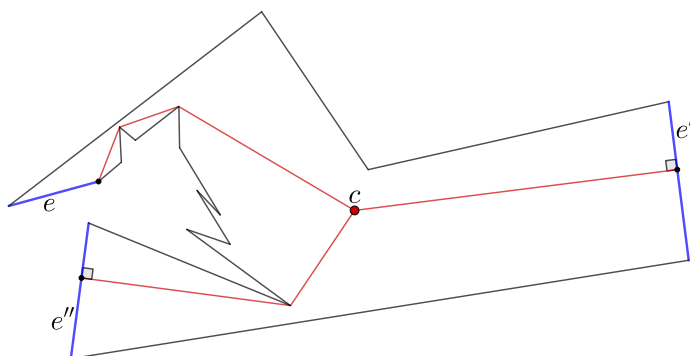
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 49; pp. 49:1–49:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Point c is the edge center of this polygon. Edges e, e', e'' (in blue) are geodesically farthest from c – the geodesic paths (in red) from c to these edges all have the same length.

that minimizes the maximum geodesic distance to a polygon vertex. In 1989, Pollack, Sharir and Rote [25] gave an $O(n \log n)$ time algorithm for the geodesic vertex center problem. A main tool – which is used in all subsequent algorithms – is a linear-time *chord oracle* that finds, given a chord, which side of the chord contains the center. In 2016, Ahn, Barba, Bose, De Carufel, Korman, and Oh [2] improved the runtime for the geodesic vertex center to $O(n)$. Their most important new contribution is the use of ϵ -nets to perform a divide-and-conquer search. Our algorithm follows the approach of Ahn et al., modified to deal with farthest edges rather than farthest vertices. We simplify some aspects and we repair some errors in their approach. The edge-center problem is more general than the vertex center problem via the reduction of splitting each vertex into two vertices joined by a very short edge.

In general, the center of a set of sites can be determined from the farthest Voronoi diagram of those sites, but computing the Voronoi diagram can be more costly. As the first step of our center algorithm we give a linear-time algorithm to compute the geodesic farthest-edge Voronoi diagram restricted to the boundary of the polygon. Computing the whole geodesic farthest-edge Voronoi diagram in linear time is an open problem.

Background on centers and farthest Voronoi diagrams. Megiddo [19] gave a linear-time algorithm to find the center of a set of points in the plane (Sylvester’s problem) using the “prune-and-search” technique (see also Dyer [11]), which is used in the final stages of all geodesic center algorithms. However, computing the farthest Voronoi diagram of points in the plane takes $\Theta(n \log n)$ time [26].

Our problem involves distances that are geodesic rather than Euclidean, and sites that are segments (edges) rather than points. These have been studied separately, although there is almost no work combining them.

For Euclidean distances, Megiddo’s method extends to linear-time algorithms to find the center of line segments or lines in the plane [6]. The farthest Voronoi diagram of segments in the plane was considered by Aurenhammer et al. [4], who called it a “stepchild in the vast Voronoi diagram literature”. They gave an $O(n \log n)$ time algorithm which was improved to output-sensitive time $O(n \log h)$, where h is the number of faces of the diagram [24].

For geodesic distances with point sites Ahn et al. gave a linear-time algorithm to find the geodesic center of the vertices of a polygon [2]. The corresponding farthest Voronoi diagram can be found in time $O(n \log \log n)$ [23], and in expected linear time [5]. More generally, for m points inside an n -vertex polygon, an algorithm to find their farthest Voronoi diagram was first given by Aronov et al. [3] with run-time $O((n + m) \log(n + m))$, and improved in a

sequence of papers [23, 5, 22], culminating in an optimal run time of $O(n + m \log m)$ [29]. This is also the best-known bound for finding the center of m points in a simple polygon. For sites more general than point sites inside a polygon, the only result we are aware of is our $O((n + m) \log(n + m))$ time algorithm to find the geodesic center of m *half-polygons* [15], with edges being a special case.

Finally, we mention a curious difference between nearest and farthest site Voronoi diagrams of edges in a polygon. The nearest Voronoi diagram of the edges of a polygon is the medial axis, one of the most famous and useful Voronoi diagrams. The medial axis can be found in linear time [9]. By contrast, the *farthest* Voronoi diagram of edges in a polygon has received virtually no attention, except for a convex polygon (which avoids geodesic issues) where there is an $O(n \log n)$ time algorithm [10], and a recent expected linear-time algorithm [14].

2 Overview of the algorithm

Before giving the overview of our algorithm, we outline the previous work that our algorithm builds upon, and explain what is novel about our contributions.

Pollack et al. [25] gave an $O(n \log n)$ time algorithm to find the geodesic vertex center of a simple polygon. A main ingredient is to solve the problem one dimension down. In particular, they develop an $O(n)$ time *chord oracle* that, given a chord of the polygon, finds the *relative center* restricted to the chord and from that, determines whether the center of the polygon lies to left or right of the chord. By applying the chord oracle $O(\log n)$ times, they limit the search to a convex subpolygon where Euclidean distances can be used. This reduces the problem to finding a minimum disc that encloses some disks, which Megiddo [20] solved in linear time using the same approach as for his linear programming algorithm. We extended the chord oracle to handle farthest *edges* instead of vertices [15].

The idea used in the chord oracle algorithm is central to further developments. Expressed in general terms, the goal is to find a point in a domain (either a chord or the whole polygon) that minimizes the maximum distance to a site (a vertex or edge of the polygon). The idea is to first find what we will call a *coarse cover* of the domain by a linear number of elementary regions R (intervals or triangles), each with an associated easy-to-compute convex function f_R that captures the geodesic distance to a potential farthest edge, and with the property that the upper envelope of the functions is the geodesic radius function. Thus, the goal is to find the point x that minimizes the upper envelope of the functions f_R . When the domain is a chord, the chord oracle solves this in linear time.

When the domain is the whole polygon, and the sites are vertices, Ahn et al. [2] gave a linear-time algorithm. They find a coarse cover of the whole polygon starting from Hershberger and Suri's algorithm [13] (based on matrix-searching techniques [1]) to find the farthest vertex from each vertex. They then use divide-and-conquer based on ϵ -nets – their big innovation – to reduce the domain to a triangle. After that, the vertex center is found using Megiddo-style prune-and-search techniques like those used by Pollack et al.

Our algorithm uses a similar approach, modified to deal with farthest *edges* rather than vertices. Another difference is that we give a simpler method of finding a coarse cover of the polygon by first finding the geodesic farthest-edge Voronoi diagram on the polygon boundary. There is a linear-time algorithm to find the geodesic farthest *vertex* Voronoi diagram on the polygon boundary by Oh, Barba, and Ahn [22]. Our algorithm is considerably simpler, and it is a novel idea to use the boundary Voronoi diagram to find the center.

Other differences between our approach and that of Ahn et al. are introduced in order to repair some flaws in their paper. They use ϵ -net techniques, but their range space does not have the necessary properties for finding ϵ -nets in deterministic linear time. We remedy this by using a different range space, thereby repairing and generalizing their result.

Algorithm overview**Phase I: Finding the farthest-edge Voronoi diagram restricted to the polygon boundary (Section 4)**

We first show that the linear-time algorithm of Hershberger and Suri [13] that finds the farthest *vertex* from each vertex can be modified to find the farthest *edge* from each vertex. A polygon edge e whose endpoints have the same farthest edge g is then part of the farthest Voronoi region of g . To find the Voronoi diagram on a *transition edge* e that has different farthest edges at its endpoints, we must find the upper envelope of the coarse cover of e . We use the fact that the coarse cover of e is constructed from two shortest path trees inside a smaller subpolygon called the *hourglass* of e . The hourglasses of all transition edges can be found in linear time. In each hourglass, the shortest path trees allow us to construct the upper envelope incrementally in linear time – this is a main new aspect of our work.

Phase II: Finding the geodesic edge center (Section 5)

We first find a coarse cover of the polygon by triangles, each bounded by two polygon chords plus a segment of an edge, and each with an associated convex function that captures the geodesic distance to a potential farthest edge – the potential farthest edges are those that have non-empty Voronoi regions on the boundary of P . The problem of finding the edge center is then reduced to the problem of finding the point that minimizes the upper envelope of the coarse cover functions.

To find this point we use divide-and-conquer, reducing in each step to a smaller subpolygon with a constant fraction of the coarse cover elements. There are two stages. In Stage 2, once the subpolygon is a triangle, the prune-and-search approach of Megiddo’s can be applied. In Stage 1 every coarse cover triangle that intersects the subpolygon has a boundary chord crossing the subpolygon, and ϵ -net techniques are used to reduce the number of such chords, and hence the number of coarse cover elements. Our approach follows that of Ahn et al. [2] but we repair some flaws – another main new aspect of our work. Ahn et al. recurse on subpolygons called “4-cells” that are the intersection of four half-polygons (a *half-polygon* is the subpolygon to one side of a chord). We instead recurse on “3-anchor hulls” that are the geodesic convex hulls of at most three points or subchains of the polygon boundary. These define a range space whose ground set is a set of chords and whose ranges are subsets of chords that cross a 3-anchor hull. We prove that our range space has finite VC-dimension, which repairs the faulty proof in Ahn et al. for 4-cells. Even more crucially, we give a “subspace oracle” that permits an ϵ -net to be found in *deterministic* linear time, something missing from their approach.

3 Preliminaries

Although our algorithm follows the pattern of the geodesic vertex center algorithm by Ahn et al. [2], we must re-do everything from the ground up to deal with farthest edges. In this section we summarize some basic results, deferring proofs and details to the full version [17].

Notation and definitions. A *chord* of a polygon is a straight line segment in the (closed) polygon with both endpoints on the boundary, ∂P . For a point $p \in P$ and a point or line segment s in P , $\pi(p, s)$ is the unique *shortest* (or *geodesic*) path from p to s , and $t(p, s)$ is its *terminal point*. The length of $\pi(p, s)$, denoted $d(p, s)$, is the *geodesic distance* from

p to s . For point p in P , a **farthest edge**, $F(p)$, is an edge for which $d(p, F(p)) \geq d(p, e)$, for every edge e of P . The **geodesic radius** of point p is $r(p) := d(p, F(p))$. The **geodesic edge center** is a point $p \in P$ that minimizes $r(p)$.

General position assumptions. As is standard for Voronoi diagrams of segments, e.g., see [4], we use the following tie-breaking rule to prevent 2-dimensional Voronoi regions with more than one farthest edge.

Tie-breaking rule. Suppose that p is a point of P , e and f are two edges that meet at reflex vertex u , and $\pi(p, e) = \pi(p, f) = \pi(p, u)$. Let line b be the angle bisector of u . For p not on b , break the tie $d(p, e) = d(p, f)$ by saying that the distance to the edge on the opposite side of b is greater.

We make the following general position assumptions, which we claim can be effected by perturbing vertices.

► **Assumptions 1.** (1) No three vertices of P are collinear. (2) After imposing the tie-breaking rule, no vertex is equidistant from two or more edges. (3) No point on the polygon boundary has more than two farthest edges and no point in the interior of the polygon has more than a constant number (six) of farthest edges.

It follows that the set of points with more than one farthest edge is 1-dimensional, does not contain any vertex of P , and intersects ∂P in isolated points; see the full version [17].

Properties of farthest edges. We need the following basic “triangle property” (proved in the full version [17]) about shortest paths that cross.

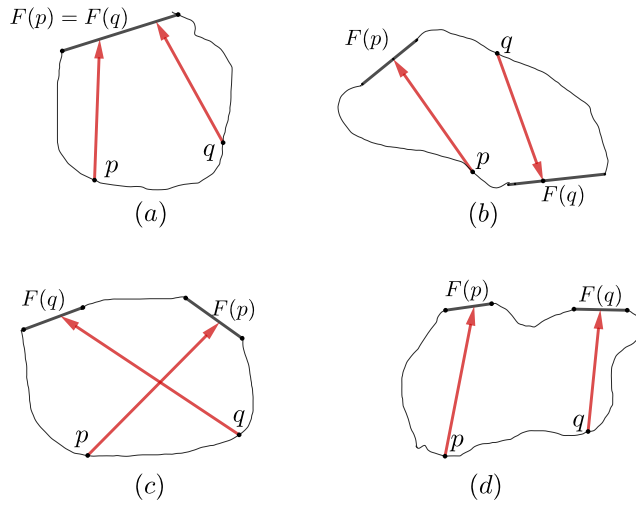
► **Lemma 1.** Suppose the points p, q and the edges e, f occur in the order p, q, e, f along the polygon boundary ∂P . Then $d(p, e) + d(q, f) \geq d(p, f) + d(q, e)$.

We then characterize what two paths to farthest edges are like, see Figure 2 and the full version [17]. In particular, we generalize the farthest-vertex Ordering Property [3] as follows.

The ordering property. As p moves clockwise around ∂P , so does $F(p)$.

Shortest paths to/from edges. As basic tools, we need linear-time algorithms to find shortest paths from a given point to all edges of the polygon, and to find shortest paths from a given edge to all vertices of the polygon. See the full version [17].

Separators and funnels. A geodesic path between two vertices of P separates ∂P into two parts, and when we focus on which vertices/edges are in opposite parts, we call the geodesic path a “separator”. Separators, first introduced by Suri [27], are a main tool for finding all farthest vertices in a polygon. In the full version [17] we extend the basic properties of separators to the case of farthest edges and prove: (1) If vertex v and edge e (e need not be farthest from v) are separated by a geodesic path $\pi(a, b)$, then the shortest path from v to e is contained, except for one edge, in the shortest path trees of a and b ; (2) A constant number of separators suffice to separate every vertex from its farthest edge.



■ **Figure 2** Schematics for possible and impossible orderings of points p, q and their farthest edges $F(p)$ and $F(q)$. (a) The only possible ordering if $F(p) = F(q)$. (b),(c) The two possible orderings if $F(p) \neq F(q)$. (d) The impossible ordering if $F(p) \neq F(q)$.

3.1 Chord oracles and coarse covers

In this section we describe the chord oracle results that we need from previous work, and we give a unified explanation of those algorithms and our current algorithm in terms of coarse covers. The basic function of a chord oracle is to decide, given a chord K , whether the center lies to the left or right (or on) the chord. Pollack et al. [25] gave a linear-time chord oracle for the geodesic vertex center, which is at the heart of all further geodesic center algorithms. We extended the chord oracle to the case of the geodesic *edge* center [15].

In both cases, a main step is the “one-dimension down” problem of finding the *relative center*, which is a point c_K on K that minimizes the geodesic radius function $r(x)$. The directions of the first segments of the paths from c_K to its farthest sites determine whether the center of P lies left/right/on K (see the full version [17]).

Algorithms to find the relative center of a chord or the center of a polygon rely on a basic convexity property of the geodesic radius function (see the full version [17]) and all follow the same pattern, which can be formalized via the concept of a *coarse cover* of the chord/polygon. The idea is that a *coarse cover* for a domain (a chord/polygon) is a set of elementary regions R (intervals/triangles) covering the domain, where each region R has an associated easy-to-compute convex function f_R , such that the upper envelope of the f_R 's is the geodesic radius function. We give a precise definition for the case of farthest edges (following [15] and specialized for our Assumptions 1).

► **Definition 2.** A *coarse cover* of chord K [or polygon P] is a set of triples (R, f, e) where

1. R is a subinterval of K [or a triangle of P], f is a function defined on domain R , and e is an edge of P .
2. For all $x \in R$, $f(x) = d(x, e)$ and either: $f(x) = d_2(x, v) + \kappa$ where d_2 is Euclidean distance, κ is a constant and v is a vertex of P ; or $f(x) = d_2(x, \bar{e})$, where d_2 is Euclidean distance and \bar{e} is the line through e .
3. For any point $x \in K$ [or P], and any edge e that is farthest from x , there is a triple (R, f, e) in the coarse cover with $x \in R$.

Condition (3) implies that the upper envelope of the functions of the coarse cover is the geodesic radius function. Thus the [relative] center problem breaks into two subproblems: (1) find a coarse cover; and (2) find the point x that minimizes the upper envelope of the coarse cover functions. The high-level idea for solving step (2) in linear time (for a chord or polygon domain) is to recursively reduce the domain (the search space) to a subinterval or subpolygon while eliminating elements of the coarse cover whose functions are strictly dominated by others. As for step (1) – constructing a coarse cover – see Section 4 for a chord and Section 5 for a polygon.

We call the chord oracle in Phase II when we use divide-and-conquer to search for the center in successively smaller subpolygons. We actually need two variations of the basic chord oracle. First, we need a *geodesic oracle* that tests which side of a geodesic contains the center. Secondly, we do not construct a coarse cover of a chord/geodesic from scratch; rather, we intersect the triangles of the coarse cover of the subpolygon with the chord/geodesic, thus avoiding runtime dependence on n . These variations are described in the full version [17].

4 Phase I: Finding the farthest-edge Voronoi diagram restricted to the polygon boundary

Based on Assumptions 1, the boundary of P consists of chains with a single farthest edge, separated by points (not vertices) that have two farthest edges (see Figure 4). Our goal is to find these points. The first step of the algorithm is to find the farthest edge from each vertex of the polygon in linear time. To do this, we extend the algorithm of Hershberger and Suri [13] that finds the farthest *vertex* from each vertex. Details are in the full version [17]. The next step is to fill in the Voronoi diagram along the polygon edges. For an edge ab where vertices a and b have the same farthest edge, i.e., $F(a) = F(b)$, all points on the edge ab have the same farthest edge, by the Ordering Property. An edge ab with $F(a) \neq F(b)$ is a **transition edge**. We will find the farthest-edge Voronoi diagram on one transition edge in linear time. To handle *all* the transition edges in linear time, we will show that for each transition edge ab we can restrict our attention to the **hourglass** $H(a, b)$ which is the subpolygon of P bounded by ab , $\pi(a, F(b))$, $\pi(b, F(a))$ and the portion of ∂P between the terminals $t(a, F(b))$ and $t(b, F(a))$. In the full version [17] we show that the hourglasses of all transition edges can be found in linear time and that the sum of their sizes is linear.

In this section, we show how to construct the farthest-edge Voronoi diagram along one polygon edge ab in time linear in the size of the polygon. We do not assume that the polygon is an hourglass. For purposes of description, imagine ab horizontal with a at the left, and the polygon interior above ab . We use the **coarse cover** (Definition 2) of the edge ab , which can be found in linear time (Lubiw and Naredla [16]). Elements of the coarse cover are triples (I, f, e) where I is a subinterval of ab and $f(x) = d(x, e)$ for any $x \in I$. By resolving overlaps of coarse cover intervals I , we find the upper envelope of the coarse cover functions f , which immediately gives the Voronoi diagram on ab . This is easy if we sort the endpoints of the intervals I , but we cannot afford to sort. Instead, we will insert the coarse cover elements one by one, maintaining a list M of [pairwise internally] disjoint subintervals of ab together with an associated distance function $f_M(x)$. An efficient insertion order depends on the fact that elements of the coarse cover of edge ab are associated with edges of the shortest path trees T_a and T_b (that consist of the shortest paths from a and b , respectively, to all the edges of P). We will use the ordering of the trees as embedded in the plane.

Oh, Barba, Ahn [23] gave a linear-time algorithm to find the farthest *vertex* Voronoi diagram on the boundary of P . The approach is similar, but they add coarse cover elements by iterating over the sites (the vertices in their case), which involves a complicated algorithm to sweep back and forth along M maintaining a shortest path to the current vertex, and a tricky amortized analysis (see [23, Lemma 7]). Our approach is simpler and more general.

4.1 Farthest-edge Voronoi diagram on one edge

In previous work [15, 16] we constructed a coarse cover (see Definition 2) of an edge ab from the shortest path trees T_a and T_b . The trees are first augmented with 0-length edges so that the paths to every polygon edge e end with a tree edge perpendicular to e . In particular, every polygon edge corresponds to a leaf in each tree.

Direct edges of T_a and T_b away from their roots. Each edge uv of $T_a \setminus T_b$ with $u \neq a$ corresponds to an ***a-side*** coarse cover element (I, f, e) where e corresponds to the farthest leaf of T_a descended from v . For example, in Figure 3, see edge a_3 of T_a and interval I_{a_3} . There are symmetrically defined ***b-side*** coarse cover elements. Each edge uv of $T_a \cap T_b$ with u visible from ab corresponds to a ***central triangle*** coarse cover element (I, f, e) where e corresponds to the farthest leaf of T_a descended from v . For example, see edge $a_5 = b_5$ and interval I_{a_5} . Each polygon edge e that has an interior point visible from ab corresponds to a ***central trapezoid*** coarse cover element (I, f, e) where I consists of the points on ab whose shortest paths to e arrive perpendicularly. For example, see edge e_4 and interval I_{e_4} .

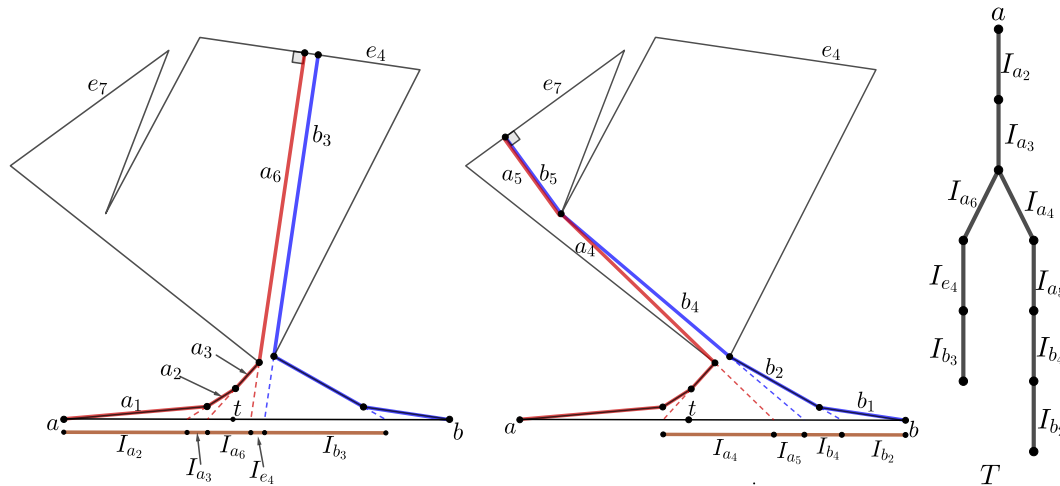
► **Lemma 3** (proved in the full version [17]). *For any edge e of P , let $C(e)$ be the set of coarse cover elements (I, f, e) for e . If $C(e)$ is nonempty, then its elements correspond to a (possibly empty) path in T_a directed towards a leaf, followed by a central triangle or trapezoid, followed by a (possibly empty) path in T_b directed towards the root. Furthermore, the corresponding intervals on ab appear in order, are [internally] disjoint, and their union is an interval.*

We next construct a single tree T whose edges correspond to coarse cover elements of ab . Then we incrementally construct the farthest-edge Voronoi diagram on ab by adding coarse cover elements in a depth first search (DFS) order of T .

Constructing tree T . Starting with T_a , attach an edge for each central trapezoid element to the associated leaf vertex of T_a ; add the path of b -side triangle elements for each polygon edge e after the central triangle or trapezoid for e ; and contract original edges of T_a that are not associated with coarse cover elements. See Figure 3(right). We give more detail of these steps in the full version [17]. The resulting tree T can be constructed in linear time and its edges are in one-to-one correspondence with the coarse cover elements.

► **Observation 4.** *If uv and vw are edges of T , then the corresponding coarse cover intervals I_1 and I_2 appear in that order along ab and intersect in a single point.*

DFS algorithm for the Voronoi diagram. We add the coarse cover elements following a DFS of T with children of a node in clockwise order. We maintain a list M of interior disjoint subintervals of ab whose union is an interval starting at a . Each subinterval in M records the coarse cover element it came from. Define f_M to be the distance function determined by the intervals of M . Initially, M is the single point a , and f_M is $-\infty$. At the end M will be the upper envelope of the coarse cover functions (though this property is not guaranteed throughout). To handle edge uv of T with associated coarse cover element (I, f, e) , we



■ **Figure 3** Coarse cover elements corresponding to some (not all) edges of T_a (red) and T_b (blue): (left) coarse cover elements for e_4 ; (middle) coarse cover elements for e_7 ; (right) the corresponding part of tree T . When I_{a_6} is handled by **Insert** it wins the comparison with I_{a_4} so it replaces I_{a_4} up to the cross-over point t , and the algorithm discards the rest of I_{a_6} , together with I_{e_4} and I_{b_3} .

compare f to f_M beginning at the left endpoint of I . We maintain a pointer p_u that gives an interval of M containing this endpoint. The recursive routine **Insert**(u, p_u) inserts into M the portions of coarse cover elements that are associated with u 's subtree and that define the upper envelope. At the top level, we call **Insert**(a, p_a), where p_a points to a .

```

Insert( $u, p_u$ )  #  $u$  is a node of  $T$  and  $p_u$  is a pointer to an interval of  $M$ 
  for each child  $v$  of  $u$  in clockwise order do
    ( $I, f, e$ ) := the coarse cover element associated with the edge  $uv$  of  $T$ 
     $l$  := left endpoint of  $I$ ;  $r$  := right endpoint of  $I$ 
    Invariant:  $p_u$  points to an interval of  $M$  that contains  $l$ 
    if  $f(l^+) > f_M(l^+)$  where  $l^+$  is just to the right of  $l$  then
      replace intervals of  $M$  starting at  $p_u$  with a subinterval of  $I$  ending at
      the ‘‘cross-over’’ point  $t < r$  where  $f_M$  starts to dominate  $f$ , or at  $r$ 
    if  $f$  dominates until  $r$  and  $v$  is not a leaf of  $T$  then
      call Insert( $v, p_v$ ), where  $p_v$  is a pointer to interval  $I$  in  $M$ 
  
```

Runtime. Each edge of T is handled once, and causes at most one new interval to be inserted into M , so the total number of endpoints inserted into M is $O(n)$. We can access $f_M(l^+)$ in constant time using the pointer p_u . Then the endpoints of intervals of M that we traverse as we do the insertion vanish from M . Thus the runtime is $O(n)$.

Correctness. The following lemma implies that the final M is the upper envelope of the coarse cover functions.

► **Lemma 5.** *The algorithm only discards pieces of coarse cover elements that do not form part of the final upper envelope.*

Proof. We examine the behaviour of the algorithm for edge uv of T with associated coarse cover element (I, f, e) , where $I = [l, r]$. We insert the subinterval $[l, t]$ into M (or no subinterval). Because $f(x) \geq f_M(x)$ for $x \in [l, t]$, any subintervals of M that are removed due to the insertion do not determine the upper envelope, so their removal is correct.

49:10 The Geodesic Edge Center of a Simple Polygon

If we insert all of interval I into M and recursively call $\text{Insert}(v, p_v)$, then this is correct by induction. So suppose we insert a proper subinterval of I or none of I . We must prove that no later part of I , and no element of the coarse cover associated with edges of the subtree rooted at v determines the upper envelope. Let t^+ be a point just to the right of t (or just to the right of l if we insert no part of I). Then $f_M(t^+) > f(t^+)$. Number the polygon edges e_1, e_2, \dots, e_m clockwise from a to b . Suppose that $e = e_i$, so $f(x) = d(x, e_i)$ for $x \in I$, in particular, $f(t^+) = d(t^+, e_i)$. Suppose that $f_M(t^+) = d(t^+, e_k)$. Then $d(t^+, e_k) > d(t^+, e_i)$.

Now consider the edges of T descended from v plus the edge uv . Consider the corresponding coarse cover elements, C_v , and let e_j be any polygon edge associated with any element in C_v . Note that the intervals on ab associated with coarse cover elements of C_v lie to the right of r , except for I associated with uv . We will prove that for any point $x \in ab$ to the right of t^+ , $d(x, e_k) > d(x, e_j)$, which implies that none of the coarse cover elements in C_v determines the upper envelope, nor does any part of I to the right of t . Thus the algorithm is correct to discard them.

We first prove the result for $x = t^+$. If uv corresponds to a central triangle/trapezoid for e_i or a b -side triangle, then T has a single path descending from v , all of whose edges are associated with e_i , i.e., $j = i$. Otherwise, by the definition of an a -side coarse cover element, e_i corresponds to the farthest leaf of T_a descended from v , which implies that $d(x, e_i) \geq d(x, e_j)$ for all $x \in I$, and in particular for $x = t^+$. Thus, in either case we have $d(t^+, e_k) > d(t^+, e_i) \geq d(t^+, e_j)$.

We next claim that $k < j$. The current f_M values arise from tree edges already processed. These consist of: (1) edges on the path from a to u ; and (2) edges of T counterclockwise from this path. Edges on the path from a to u have coarse cover intervals on ab to the left of l , by Observation 4. Thus type (1) edges do not determine $f_M(t^+)$. By the depth-first-search order, type (2) edges have coarse cover elements corresponding to polygon edges counterclockwise from e_j . Thus $k < j$.

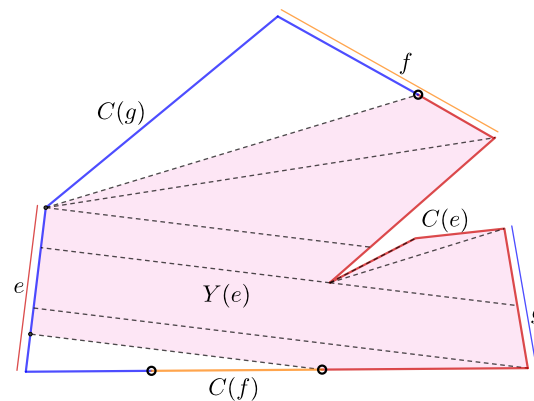
To complete the proof of Lemma 5, consider any point $x \in ab$ to the right of t^+ . The clockwise ordering around the polygon boundary is x, t^+, e_k, e_j , so by Lemma 1 and the fact that $d(t^+, e_k) > d(t^+, e_j)$, we get $d(x, e_k) > d(x, e_j)$, as required. ◀

5 Phase II: Finding the geodesic edge center

The first step of Phase II is to construct a **coarse cover** (Definition 2) of the polygon in linear time. As shown in Figure 4 the **funnel** $Y(e)$ that consists of shortest paths between a chain on ∂P with farthest edge e and e itself can be partitioned into its shortest path map. If the result includes trapezoids, we partition each one into two triangles¹. Each triangle is bounded by two polygon chords and a segment of a polygon edge, and the distance to e has the form required by Definition 2 (see the full version [17]). We seek the point inside P that minimizes the upper envelope of the functions of the coarse cover. Note that Phase I can detect if the edge center lies *on* ∂P so we may assume that the center is interior to P .

Our final divide-and-conquer algorithm follows the vertex center algorithm of Ahn et al. [2], generalized to farthest edges, and repairing flaws in their approach. At each step of the algorithm we have a subpolygon Q whose interior contains the center together with the coarse cover elements needed to compute the edge center and we shrink the subpolygon and eliminate a constant fraction of the coarse cover. Each recursive step takes time linear in the size of the subproblem (the size of Q plus the size of its coarse cover). The subpolygons

¹ Thus our triangles are not necessarily “apexed” triangles as in [2].



■ **Figure 4** The farthest-edge Voronoi diagram restricted to the polygon boundary consists of chains $C(e)$, $C(f)$, $C(g)$ farthest from edges e , f , g , respectively. To construct the coarse cover, the funnel $Y(e)$ (shaded) is partitioned (by dashed segments) into a shortest path map from e to $C(e)$.

we work with are **3-anchor hulls** defined as follows (see Figure 6). An **anchor** is a point inside P , or a subchain of ∂P . A **3-anchor hull** is the geodesic convex hull of at most three anchors. These are weakly simple in general, but we only recurse on **simple** 3-anchor hulls.

The algorithm has two stages. In Stage 1 no triangle of the coarse cover contains Q (this is true initially when $Q = P$), so every triangle has a chord crossing Q and we use ϵ -net techniques on the set of such chords to reduce to a smaller cell Q' that is crossed by a fraction of the chords, and hence by a fraction of the coarse cover triangles. Once Q is contained in a triangle of the coarse cover we show (see the full version [17]) that the size of Q , denoted $|Q|$, is at most 6. In fact, we will exit Stage 1 as soon as $|Q| \leq 6$. It is then easy to reduce Q to a triangle. After that, we switch to Stage 2, where the convexity of Q allows us to use a Megiddo-style prune-and-search technique (as Ahn et al. do) to recursively reduce the size of the subproblem. Stage 2 is deferred to the full version [17].

5.1 Stage 1: Algorithm for large Q

Consider a subproblem corresponding to a simple 3-anchor hull Q with $|Q| > 6$. We give an algorithm that either finds the edge center or reduces to a subproblem with $|Q| \leq 6$, which is handled by Stage 2. In Stage 1, no triangle of the coarse cover contains Q (as proved in the full version [17]) so each one has a chord crossing Q – we denote this set of chords by $\mathcal{K}(Q)$.

To apply ϵ -net techniques we define a **3-anchor range space** as follows. The ground set is a set \mathcal{K} of chords of P , and for each 3-anchor hull H of P there is a range $\mathcal{K}(H)$ consisting of all chords of \mathcal{K} that **cross** H . Here a chord **crosses** a set if both open half-polygons of the chord contain points of the set.

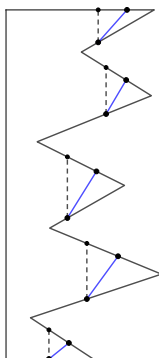
The algorithm finds a constant size ϵ -net of the 3-anchor range space on $\mathcal{K}(Q)$, which is a set $N \subseteq \mathcal{K}(Q)$ such that any 3-anchor hull not intersected by a chord of N is intersected by only a constant fraction of the chords of $\mathcal{K}(Q)$ – this is the important property that allows us to discard a fraction of the chords. The set of chords N forms an arrangement that partitions Q into cells. We use the chord oracle to determine which cell contains the center. We then add geodesic paths to subdivide this cell into a constant number of 3-anchor hulls and use a **geodesic oracle** (see the full version [17]) to find which 3-anchor hull contains the center, and to shrink it to a simple 3-anchor hull Q' . The algorithm recurses on Q' , whose coarse cover is a fraction of the size.

More details of the algorithm can be found in the full version [17]. For now, we expand on the aspects of the algorithm that differ from the approach of Ahn et al. [2]. Instead of 3-anchor hulls, their algorithm works with 4-cells, formed by taking the intersection of at most four half-polygons, where a half-polygon is the part of P to one side of a chord. The number (three versus four) is not significant, but we bound our regions by geodesics instead of chords in order to obtain the following two results.

1. The 3-anchor range space has finite VC-dimension. This implies that constant-sized ϵ -nets exist. Furthermore, there is a “subspace oracle” that allows us to find an ϵ -net N in deterministic linear time [28, Chapter 47, Theorem 47.4.3]. For further background see the full version [17], and the paper by Chazelle and Matoušek [8].

Ahn et al. claim that their range space (of chords crossing 4-cells) has finite VC-dimension but their proof is flawed. Our proof shows that their range space does in fact have finite VC-dimension. They do not mention subspace oracles, without which their algorithm runs in expected linear time rather than deterministic linear time as claimed. We expand on these aspects in Section 5.2 below.

2. A cell of the arrangement of N can be partitioned into constantly many 3-anchor hulls. The method used by Ahn et al. to subdivide a cell of N into 4-cells by adding a constant number of chords is incomplete, see Figure 5. We see how to repair their partition step but we find 3-anchor hulls more natural.



■ **Figure 5** Ahn et al. [2] subdivide a cell of N by adding vertical chords (dashed) at endpoints and intersection points of chords of N (blue), which leaves a 5-cell in this example.

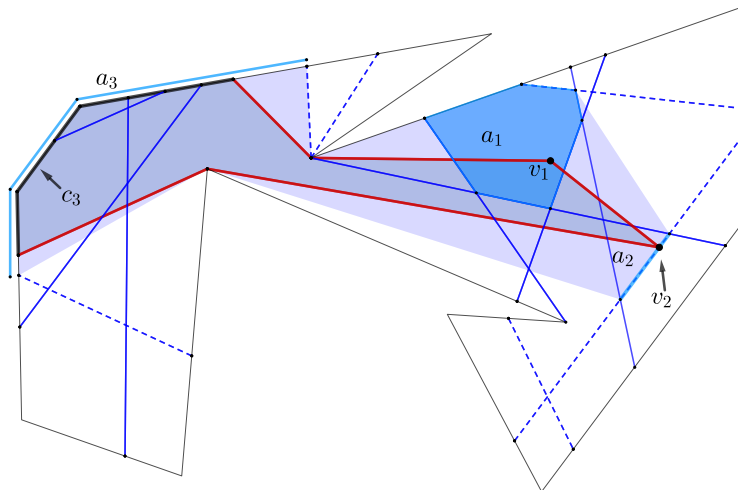
5.2 Epsilon-net results for Stage 1

In this section we expand on the ϵ -net results needed for Stage 1 of the algorithm as described above. We also give details of the flaws in the approach of Ahn et al. [2]. For an overview of ϵ -nets as used for geometric divide-and-conquer, see the full version [17] (also see Chazelle [7] or Mustafa [21]). To show that ϵ -nets of constant size exist we need the following result.

► **Lemma 6.** *The 3-anchor range space has VC-dimension less than 259.*

Our proof of Lemma 6 works equally well for 4-anchor hulls – the bound becomes 372. A 4-cell is a special case of a 4-anchor hull so our proof implies finite VC-dimension (≤ 372) for the 4-cell range space, which repairs the claim by Ahn et al.² We explain the flaw in

² In response to our enquiries, Eunjin Oh independently suggested a similar remedy.



■ **Figure 6** A simple 3-anchor hull Q (outlined in red) with anchors v_1, v_2 and the polygon chain c_3 (thick black). Solid blue chords cross Q , while dashed blue chords do not. The expanded 3-anchor hull $\psi(Q)$ (lightly shaded) has expanded anchors: a_1 , the face in the chord arrangement containing v_1 ; a_2 , the edge with v_2 in its interior; and a_3 , the polygon chain extending c_3 to chord endpoints. The same chords cross Q and $\psi(Q)$.

their proof. Let us refer to the set of chords intersecting a 4-cell as a “4-cell range”. Ahn et al. prove that the 1-cell range space has VC-dimension at most 65,535. They note that a 4-cell is the intersection of four 1-cells, and then claim in their Lemma 9.1 that this implies finite VC-dimension for the 4-cell range space. As justification, they refer to Proposition 10.3.3 of Matousek’s text [18], which states that the VC-dimension is bounded for any family whose sets can be defined by a formula of Boolean connectives (union, intersection, set difference). However, Matousek’s proposition cannot be applied in this situation because, although a 4-cell is the intersection of four 1-cells, it is not true that a 4-cell *range* is the intersection of four 1-cell *ranges*. In particular, a chord can intersect two 1-cells, but not intersect the intersection of the two 1-cells. For example, a line of slope -1 can intersect the $+x$ half-plane and the $+y$ half-plane without intersecting the $+x, +y$ quadrant.

Proof of Lemma 6. We will prove that the shattering dimension is 6 and then apply the result that a range space with shattering dimension d has VC-dimension bounded by $12d \ln(6d)$ (Lemma 5.14 from Har-Peled [12]). For $d = 6$, this is less than 259.

We must show that for a set \mathcal{K} of chords with $|\mathcal{K}| = m$, the number of distinct ranges is $O(m^6)$. We prove that the range space for \mathcal{K} is the same if we replace 3-anchor hulls by “expanded 3-anchor hulls” that are defined in terms of \mathcal{K} , more precisely, in terms of the arrangement $A(\mathcal{K})$ of the chords \mathcal{K} plus the edges of P . Define an *expanded anchor* to be an internal face, edge, or vertex of $A(\mathcal{K})$, or a polygon chain with endpoints in $V(\mathcal{K})$, the set of endpoints of chords \mathcal{K} . An *expanded 3-anchor hull* is the geodesic convex hull of at most three expanded anchors.

► **Lemma 7.** *The set of ranges $\mathcal{R} = \{\mathcal{K}(Q) \mid Q \text{ is a 3-anchor hull}\}$ is the same as the set of ranges $\overline{\mathcal{R}} = \{\mathcal{K}(Q) \mid Q \text{ is an expanded 3-anchor hull}\}$.*

Proof. To prove $\mathcal{R} \subseteq \overline{\mathcal{R}}$, consider a 3-anchor hull Q . Replace any point anchor p by the smallest (by containment) internal vertex, edge, or face of $A(\mathcal{K})$ that contains p . See Figure 6. Replace any polygon chain anchor C by the smallest chain of ∂P containing C and with

endpoints in $V(\mathcal{K})$. Let $\psi(Q)$ be the geodesic convex hull of these expanded anchors. Then $\psi(Q)$ is an expanded 3-anchor hull that contains Q , and it is straight-forward to prove that $\mathcal{K}(Q) = \mathcal{K}(\psi(Q))$ (see the full version [17]).

For the other direction, let Q be an expanded 3-anchor hull. Replace an expanded anchor that is a face, edge, or vertex of $A(\mathcal{K})$ by a point anchor in the interior of that face, edge, or vertex. An expanded anchor that is a polygon chain remains unchanged. Let $\gamma(Q)$ be the geodesic convex hull of the resulting anchors. Observe that $\gamma(Q)$ is a 3-anchor hull and $\psi(\gamma(Q)) = Q$. As above, this implies that $\mathcal{K}(Q) = \mathcal{K}(\gamma(Q))$. ◀

To complete the proof of Lemma 6 we claim that the number of expanded 3-anchor hulls of \mathcal{K} is $O(m^6)$. An expanded anchor may be an internal vertex, edge, or face of $A(\mathcal{K})$, of which there are $O(m^2)$ possibilities. Otherwise, an expanded anchor is a chain of ∂P between vertices of $V(\mathcal{K})$, also with $O(m^2)$ possibilities. Thus the number of expanded 3-anchor hulls is $O((m^2)^3) = O(m^6)$. ◀

Subspace oracle

To prove that the 3-anchor range space has a subspace oracle, we present a deterministic algorithm that, given a subset $\mathcal{K}' \subseteq \mathcal{K}$ with $|\mathcal{K}'| = m$, computes the set of ranges $\mathcal{R} = \{\mathcal{K}'(Q) \mid Q \text{ is a 3-anchor hull}\}$ in time $O(m^7)$. The idea is to use Lemma 7 and to construct $A(\mathcal{K}')$ minus the edges of P , and find, for each chord $K \in \mathcal{K}'$, which of the $O(m^2)$ expanded anchors in $A(\mathcal{K}')$ intersects each side of K , and then, for each of the $O(m^6)$ expanded 3-anchor hulls, eliminate the chords that have all three expanded anchors to one side, leaving the chords that cross the hull. For further details see [17].

References

- 1 Alok Aggarwal, Maria M Klawe, Shlomo Moran, Peter Shor, and Robert Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2(1-4):195–208, 1987. doi:10.1007/bf01840359.
- 2 Hee-Kap Ahn, Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, Matias Korman, and Eunjin Oh. A linear-time algorithm for the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 56(4):836–859, 2016. doi:10.1007/s00454-016-9796-0.
- 3 Boris Aronov, Steven Fortune, and Gordon Wilfong. The furthest-site geodesic Voronoi diagram. *Discrete & Computational Geometry*, 9(3):217–255, 1993. doi:10.1007/bf02189321.
- 4 Franz Aurenhammer, Robert L Scot Drysdale, and Hannes Krasser. Farthest line segment Voronoi diagrams. *Information Processing Letters*, 100(6):220–225, 2006. doi:10.1016/j.ipl.2006.07.008.
- 5 Luis Barba. Optimal algorithm for geodesic farthest-point Voronoi diagrams. In *35th International Symposium on Computational Geometry (SoCG 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019. doi:10.4230/LIPIcs.SocG.2019.12.
- 6 Binay K Bhattacharya, Shreesh Jadhav, Asish Mukhopadhyay, and J-M Robert. Optimal algorithms for some intersection radius problems. *Computing*, 52(3):269–279, 1994. doi:10.1007/bf02246508.
- 7 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete & Computational Geometry*, 9(2):145–158, 1993. doi:10.1007/bf02189314.
- 8 Bernard Chazelle and Jiří Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *Journal of Algorithms*, 21(3):579–597, 1996. doi:10.1006/jagm.1996.0060.
- 9 Francis Chin, Jack Snoeyink, and Cao An Wang. Finding the medial axis of a simple polygon in linear time. *Discrete & Computational Geometry*, 21(3):405–420, 1999. doi:10.1007/p100009429.

- 10 R L Scot Drysdale and Asish Mukhopadhyay. An $O(n \log n)$ algorithm for the all-farthest-segments problem for a planar set of points. *Information Processing Letters*, 105(2):47–51, 2008. doi:10.1016/j.ipl.2007.08.004.
- 11 Martin E Dyer. Linear time algorithms for two- and three-variable linear programs. *SIAM Journal on Computing*, 13(1):31–45, 1984. doi:10.1137/0213003.
- 12 Sariel Har-Peled. *Geometric Approximation Algorithms*. American Mathematical Society, 2011. doi:10.1090/surv/173.
- 13 John Hershberger and Subhash Suri. Matrix searching with the shortest-path metric. *SIAM Journal on Computing*, 26(6):1612–1634, 1997. doi:10.1137/s0097539793253577.
- 14 Elena Khramtcova and Evanthia Papadopoulou. An expected linear-time algorithm for the farthest-segment Voronoi diagram. arXiv, 2014. doi:10.48550/arxiv.1411.2816.
- 15 Anna Lubiw and Anurag Murty Naredla. The visibility center of a simple polygon. arXiv, 2021. doi:10.48550/arxiv.2108.07366.
- 16 Anna Lubiw and Anurag Murty Naredla. The visibility center of a simple polygon. In Petra Mutzel, Rasmus Pagh, and Grzegorz Herman, editors, *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 65:1–65:14, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ESA.2021.65.
- 17 Anna Lubiw and Anurag Murty Naredla. The geodesic edge center of a simple polygon. arXiv, 2023. doi:10.48550/arXiv.2303.09702.
- 18 Jiří Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer Verlag, 2002. doi:10.1007/978-1-4613-0039-7.
- 19 Nimrod Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983. doi:10.1137/0212052.
- 20 Nimrod Megiddo. On the ball spanned by balls. *Discrete & Computational Geometry*, 4(6):605–610, 1989. doi:10.1007/bf02187750.
- 21 Nabil H Mustafa. *Sampling in Combinatorial and Geometric Set Systems*, volume 265 of *Mathematical Surveys and Monographs*. American Mathematical Society, 2022. doi:10.1090/surv/265.
- 22 Eunjin Oh and Hee-Kap Ahn. Voronoi diagrams for a moderate-sized point-set in a simple polygon. *Discrete & Computational Geometry*, 63(2):418–454, 2020. doi:10.1007/s00454-019-00063-4.
- 23 Eunjin Oh, Luis Barba, and Hee-Kap Ahn. The geodesic farthest-point Voronoi diagram in a simple polygon. *Algorithmica*, 82(5):1434–1473, 2020. doi:10.1007/s00453-019-00651-z.
- 24 Evanthia Papadopoulou and Sandeep Kumar Dey. On the farthest line-segment Voronoi diagram. *International Journal of Computational Geometry & Applications*, 23(06):443–459, 2013. doi:10.1007/978-3-642-35261-4_22.
- 25 Richard Pollack, Micha Sharir, and Günter Rote. Computing the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 4(6):611–626, 1989. doi:10.1007/bf02187751.
- 26 Michael Ian Shamos and Dan Hoey. Closest-point problems. In *16th Annual Symposium on Foundations of Computer Science (FOCS 1975)*, pages 151–162. IEEE, 1975. doi:10.1109/sfcs.1975.8.
- 27 Subhash Suri. Computing geodesic furthest neighbors in simple polygons. *Journal of Computer and System Sciences*, 39:220–235, 1989. doi:10.1016/0022-0000(89)90045-7.
- 28 Csaba D Toth, Joseph O’Rourke, and Jacob E Goodman, editors. *Handbook of Discrete and Computational Geometry*. CRC press, 2017. doi:10.1201/9781315119601.
- 29 Haitao Wang. An optimal deterministic algorithm for geodesic farthest-point Voronoi diagrams in simple polygons. *Discrete & Computational Geometry*, 2022. doi:10.1007/s00454-022-00424-6.

A Structural Approach to Tree Decompositions of Knots and Spatial Graphs

Corentin Lunel ✉

LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, F-77454 Marne-la-Vallée, France

Arnaud de Mesmay ✉

LIGM, CNRS, Univ. Gustave Eiffel, ESIEE Paris, F-77454 Marne-la-Vallée, France

Abstract

Knots are commonly represented and manipulated via diagrams, which are decorated planar graphs. When such a knot diagram has low treewidth, parameterized graph algorithms can be leveraged to ensure the fast computation of many invariants and properties of the knot. It was recently proved that there exist knots which do not admit any diagram of low treewidth, and the proof relied on intricate low-dimensional topology techniques. In this work, we initiate a thorough investigation of tree decompositions of knot diagrams (or more generally, diagrams of spatial graphs) using ideas from structural graph theory. We define an obstruction on spatial embeddings that forbids low tree width diagrams, and we prove that it is optimal with respect to a related width invariant. We then show the existence of this obstruction for knots of high representativity, which include for example torus knots, providing a new and self-contained proof that those do not admit diagrams of low treewidth. This last step is inspired by a result of Pardon on knot distortion.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Geometric topology

Keywords and phrases Knots, Spatial Graphs, Tree Decompositions, Tangle, Representativity

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.50

Related Version *The full version of this article is available on arXiv:* <https://arxiv.org/abs/2303.07982> [31]

Funding This work was partially supported by the ANR project SoS (ANR-17-CE40-0033).

Acknowledgements We would like to thank Pierre Dehornoy and Saul Schleimer for helpful discussions, and the anonymous reviewers for their feedback which allowed us to significantly improve the paper.

1 Introduction

A (tame) **knot** is a polygonal embedding of the circle S^1 into \mathbb{R}^3 , and two knots are considered equivalent if they are **isotopic**, i.e., if they can be continuously deformed one into the other without introducing self-intersections. The **trivial knot**, or **unknot**, is, up to equivalence, the embedding of S^1 as a triangle. The investigation of knots and their mathematical properties dates back to at least the nineteenth century [1] and has developed over the years into a very rich and mature mathematical theory. From a computational perspective, a fundamental question is to figure out the best algorithm testing whether a given knot is the unknot. Note that it is neither obvious from the definitions that a non-trivial knot exists, nor that the problem is decidable. This was famously posed as an open problem by Turing [50]. The current state of the art on this problem is that it lies in **NP** [21] and **co-NP** [29], a quasipolynomial time algorithm has been announced [30] but no polynomial-time algorithm is known. More generally, algorithmic questions surrounding knots typically display a wide gap between the best known algorithms (which are almost never polynomial-time, and sometimes the complexity is a tower of exponentials) and the best known complexity lower bounds. We refer to the survey of Lackenby for a panorama of algorithms in knot theory [28].



© Corentin Lunel and Arnaud de Mesmay;

licensed under Creative Commons License CC-BY 4.0

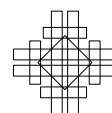
39th International Symposium on Computational Geometry (SoCG 2023).

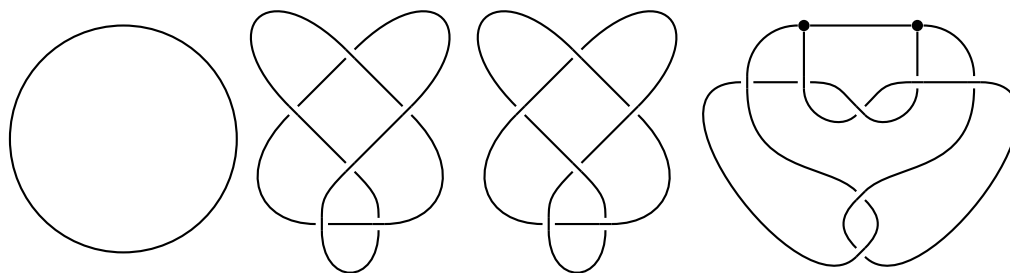
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 50; pp. 50:1–50:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** Diagrams of two trivial knots on the left, a bowline knot and a knotted spatial graph.

In recent years, many attempts have been made to attack such seemingly hard problems via the route of parameterized algorithms. In particular, the **treewidth** of a graph is a parameter quantifying how close a graph is to a tree, and thus algorithmic problems on graphs of low treewidth can often be solved very efficiently using dynamic programming techniques on the underlying tree structure of instance. The concept of **branchwidth**, which we also use below, is somewhat equivalent and always within a constant factor of treewidth [39]. One approach is to study a knot via one of its **diagrams** (see Figure 1), that is, a decorated graph obtained by a planar projection where it is indicated on each vertex which strand goes over or under. Then, if such a diagram has low treewidth, one can apply these standard dynamic programming techniques to solve seemingly hard problems very efficiently. While this approach has not yet been successful for unknot recognition beyond treewidth 2 [5], it has proved effective for the computation of many knot invariants, including: Jones and Kauffman polynomials [33] (which are known to be $\#P$ -hard to compute in general [27]), HOMFLY-PT polynomials [6], and quantum invariants [34, 8]. Since any knot admits infinitely many diagrams, these algorithms naturally lead to the following question raised by Burton [7, p.2694], and Makowsky and Mariño [33, p.755]: do all knots admit diagrams of constant treewidth, or conversely does there exist a family of knots for which *all* the diagrams have treewidth going to infinity. This question was answered recently by de Mesmay, Purcell, Schleimer and Sedgwick [9] who proved that, among other examples, torus knots $T_{p,q}$ are such a family. The proof relies at its core on an intricate result of Hayashi and Shimokawa [23] on *thin position of multiple Heegaard splittings*.

Our results. The main purpose of this work is to provide new techniques to characterize which knots, or more generally which **spatial graphs** (polygonal embeddings of graphs into \mathbb{R}^3 , considered up to isotopy, see for example Figure 1), do not admit diagrams of low treewidth. Our starting point is similar to the one in [9]: we first observe that if a knot or a spatial graph admits a diagram of low treewidth, then there is a way to sweep \mathbb{R}^3 using spheres arranged in a tree-like fashion which intersect the knot a small number of times (Proposition 5). This corresponds roughly to a map $f : \mathbb{R}^3 \rightarrow T$ where T is a trivalent tree, where the preimage of each point interior to an edge is a sphere with a small number of intersections with the knot (we refer to Section 2 for the precise technical definitions of all the concepts discussed in this introduction). We call this a **sphere decomposition**¹, and the resulting measure (maximal number of intersections) the **spherewidth** of the knot.

¹ Our sphere decompositions are different from the ones in [9] but functionally equivalent for knots.

Thus, in order to lower bound the treewidth of all the diagrams of a knot, it suffices to lower bound its spherewidth. We provide a systematic technique to do so using a perspective taken from structural graph theory. In the proof of the celebrated Graph Minor Theorem of Robertson and Seymour [41], handling families of graphs with bounded treewidth turns out not to be too hard [38], and in contrast, a large part is devoted to analyzing the structure shared by graphs of large treewidth. There, a fundamental contribution is the concept of **tangle**². We refer to Diestel [13] or Grohe [18] for nice introductions to tangles and their applications. Informally, a tangle of order k in a graph G is a choice, for each separation of size at most k , of a “big side” of that separation, where the highly-connected part of the graph lies. In addition, there are some compatibility properties, in particular no three “small sides” should cover the whole graph. Such a tangle turns out to be exactly the structure dual to branchwidth, in the sense that, as is proved in [39], for any graph G , the maximal possible order of a tangle is exactly equal to its branchwidth. We develop a similar concept dual to sphere decompositions which we call a **bubble tangle**. Informally, a bubble tangle of order k for a knot or spatial graph K is a map that, for each sphere intersecting K at most k times, chooses a “big side” indicating where the complicated part of K lies. There are again some compatibility conditions which add topological information to the collection of “small sides”. Then our first result is the following.

► **Theorem 1.** *For any knot or spatial graph K , the maximum order of a bubble tangle for K is equal to the spherewidth of K .*

This provides a convenient and systematic pathway to prove lower bounds on the spherewidth, and thus on the treewidth of all possible diagrams: it suffices to prove the existence of a bubble tangle of high order. However, making choices for the uncountable family of spheres with a small number of intersections with K , and then verifying the needed compatibility conditions is very unwieldy. Our second contribution is to provide a way to define such a bubble tangle in the case of knots (or spatial graphs) which are embedded on some surface Σ in \mathbb{R}^3 . Given a surface Σ in \mathbb{R}^3 , a **compression disk** is a disk properly embedded in $\mathbb{R}^3 \setminus \Sigma$ whose boundary is a non-contractible curve on Σ . The **compression-representativity of an embedding** of a knot or spatial graph K on a surface Σ in \mathbb{R}^3 is the smallest number of intersections between K and a cycle on Σ that bounds a compression disk. The **compression-representativity** of a knot or spatial graph is the supremum of that quantity over all embeddings on surfaces (this was originally defined by Ozawa [36]). Our second theorem is the following.

► **Theorem 2.** *For any knot or spatial graph K embedded on a surface Σ in \mathbb{R}^3 , there exists a bubble tangle of order $2/3$ of the compression-representativity of the embedding. Therefore, for any knot or spatial graph K , there exists a bubble tangle of order $2/3$ of the compression-representativity of K .*

Combining together Theorems 1 and 2 with Proposition 5 provides a large class of knots of high spherewidth, and our tools are versatile enough to apply to spatial graphs, while previous ones did not. In particular, observing that torus knots $T_{p,q}$ have high compression-representativity, we obtain the following corollary, which improves the lower bound obtained by [9], without relying on deep knot-theoretical tools.

² It turns out that the word *tangle* holds a completely different meaning in knot theory, and, to avoid confusion, in this article we will always use it with the graph theory meaning.

► **Corollary 3.** *A torus knot $T(p, q)$ has spherewidth at least $2/3 \min(p, q)$, and thus any diagram of $T(p, q)$ has treewidth at least $1/3 \min(p, q)$.*

Related work and proof techniques. The results in this article and many of their proof techniques stem from two very distinct lineages in quite distant communities, the first one being knot theory or more generally low-dimensional topology, and the second one being structural graph theory. While there have been some recent works aiming at building bridges between combinatorial width parameters and topological quantities (for example the aforementioned [9], but also [25, 26, 35] for related problems in 3-manifold theory), the main contribution in this article is that we dive deeper in the structural graph theory perspective via the concept of a tangle. The latter has now proved to be a fundamental tool in graph theory and beyond (see for example Diestel [10, Preface to the 5th edition]).

The duality theorem of Robertson and Seymour between branchwidth and tangles in [39] has been generalized many times since its inception, for example in order to encompass other notions of decompositions and their obstructions [2, 32], to apply more generally to matroids [17] and to the wide-ranging setting of abstract separations systems [11, 12]. The key difference in our work, and why it does not fit into these generalizations, is that our notions of sphere decomposition and bubble tangles inherently feature the *topological* constraint of working with 2-spheres. This is a crucial constraint, as it would be easy to sweep any knot with width at most 2 if one were allowed to use arbitrary surfaces during the sweeping process. Furthermore, in planar graphs, it was shown by Seymour and Thomas [48] that the separations involved in an optimal branch decomposition can always be assumed to take the form of 1-spheres, i.e., Jordan curves. This property led to the ratcatcher algorithm to compute the branchwidth of planar graphs in polynomial time [48], and to sphere-cut decompositions and their algorithmic applications (see for example [14]). Our sphere decompositions are the generalization one dimension higher of these sphere-cut decompositions, and Theorem 1 identifies bubble tangles as a correct notion of dual obstruction for those. We believe that these notions could be of further interest beyond knots, in the study of graphs embedded in \mathbb{R}^3 with some topological constraints, e.g., linkless graphs [42].

The **representativity** (also called **facewidth**) of a graph embedded on a surface S is the smallest number of intersections of a non-contractible curve with that graph. Theorem 2 will not come as a surprise for readers accustomed to graph minor theory, as Robertson and Seymour proved a very similar-looking theorem in Graph Minors XI [40, Theorem 4.1], showing that the branchwidth of a graph embedded on a surface is lower bounded by its representativity, which they prove by exhibiting a tangle. The key difference is that our notion of compression-representativity only takes into account the length of cycles bounding compression disks, instead of all the non-contractible cycles. Here again, this topological distinction is crucial to give a meaningful concept for knots, as for example the graph-theoretical representativity of a torus knot is zero. Due to this difference, the proof technique of Robertson and Seymour does not readily apply to prove Theorem 2; instead we have to rely on more topological arguments.

From the knot theory side, there is a long history in the study of the “best” way to sweep a knot while trying to minimize the number of intersections in this sweepout. One of the oldest knot invariants, the bridge number, can be seen through this lens (see for example [47]). A key concept in modern knot theory, introduced by Gabai in his proof of the Property R conjecture [16], is the notion of **thin position** which more precisely quantifies the best way to place a knot to minimize its width. It is at the core of many advances in modern knot theory (see for example Scharlemann [43]). Recent developments in thin position have

highlighted that in order to obtain the best topological properties, it can be helpful to sweep the knot in a tree-like fashion compared to the classical linear one. This approach leads to definitions bearing close similarities to our sphere decompositions (this is one of the ideas behind generalized Heegaard splittings [45, 44], see also [23, 24, 49]). The concept of compression-representativity of a knot or a spatial graph finds its roots in the works of Ozawa [36], and Blair and Ozawa [4] who defined it under the simple name of representativity, taking inspiration from graph theory. They proved that it provides a lower bound on the bridge number and on more general linear width quantities. Our Theorem 2 strengthens their results by showing that it also lower bounds the width of tree-like decompositions. Furthermore, while specific tools have been developed to lower bound various notions of width of knots or 3-manifolds, we are not aware of duality theorems like our Theorem 1. It shows that our bubble tangles constitute an obstruction that is, in a precise sense, the optimal tool for the purpose of lower bounding spherewidth.

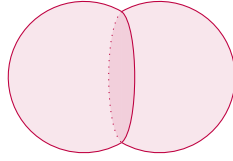
Finally, an important inspiration for our proof of Theorem 2 comes from a seemingly unrelated breakthrough of Pardon [37], who solved a famous open problem of Gromov [19] by proving the existence of knots with arbitrarily high **distortion**. The distortion for two points on an embedded curve in \mathbb{R}^3 is the ratio between the intrinsic and Euclidean distance between the points. The distortion of the entire curve is the supremum over all pairs of points. The distortion of a knot is the minimal distortion over all embeddings of the knot. While this metric quantity seems to have nothing to do with tree decompositions, it turns out that the technique developed by Pardon can be reinterpreted in our framework. With our terminology, his proofs amounts to first lower bounding the distortion by a constant factor of the spherewidth, and then defining a bubble tangle for knots of high representativity. The lower bound is nicely explained by Gromov and Guth [20, Lemma 4.2], where the simplicial map is similar to our sphere decompositions, up to a constant factor. Then our proof of Theorem 2 is inspired by the second part of Pardon's argument, with a quantitative strengthening to obtain the $2/3$ factor, whereas his argument would only yield $1/2$.

Organization of this paper. After providing background and defining our key concepts in Section 2, we prove Theorem 1 in Section 3, and Theorem 2 in Section 4. We provide examples in Section 5. Due to the line limitations, some proofs are not included in this extended abstract and are deferred to the full version [31].

2 Preliminaries

We include the most relevant definitions, but some familiarity with low-dimensional topology will help, see for example in the textbook of Schultens [46]. We refer to Diestel [10] for a nice introduction to graph theory and in particular its structural aspects. We denote by $V(G)$, $E(G)$, and $L(G)$ the vertices, edges and leaves (degree one vertices) of a graph G .

Low-dimensional topology. Following standard practice, instead of considering knots and spatial graphs within \mathbb{R}^3 , we compactify it and work within \mathbb{S}^3 . We denote by $C(A)$ the connected components of a subset A of \mathbb{S}^3 , and thus by $|C(A)|$ its number of connected components. As is standard in low-dimensional topology, we work in the Piecewise-Linear (PL) category, which means that *all the objects that we use in this article are assumed to be piecewise-linear*, i.e., made of a finite number of linear pieces with respect to a fixed triangulation of \mathbb{S}^3 . This allows us to avoid pathologies such as wild knots or the Alexander horned sphere. An **embedding** of a compact topological space X into another one Y is a



■ **Figure 2** A double bubble: two spheres that intersect in a single disk.

continuous injective map, and it is **proper** if it maps the boundary ∂X within the boundary ∂Y . A 3-dimensional version of the Schoenflies theorem guarantees that for any PL 2-sphere S embedded in \mathbb{S}^3 , both components of $\mathbb{S}^3 \setminus S$ are balls (see for example [3, Theorem XIV.1]). A **knot** is a PL embedding of S^1 into \mathbb{S}^3 , a **link** is a disjoint union of knots, and a **spatial graph** is a PL embedding of a graph G into \mathbb{S}^3 . All these objects are considered equivalent when they are ambient isotopic, i.e., when there exists a continuous deformation preserving the embeddedness. Knots and links are a special instance of spatial graphs, and henceforth we will mostly focus on spatial graphs, generally denoted by the letter G . For technical reasons, it is convenient to thicken our embedded graphs as follows. A **thickened embedding** φ of a graph G is an embedding of G in \mathbb{S}^3 where each vertex is thickened to a small ball, two balls are connected by a polygonal edge if and only if they are adjacent in the graph G , and pairs of edges are disjoint. We emphasize that we do not thicken edges, which might be considered nonstandard. We will also work with graphs embedded on surfaces which are themselves embedded in \mathbb{S}^3 : such embeddings will also always be thickened, that is, vertices on the surface are thickened into small disks. *From now on, all the graph embeddings will be thickened, and thus to ease notations we will omit the word thickened.*

As mentioned in the introduction, for Σ a surface embedded in \mathbb{S}^3 , a **compression disk** is a properly embedded disk D in $\mathbb{S}^3 \setminus \Sigma$ such that the boundary ∂D is a non-contractible curve on Σ . A **compressible curve** γ of Σ is the boundary of a compressing disk of Σ . For a spatial graph G embedded on a surface Σ in \mathbb{S}^3 , the **compression representativity** of G on Σ , written $c\text{-rep}(G, \Sigma)$ is $\min \{ |C(\alpha \cap G)| \mid \alpha \text{ compressible curve of } \Sigma \}$ (we count connected components to correctly handle thickened vertices). The compression representativity $c\text{-rep}(G)$ of G is the supremum of $c\text{-rep}(G, \Sigma)$ over all nested embeddings $G \hookrightarrow \Sigma \hookrightarrow \mathbb{S}^3$.

In order to define spherewidth and bubble tangles, we require a precise control of the event when two spheres merge together to yield a third one, which is mainly encapsulated in the concept of double bubble. A **double bubble** is a triplet of closed disks (D_1, D_2, D_3) in \mathbb{S}^3 , disjoint except on their boundaries, that they share: $D_1 \cap D_2 = D_1 \cap D_3 = D_2 \cap D_3 = D_1 \cap D_2 \cap D_3 = \partial D_1 = \partial D_2 = \partial D_3$, see Figure 2. Such a double bubble defines three spheres, which, by the PL Schoenflies theorem, bound three balls.

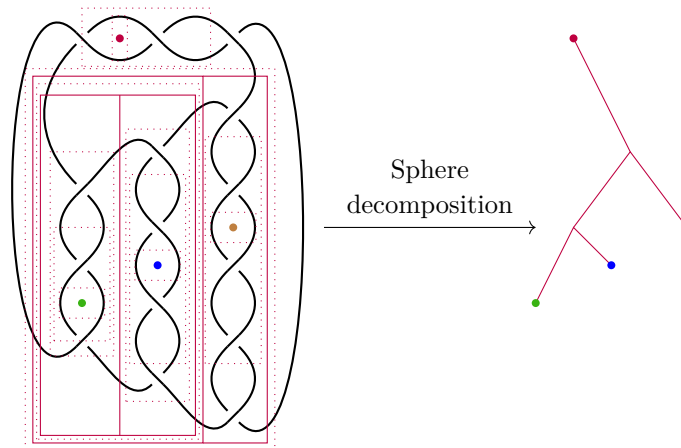
Two surfaces (resp. a knot and a surface) embedded in \mathbb{S}^3 are **transverse** if they intersect in a finite number of connected components, where the intersection is locally homeomorphic to the intersection of two orthogonal planes (resp. to the intersection of a plane and an orthogonal line). Likewise, we say that a surface is transverse to a ball if it is transverse to its boundary. A surface is transverse to a graph if it is transverse to all the thickened vertices and edges it intersects. A double bubble is transverse to a graph or a surface if each of its three spheres is and if the vertices of the graph do not intersect the spheres on their shared circle ∂D_i . Intersections are **tangent** when they are not transverse, and a sphere S is said **finitely tangent** to a graph G embedded in \mathbb{S}^3 if they do not intersect transversely but the number of intersections $|E(G) \cap S|$ is still finite.

Spherewidth. In this paragraph, we introduce sphere decompositions, which are the main way that we use to sweep knots and spatial graphs using spheres.

► **Definition 4** (Sphere decomposition). Let G be a graph embedded in \mathbb{S}^3 . A **sphere decomposition** of G is a continuous map $f : \mathbb{S}^3 \rightarrow T$ where T is a trivalent tree with at least one edge:

- For all $x \in L(T)$, $f^{-1}(x)$ is a point disjoint from G .
- For all $x \in V(T) \setminus L(T)$, $f^{-1}(x)$ is a PL double bubble transverse to G .
- For all x interior to an edge, $f^{-1}(x)$ is a sphere transverse or finitely tangent to G .

The **weight** of a sphere S (with respect to G) is the number of connected components in its intersection with G . The **width** of a sphere decomposition f is the supremum of the weight of $f^{-1}(x)$ over all points x interior to edges of the tree T . The **spherewidth** of the graph G , denoted by $sw(G)$, is the infimum, over all sphere decompositions f , of the width of f : $sw(G) = \inf_{f: \mathbb{S}^3 \rightarrow T} \sup_{x \in \mathring{e} \in E(T)} |C(f^{-1}(x) \cap G)|$. Therefore, a sphere decomposition is a way to continuously sweep \mathbb{S}^3 using spheres, which will occasionally merge or split in the form of double bubbles, and the spherewidth is a measure of how well we can sweep a graph G using sphere decompositions. This is similar to the level sets given by a Morse function, but note that our double-bubble singularities are not of Morse type, and those are key to the proof of Theorem 1.



■ **Figure 3** A width-4 sphere decomposition of a pretzel knot.

The point of using thickened embeddings instead of usual ones is that this allows disjoint spheres of a sphere decomposition to intersect a same vertex of a graph embedding. This is motivated by the following proposition, which provides a bridge between sphere decompositions and tree decompositions of diagrams of knots and spatial graphs.

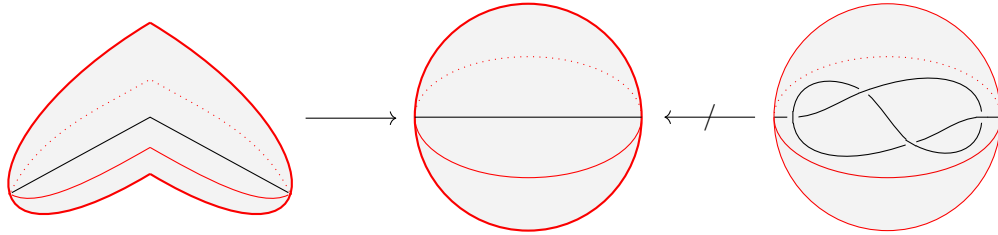
► **Proposition 5.** Let G be a knot or a graph embedded in \mathbb{S}^3 , and D be a diagram of G . Then the spherewidth of G is at most twice the tree-width of D .

The proof is very similar to that of Lemma 3.4 in [9] and is included to the full version [31].

Bubble tangle. Bubble tangles are our second main concept in this article. They will constitute an obstruction to spherewidth, by designating, for each sphere in \mathbb{S}^3 not intersecting the graph too many times, the side of the sphere that is easy to sweep. We first observe

that some balls have to be easy to sweep: intuitively this will be the case of any unknotted segment or empty ball (see Figure 4). Let G be a graph embedded in \mathbb{S}^3 . A closed ball B in \mathbb{S}^3 is said to be G -trivial if its boundary is transverse to G and one of the following holds (where $B(0, 1)$ is the unit ball of \mathbb{R}^3):

- $B \cap G = \emptyset$.
- $B \setminus G$ is homeomorphic to $B(0, 1) \setminus [-1, 1] \times \{(0, 0)\} \subset \mathbb{R}^3$.
- $B \setminus G$ is homeomorphic to $B(0, 1) \setminus [-1, 0] \times \{(0, 0)\} \subset \mathbb{R}^3$.



■ **Figure 4** Representation of a G -trivial ball and a non G -trivial ball.

We can now introduce bubble tangles.

► **Definition 6.** Let G be an embedding of a graph in \mathbb{S}^3 and $n \in \mathbb{N}$. A **bubble tangle \mathcal{T} of order $n \geq 2$** , is a collection of closed balls in \mathbb{S}^3 such that:

- (T1) For every closed ball B in \mathcal{T} , $|C(\partial B \cap G)| < n$.
- (T2) For every sphere S in \mathbb{S}^3 transverse to G , if $|C(S \cap G)| < n$ then exactly one of the two closed balls \bar{B}_1 is in \mathcal{T} or \bar{B}_2 is in \mathcal{T} , where $\mathbb{S}^3 \setminus S = \{B_1, B_2\}$.
- (T3) For every triple of balls B_1, B_2 and B_3 induced by a double bubble transverse to G , $\{B_1, B_2, B_3\} \not\subset \mathcal{T}$.
- (T4) For every closed ball B in \mathbb{S}^3 , if B is G -trivial and $|C(\partial B \cap G)| < n$, then $B \in \mathcal{T}$.

For every transverse sphere S such that $|C(S \cap G)| < n$, a bubble tangle chooses one of the two balls having S as the boundary. We think of the ball in \mathcal{T} as being a “small side”, since T4 stipulates that balls containing trivial parts of G are in \mathcal{T} , while the other one is the “big side”. Then the key property T3 enforces that no three small sides forming a double bubble should cover the entire \mathbb{S}^3 .

► **Remark 7.** Tangles in graph theory are often endowed with an additional axiom, specifying that small sides should be stable under inclusion (see e.g., [17, Axiom (T3A)]). Our bubble tangles are weaker in the sense that we do not enforce this axiom, but still strong enough to guarantee duality (Theorem 1) and the connection to compression-representativity (Theorem 2). Whether such an axiom can be additionally enforced in our definition of bubble tangle while preserving these properties is left as an open problem.

3 Obstruction and duality

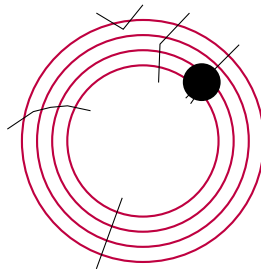
In this section, we prove Theorem 1: given a graph G embedded in \mathbb{S}^3 , the highest possible order of a bubble tangle is equal to the spherewidth of G . In the following, G is an embedding of a graph in \mathbb{S}^3 and the order of all bubble tangles that we consider is at least 3, the theorem being trivial otherwise. The proof is split into two inequalities: Proposition 8 and Proposition 11 which together immediately imply Theorem 1.

Bubble tangles as obstruction. We first show that a bubble tangle of order k and a sphere decomposition of width less than k cannot both exist at the same time.

► **Proposition 8.** *Let G be an embedding of a graph in \mathbb{S}^3 . If G admits a bubble tangle \mathcal{T} of order k then $sw(G) \geq k$.*

The proof of this proposition is similar to its graph-theoretical counterparts showing that tangles are an obstruction to branchwidth (see, e.g., [39]). The main difference with these proofs lies in the continuous aspects of our sphere decomposition, which we control using Lemmas 9 and 10.

Let S and S' be two disjoint spheres in \mathbb{S}^3 . Then $\mathbb{S}^3 \setminus (S \cup S')$ has three connected components: two balls and a space I homeomorphic to $\mathbb{S}^2 \times [0, 1]$. The spheres S and S' are said to be **braid-equivalent** if $(I \cup S \cup S') \setminus G$ is homeomorphic to $S_k \times [0, 1]$ where S_k is the 2-sphere with k holes. The intuition behind this definition is that it means that G forms a braid between S and S' . The following lemma explains how braid-equivalent spheres interact with a bubble tangle.



■ **Figure 5** The three innermost spheres are braid-equivalent, not the fourth one.

► **Lemma 9.** *Let \mathcal{T} be a bubble tangle and S, S' be two braid-equivalent spheres. Let us write $\mathbb{S}^3 \setminus S = \{B_1, B_2\}$ and $\mathbb{S}^3 \setminus S' = \{B'_1, B'_2\}$ such that $B_1 \subset B'_1$. If $B_1 \in \mathcal{T}$ then $B'_1 \in \mathcal{T}$.*

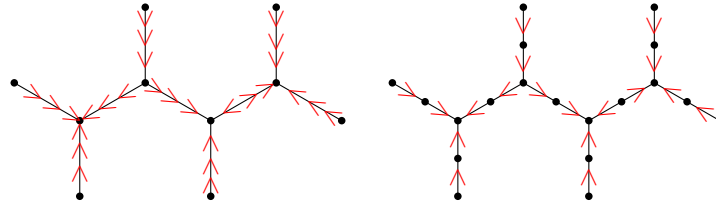
In the following, we will assume that there exists a bubble tangle \mathcal{T} of order k and a sphere decomposition $f : \mathbb{S}^3 \rightarrow T$ of G of width less than k in order to reach a contradiction. Let $e = (u, v) \in E(T)$ be an edge and x be a point of e so that $f^{-1}(x)$ is transverse to G . Notice that x cuts T in two trees : $T_u(x)$ and $T_v(x)$ where $T_u(x)$ is the tree containing the endpoint u . By definition $f^{-1}(x) = S$ is a sphere in \mathbb{S}^3 such that $|C(G \cap S)| < k$. It follows by T2 that exactly one of $f^{-1}(T_u(x))$ or $f^{-1}(T_v(x))$ belongs to \mathcal{T} . We define an **orientation** $o : T \rightarrow V(T)$ induced by \mathcal{T} as follows: if $f^{-1}(x)$ is transverse to G , $o(x) := v$ if $f^{-1}(T_u(x)) \in \mathcal{T}$, or $o(x) := u$ if $f^{-1}(T_v(x)) \in \mathcal{T}$. In other words, at a point x where $f^{-1}(x)$ is transverse to G the orientation o orients x outwards, toward the “big side”. If $f^{-1}(x)$ has a tangency with G , note that for any close enough neighbor y of x , $f^{-1}(y)$ is transverse to G , and we define $o(x) := o(y)$, making an arbitrary choice if needed. As we consider edges of the tree T to be intervals, we will use interval notations: we write $[u, v]$ for the edge (u, v) , and more generally $[x, y]$ to describe all the points on the edge between x and y . We say that an orientation o is **consistent** if for any x on some edge such that $f^{-1}(x)$ is transverse to G , o is constant on $[x, o(x)]$. The following lemma shows that the orientation o can be assumed to be consistent on all the edges of the tree T .

► **Lemma 10.** *Let us assume that there exists a bubble tangle \mathcal{T} of order k and a sphere decomposition $f : \mathbb{S}^3 \rightarrow T$ of G of width less than k . Then there exists a sphere decomposition to the same tree such that o is consistent on T .*

Lemma 10 ensures that for any edge $e = (u, v)$ of T , there exists a point x_e so that all the points in (x_e, v) are oriented towards v , while all the points in (u, x_e) are oriented towards u . Hence, by subdividing each edge e of T at this x_e , we can think of o as assigning a direction to each edge. This directed tree is the main tool that we use in the proof of Proposition 8.

Proof of Proposition 8. Let us assume that there exists both a bubble tangle of order k and a sphere decomposition $f : \mathbb{S}^3 \rightarrow T$ of width less than k . By Lemma 10, there exists a sphere decomposition of width less than k so that the orientation o as defined above is consistent. Denoting by T' the tree T where each edge has been subdivided once, this orientation corresponds to a choice of direction for each edge of T' . Every directed acyclic graph, and thus in particular the tree T' contains at least one sink, see Figure 6.

This sink cannot be a leaf of the tree. Indeed, let $e = [\ell, u]$ be an edge of T incident to a leaf ℓ . By definition, $f^{-1}(\ell)$ is a point disjoint from G , and thus for any y in (ℓ, u) close enough to ℓ , $f^{-1}(y)$ is a sphere disjoint from G . Hence $f^{-1}(T_\ell(y))$ is a G -trivial ball and belongs to \mathcal{T} . It follows that all edges incident to leaves of T' are oriented inward. This sink cannot be a degree-two vertex either, as the tree T' was defined in such a way that the two edges adjacent to a degree-two vertex are always oriented outwards. Finally, this sink cannot be a degree-three vertex as this would mean that the three balls induced by a double bubble are in \mathcal{T} , which would violate T3. We have thus reached a contradiction. ◀



■ **Figure 6** An example of T' from T leading to at least one sink.

Tightness of the obstruction. In this paragraph, we show that bubble tangles form a tight obstruction to sphere decompositions, in the sense that a bubble tangle of order k exists whenever a sphere decomposition of width less than k does not exist.

► **Proposition 11.** *Let G be an embedding of a graph in \mathbb{S}^3 and k be an integer at least three. If G does not admit a sphere decomposition of width less than k , then there exists a bubble tangle of order k .*

The idea of the proof is to show that, given a collection of closed balls satisfying the axioms T1 and T4 of bubble tangles, then either we can extend this collection to a bubble tangle, or there exists a **partial sphere decomposition** of width k which sweeps the space “between” the balls of the collection. We first introduce the relevant definition.

Let G a graph embedded in \mathbb{S}^3 . A **partial sphere decomposition** of G is a continuous map $f : \mathbb{S}^3 \rightarrow T$ where T is a trivalent tree with at least one edge such that:

- For all $x \in L(T)$, $f^{-1}(x)$ is a point disjoint from G or a closed ball B .
- For all $x \in V(T) \setminus L(T)$, $f^{-1}(x)$ is a double bubble transverse to G .
- For all x interior to an edge, $f^{-1}(x)$ is a sphere transverse or finitely tangent to G .

The leaves of T having preimages by f which are not points are called **non-trivial leaves**. Let G be a graph embedding in \mathbb{S}^3 and \mathcal{A} be a collection of closed balls in \mathbb{S}^3 . A partial sphere decomposition f **conforms** to \mathcal{A} if, for all $x \in L(T)$, $f^{-1}(x)$ is either a point disjoint

from G , or a closed ball B such that there exists $A \in \mathcal{A}$ such that ∂B and ∂A are braid equivalent and $B \subset A$. In the latter case we say that x conforms to A . The width of a partial sphere decomposition is defined like the width of standard sphere decompositions: it is the supremal weight of spheres that are pre-images of points in the interiors of edges of T .

Now, the proof of Proposition 11 hinges on the following key lemma. Its proof is similar to branchwidth-tangle duality proofs [39] in that it builds a bubble tangle inductively, but the continuous nature of our objects makes us rely on transfinite induction in the form of Zorn's lemma.

► **Lemma 12.** *Let G be an embedding of a graph in \mathbb{S}^3 , k be an integer at least 3 and \mathcal{A} be a collection of closed balls in \mathbb{S}^3 satisfying $T1$ and $T4$. Then one of the following is true :*

- \mathcal{A} extends to a bubble tangle of order k .
- there is a partial sphere decomposition of width less than k that conforms to \mathcal{A} .

Proof of Proposition 11. We denote by \mathcal{A} the collection of G -trivial balls. By definition, \mathcal{A} satisfies $T4$, and since G -trivial balls have weight at most two, it also satisfies $T1$ for k at least three. Therefore, by Lemma 12, either \mathcal{A} extends to a bubble tangle of order k , or there exists a partial sphere decomposition of width less than k conforming to it. In the first case, we are done. In the second case, we are also done, since, given a partial sphere decomposition of width less than k conforming to G -trivial balls, it is straightforward to sweep within the G -trivial balls so as to obtain a sphere decomposition of width less than k . ◀

4 From compression representativity to bubble tangles

The goal of this section is to show Theorem 2: when a graph G is embedded on a compact, orientable, and non-zero genus surface Σ , there exists a bubble tangle naturally arising from the compression representativity of G on Σ . In the following, we assume Σ is compact, orientable, and not a sphere.

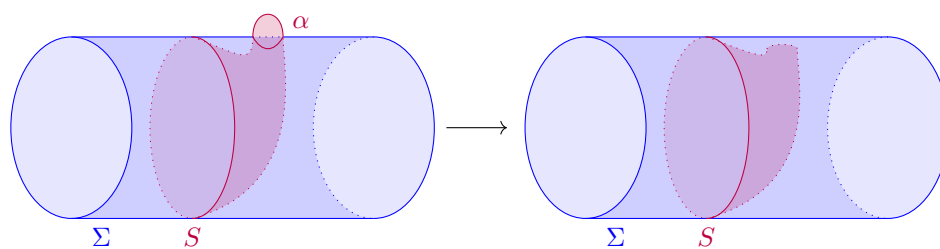
Under these hypotheses, the idea of the proof is to show that there exists a natural choice of small side for every sphere with fewer intersections with G than the compression representativity. Intuitively, such a sphere will only cut disks or “trivial parts” of Σ on one of its sides, which we will designate as the small one. That is justified by the following lemma.

► **Lemma 13.** *Let Σ be a surface embedded in \mathbb{S}^3 and S be a sphere in \mathbb{S}^3 that intersects Σ transversely such that there is at least one non-contractible curve in the intersection. Then one of the non-contractible curves is compressible.*

Proof. As Σ and S are transverse, the intersection of S and Σ consists of a disjoint union of simple closed curves. Each one of these curves bounds two disks on S . Let α be a curve of $S \cap \Sigma$ that is innermost in S , i.e. it bounds a disk D in S that does not contain any other curve of $S \cap \Sigma$. If α is non-contractible, then the disk D is a compression disk for α , and thus α is compressible. Otherwise, α bounds a disk D_Σ in Σ (see for example Epstein [15, Theorem 1.7]). We deform S continuously by “pushing” D through D_Σ while keeping S embedded (see Figure 7) until α disappears from $\Sigma \cap S$.

Repeating this process on a new innermost curve of S will eventually yield a non-contractible compressible curve. Indeed, the number of curves in the intersection is finite (recall that both surfaces are piecewise linear), decreases at each step, and one of the curves in $\Sigma \cap S$ is non-contractible. ◀

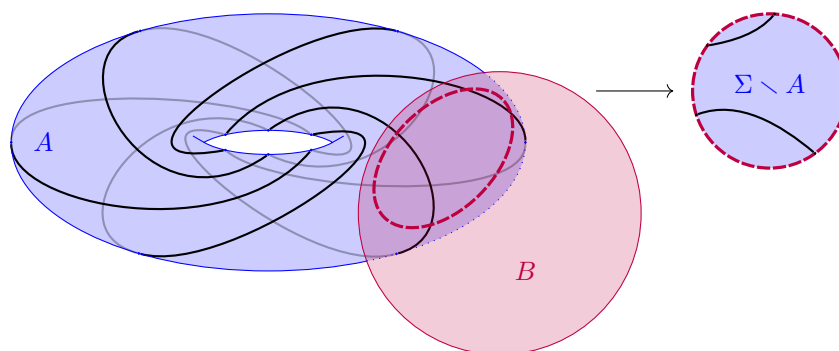
A direct consequence of this lemma is that if G is embedded on a surface Σ , a sphere S intersects Σ , and the intersection has weight less than $c\text{-rep}(G, \Sigma)$, then all the simple closed curves in the intersection are contractible. Therefore, one of the two balls bounded



■ **Figure 7** Removing a trivial curve from $S \cap \Sigma$.

by S contains the meaningful topology of Σ , while the other one only contains spheres with holes (see Figure 8). In order to formalize this, we will rely on fundamental groups (see for example Hatcher [22] for an introduction to this concept). The inclusion of a subsurface X on Σ induces a morphism $i_* : \pi_1(X) \rightarrow \pi_1(\Sigma)$. If this morphism is trivial, we say that X is π_1 -trivial with respect to Σ .

► **Definition 14** (Compression bubble tangle on an embedded surface). *Let G be a graph embedded on Σ , a surface embedded in \mathbb{S}^3 such that $c\text{-rep}(G, \Sigma) \geq 3$ and set $k = \frac{2}{3}c\text{-rep}(G, \Sigma)$. The **compression bubble tangle** $c\mathcal{T}$, is the collection of balls in \mathbb{S}^3 defined as follows: for any sphere S in \mathbb{S}^3 transverse to G such that $|C(S \cap G)| < k$, by Lemma 13, there is exactly one connected component A of $\Sigma \setminus S$ that is π_1 -trivial. Exactly one of the open balls B of $\mathbb{S}^3 \setminus S$ contains A , and we put the closed ball in $c\mathcal{T}$: $\bar{B} \in c\mathcal{T}$.*



■ **Figure 8** Intersection between a torus knot $T_{6,5}$ embedded on a torus and a sphere. Here the ball B containing the disk on the right is in the compression bubble tangle.

The main step in the proof of Theorem 2 is to prove that a compression bubble tangle on the torus is indeed a bubble tangle.

► **Proposition 15.** *A compression bubble tangle is a bubble tangle.*

Note that Propositions 15 and 8 directly imply Theorem 2 (the theorem is trivial if $c\text{-rep}(G, \Sigma) < 3$). Therefore, the rest of this section is devoted to proving Proposition 15.

By definition, a compression bubble tangle satisfies T1 and T2. We then notice that T4 is verified whenever the compression representativity of G on Σ is greater than 2.

► **Lemma 16.** *If $c\text{-rep}(G, \Sigma) \geq 3$ then for all G -trivial balls B , $B \cap \Sigma$ is π_1 -trivial.*

The hard part of the proof is to show that T3 is satisfied. This is more delicate than it seems at first glance, since any surface can be obtained by gluing three disks, and these three disks can even come from a double bubble: we provide an example in the appendix of the full version [31].

Henceforth, we will proceed by contradiction and assume that we can cover \mathbb{S}^3 by three closed balls B_1, B_2, B_3 of $c\mathcal{T}$ that induce a double bubble DB transverse to Σ and G . Thus Σ is covered by three surfaces with boundary: $\Sigma \cap B_1, \Sigma \cap B_2$ and $\Sigma \cap B_3$ which are π_1 -trivial by definition of $c\mathcal{T}$. In the following, we write $S_i = \partial B_i$. We first show that we can furthermore assume that these surfaces are a disjoint union of closed disks on Σ .

► **Lemma 17.** *Let G be a graph embedded on Σ , a surface embedded in \mathbb{S}^3 . Let $c\mathcal{T}$ be the compression bubble tangle associated to G and Σ . If there is a double bubble DB transverse to Σ , inducing three balls $B_1, B_2, B_3 \in c\mathcal{T}^3$ such that $B_1 \cup B_2 \cup B_3 = \mathbb{S}^3$, then we can isotope the double bubble so that we additionally have that $B_i \cap \Sigma$ is a union of closed disks.*

Then we define Γ **induced by the double bubble DB** to be the intersection of the double bubble with Σ : where vertices are the intersection of the common boundary of the three disks with Σ and edges are the intersections of Σ with the disks. By Lemma 17, we can assume that this graph is trivalent and cellularly embedded. It is naturally weighted by endowing each edge with its weight, i.e., the number of connected components in its intersection with G . Let us now state the lemma we will use for the sake of contradiction.

► **Lemma 18.** *The total weight of Γ is less than $c\text{-rep}(G, \Sigma)$:*

$$\sum_{e \in E(\Gamma)} |C(e \cap G)| < c\text{-rep}(G, \Sigma).$$

Proof. Since each edge of Γ bounds exactly two faces of Γ , i.e, disks of Σ ; and $\Gamma = DB \cap \Sigma$ we get the following equality:

$$|C(S_1 \cap G)| + |C(S_2 \cap G)| + |C(S_3 \cap G)| = 2 \sum_{e \in E(\Gamma)} |C(e \cap G)| \tag{1}$$

By definition of $c\mathcal{T}$, each ball B_i satisfies $|C(S_i \cap G)| < \frac{2}{3}c\text{-rep}(G, \Sigma)$ so that:

$$|C(S_1 \cap G)| + |C(S_2 \cap G)| + |C(S_3 \cap G)| < 3 \cdot \frac{2}{3}c\text{-rep}(G, \Sigma) = 2c\text{-rep}(G, \Sigma). \tag{2}$$

Combining (1) and (2) concludes the proof: $2 \sum_{e \in E(\Gamma)} |C(e \cap G)| < 2c\text{-rep}(G, \Sigma)$. ◀

Hence, if Γ contained a simple closed curve that is compressible, we would obtain the contradiction that we are looking for. The rest of the proof almost consists of finding such a compressible curve, leading to the following proposition.

► **Proposition 19.** *There exists a set of edges X on Γ such that:*

$$\sum_{e \in X} |C(e \cap G)| \geq c\text{-rep}(G, \Sigma).$$

The proof of Proposition 19 is the technical crux of Theorem 2. It consists in defining a merging process, which gradually merges two balls of a double bubble, and proving that at some point in this merging process, one ball will intersect Σ in a non-trivial way, and

thus yield a compressible curve via Lemma 13. An additional difficulty is that this curve might be non-simple in Γ ; we circumvent this issue by finding a fractional version of such a curve instead, which will be strong enough to prove Proposition 19. This proposition directly implies Proposition 15, and thus Theorem 2:

Proof of Proposition 15. A compression bubble tangle immediately satisfies the bubble tangle axioms T1 and T2 by definition, and T4 by Lemma 16. For the axiom T3, assume by contradiction that there exist three closed balls $B_1, B_2, B_3 \in \mathcal{c}\mathcal{T}$ covering \mathbb{S}^3 and inducing a double bubble transverse to Σ . By Lemma 17, we can assume the graph Γ induced by the intersection of the double bubble with Σ is cellularly embedded. Then by Proposition 19, the total weight of Γ is at least $\mathcal{c}\text{-rep}(G, \Sigma)$. This is a contradiction with Lemma 18. ◀

5 Examples

A torus knot $T_{p,q}$ is a knot embedded on an unknotted torus \mathbb{T} in \mathbb{S}^3 , for example a standard torus of revolution. It winds p times around the revolution axis, and q times around the core of the torus. We refer to Figure 8 for an illustration of $T_{6,5}$. The proof of Corollary 3 (see [31, Corollary 1.3]) follows by combining Proposition 5 and Theorems 1 and 2.

More generally, the same argument can be applied to lower bound the treewidth of the (p, q) -cabling [1, Section 5.2] of any nontrivial knot. We refer to Ozawa [36, Theorem 6] for examples of spatial embeddings of any graph with high compression representativity, and thus high spherewidth.

We conclude by observing that the proof of Theorem 2 offers more flexibility than what the theorem states and can also be applied in some settings where the compression-representativity is low. For example, a connected sum of two knots $K_1 \# K_2$ has compression-representativity two (see [36, Corollary 9]), but if one these two knots, say K_1 , has high compression-representativity separately, then we can still define a bubble-tangle of high order by considering as big sides the balls containing the surface that K_1 is embedded on.

References

- 1 Colin C. Adams. *The knot book*. American Mathematical Society, 1994.
- 2 Omid Amini, Frédéric Mazoit, Nicolas Nisse, and Stéphan Thomassé. Submodular partition functions. *Discrete Mathematics*, 309(20):6000–6008, 2009.
- 3 Rudolph H Bing. *The geometric topology of 3-manifolds*, volume 40. American Mathematical Society, 1983.
- 4 Ryan Blair and Makoto Ozawa. Height, trunk and representativity of knots. *Journal of the Mathematical Society of Japan*, 71(4):1105–1121, 2019.
- 5 Hans L. Bodlaender, Benjamin A. Burton, Fedor V. Fomin, and Alexander Grigoriev. Knot diagrams of treewidth two. In *International Workshop on Graph-Theoretic Concepts in Computer Science*, pages 80–91. Springer, 2020.
- 6 Benjamin A. Burton. The HOMFLY-PT polynomial is fixed-parameter tractable. In *34th International Symposium on Computational Geometry (SoCG 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.
- 7 Benjamin A. Burton, Herbert Edelsbrunner, Jeff Erickson, and Stephan Tillmann. Computational geometric and algebraic topology. *Oberwolfach Reports*, 12(4):2637–2699, 2016.
- 8 Benjamin A. Burton, Clément Maria, and Jonathan Spreer. Algorithms and complexity for Turaev–Viro invariants. *Journal of Applied and Computational Topology*, 2(1-2):33–53, 2018.
- 9 Arnaud de Mesmay, Jessica Purcell, Saul Schleimer, and Eric Sedgwick. On the tree-width of knot diagrams. *Journal of Computational Geometry*, 10(1):164–180, 2019.

- 10 Reinhard Diestel. *Graph theory*. Number 173 in Graduate texts in mathematics. Springer, New York, 5th edition, 2016.
- 11 Reinhard Diestel and Sang-il Oum. Tangle-Tree Duality: In Graphs, Matroids and Beyond. *Combinatorica*, 39(4):879–910, August 2019.
- 12 Reinhard Diestel and Sang-il Oum. Tangle-tree duality in abstract separation systems. *Advances in Mathematics*, 377:107470, 2021.
- 13 Reinhard Diestel and Geoff Whittle. Tangles and the Mona Lisa. *arXiv preprint arXiv:1603.06652*, 2016.
- 14 Frederic Dorn, Eelko Penninx, Hans L Bodlaender, and Fedor V Fomin. Efficient exact algorithms on planar graphs: Exploiting sphere cut branch decompositions. In *European Symposium on Algorithms*, pages 95–106. Springer, 2005.
- 15 David B.A. Epstein. Curves on 2-manifolds and isotopies. *Acta Mathematica*, 115:83–107, 1966.
- 16 David Gabai. Foliations and the topology of 3-manifolds. ii. *Journal of Differential Geometry*, 26(3):461–478, 1987.
- 17 James Geelen, Bert Gerards, Neil Robertson, and Geoff Whittle. Obstructions to branch-decomposition of matroids. *Journal of Combinatorial Theory*, page 11, 2006.
- 18 Martin Grohe. Tangles and connectivity in graphs. In *Language and Automata Theory and Applications: 10th International Conference, LATA 2016, Prague, Czech Republic, March 14-18, 2016, Proceedings 10*, pages 24–41. Springer, 2016.
- 19 Mikhael Gromov. Filling Riemannian manifolds. *Journal of Differential Geometry*, 18(1):1–147, 1983.
- 20 Misha Gromov and Larry Guth. Generalizations of the Kolmogorov–Barzdin embedding estimates. *Duke Mathematical Journal*, 161(13):2549–2603, 2012.
- 21 Joel Hass, Jeffrey C. Lagarias, and Nicholas Pippenger. The computational complexity of knot and link problems. *Journal of the ACM (JACM)*, 46(2):185–211, 1999.
- 22 Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge ; New York, 2002.
- 23 Chuichiro Hayashi and Koya Shimokawa. Thin position of a pair (3-manifold, 1-submanifold). *Pacific Journal of Mathematics*, 197(2):301–324, February 2001.
- 24 Qidong He and Scott A Taylor. Links, bridge number, and width trees. *Journal of the Mathematical Society of Japan*, 1(1):1–39, 2022.
- 25 Kristóf Huszár and Jonathan Spreer. 3-manifold triangulations with small treewidth. In *35th International Symposium on Computational Geometry (SoCG 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.
- 26 Kristóf Huszár, Jonathan Spreer, and Uli Wagner. On the treewidth of triangulated 3-manifolds. *Journal of Computational Geometry*, 10(2):70–98, 2019.
- 27 François Jaeger, Dirk L. Vertigan, and Dominic J.A. Welsh. On the computational complexity of the Jones and Tutte polynomials. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 108, pages 35–53. Cambridge University Press, 1990.
- 28 Marc Lackenby. Algorithms in 3-manifold theory. *Surveys in Differential Geometry*, 2020.
- 29 Marc Lackenby. The efficient certification of knottedness and Thurston norm. *Advances in Mathematics*, 387:107796, 2021.
- 30 Marc Lackenby. Unknot recognition in quasi-polynomial time, 2021. Talk with slides available on the author’s webpage : <http://people.maths.ox.ac.uk/lackenby/quasipolynomial-talk.pdf>.
- 31 Corentin Lunel and Arnaud de Mesmay. A structural approach to tree decompositions of knots and spatial graphs. *arXiv preprint arXiv:2303.07982*, 2023.
- 32 Laurent Lyaudet, Frédéric Mazoit, and Stéphan Thomassé. Partitions versus sets: a case of duality. *European journal of Combinatorics*, 31(3):681–687, 2010.
- 33 J.A. Makowsky and J.P. Mariño. The parametrized complexity of knot polynomials. *Journal of Computer and System Sciences*, 67(4):742–756, December 2003.

- 34 Clément Maria. Parameterized Complexity of Quantum Knot Invariants. In Kevin Buchin and Éric Colin de Verdière, editors, *37th International Symposium on Computational Geometry (SoCG 2021)*, volume 189 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 53:1–53:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- 35 Clément Maria and Jessica Purcell. Treewidth, crushing and hyperbolic volume. *Algebraic & Geometric Topology*, 19(5):2625–2652, 2019.
- 36 Makoto Ozawa. Bridge position and the representativity of spatial graphs. *Topology and its Applications*, 159(4):936–947, 2012.
- 37 John Pardon. On the distortion of knots on embedded surfaces. *Annals of Mathematics*, 174(1):637–646, July 2011.
- 38 Neil Robertson and Paul D. Seymour. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, August 1986.
- 39 Neil Robertson and Paul D. Seymour. Graph minors. X. Obstructions to tree-decomposition. *Journal of Combinatorial Theory, Series B*, 52(2):153–190, July 1991.
- 40 Neil Robertson and Paul D. Seymour. Graph minors. XI. Circuits on a Surface. *Journal of Combinatorial Theory. Series B*, 60(1):72–106, January 1994.
- 41 Neil Robertson and Paul D. Seymour. Graph minors. XX. Wagner’s conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.
- 42 Horst Sachs. On a spatial analogue of Kuratowski’s theorem on planar graphs—an open problem. In *Graph theory*, pages 230–241. Springer, 1983.
- 43 Martin Scharlemann. Thin position in the theory of classical knots. In *Handbook of knot theory*, pages 429–459. Elsevier, 2005.
- 44 Martin Scharlemann, Jennifer Schultens, and Toshio Saito. *Lecture notes on generalized Heegaard splittings*. World Scientific, 2016.
- 45 Martin Scharlemann and Abigail Thompson. Thin position for 3-manifolds. In *Geometric Topology: Joint US-Israel Workshop on Geometric Topology, June 10-16, 1992, Technion, Haifa, Israel*, volume 164, page 231. American Mathematical Society, 1994.
- 46 Jennifer Schultens. *Introduction to 3-manifolds*, volume 151. American Mathematical Society, 2014.
- 47 Jennifer Schultens. The bridge number of a knot. In *Encyclopedia of Knot Theory*, pages 229–242. Chapman and Hall/CRC, 2021.
- 48 Paul D. Seymour and Robin Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.
- 49 Scott Taylor and Maggy Tomova. Additive invariants for knots, links and graphs in 3-manifolds. *Geometry & Topology*, 22(6):3235–3286, 2018.
- 50 Alan Mathison Turing. *Solvable and unsolvable problems*. Penguin Books London, 1954.

Ephemeral Persistence Features and the Stability of Filtered Chain Complexes

Facundo Mémoli ✉ 🏠

Department of Mathematics and Department of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA

Ling Zhou ✉ 🏠 

Department of Mathematics, The Ohio State University, Columbus, OH, USA

Abstract

We strengthen the usual stability theorem for Vietoris-Rips (VR) persistent homology of finite metric spaces by building upon constructions due to Usher and Zhang in the context of filtered chain complexes. The information present at the level of filtered chain complexes includes ephemeral points, i.e. points with zero persistence, which provide additional information to that present at homology level. The resulting invariant, called verbose barcode, which has a stronger discriminating power than the usual barcode, is proved to be stable under certain metrics which are sensitive to these ephemeral points. In some situations, we provide ways to compute such metrics between verbose barcodes. We also exhibit several examples of finite metric spaces with identical (standard) VR barcodes yet with different verbose VR barcodes thus confirming that these ephemeral points strengthen the discriminating power of the standard VR barcode.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology; Mathematics of computing → Topology

Keywords and phrases filtered chain complexes, Vietoris-Rips complexes, barcode, bottleneck distance, matching distance, Gromov-Hausdorff distance

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.51

Related Version *Full Version*: <https://arxiv.org/abs/2208.11770>

Funding FM and LZ were partially supported by the NSF through grants RI-1901360, CCF-1740761, and CCF-1526513, and DMS-1723003.

1 Introduction

In topological data analysis, *persistent homology* is one of the main tools used for extracting and analyzing multiscale geometric and topological information from metric spaces.

Typically, the *persistent homology pipeline* (as induced by the Vietoris-Rips filtration) is explained via the diagram:

$$\text{Metric Spaces} \rightarrow \text{Simplicial Filtrations} \rightarrow \text{Persistence Modules}$$

where, from left to right, the second map is homology with field coefficients. Throughout the paper, we fix a base field \mathbb{F} .

Pairs of birth and death times of topological features (such as connected components, loops, voids and so on) give rise to the *barcode*, or also called the *persistence diagram*, of a given metric space [13, 4]. The so-called *bottleneck distance* d_B between the persistent homology barcodes arising from the Vietoris-Rips filtration of metric spaces provides a polynomial time computable lower bound for the *Gromov-Hausdorff distance* d_{GH} between the underlying metric spaces [8, 10]. However, this bound is not tight, in general (cf. [17, Example 6.6]). A restricted version of this theorem states:



© Facundo Mémoli and Ling Zhou;

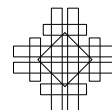
licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 51; pp. 51:1–51:18

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



► **Theorem 1** (Stability Theorem for d_B). *Let X and Y be two finite metric spaces. Let $\mathcal{B}_k(X)$ (resp. $\mathcal{B}_k(Y)$) denote the barcode of the persistence module $H_k(\text{VR}_\bullet(X))$ (resp. $H_k(\text{VR}_\bullet(Y))$). Then, we have*

$$\sup_{k \in \mathbb{Z}_{\geq 0}} d_B(\mathcal{B}_k(X), \mathcal{B}_k(Y)) \leq 2 \cdot d_{\text{GH}}(X, Y).$$

In this paper, with the goal of refining the standard stability result alluded to above, we concentrate on the usually implicit but conceptually important intermediate step which assigns a *filtered chain complex* (FCC) to a given simplicial filtration:

$$\text{Metric spaces} \rightarrow \text{Simplicial Filtrations} \rightarrow \boxed{\text{FCCs}} \rightarrow \text{Persistence Modules}.$$

Related work on FCCs

An FCC is an ascending sequence of chain complexes connected by monomorphisms. For instance, an FCC induced by a simplicial filtration $\{X_t\}_{t \in \mathbb{R}}$ can be represented by the following commutative diagram: for any $t \leq t'$,

$$\begin{array}{ccccccc} C_*(X_t) : & \cdots & \xrightarrow{\partial_{k+2}} & C_{k+1}(X_t) & \xrightarrow{\partial_{k+1}} & C_k(X_t) & \xrightarrow{\partial_k} \cdots \\ \downarrow & & & \downarrow & & \downarrow & \\ C_*(X_{t'}) : & \cdots & \xrightarrow{\partial_{k+2}} & C_{k+1}(X_{t'}) & \xrightarrow{\partial_{k+1}} & C_k(X_{t'}) & \xrightarrow{\partial_k} \cdots \end{array},$$

where each X_t is a simplicial complex and $C_*(X_t)$ denotes the simplicial chain complex of X_t .

Studies of the decomposition of FCCs in several different settings can be found in [18, 12, 15, 5, 6]. We follow the convention of Usher and Zhang [18], where they study a notion of Floer-type complexes as a generalization of FCCs and prove a stability result for the usual bottleneck distance of concise barcodes of Floer-type complexes. In particular, they studied FCCs in detail and considered the notion of *verbose barcode* $\mathcal{B}_{\text{Ver},k}$ of FCCs, which consists of the standard barcode (which the authors call *concise barcode* and denote as $\mathcal{B}_{\text{Con},k} := \mathcal{B}_k$) together with additional *ephemeral* bars, i.e. bars of length 0.

They also proved that every FCC decomposes into the direct sum of indecomposables $\mathcal{E}(a, a + L, k)$, which they called *elementary FCCs*, of the following form (see [18, Definition 7.2]): if $L \in [0, \infty)$ and $a \in \mathbb{R}$, then $\mathcal{E}(a, a + L, k)$ is given by

$$\begin{array}{l} t < a : \quad \cdots \rightarrow 0 \xrightarrow{\partial_{k+2}=0} 0 \xrightarrow{\partial_{k+1}=0} 0 \xrightarrow{\partial_k=0} 0 \rightarrow \cdots \\ t \in [a, a + L) : \quad \cdots \rightarrow 0 \xrightarrow{\partial_{k+2}=0} 0 \xrightarrow{\partial_{k+1}=0} \mathbb{F}x \xrightarrow{\partial_k=0} 0 \rightarrow \cdots \\ t \in [a + L, \infty) : \quad \cdots \rightarrow 0 \xrightarrow{\partial_{k+2}=0} \mathbb{F}y \xrightarrow{\partial_{k+1}: y \mapsto x} \mathbb{F}x \xrightarrow{\partial_k=0} 0 \rightarrow \cdots \end{array}$$

If $L = \infty$, then $\mathcal{E}(a, \infty, k)$ (with the convention that $a + \infty = \infty$) is given by

$$\begin{array}{l} t < a : \quad \cdots \rightarrow 0 \xrightarrow{\partial_{k+2}=0} 0 \xrightarrow{\partial_{k+1}=0} 0 \xrightarrow{\partial_k=0} 0 \rightarrow \cdots \\ t \in [a, \infty) : \quad \cdots \rightarrow 0 \xrightarrow{\partial_{k+2}=0} 0 \xrightarrow{\partial_{k+1}=0} \mathbb{F}x \xrightarrow{\partial_k=0} 0 \rightarrow \cdots \end{array}$$

The degree- l verbose barcode of the elementary FCC $\mathcal{E}(a, a + L, k)$ is $\{(a, a + L)\}$ for $l = k$ and is empty for $l \neq k$.

The concise barcode of an FCC is defined as the collection of non-ephemeral bars, i.e. bars corresponding to elementary FCCs with $L \neq 0$ in its decomposition, which agrees with the

standard barcode. Indeed, the k -th persistent homology of the elementary FCC $\mathcal{E}(a, a + L, k)$ is the interval persistence module associated to the interval $[a, a + L]$, for $L \in [0, \infty]$. In particular, $H_k(\mathcal{E}(a, a, k))$ is the trivial persistence module.

In real calculations, barcodes are often computed for simplexwise filtrations first (i.e., simplices are assumed to enter the filtration one at a time), in which case all elementary FCCs corresponds to intervals with positive length. This implies that, although not outputted, verbose barcodes are computed in many persistence algorithms. For VR FCCs, we made a small modification of the software Ripser introduced by Bauer (see [1]) to extract verbose barcodes of finite metric spaces.

In this paper, we focus on the *ephemeral* bars in the barcode, or equivalently, on the diagonal points in the persistence diagram.

Overview of our results

One drawback of the bottleneck stability result described in Theorem 1 is that one asks for optimal matchings between the concise (i.e. standard) barcodes $\mathcal{B}_{\text{Con},k}(X)$ and $\mathcal{B}_{\text{Con},k}(Y)$ for each individual degree k *independently*.

With the goal of finding a *coherent* or *simultaneous* matching of barcodes across all degrees at once, we study the interleaving distance d_I between FCCs and establish an isometry theorem between d_I and the *matching distance* d_M between the verbose barcodes (see Definition 18):

► **Theorem 2 (Isometry theorem).** *For any two FCCs $(C_*, \partial_C, \ell_C)$ and $(D_*, \partial_D, \ell_D)$, let $\mathcal{B}_{\text{Ver},k}^C$ and $\mathcal{B}_{\text{Ver},k}^D$ denote their degree- k verbose barcodes, respectively, and let $d_M(\mathcal{B}_{\text{Ver},k}^C, \mathcal{B}_{\text{Ver},k}^D) := \sup_{k \in \mathbb{Z}_{\geq 0}} d_M(\mathcal{B}_{\text{Ver},k}^C, \mathcal{B}_{\text{Ver},k}^D)$. Then,*

$$d_M(\mathcal{B}_{\text{Ver}}^C, \mathcal{B}_{\text{Ver}}^D) = d_I((C_*, \partial_C, \ell_C), (D_*, \partial_D, \ell_D)).$$

To prove that $d_M \leq d_I$ (see §3.3.1), we adapted ideas implicit in [18, Proposition 9.3] which the authors used to establish the stability of *Floer*-type complexes (on the same underlying chain complex). For the other direction, $d_M \geq d_I$ (see §3.3.2), we use an idea similar to the one used for proving that the bottleneck distance d_B between concise barcodes is upper bounded by d_I between persistent modules, cf. [14, Theorem 3.4].

Unlike d_B between concise barcodes, d_M between verbose barcodes of VR RCCs is not stable under the Gromov-Hausdorff distance between metric spaces. Indeed, d_M is only finite if the two underlying metric spaces have the same cardinality. We remedy this issue in §4.2 by incorporating the notion of tripods as in [16].

For a surjection $\phi_X : Z \rightarrow X$, we equip Z with the pullback $\phi_X^* d_X$ of the distance function d_X and call the pair $(Z, \phi_X^* d_X)$ the *pullback (pseudo) metric space* (induced by ϕ_X). We call the degree- k verbose barcode of $(Z, \phi_X^* d_X)$ a *degree- k pullback barcode* of X . We define the *pullback bottleneck distance* between verbose barcodes of two finite metric spaces X and Y to be the infimum of the matching distance between the verbose barcodes of the VR FCCs induced by the respective pullbacks $(Z, \phi_X^* d_X)$ and $(Z, \phi_Y^* d_Y)$, where the infimum is taken over *tripods* $R : X \xleftarrow{\phi_X} Z \xrightarrow{\phi_Y} Y$. We denote the result by \hat{d}_B ; see Definition 24. Similarly, we define the *pullback interleaving distance* between two VR FCCs, and denote it by \hat{d}_I (see Definition 23).

► **Remark 3 (Terminology).** We point out the following regarding the use of the term “distance” when referring to \hat{d}_B and \hat{d}_I :

51:4 Ephemeral Persistence Features and the Stability of Filtered Chain Complexes

- (1) \hat{d}_B between degree-0 verbose barcodes satisfies the triangle inequality [17, Corollary 6.7].
- (2) The question whether \hat{d}_B between positive-degree verbose barcodes satisfies the triangle inequality is still open.
- (3) \hat{d}_I does not satisfy the triangle inequality; see [17, Remark 6.8] for details.

Due to Items (2) and (3), the term “distance” is being abused through the use of the terminology “pullback bottleneck distance” and “pullback interleaving distance”. We do so for consistency with Item (1) and due to the fact that in [17, Remark 6.8] we provide a way to modify \hat{d}_I and \hat{d}_B so that they do satisfy the triangle inequality (while still being Gromov-Hausdorff stable).

It is important to note that in general, the pullback bottleneck distance \hat{d}_B (or the pullback interleaving distance \hat{d}_I) depends on the underlying metric spaces, rather than solely on the verbose barcodes (or FCCs). Nonetheless, we use the current terminology to emphasize the roles of verbose barcodes and FCCs in our discussion.

It follows from Theorem 2 and the definitions of \hat{d}_B and \hat{d}_I that we have the following:

► **Corollary 4.** *Let (X, d_X) and (Y, d_Y) be two finite metric spaces. Then,*

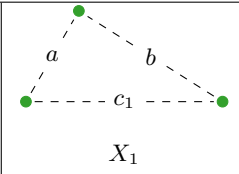
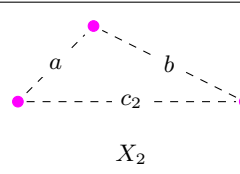
$$\sup_{k \in \mathbb{Z}_{\geq 0}} \hat{d}_B(\mathcal{B}_{\text{Ver},k}(X), \mathcal{B}_{\text{Ver},k}(Y)) \leq \hat{d}_I((C_*(\text{VR}(X)), \partial^X, \ell^X), (C_*(\text{VR}(Y)), \partial^Y, \ell^Y)).$$

In the theorem below, we show that the pullback bottleneck distance \hat{d}_B is stable under the Gromov-Hausdorff distance d_{GH} , and that the bottleneck distance d_B between concise barcodes is not larger than \hat{d}_B between verbose barcodes. We show in several examples below and in §4.3 that \hat{d}_B between verbose barcodes can be strictly larger than d_B between concise barcodes. Thus, the stability of \hat{d}_B improves the stability of the standard bottleneck distance d_B (cf. Theorem 1). See §4.2 for the proof of Theorem 5.

► **Theorem 5 (Pullback stability theorem).** *Let (X, d_X) and (Y, d_Y) be two finite metric spaces. Then, for any $k \in \mathbb{Z}_{\geq 0}$,*

$$d_B(\mathcal{B}_{\text{Con},k}(X), \mathcal{B}_{\text{Con},k}(Y)) \leq \hat{d}_B(\mathcal{B}_{\text{Ver},k}(X), \mathcal{B}_{\text{Ver},k}(Y)) \leq 2 \cdot d_{\text{GH}}(X, Y). \quad (1)$$

See Figure 1 for a pair of 3-point metric spaces which d_B between concise barcodes fails to distinguish, but the \hat{d}_B between verbose barcodes succeeds at telling apart.

		
	X_1	X_2
$\mathcal{B}_{\text{Ver},0}$	$(0, a), (0, b), (0, \infty)$	$(0, a), (0, b), (0, \infty)$
$\mathcal{B}_{\text{Ver},1}$	(c_1, c_1)	(c_2, c_2)

$d_B(\mathcal{B}_{\text{Con}}(X_1), \mathcal{B}_{\text{Con}}(X_2))$	$\hat{d}_B(\mathcal{B}_{\text{Ver}}(X_1), \mathcal{B}_{\text{Ver}}(X_2))$	$2 \cdot d_{\text{GH}}(X_1, X_2)$
0	$ c_1 - c_2 $	$ c_1 - c_2 $

■ **Figure 1** *First table:* three-point metric spaces X_1 and X_2 together with their verbose barcodes. Here $a \leq b \leq c_i$ for $i = 1, 2$. *Second table:* the bottleneck distance between concise barcodes, the pullback bottleneck distance between verbose barcodes and twice of the Gromov-Hausdorff distance between X_1 and X_2 . See Example 28.

In order to have a more concrete understanding of the pullback bottleneck distance and in order to explore the possibility of computing it, we study the relation between the verbose barcode of a pullback metric space $(Z, \phi_X^* d_X)$ with the verbose barcode of the original space X . We conclude that the verbose barcodes of Z and X only differ on some distinguished diagonal points; see Proposition 6 below.

We now set up some notation about multisets¹. For a non-negative integer m , by $\{x\}^m$ we will denote the multiset containing exactly m copies of x . For any multiset A and any $l \geq 1$, we let $P_l(A)$ be the multiset consisting of sub-multisets of A each with cardinality l .

► **Proposition 6** (Pullback barcodes). *Let $k \geq 0$, $m \geq 1$ and $Z = X \sqcup \{x_{j_1}, \dots, x_{j_m}\}$ for some $j_1 \leq \dots \leq j_m$. Then, for $k \geq 0$,*

$$\mathcal{B}_{\text{Ver},k}(Z) = \mathcal{B}_{\text{Ver},k}(X) \sqcup \bigsqcup_{i=0}^{m-1} \{ \text{diam}([x_{j_{i+1}}, \beta_i]) \cdot (1, 1) : \beta_i \in P_k((X \setminus \{x_{j_{i+1}}\}) \sqcup \{x_{j_1}, \dots, x_{j_i}\}) \}. \tag{2}$$

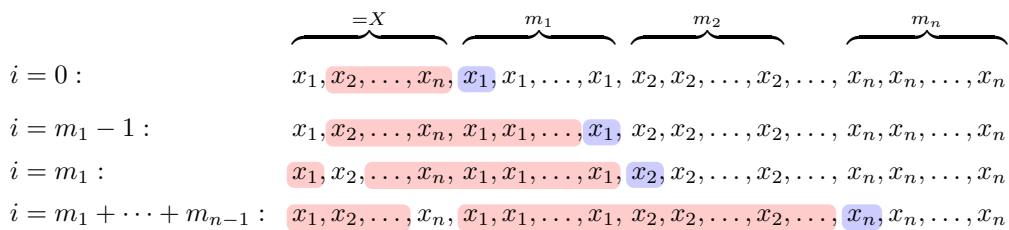
In particular, $\mathcal{B}_{\text{Ver},0}(Z) = \mathcal{B}_{\text{Ver},0}(X) \sqcup \{(0, 0)\}^m$.

Because concise barcodes can be obtained from verbose barcodes by excluding all diagonal points, the above proposition interestingly implies that $\mathcal{B}_{\text{Con},k}(Z) = \mathcal{B}_{\text{Con},k}(X)$ for any k .

To better understand Equation (2) in the case when $k \geq 1$, we give a graphical explanation in Figure 2. Let (X, d_X) be a finite metric space with $X = \{x_1, \dots, x_n\}$. Each finite pullback metric space $(Z, \phi_X^* d_X)$ of X can be written as a multiset $Z = X \sqcup \{x_{j_1}, \dots, x_{j_m}\}$ equipped with the pullback pseudo-metric $\phi_X^* d_X$ induced from d_X , for some $m \geq 0$ and $1 \leq j_1 \leq \dots \leq j_m \leq n$. In other words, the extra points in Z are “repeats” of the points in X . We will call each point in X the *parent* of its repeated copies: to be more precise, for each $z \in Z$, the point $\phi_X(z) \in X$ will be called the parent of z . Write

$$Z = X \sqcup \left\{ \underbrace{x_1, \dots, x_1}_{m_1}, \dots, \underbrace{x_n, \dots, x_n}_{m_n} \right\},$$

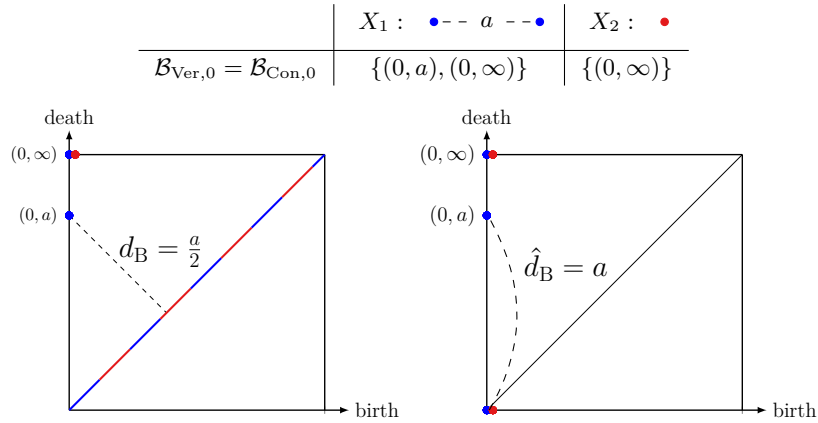
where each $m_j \geq 0$ is the multiplicity of the extra copies of x_j in Z and $m_1 + \dots + m_n = m$.



■ **Figure 2** With the same notation as in Equation (2), for each i (i.e. for each row), the point $x_{j_{i+1}}$ is colored in blue. For each i the multiset β_i in Equation (2) ranges over all k element sub-multisets of the red-colored multiset. Notice that each red-colored multiset consists of every point before $x_{j_{i+1}}$ (from left to right) excluding the parent of $x_{j_{i+1}}$.

We examine the relationship between \hat{d}_B and d_B , and obtain an interpretation of \hat{d}_B in terms of matchings of points in the barcodes. To compute d_B , one looks for an optimal matching where points from a barcode can be matched to any points on the diagonal. However,

¹ We use the notation $\{\cdot\}$ for multisets as well when its meaning is clear from the content.



■ **Figure 3** *Top:* X_1 a two-point space, X_2 the one-point space, and their 0-th verbose (or concise) barcode. *Bottom:* visualization of d_B and \hat{d}_B , where in both figures the point $(0, \infty)$ is matched with $(0, \infty)$ and the ℓ_∞ -metric is used to compute the distances.

in the computation of \hat{d}_B , points are only allowed to be matched to verbose barcodes and a *particular sub-multiset* of the diagonal points, where the choice of these diagonal points depends on the metric structure of the two underlying metric spaces.

For degree-0, since the verbose barcode of any pullback (pseudo-)metric space Z of X only differs from the verbose barcode of X in multiple copies of the point $(0, 0)$, the distance \hat{d}_B is indeed computing an optimal matching between concise barcodes which only allows bars to be matched to other bars or to the origin $(0, 0)$ (see Figure 3). Combined with the fact that degree-0 bars are all born at 0, we obtain the following explicit formula for computing the distance \hat{d}_B for degree-0 (see [17, §6.2.1] for the proof):

► **Proposition 7** (Pullback bottleneck distance in degree 0). *Let X and Y be two finite metric spaces such that $\text{card}(X) = n \leq n' = \text{card}(Y)$. Suppose the death time of finite-length degree-0 bars of X and Y are given by the sequences $a_1 \geq \dots \geq a_{n-1}$ and $b_1 \geq \dots \geq b_{n'-1}$, respectively. Then*

$$\hat{d}_B(\mathcal{B}_{\text{Ver},0}(X), \mathcal{B}_{\text{Ver},0}(Y)) = \max \left\{ \max_{1 \leq i \leq n-1} |a_i - b_i|, \max_{n \leq i \leq n'-1} b_i \right\}.$$

For higher degrees, the situation becomes more complicated because in addition to the point $(0, 0)$, other choices of diagonal points need to be considered, as evidenced by the formula for pullback barcodes in Proposition 6. We leave this as our future work.

2 Filtered chain complexes (FCCs)

We recall the notion of FCCs and provide some properties and examples for VR FCCs.

Usher and Zhang express FCCs as the triples $(C_*, \partial_C, \ell_C)$, where (C_*, ∂_C) denotes a chain complex and $\ell_C : C_* \rightarrow \mathbb{R} \sqcup \{-\infty\}$ is a *filtration* function such that (1) $\ell_C \circ \partial_C \leq \ell_C$, and (2) $\ell_C(x) = -\infty$ iff $x = 0$, $\ell_C(\lambda x) = \ell_C(x)$ for $\lambda \neq 0$, and $\ell_C(x+y) \leq \max\{\ell_C(x), \ell_C(y)\}$, $\forall x, y \in C$. A *morphism of FCCs* from $(C_*, \partial_C, \ell_C)$ to $(D_*, \partial_D, \ell_D)$ is a chain map $\Phi_* : C_* \rightarrow D_*$ that is *filtration preserving*, i.e. $\ell_D \circ \Phi_* \leq \ell_C$. Let **FCC** denote the category of FCCs. We refer readers to [18] or [17, §3] for more details about general FCCs.

VR FCCs. A pseudo-metric d_X on X is a function $d_X : X \times X \rightarrow [0, +\infty)$ satisfying the axioms for a metric, except that different points are allowed to have distance 0.

Given a finite pseudo-metric space (X, d_X) and $\epsilon \geq 0$, the ϵ -Vietoris-Rips complex $\text{VR}_\epsilon(X)$ is the simplicial complex with vertex set X , where

$$\text{a finite subset } \sigma \subset X \text{ is a simplex of } \text{VR}_\epsilon(X) \iff \text{diam}(\sigma) \leq \epsilon.$$

Let $\text{diam}(X)$ be the diameter of X . Let $\text{VR}(X) := \text{VR}_{\text{diam}(X)}(X)$, which is the full complex on X . For each $k \in \mathbb{Z}_{\geq 0}$, we denote by $C_k(\text{VR}(X))$ the free \mathbb{F} -vector space generated by k -simplices in $\text{VR}(X)$, and let $C_*(\text{VR}(X))$ be the free simplicial chain complex induced by $\text{VR}(X)$ over coefficients in \mathbb{F} , with the standard simplicial boundary operator ∂^X . Notice that up to homotopy equivalence the simplicial complex $\text{VR}(X)$ only depends on the cardinality of X , so does the chain complex $(C_*(\text{VR}(X)), \partial^X)$.

Define the filtration function $\ell^X : C_*(\text{VR}(X)) \rightarrow \mathbb{R} \sqcup \{-\infty\}$ by

$$\ell^X \left(\sum_{i=1}^r \lambda_i \sigma_i \right) := \max_{\lambda_i \neq 0} \{\text{diam}(\sigma_i)\},$$

where the σ_i are simplices, and $\ell^X(0) := -\infty$. Then $(C_*(\text{VR}(X)), \partial^X, \ell^X)$ is an FCC.

2.1 Verbose and concise barcodes

For a vector space equipped with a filtration function ℓ , a finite collection (x_1, \dots, x_r) of elements C is said to be (ℓ) -orthogonal if, for all $\lambda_1, \dots, \lambda_r \in \mathbb{F}$,

$$\ell \left(\sum_{i=1}^r \lambda_i x_i \right) = \max_{\lambda_i \neq 0} \ell(x_i).$$

Let $A : C \rightarrow D$ be a linear map with rank r . A (unsorted) singular value decomposition of A is a choice of orthogonal ordered bases (y_1, \dots, y_n) for C and (x_1, \dots, x_m) for D such that (see [18, Definition 3.1]):

- (y_{r+1}, \dots, y_n) is an orthogonal ordered basis for $\text{Ker } A$;
- (x_1, \dots, x_r) is an orthogonal ordered basis for $\text{Im } A$;
- $Ay_i = x_i$ for $i = 1, \dots, r$.

The existence of a singular value decomposition for linear maps between finite-dimensional orthogonalizable \mathbb{F} -spaces is guaranteed by [18, Theorem 3.4].

► **Definition 8** (Verbose barcode and concise barcode, [18, Definition 6.3]). Let $(C_*, \partial_C, \ell_C)$ be an FCC over \mathbb{F} and for each $k \in \mathbb{Z}$ write $\partial_k = \partial_C|_{C_k}$. Given any $k \in \mathbb{Z}$ choose a singular value decomposition $((y_1, \dots, y_n), (x_1, \dots, x_m))$ for the \mathbb{F} -linear map $\partial_{k+1} : C_{k+1} \rightarrow \text{Ker } \partial_k$ and let r denote the rank of ∂_{k+1} . Then the degree- k verbose barcode of $(C_*, \partial_C, \ell_C)$ is the multiset $\mathcal{B}_{\text{Ver},k}^{(C_*, \partial_C, \ell_C)}$ (or $\mathcal{B}_{\text{Ver},k}^C$ for simplicity) of elements of $\mathbb{R} \times [0, \infty]$ consisting of

- a pair $(\ell(x_i), \ell(y_i))$ for each $i = 1, \dots, r = \text{rank}(\partial_{k+1})$; and
- a pair $(\ell(x_i), \infty)$ for each $i = r + 1, \dots, m = \dim(\text{Ker } \partial_k)$.

The concise barcode of $(C_*, \partial_C, \ell_C)$ is the submultiset of the verbose barcode consisting of those elements where $\ell(y_i) - \ell(x_i) \neq 0$.

► **Remark 9.** Let X be a finite metric space. The degree-0 verbose barcode $\mathcal{B}_{\text{Ver},0}$ and the degree-0 concise barcode $\mathcal{B}_{\text{Con},0}$ of the VR FCC $(C_*(\text{VR}(X)), \partial^X, \ell^X)$ are the same. Notice that this is not necessarily true for pseudo-metric spaces, in which case verbose barcode may contain several copies of $(0, 0)$.

► **Example 10** (Verbose barcodes of VR FCCs). Let $n := \text{card}(X)$. The number of k -verbose barcodes (with multiplicity) of the VR FCC of a finite pseudo-metric space X is

$$\text{card}(\mathcal{B}_{\text{Ver},k}(X)) = \dim(\text{Ker}(\partial_k)) = \begin{cases} n, & k = 0, \\ \binom{n-1}{k+1}, & \text{for } 1 \leq k \leq n-2, \\ 0, & \text{for } k \geq n-1. \end{cases}$$

2.2 Decomposition of FCCs

We recall from [18] that the collection of verbose barcodes is a *complete* invariant of FCCs, because every FCC decomposes uniquely up to isomorphism into the following form:

$$(C_*, \partial_C, \ell_C) \cong \bigoplus_{k \in \mathbb{Z}} \bigoplus_{(a, a+L) \in \mathcal{B}_{\text{Ver},k}} \mathcal{E}(a, a+L, k).$$

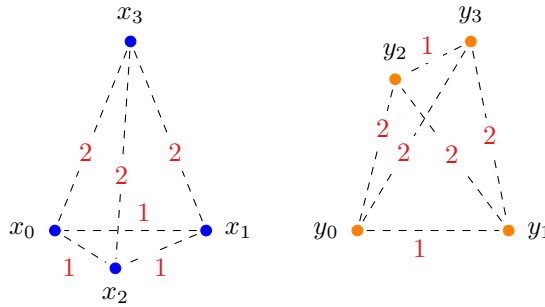
Also, the collection of concise barcodes is an invariant up to the so-called filtered homotopy equivalence. In addition, for the case of VR FCCs, we show that isometry implies filtered chain isomorphism while the inverse is not true.

For the purpose of this paper, we use the theorem below as definitions of *filtered chain isomorphism (f.c.i.)* and *filtered homotopy equivalence (f.h.e.)* between FCCs, and refer readers to [18] for the original definitions of these two concepts.

► **Theorem 11** (Theorem A & B, [18]). *Two FCCs $(C_*, \partial_C, \ell_C)$ and $(D_*, \partial_D, \ell_D)$ are*

1. *filtered chain isomorphic iff they have identical verbose barcodes in all degrees;*
2. *filtered homotopy equivalent iff they have identical concise barcodes in all degrees.*

► **Example 12** (f.h.e. but not f.c.i.). Let X and Y be (ultra-)metric spaces of 4 points given in Figure 4. The FCCs $(C_*(\text{VR}(X)), \partial^X, \ell^X)$ and $(C_*(\text{VR}(Y)), \partial^Y, \ell^Y)$ arising from Vietoris-Rips complexes have the same concise barcodes but different verbose barcodes.



■ **Figure 4** Four-point metric spaces X (left) and Y (right).

We compute from Definition 8 that the verbose barcodes of X and Y are

$$\mathcal{B}_{\text{Ver},k}(X) = \begin{cases} \{(0, 1), (0, 1), (0, 2), (0, \infty)\}, & k = 0 \\ \{(1, 1), (2, 2), (2, 2)\}, & k = 1 \\ \{(2, 2)\}, & k = 2 \\ \emptyset, & \text{otherwise.} \end{cases}$$

and

$$\mathcal{B}_{\text{Ver},k}(Y) = \begin{cases} \{(0, 1), (0, 1), (0, 2), (0, \infty)\}, & k = 0 \\ \{(2, 2), (2, 2), (2, 2)\}, & k = 1 \\ \{(2, 2)\}, & k = 2 \\ \emptyset, & \text{otherwise,} \end{cases}$$

respectively. The concise barcodes of X and Y are

$$\mathcal{B}_{\text{Con},k}(X) = \mathcal{B}_{\text{Con},k}(Y) = \begin{cases} \{(0, 1), (0, 1), (0, 2), (0, \infty)\}, & k = 0 \\ \emptyset, & \text{otherwise.} \end{cases}$$

Let (X, d_X) and (Y, d_Y) be two finite pseudo-metric spaces with $|X| = |Y|$. Then, any bijection $f : X \rightarrow Y$ induces a chain isomorphism $f_* : C_*(\text{VR}(X)) \xrightarrow{\cong} C_*(\text{VR}(Y))$. It is not difficult to check that the respective VR FCCs of two isometric pseudo-metric spaces are filtered chain isomorphic.

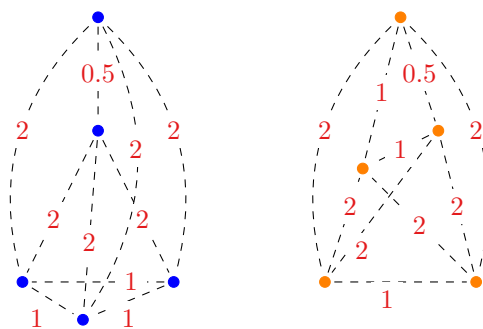
► **Proposition 13** (Isometry implies f.c.i.). *Let (X, d_X) and (Y, d_Y) be two finite pseudo-metric spaces. If (X, d_X) and (Y, d_Y) are isometric, then the FCCs $(C_*(\text{VR}(X)), \partial^X, \ell^X)$ and $(C_*(\text{VR}(Y)), \partial^Y, \ell^Y)$ are filtered chain isomorphic.*

However, the converse of Proposition 13 is not true.

► **Example 14** (f.c.i. but not isometric). Let X and Y be (ultra-)metric spaces of 5 points given in Figure 5. The distance matrices for X and Y are, respectively:

$$\begin{pmatrix} 0 & 0.5 & 2 & 2 & 2 \\ 0.5 & 0 & 2 & 2 & 2 \\ 2 & 2 & 0 & 1 & 1 \\ 2 & 2 & 1 & 0 & 1 \\ 2 & 2 & 1 & 1 & 0 \end{pmatrix} \text{ and } \begin{pmatrix} 0 & 0.5 & 1 & 2 & 2 \\ 0.5 & 0 & 1 & 2 & 2 \\ 1 & 1 & 0 & 2 & 2 \\ 2 & 2 & 2 & 0 & 1 \\ 2 & 2 & 2 & 1 & 0 \end{pmatrix}.$$

Notice that X and Y are not isometric. Indeed, in Y every vertex belongs to an edge of length 1, but the top point in X only belongs to edges of length 0.5 and 2.



■ **Figure 5** Five-point metric spaces X (left) and Y (right).

However, the VR FCCs of X and Y have the same verbose barcodes:

$$\mathcal{B}_{\text{Ver},k}(X) = \mathcal{B}_{\text{Ver},k}(Y) = \begin{cases} \{(0, 0.5), (0, 1), (0, 1), (0, 2), (0, \infty)\}, & k = 0 \\ \{(1, 1), (2, 2), (2, 2), (2, 2), (2, 2), (2, 2)\}, & k = 1 \\ \{(2, 2), (2, 2), (2, 2), (2, 2)\}, & k = 2 \\ \{(2, 2)\}, & k = 3 \\ \emptyset, & \text{otherwise.} \end{cases}$$

3 Isometry theorem ($d_I = d_M$)

In TDA, it is well-known that, under mild conditions (e.g. q -tameness, see [9]), an isometry theorem holds: the interleaving distance between persistence modules is equal to the bottleneck distance between their *concise* barcodes (cf. [11, 7, 11]). In our notation, this means that for any degree k and any two FCCs $(C_*, \partial_C, \ell_C)$ and $(D_*, \partial_D, \ell_D)$,

$$d_B(\mathcal{B}_{\text{Con},k}^C, \mathcal{B}_{\text{Con},k}^D) = d_I(\mathbb{H}_k \circ (C_*, \partial_C, \ell_C), \mathbb{H}_k \circ (D_*, \partial_D, \ell_D)).$$

We prove an analogous isometry theorem for the *verbose* barcode, i.e., Theorem 2.

3.1 Interleaving distance d_I between FCCs

For detailed proofs of results in this subsection, see [17, §4.1]. Let d_I be the categorical interleaving distance in the category of filtered chain complexes given by [2, Definition 3.2].

► **Proposition 15.** *Let $(C_*, \partial_C, \ell_C)$ and $(D_*, \partial_D, \ell_D)$ be two FCCs. Then*

$$d_I((C_*, \partial_C, \ell_C), (D_*, \partial_D, \ell_D)) < \infty \iff (C_*, \partial_C) \cong (D_*, \partial_D).$$

Because of Proposition 15, the interleaving distance between FCCs is only interesting when we consider the case when two FCCs have the *same* underlying chain complexes. Let (C_*, ∂_C) be a finite-dimensional non-zero chain complex over \mathbb{F} , and let $\text{Iso}((C_*, \partial_C))$ be the set of chain isomorphisms on (C_*, ∂_C) .

► **Theorem 16.** *Let (C_*, ∂_C) be a non-zero chain complex and let $\ell_1, \ell_2 : C_* \rightarrow \mathbb{R} \sqcup \{-\infty\}$ be two filtration functions such that both $(C_*, \partial_C, \ell_1)$ and $(C_*, \partial_C, \ell_2)$ are FCCs. Then*

$$d_I((C_*, \partial_C, \ell_1), (C_*, \partial_C, \ell_2)) = \inf_{\Phi_* \in \text{Iso}(C_*, \partial_C)} \|\ell_1 - \ell_2 \circ \Phi_*\|_\infty.$$

Here we follow the convention $(-\infty) - (-\infty) = 0$ when computing $\|\ell_1 - \ell_2\|_\infty$. When ℓ_1 is the trivial filtration function, we have $d_I((C_*, \partial_C, \ell_1), (C_*, \partial_C, \ell_2)) = \|\ell_2\|_\infty$.

► **Example 17** (d_I between Elementary FCCs). For $L_a, L_b < \infty$, the interleaving distance between elementary FCCs $\mathcal{E}(a, a + L_a, k)$ and $\mathcal{E}(b, b + L_b, l)$ is finite iff $k = l$. And

$$d_I(\mathcal{E}(a, a + L_a, k), \mathcal{E}(b, b + L_b, k)) = \max\{|a - b|, |(a + L_a) - (b + L_b)|\}.$$

3.2 Matching distance d_M between verbose barcodes

Let $\mathcal{H} := \{(p, q) : 0 \leq p < q \leq \infty\}$, and let $\Delta := \{(r, r) : r \in \mathbb{R}_{\geq 0} \sqcup \{+\infty\}\}$. We denote $\overline{\mathcal{H}} := \mathcal{H} \sqcup \Delta$ the extended real upper plane. Let d_∞ be the metric on $\overline{\mathcal{H}}$ inherited from the l_∞ -metric, where for $p, q, p', q' \in \mathbb{R}_{\geq 0} \sqcup \{\infty\}$,

$$d_\infty((p, q), (p', q')) = \begin{cases} \max\{|p - p'|, |q - q'|\}, & q, q' < \infty, \\ |p - p'|, & q = q' = \infty, \\ \infty, & \text{otherwise.} \end{cases}$$

Denote by Δ^∞ (or \mathcal{H}^∞ and $\overline{\mathcal{H}}^\infty$, respectively) the multiset consisting of each point on Δ (or \mathcal{H} and $\overline{\mathcal{H}}$, respectively), taken with (countably) infinite multiplicity. Let $\overline{\mathcal{H}}^\infty$ be equipped with the metric d_∞ inherited from $\overline{\mathcal{H}}$.

► **Definition 18** (The Matching Distance d_M). *Let A and B be two non-empty sub-multisets of $\overline{\mathcal{H}}^\infty$. The matching distance between A and B is*

$$d_M(A, B) := \min \left\{ \max_{a \in A} d(a, \phi(a)) : A \xrightarrow{\phi} B \text{ a bijection} \right\},$$

where $d_M(A, B) = \infty$ if $\text{card}(A) \neq \text{card}(B)$.

► **Definition 19** (The Bottleneck Distance d_B). *Let A and B be two finite non-empty sub-multisets of \mathcal{H}^∞ . The bottleneck distance between A and B is*

$$d_B(A, B) := d_M(A \sqcup \Delta^\infty, B \sqcup \Delta^\infty).$$

Unlike the bottleneck distance d_B , the diagonal points that can be matched in d_M between verbose barcodes are limited (see also [17, Proposition 4.11]). Thus,

► **Proposition 20.** *Given two FCCs $(C_*, \partial_C, \ell_C)$ and $(D_*, \partial_D, \ell_D)$ and any degree k , we have*

$$d_B(\mathcal{B}_{\text{Con},k}^C, \mathcal{B}_{\text{Con},k}^D) \leq d_M(\mathcal{B}_{\text{Ver},k}^C, \mathcal{B}_{\text{Ver},k}^D).$$

Given $(C_*, \partial_C, \ell_C)$ and a chain isomorphism Φ_* on (C_*, ∂_C) , because (C_*, ∂_C, ℓ) and $(C_*, \partial_C, \ell \circ \Phi_*)$ are filtered chain isomorphic, they have the same verbose barcode (see [17, Proposition 4.14]). By checking that Φ_* maps a singular value decomposition of (C_*, ∂_C, ℓ) to a singular value decomposition of $(C_*, \partial_C, \ell \circ \Phi_*)$, we see that chain isomorphisms induce permutations of verbose barcodes. For more details, see [17, §4.2].

3.3 Proof of the isometry theorem

We now prove Theorem 2. If two FCCs have non-isomorphic underlying chain complexes, then d_I between the two FCCs is ∞ , and so is d_M between their verbose barcodes. Thus, it remains to consider the case when two FCCs have the same (or isomorphic) underlying chain complexes.

3.3.1 The inequality $d_M \leq d_I$

Although [18, Proposition 9.3] states a weaker result than the lemma below, their proof indeed implies the following (see [17, §4.3.1] for more details):

► **Lemma 21.** *Let (C_*, ∂_C) be a finite-dimensional non-zero chain complex over \mathbb{F} and let $\ell_1, \ell_2 : C_* \rightarrow \mathbb{R} \sqcup \{-\infty\}$ be two filtration functions. Denote by $\mathcal{B}_{\text{Ver}}^1$ and $\mathcal{B}_{\text{Ver}}^2$ the verbose barcodes of $(C_*, \partial_C, \ell_1)$ and $(C_*, \partial_C, \ell_2)$, respectively. Then, we have*

$$d_M(\mathcal{B}_{\text{Ver}}^1, \mathcal{B}_{\text{Ver}}^2) = \sup_{k \in \mathbb{Z}_{\geq 0}} d_M(\mathcal{B}_{\text{Ver},k}^1, \mathcal{B}_{\text{Ver},k}^2) \leq \|\ell_1 - \ell_2\|_\infty.$$

► **Proposition 22.** *With the same notation as in Lemma 21, we have*

$$d_M(\mathcal{B}_{\text{Ver}}^1, \mathcal{B}_{\text{Ver}}^2) \leq d_I((C_*, \partial_C, \ell_1), (C_*, \partial_C, \ell_2)).$$

51:12 Ephemeral Persistence Features and the Stability of Filtered Chain Complexes

Proof. Given any $\Phi_* \in \text{Iso}(C_*, \partial_C)$, [17, Proposition 4.14] implies that $\mathcal{B}_{\text{Ver}}^2 = \mathcal{B}_{\text{Ver}}^{(C_*, \partial_C, \ell_2 \circ \Phi_*)}$ agrees with the verbose barcodes of $(C_*, \partial_C, \ell_2 \circ \Phi_*)$. Combined with Lemma 21, we have

$$d_M(\mathcal{B}_{\text{Ver}}^1, \mathcal{B}_{\text{Ver}}^2) = d_M(\mathcal{B}_{\text{Ver}}^1, \mathcal{B}_{\text{Ver}}^{(C_*, \partial_C, \ell_2 \circ \Phi_*)}) \leq \|\ell_1 - \ell_2 \circ \Phi_*\|_\infty,$$

for any $\Phi_* \in \text{Iso}(C_*, \partial_C)$. Therefore,

$$d_M(\mathcal{B}_{\text{Ver}}^1, \mathcal{B}_{\text{Ver}}^2) \leq \min_{\Phi_* \in \text{Iso}(C_*, \partial_C)} \|\ell_1 - \ell_2 \circ \Phi_*\|_\infty = d_I((C_*, \partial_C, \ell_1), (C_*, \partial_C, \ell_2)),$$

where the equality follows from Theorem 16. ◀

3.3.2 The inequality $d_M \geq d_I$

We prove $d_M \geq d_I$ via an idea similar to the one used for proving that d_B of concise barcodes is no larger than d_I between persistent modules, cf. [14, Theorem 3.4].

Proof of Theorem 2 “ $d_M \geq d_I$ ”. The proof is trivial if $(C_*, \partial_C, \ell_C)$ and $(D_*, \partial_D, \ell_D)$ have non-isomorphic underlying chain complexes. We now consider the case when the chain complexes (C_*, ∂_C) and (D_*, ∂_D) are isomorphic, and we assume without loss of generality that $(D_*, \partial_D) = (C_*, \partial_C)$ and write $\ell_1 := \ell_C, \ell_2 := \ell_D$.

Take any number $\delta \geq d_M(\mathcal{B}_{\text{Ver}}^1, \mathcal{B}_{\text{Ver}}^2)$. Then for any $k \in \mathbb{Z}_{\geq 0}$, there is a bijection $f_k : \mathcal{B}_{\text{Ver},k}^1 \rightarrow \mathcal{B}_{\text{Ver},k}^2$ such that

$$\max_{a \in \mathcal{B}_{\text{Ver},k}^1} d_\infty(a, f_k(a)) \leq \delta. \quad (3)$$

For $a \in \mathcal{B}_{\text{Ver},k}^1 \subset \overline{\mathcal{H}}^\infty$, assume that $a = (a_1, a_2)$. Also, write $b = f_k(a)$ and assume that $b = (b_1, b_2)$. Next we construct an isomorphism between the following elementary FCCs:

$$h_k : \mathcal{E}(a_1, a_2, k) \rightarrow \mathcal{E}(b_1, b_2, k).$$

Notice that a_2 and b_2 are either both finite or both infinite, otherwise the left hand side of Equation (3) is equal to ∞ , which contradicts with $\delta < \infty$.

Case (1): $a_2 = b_2 = \infty$, so $\mathcal{E}(a_1, a_2, k)$ and $\mathcal{E}(b_1, b_2, k)$ have the same underlying chain complex:

$$\dots \longrightarrow 0 \longrightarrow \mathbb{F}x_k \xrightarrow{\partial_k=0} 0 \longrightarrow \dots,$$

and the filtration functions are given by $\ell_1(x_k) = a_1$ and $\ell_2(x_k) = b_1$, respectively. We define the chain isomorphism to be

$$h_k : \mathcal{E}(a_1, \infty, k) \rightarrow \mathcal{E}(b_1, \infty, k) \text{ with } x_k \mapsto x_k.$$

Case (2): $a_2, b_2 < \infty$, so $\mathcal{E}(a_1, a_2, k)$ and $\mathcal{E}(b_1, b_2, k)$ have the same underlying chain complex:

$$\dots \longrightarrow 0 \longrightarrow \mathbb{F}y_{k+1} \xrightarrow{\partial_{k+1}: y_{k+1} \mapsto x_k} \mathbb{F}x_k \xrightarrow{\partial_k=0} 0 \longrightarrow \dots,$$

and the filtration functions are given by $\ell_1(x_k) = a_1$, $\ell_1(y_{k+1}) = a_2$ and $\ell_2(x_k) = b_1$, $\ell_2(y_{k+1}) = b_2$, respectively. We define the chain isomorphism to be

$$h_k : \mathcal{E}(a_1, a_2, k) \rightarrow \mathcal{E}(b_1, b_2, k) \text{ with } x_k \mapsto x_k, y_{k+1} \mapsto y_{k+1}.$$

In either case, it is straightforward to check that h_k satisfies the following condition

$$\|\ell_1 - \ell_2 \circ h_k\|_\infty \leq \max\{|a_1 - a_2, b_1 - b_2|\} = d_\infty(a, f(a)) \leq \delta.$$

We write $h_{k,a}$ whenever it is needed to emphasize that h_k depends on a .

By [18, Proposition 7.4] we have the following decomposition of FCCs

$$(C_*, \partial_C, \ell_1) \cong \bigoplus_{k \in \mathbb{Z}_{\geq 0}} \bigoplus_{a \in \mathcal{B}_{\text{Ver},k}^1} \mathcal{E}(a_1, a_2, k) \text{ and } (C_*, \partial_C, \ell_2) \cong \bigoplus_{k \in \mathbb{Z}_{\geq 0}} \bigoplus_{b \in \mathcal{B}_{\text{Ver},k}^2} \mathcal{E}(b_1, b_2, k).$$

Let $h := \bigoplus_{k \in \mathbb{Z}_{\geq 0}} \bigoplus_{a \in \mathcal{B}_{\text{Ver},k}^1} h_{k,a} : (C_*, \partial_C, \ell_1) \rightarrow (C_*, \partial_C, \ell_2)$, which is then a chain isomorphism such that

$$\|\ell_1 - \ell_2 \circ h\|_\infty = \max_{k \in \mathbb{Z}_{\geq 0}} \max_{a \in \mathcal{B}_{\text{Ver},k}^1} \|\ell_1 - \ell_2 \circ h_{k,a}\|_\infty \leq \delta.$$

It then follows from Theorem 16 that

$$d_1((C_*, \partial_C, \ell_1), (C_*, \partial_C, \ell_2)) = \min_{\Phi_* \in \text{Iso}(C_*, \partial_C)} \|\ell_1 - \ell_2 \circ \Phi_*\|_\infty \leq \|\ell_1 - \ell_2 \circ h\|_\infty \leq \delta.$$

Letting $\delta \searrow d_M(\mathcal{B}_{\text{Ver}}^1, \mathcal{B}_{\text{Ver}}^2)$, we obtain the desired inequality $d_1 \leq d_M$. ◀

4 Improved stability result for VR FCCs

In this section, we overcome the problem that the matching distance between verbose barcodes are not stable under the Gromov-Hausdorff distance, by incorporating the notion of tripods (see [16]). A *tripod* between two sets X and Y is a pair of surjections from another set Z to X and Y respectively, cf. [16]. We will express this by the diagram

$$R : X \xleftarrow{\phi_X} Z \xrightarrow{\phi_Y} Y.$$

Define $\text{dis}(R) := \sup_{z, z' \in Z} |d_X(\phi_X(z), \phi_X(z')) - d_Y(\phi_Y(z), \phi_Y(z'))|$.

In §4.1, we define the *pullback bottleneck distance* between verbose barcodes of VR FCCs of two finite metric spaces X and Y and the *pullback interleaving distance* between VR FCCs. We prove the pullback stability theorem (Theorem 5) in §4.2, and provide examples in §4.3 to show that verbose barcodes improve the stability of concise barcodes in many cases.

For notational simplicity, we will omit the differential map ∂^X for VR FCC of X .

4.1 Pullback interleaving distance and pullback bottleneck distance

Using the notion of tripod, we construct a new distance between filtered chain complexes:

► **Definition 23** (Pullback interleaving distance). *For two finite metric spaces X and Y , we define the pullback interleaving distance between the VR FCCs of X and Y to be*

$$\hat{d}_1((C_*(\text{VR}(X)), \ell^X), (C_*(\text{VR}(Y)), \ell^Y)) := \inf \left\{ d_1((C_*(\text{VR}(Z)), \ell^{Z_X}), (C_*(\text{VR}(Z)), \ell^{Z_Y})) \mid X \xleftarrow{\phi_X} Z \xrightarrow{\phi_Y} Y \text{ a tripod} \right\},$$

where $Z_X := (Z, \phi_X^* d_X)$ and $Z_Y := (Z, \phi_Y^* d_Y)$.

With a similar idea and again using tripods, we refine the standard bottleneck distance and introduce a new notion of distance between verbose barcodes:

► **Definition 24** (Pullback bottleneck distance). *Let $k \in \mathbb{Z}_{\geq 0}$. For two finite metric spaces X and Y , the pullback bottleneck distance between $\mathcal{B}_{\text{Ver},k}(X)$ and $\mathcal{B}_{\text{Ver},k}(Y)$ is defined to be*

$$\hat{d}_{\text{B}}((\mathcal{B}_{\text{Ver},k}(X), \mathcal{B}_{\text{Ver},k}(Y))) := \inf \left\{ d_{\text{M}}(\mathcal{B}_{\text{Ver},k}(Z_X), \mathcal{B}_{\text{Ver},k}(Z_Y)) \mid X \xleftarrow{\phi_X} Z \xrightarrow{\phi_Y} Y \text{ a tripod} \right\},$$

where $Z_X := (Z, \phi_X^* d_X)$ and $Z_Y := (Z, \phi_Y^* d_Y)$. In addition, we define

$$\hat{d}_{\text{B}}(\mathcal{B}_{\text{Ver}}(X), \mathcal{B}_{\text{Ver}}(Y)) := \sup_{k \in \mathbb{Z}_{\geq 0}} \hat{d}_{\text{B}}(\mathcal{B}_{\text{Ver},k}(X), \mathcal{B}_{\text{Ver},k}(Y)).$$

We refer readers to Remark 3 for clarification regarding the usage of terminology, especially the term “distance”, when referring to \hat{d}_{B} and \hat{d}_{I} .

► **Remark 25.** For two finite metric spaces X and Y with the same cardinality, we have

$$\hat{d}_{\text{B}}(\mathcal{B}_{\text{Ver}}(X), \mathcal{B}_{\text{Ver}}(Y)) \leq d_{\text{M}}(\mathcal{B}_{\text{Ver}}(X), \mathcal{B}_{\text{Ver}}(Y)).$$

The above inequality can be strict. For instance, consider the four-point metric spaces X and Y given in Example 12, for which we have (see [17, Remark 5.4])

$$\hat{d}_{\text{B}}(\mathcal{B}_{\text{Ver}}(X), \mathcal{B}_{\text{Ver}}(Y)) = 0 < 1 = d_{\text{M}}(\mathcal{B}_{\text{Ver}}(X), \mathcal{B}_{\text{Ver}}(Y)).$$

4.2 Pullback stability theorem

In this section, we prove that the pullback interleaving distance \hat{d}_{I} and the pullback bottleneck distance \hat{d}_{B} are stable under the Gromov-Hausdorff distance d_{GH} (cf. Theorem 5) and see that it improves that stability of the standard bottleneck distance d_{B} (cf. Theorem 1).

We first show that \hat{d}_{I} is stable.

► **Proposition 26** (Stability of Pullback Interleaving Distance). *Let (X, d_X) and (Y, d_Y) be two finite metric spaces. Then,*

$$\hat{d}_{\text{I}}((C_*(\text{VR}(X)), \ell^X), (C_*(\text{VR}(Y)), \ell^Y)) \leq 2 \cdot d_{\text{GH}}(X, Y).$$

Corollary 4 and Proposition 26 together yield the stability of \hat{d}_{B} . In addition, we prove that \hat{d}_{B} is an improvement of d_{B} , as lower bounds of d_{GH} between metric spaces:

Proof of Theorem 5. It remains to prove $d_{\text{B}}(\mathcal{B}_{\text{Con},k}(X), \mathcal{B}_{\text{Con},k}(Y)) \leq \hat{d}_{\text{B}}(\mathcal{B}_{\text{Ver},k}(X), \mathcal{B}_{\text{Ver},k}(Y))$. For any tripod $X \xleftarrow{\phi_X} Z \xrightarrow{\phi_Y} Y$, let $Z_X := (Z, \phi_X^* d_X)$ and $Z_Y := (Z, \phi_Y^* d_Y)$. By Proposition 6 and the fact that concise barcode is the corresponding verbose barcode excluding the diagonal points, we have that $\mathcal{B}_{\text{Con},k}(X) = \mathcal{B}_{\text{Con},k}(Z_X)$ and $\mathcal{B}_{\text{Con},k}(Y) = \mathcal{B}_{\text{Con},k}(Z_Y)$. Combined with Proposition 20, we have

$$d_{\text{B}}(\mathcal{B}_{\text{Con},k}(X), \mathcal{B}_{\text{Con},k}(Y)) = d_{\text{B}}(\mathcal{B}_{\text{Con},k}(Z_X), \mathcal{B}_{\text{Con},k}(Z_Y)) \leq d_{\text{M}}(\mathcal{B}_{\text{Ver},k}(Z_X), \mathcal{B}_{\text{Ver},k}(Z_Y)).$$

◀

To prove Proposition 26, we first establish the stability of the interleaving distance between VR FCCs by showing that it is stable under the ℓ_{∞} metric between two metrics over the same underlying set.

► **Proposition 27.** Let X be a finite set. Let d_1 and d_2 be two distance functions on X , and let ℓ^1 and ℓ^2 be the filtration functions induced by d_1 and d_2 respectively. Then,

$$|\|d_1\|_\infty - \|d_2\|_\infty| \leq d_I((C_*(\text{VR}(X)), \ell^1), (C_*(\text{VR}(X)), \ell^2)) \leq \|d_1 - d_2\|_\infty.$$

Proof of Proposition 26. Suppose $R : X \xleftarrow{\phi_X} Z \xrightarrow{\phi_Y} Y$ is a tripod between X and Y with distortion $\text{dis}(R) \leq \delta$. By Proposition 27, we obtain

$$d_I((C_*(\text{VR}(Z)), \ell^{Z_X}), (C_*(\text{VR}(Z)), \ell^{Z_Y})) \leq \|\phi_X^* d_X - \phi_Y^* d_Y\|_\infty = \text{dis}(R).$$

We finish the proof, by taking infimum over all tripods R on the above inequality and using the fact that $2 \cdot d_{\text{GH}}(X, Y) = \inf_R \text{dis}(R)$ (see [3, §7.3.3]). ◀

See [17, §5.3] for the remaining proofs and examples for results in this subsection.

4.3 Tightness and strictness of the pullback stability theorem

We show through examples that both inequalities in Theorem 5 are tight and can be strict.

► **Example 28.** Recall the 3-point metric spaces X_1 and X_2 from Figure 1, assuming $a \leq b \leq c_i$ for $i = 1, 2$. Computing each of the distance given in Theorem 5, we obtain:

$d_B(\mathcal{B}_{\text{Con}}(X_1), \mathcal{B}_{\text{Con}}(X_2))$	$\hat{d}_B(\mathcal{B}_{\text{Ver}}(X_1), \mathcal{B}_{\text{Ver}}(X_2))$	$2 \cdot d_{\text{GH}}(X_1, X_2)$
0	$ c_1 - c_2 $	$ c_1 - c_2 $

The first and third column in the above table are straightforward calculations. For the second column, notice that for any tripod $X_1 \xleftarrow{\phi_1} Z \xrightarrow{\phi_2} X_2$, we have

$$\mathcal{B}_{\text{Ver}, \text{card}(Z)-2}(Z_1) = \{(c_1, c_1)\} \text{ and } \mathcal{B}_{\text{Ver}, \text{card}(Z)-2}(Z_2) = \{(c_2, c_2)\},$$

where $Z_1 := (Z, \phi_1^* d_{X_1})$ and $Z_2 := (Z, \phi_2^* d_{X_2})$. Thus,

$$\hat{d}_B(\mathcal{B}_{\text{Ver}}(X_1), \mathcal{B}_{\text{Ver}}(X_2)) \geq d_M(\mathcal{B}_{\text{Ver}, \text{card}(Z)-2}(Z_1), \mathcal{B}_{\text{Ver}, \text{card}(Z)-2}(Z_2)) = |c_1 - c_2|.$$

This example shows that \hat{d}_B between verbose barcodes gives a better bound for the Gromov-Hausdorff distance d_{GH} , compared with d_B between concise barcodes.

► **Example 29.** Let X and Y be metric spaces of 4 points given in Figure 4. Let Z be the complete graph on 4 vertices with edge length 1, and W be the cycle graph on 4 vertices with edge length 1. See Figure 6 for the illustration of all 4 spaces and their verbose barcodes.

From Figure 7, we notice that the pair of metric spaces (X, Y) is such that

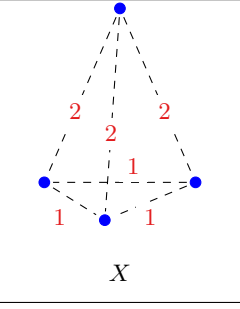
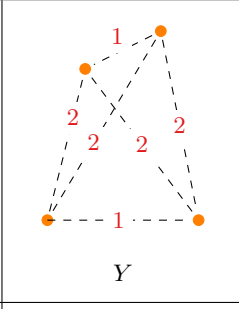
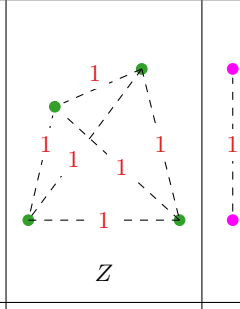
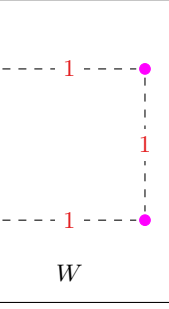
$$d_B(\mathcal{B}_{\text{Con}}(X), \mathcal{B}_{\text{Con}}(Y)) = \hat{d}_B(\mathcal{B}_{\text{Ver}}(X), \mathcal{B}_{\text{Ver}}(Y)) = 0 < 1 = 2 \cdot d_{\text{GH}}(X, Y),$$

which tells us the fact that \hat{d}_B between distinct verbose barcodes can be zero. To see $\hat{d}_B(\mathcal{B}_{\text{Ver}, 1}(X), \mathcal{B}_{\text{Ver}, 1}(Y)) = 0$, consider that pullback metric space Z_X that repeats the top point in X and Z_Y that repeats any one point in Y , and see that $\mathcal{B}_{\text{Ver}, 1}(Z_X) = \mathcal{B}_{\text{Ver}, 1}(Z_Y) = \{(1, 1), (2, 2)^5\}$. The pair (X, Y) shows the tightness of $d_B \leq \hat{d}_B$.

The pair (Z, W) is such that

$$d_B(\mathcal{B}_{\text{Con}}(Z), \mathcal{B}_{\text{Con}}(W)) = \frac{1}{2} < 1 = \hat{d}_B(\mathcal{B}_{\text{Ver}}(Z), \mathcal{B}_{\text{Ver}}(W)) = 2 \cdot d_{\text{GH}}(Z, W),$$

which is another example of \hat{d}_B and \hat{d}_I providing better bounds of d_{GH} compared to the standard bottleneck distance d_B , as well as an example for the tightness of $\hat{d}_B \leq 2 \cdot d_{\text{GH}}$.

				
	X	Y	Z	W
$\mathcal{B}_{\text{Ver},0}$	$(0, 1)^2, (0, 2), (0, \infty)$	$(0, 1)^2, (0, 2), (0, \infty)$	$(0, 1)^3, (0, \infty)$	$(0, 1)^3, (0, \infty)$
$\mathcal{B}_{\text{Ver},1}$	$(1, 1), (2, 2)^2$	$(2, 2)^3$	$(1, 1)^3$	$(1, 2), (2, 2)^2$
$\mathcal{B}_{\text{Ver},2}$	$(2, 2)$	$(2, 2)$	$(1, 1)$	$(2, 2)$

■ **Figure 6** The 4-point metric spaces X, Y, Z and W ; and their verbose barcodes.

One more example that the stability of \hat{d}_B improves that of d_B (see [17, Example 5.11]):

► **Proposition 30.** *Let X be the one-point space, and Y be any finite metric space. Then,*

$$d_B(\mathcal{B}_{\text{Con}}(X), \mathcal{B}_{\text{Con}}(Y)) = \frac{\text{diam}(Y)}{2} < \text{diam}(Y) = \hat{d}_B(\mathcal{B}_{\text{Ver}}(X), \mathcal{B}_{\text{Ver}}(Y)) = 2 \cdot d_{\text{GH}}(X, Y).$$

5 About computing the pullback bottleneck distance

To have a more concrete understanding of the pullback bottleneck distance, we study verbose barcodes under pullbacks.

Let (X, d_X) be a finite metric space with $X = \{x_1, \dots, x_n\}$. For any surjection $\phi : Z \rightarrow X$, the *pullback (pseudo) metric space* (induced by ϕ) is defined as the pair (Z, ϕ^*d_X) , where ϕ^*d_X is the pullback of the distance function d_X . In other words, for any $z_1, z_2 \in Z$,

$$(\phi^*d_X)(z_1, z_2) := d_X(\phi_X(z_1), \phi_X(z_2)).$$

For each $z \in Z$, the point $\phi_X(z) \in X$ is called the *parent* of z .

► **Proposition 31.** *Assume $X = \{x_1, \dots, x_n\}$ is a pseudo-metric space and $Z = X \sqcup \{z\}$. Suppose $\phi : Z \rightarrow X$ is such that $z \mapsto x_j$ for some $j = 1, \dots, n$. Then*

$$\mathcal{B}_{\text{Ver},0}(Z) = \mathcal{B}_{\text{Ver},0}(X) \sqcup \{(0, 0)\},$$

and for $k \geq 1$,

$$\begin{aligned} \mathcal{B}_{\text{Ver},k}(Z) &= \mathcal{B}_{\text{Ver},k}(X) \sqcup \{\text{diam}([x_j, x_j, x_{i_1}, \dots, x_{i_k}]) \cdot (1, 1) : x_{i_l} \in X - \{x_j\}, \forall l = 1, \dots, k\} \\ &= \mathcal{B}_{\text{Ver},k}(X) \sqcup \{\text{diam}([x_j, x_j, \beta]) \cdot (1, 1) : \beta \in P_k(X \setminus \{x_j\})\}. \end{aligned}$$

Each finite pullback metric space Z of X can be written as a multiset $Z = X \sqcup \{x_{j_1}, \dots, x_{j_m}\}$ equipped with the inherited metric from X for some $m \geq 0$ and $j_1 \leq \dots \leq j_m$.

We apply the Proposition 31 to prove Proposition 6.

Proof of Proposition 6. We prove by induction on m . When $m = 1$, the statement follows immediately from Proposition 31. Suppose $m \geq 2$ and that the statement holds for $Z' := X \sqcup \{x_{j_1}, \dots, x_{j_{m-1}}\}$. Recall that $P_k(A)$ denotes the multiset consisting of sub-multisets of A each with cardinality k . By Proposition 31 and the induction hypothesis, we obtain:

$d_B(\mathcal{B}_{\text{Con},0}(\cdot), \mathcal{B}_{\text{Con},0}(\cdot))$	X	Y	Z	W	$\hat{d}_B(\mathcal{B}_{\text{Ver},0}(\cdot), \mathcal{B}_{\text{Ver},0}(\cdot))$	X	Y	Z	W
X	0	0	1	1	X	0	0	1	1
Y		0	1	1	Y		0	1	1
Z			0	0	Z			0	0
W				0	W				0
$d_B(\mathcal{B}_{\text{Con},1}(\cdot), \mathcal{B}_{\text{Con},1}(\cdot))$	X	Y	Z	W	$\hat{d}_B(\mathcal{B}_{\text{Ver},1}(\cdot), \mathcal{B}_{\text{Ver},1}(\cdot))$	X	Y	Z	W
X	0	0	0	$\frac{1}{2}$	X	0	0	1	1
Y		0	0	$\frac{1}{2}$	Y		0	1	1
Z			0	$\frac{1}{2}$	Z			0	1
W				0	W				0
$2 \cdot d_{\text{GH}}(\cdot, \cdot)$	X	Y	Z	W					
X	0	1	1	1					
Y		0	1	1					
Z			0	1					
W				0					

■ **Figure 7** The bottleneck distance d_B between concise barcodes, the pullback bottleneck distance \hat{d}_B between verbose barcodes, and the Gromov-Hausdorff distance between spaces.

$$\begin{aligned}
 \mathcal{B}_{\text{Ver},k}(Z) &= \mathcal{B}_{\text{Ver},k}(Z') \sqcup \left\{ \text{diam}([x_{j_m}, \beta]) \cdot (1, 1) : \beta \in P_k \left((X \setminus \{x_{j_m}\}) \sqcup \{x_{j_1}, \dots, x_{j_{m-1}}\} \right) \right\} \\
 &= \mathcal{B}_{\text{Ver},k}(X) \sqcup \bigsqcup_{i=0}^{m-2} \left\{ \text{diam}([x_{j_{i+1}}, \beta_i]) \cdot (1, 1) : \beta_i \in P_k \left((X \setminus \{x_{j_{i+1}}\}) \sqcup \{x_{j_1}, \dots, x_{j_i}\} \right) \right\} \\
 &\quad \sqcup \left\{ \text{diam}([x_{j_m}, \beta_{m-1}]) \cdot (1, 1) : \beta_{m-1} \in P_k \left((X \setminus \{x_{j_m}\}) \sqcup \{x_{j_1}, \dots, x_{j_{m-1}}\} \right) \right\} \\
 &= \mathcal{B}_{\text{Ver},k}(X) \sqcup \bigsqcup_{i=0}^{m-1} \left\{ \text{diam}([x_{j_{i+1}}, \beta_i]) \cdot (1, 1) : \beta_i \in P_k \left((X \setminus \{x_{j_{i+1}}\}) \sqcup \{x_{j_1}, \dots, x_{j_i}\} \right) \right\}.
 \end{aligned}$$

When considering degree 0, Proposition 6 implies Proposition 7, which imposes the strategy of matching bars in concise barcodes only to other bars or to the origin $(0, 0)$ unlike in the case of d_B when bars are allowed to be matched to any point on the diagonal.

See [17, §6] for proofs and further details of this section.

References

- 1 Ulrich Bauer. Ripser. <https://github.com/Ripser/ripser>, 2016.
- 2 Peter Bubenik and Jonathan A. Scott. Categorification of persistent homology. *Discrete & Computational Geometry*, 51(3):600–627, April 2014. doi:10.1007/s00454-014-9573-x.
- 3 Dmitri Burago, IU D Burago, Yuri Burago, Sergei A Ivanov, and Sergei Ivanov. *A course in metric geometry*, volume 33. American Mathematical Soc., 2001.
- 4 Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46(2):255–308, 2009.
- 5 Wojciech Chachólski, Barbara Giunti, and Claudia Landi. Invariants for tame parametrised chain complexes. *Homology, Homotopy & Applications*, 23(2):183–213, 2021. doi:10.4310/HHA.2021.v23.n2.a11.

- 6 Wojciech Chachólski, Barbara Giunti, Alvin Jin, and Claudia Landi. Decomposing filtered chain complexes: Geometry behind barcoding algorithms. *Computational Geometry*, 109:101938, 2023. doi:10.1016/j.comgeo.2022.101938.
- 7 Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J Guibas, and Steve Y Oudot. Proximity of persistence modules and their diagrams. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 237–246, 2009. doi:10.1145/1542362.1542407.
- 8 Frédéric Chazal, David Cohen-Steiner, Leonidas J. Guibas, Facundo Mémoli, and Steve Y. Oudot. Gromov-Hausdorff stable signatures for shapes using persistence. In *Proceedings of the Symposium on Geometry Processing, SGP '09*, pages 1393–1403, Aire-la-Ville, Switzerland, Switzerland, 2009. Eurographics Association. URL: <http://dl.acm.org/citation.cfm?id=1735603.1735622>.
- 9 Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. Springer, 2016.
- 10 Frédéric Chazal, Vin de Silva, and Steve Oudot. Persistence stability for geometric complexes. *Geometriae Dedicata*, 173(1):193–214, December 2014. doi:10.1007/s10711-013-9937-z.
- 11 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & computational geometry*, 37(1):103–120, 2007.
- 12 Vin De Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Dualities in persistent (co)homology. *Inverse Problems*, 27(12):124003, 2011. doi:10.1088/0266-5611/27/12/124003.
- 13 Afra Zomorodian Herbert Edelsbrunner, David Letscher. Topological persistence and simplification. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 454–463, 2000. doi:10.1109/SFCS.2000.892133.
- 14 Michael Lesnick. The theory of the interleaving distance on multidimensional persistence modules. *Foundations of Computational Mathematics*, 15(3):613–650, 2015. doi:10.1007/s10208-015-9255-y.
- 15 Killian Meehan, Andrei Pavlichenko, and Jan Segert. On the structural theorem of persistent homology. *Discrete & Computational Geometry*, 62(4):945–989, 2019. doi:10.1007/s00454-018-0042-9.
- 16 Facundo Memoli. A distance between filtered spaces via tripods. *arXiv preprint arXiv:1704.03965*, 2017.
- 17 Facundo Mémoli and Ling Zhou. Ephemeral persistence features and the stability of filtered chain complexes. *arXiv preprint arXiv:2208.11770*, 2022.
- 18 Michael Usher and Jun Zhang. Persistent homology and Floer–Novikov theory. *Geometry & Topology*, 20(6):3333–3430, 2016. doi:10.2140/gt.2016.20.3333.

Abstract Voronoi-Like Graphs: Extending Delaunay's Theorem and Applications

Evanthia Papadopoulou  

Faculty of Informatics, Università della Svizzera italiana, Lugano, Switzerland

Abstract

Any system of bisectors (in the sense of abstract Voronoi diagrams) defines an arrangement of simple curves in the plane. We define *Voronoi-like graphs* on such an arrangement, which are graphs whose vertices are *locally Voronoi*. A vertex v is called locally Voronoi, if v and its incident edges appear in the Voronoi diagram of three sites. In a so-called admissible bisector system, where Voronoi regions are connected and cover the plane, we prove that any Voronoi-like graph is indeed an abstract Voronoi diagram. The result can be seen as an abstract dual version of Delaunay's theorem on (locally) empty circles.

Further, we define Voronoi-like cycles in an admissible bisector system, and show that the Voronoi-like graph induced by such a cycle C is a unique tree (or a forest, if C is unbounded). In the special case where C is the boundary of an abstract Voronoi region, the induced Voronoi-like graph can be computed in expected linear time following the technique of [Junginger and Papadopoulou SOCG'18]. Otherwise, within the same time, the algorithm constructs the Voronoi-like graph of a cycle C' on the same set (or subset) of sites, which may equal C or be enclosed by C . Overall, the technique computes abstract Voronoi (or Voronoi-like) trees and forests in linear expected time, given the order of their leaves along a Voronoi-like cycle. We show a direct application in updating a constraint Delaunay triangulation in linear expected time, after the insertion of a new segment constraint, simplifying upon the result of [Shewchuk and Brown CGTA 2015].

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases Voronoi-like graph, abstract Voronoi diagram, Delaunay's theorem, Voronoi tree, linear-time randomized algorithm, constraint Delaunay triangulation

Digital Object Identifier 10.4230/LIPIcs.SocG.2023.52

Related Version *Full Version*: <https://arxiv.org/abs/2303.06669>

Funding Supported in part by the Swiss National Science Foundation, project 200021E-201356.

Acknowledgements I would like to thank Kolja Junginger for preliminary discussions, numerous figures and comments on earlier versions of this work. I would also like to thank Franz Aurenhammer for valuable suggestions, and Elena Arseneva for constructive comments and figures.

1 Introduction

Delaunay's theorem [6] is a well-known cornerstone in Computational Geometry: given a set of points, a triangulation is *globally Delaunay* if and only if it is *locally Delaunay*. A triangulation edge is called *locally Delaunay* if it is incident to only one triangle, or it is incident to two triangles, and appears in the Delaunay triangulation of the four related vertices. The Voronoi diagram and the Delaunay triangulation of a point set are dual to each other. These two highly influential and versatile structures are often used and computed interchangeably; see the book of Aurenhammer et al. [2] for extensive information.

Let us pose the following question: how does Delaunay's theorem extend to Voronoi diagrams of generalized (not necessarily point) sites? We are interested in simple geometric objects in the plane such as line segments, polygons, disks, or point clusters, as they often appear in application areas, and answering this question is intimately related to efficient



© Evanthia Papadopoulou;

licensed under Creative Commons License CC-BY 4.0

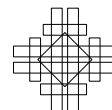
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 52; pp. 52:1–52:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



construction algorithms for Voronoi diagrams (or their duals) on these objects. We consider this question in the framework of abstract Voronoi diagrams in the plane [11] so that we can simultaneously answer it for various concrete and fundamental cases under their umbrella.

Although Voronoi diagrams and Delaunay triangulations of point sites have been widely used in many fields of science, being available in most software libraries of commonly used programming languages, practice has not been the same for their counterparts of simple geometric objects. In fact it is surprising that certain related questions may have remained open or non-optimally solved. Edelsbrunner and Seidel [7] defined Voronoi diagrams as lower envelopes of distance functions in a space one dimension higher, making a powerful link to arrangements, which made their rich combinatorial and algorithmic results applicable, e.g., [15]. However, there are different levels of difficulty concerning arrangements of planes versus more general surfaces, which play a role, especially in practice.

In this paper we define Voronoi-like graphs based on local information, inspired by Delaunay's theorem. Following the framework of abstract Voronoi diagrams (AVDs) [11], let S be a set of n abstract sites (a set of indices) and \mathcal{J} be their underlying system of bisectors, which satisfies some simple combinatorial properties (see Sections 2, 3). Consider a graph G on the arrangement of the bisector system possibly truncated within a simply connected domain D . The vertices of G are vertices of the bisector arrangement, its leaves lie on the boundary ∂D , and the edges are maximal bisector arcs connecting pairs of vertices. A vertex v in G is called *locally Voronoi*, if v and its incident edges within a small neighborhood around v appear in the Voronoi diagram of the three sites defining v (Def. 3), see Figure 4. The graph G is called *Voronoi-like*, if its vertices (other than its leaves on ∂D) are locally Voronoi vertices (Def. 4), see Figure 5. If the graph G is a simple cycle on the arrangement of bisectors related to one site p and its vertices are locally Voronoi of degree 2, then it is called a *Voronoi-like cycle*, for brevity a *site-cycle* (Def. 10).

A major difference between points in the Euclidean plane, versus non-points, such as line segments, disks, or AVDs, can be immediately pointed out: in the former case the bisector system is a line arrangement, while in the latter, the bisecting curves are not even pseudolines. On a line arrangement, it is not hard to see that a Voronoi-like graph coincides with the Voronoi diagram of the involved sites: any Voronoi-like cycle is a convex polygon, which is, in fact, a Voronoi region in the Voronoi diagram of the relevant sites. But in the arrangement of an abstract bisector system, many different Voronoi-like cycles can exist for the same set of sites, see, e.g., Figure 10. Whether a Voronoi-like graph corresponds to a Voronoi diagram is not immediately clear.

In this paper we show that a Voronoi-like graph on the arrangement of an abstract bisector system is as close as possible to being an abstract Voronoi diagram, subject to, perhaps, *missing* some faces (see Def. 5). If the graph misses no face, then it is a Voronoi diagram. Thus, in the classic AVD model [11], where abstract Voronoi regions are connected and cover the plane, any Voronoi-like graph is indeed an abstract Voronoi diagram. This result can be seen as an abstract dual version of Delaunay's theorem.

Voronoi-like graphs (and their duals) can be very useful structures to hold partial Voronoi information, either when dealing with disconnected Voronoi regions, or when considering partial information concerning some region. Building a Voronoi-like graph of partial information may be far easier than constructing the full diagram. In some cases, the full diagram may even be undesirable as in the example of Section 6 in updating a constrained Delaunay triangulation.

The term *Voronoi-like diagram* was first used, in a restricted sense, by Junginger and Papadopoulou [8], defining it as a tree (occasionally a forest) that subdivided a planar

region enclosed by a so-called *boundary curve* defined on a subset of Voronoi edges. Their Voronoi-like diagram was then used as an intermediate structure to perform deletion in an abstract Voronoi diagram in linear expected time. In this paper the formulation of a Voronoi-like graph is entirely different; we nevertheless prove that the Voronoi-like diagram of [8] remains a special case of the one defined in this paper. We thus use the results of [8] when applicable, and extend them to Voronoi-like cycles in an admissible bisector system.

In the remainder of this section we consider an *admissible* bisector system \mathcal{J} following the classic AVD model [11], where bisectors are unbounded simple curves and Voronoi regions are connected. To avoid issues with infinity, we assume a large Jordan curve Γ (e.g, a circle) bounding the computation domain, which is large enough to enclose any bisector intersection. In the sequel, we list further results, which are obtained in this paper under this model.

We consider a Voronoi-like cycle C on the arrangement of bisectors $\mathcal{J}_p \subseteq \mathcal{J} \cup \Gamma$, which are *related* to a site $p \in S$. Let $S_C \subseteq S \setminus \{p\}$ be the set of sites that (together with p) contribute to the bisector arcs in C . The cycle C encodes a *sequence of site occurrences* from S_C . We define the Voronoi-like graph $\mathcal{V}_l(C)$, which can be thought of as a Voronoi diagram of *site occurrences*, instead of sites, whose order is represented by C . We prove that $\mathcal{V}_l(C)$ is a tree, or a forest if C is unbounded, and it exists for any Voronoi-like cycle C . The uniqueness of $\mathcal{V}_l(C)$ can be inferred from the results in [8]. The same properties can be extended to Voronoi-like graphs of cycles related to a set P of k sites.

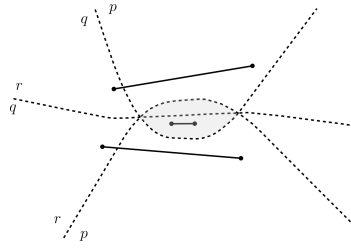
We then consider the randomized incremental construction of [8], and apply it to a Voronoi-like cycle in linear expected time. If C is the boundary of a Voronoi region then $\mathcal{V}_l(C)$, which is the part of the abstract Voronoi diagram $\mathcal{V}(S_C)$, truncated by C , can be computed in expected linear time (this was previously shown [8, 10]). Otherwise, within the same time, the Voronoi-like graph of a (possibly different) Voronoi-like cycle C' , enclosed by C , is computed by essentially the same algorithm. We give conditions under which we can force the randomized algorithm to compute $\mathcal{V}_l(C)$, if desirable, without hurting its expected-linear time complexity, using deletion [8] as a subroutine. The overall technique follows the randomized linear-time paradigm of Chew [5], originally given to compute the Voronoi diagram of points in convex position. The generalization of Chew's technique can potentially be used to convert algorithms working on point sites, which use it, to counterparts involving non-point sites that fall under the umbrella of abstract Voronoi diagrams.

Finally, we give a direct application for computing the Voronoi-like graph of a site-cycle in linear expected time, when updating a constrained Delaunay triangulation upon insertion of a new line segment, simplifying upon the corresponding result of Shewchuk and Brown[16]. The resulting algorithm is extremely simple. By modeling the problem as computing the dual of a Voronoi-like graph, given a Voronoi-like cycle (which is not a Voronoi region's boundary), the algorithmic description becomes almost trivial and explains the technicalities, such as self-intersecting subpolygons, that are listed by Shewchuk and Brown.

The overall technique computes abstract Voronoi, or Voronoi-like, trees and forests in linear expected time, given the order of their leaves along a Voronoi-like cycle. In an extended paper, we also give simple conditions under which the cycle C is an arbitrary Jordan curve of constant complexity. All omitted proofs appear in [14].

2 Preliminaries and definitions

We follow the framework of abstract Voronoi diagrams (AVDs), which have been defined by Klein [11]. Let S be a set of n abstract sites (a set of indices) and \mathcal{J} be an underlying system of bisectors that satisfy some simple combinatorial properties (some axioms). The



■ **Figure 1** Related segment bisectors intersecting twice. $\text{VR}(p, \{p, q, r\})$ is shaded.

bisector $J(p, q)$ of two sites $p, q \in S$ is a simple curve that subdivides the plane into two open domains: the *dominance region of p*, $D(p, q)$, having label p , and the *dominance region of q*, $D(q, p)$, having label q .

The *Voronoi region* of site p is

$$\text{VR}(p, S) = \bigcap_{q \in S \setminus \{p\}} D(p, q).$$

The *Voronoi diagram* of S is $\mathcal{V}(S) = \mathbb{R}^2 \setminus \bigcup_{p \in S} \text{VR}(p, S)$. The vertices and the edges of $\mathcal{V}(S)$ are called *Voronoi vertices* and *Voronoi edges*, respectively.

Variants of abstract Voronoi diagrams of different degrees of generalization have been proposed, see e.g., [12, 3]. Following the original formulation by Klein [11], the bisector system \mathcal{J} is called *admissible*, if it satisfies the following axioms, for every subset $S' \subseteq S$:

- (A1) Each Voronoi region $\text{VR}(p, S')$ is non-empty and pathwise connected.
- (A2) Each point in the plane belongs to the closure of a Voronoi region $\text{VR}(p, S')$.
- (A3) Each bisector is an unbounded simple curve homeomorphic to a line.
- (A4) Any two bisectors intersect transversally and in a finite number of points.

Under these axioms, the abstract Voronoi diagram $\mathcal{V}(S)$ is a planar graph of complexity $O(n)$, which can be computed in $O(n \log n)$ time, randomized [13] or deterministic [11].

To avoid dealing with infinity, we assume that $\mathcal{V}(S)$ is truncated within a domain D_Γ enclosed by a large Jordan curve Γ (e.g., a circle or a rectangle) such that all bisector intersections are contained in D_Γ . Each bisector crosses Γ exactly twice and transversally. All Voronoi regions are assumed to be truncated by Γ , and thus, lie within the domain D_Γ .

We make a general position assumption that no three bisectors involving one common site intersect at the same point, that is, all vertices in the arrangement of the bisector system \mathcal{J} have degree 6, and Voronoi vertices have degree 3.

Bisectors that have a site p in common are called *related*, in particular, *p-related*. Let $\mathcal{J}_p \subseteq \mathcal{J}$ denote the set of all p -related bisectors in \mathcal{J} . Under axiom A2, if related bisectors $J(p, q)$ and $J(p, s)$ intersect at a vertex v , then $J(q, s)$ must also intersect with them at the same vertex, which is a Voronoi vertex in $V(\{p, q, s\})$ (otherwise, axiom A2 would be violated in $V(\{p, q, s\})$). In an admissible bisector system, related bisectors can intersect at most twice [11]; thus, a Voronoi diagram of three sites may have at most two Voronoi vertices, see e.g., the bisectors of three line segments in Figure 1. The curve Γ can be interpreted as a p -related bisector $J(p, s_\infty)$, for a site s_∞ representing infinity, for any $p \in S$.

► **Observation 1.** *In an admissible bisector system, related bisectors that do not intersect or intersect twice must follow the patterns illustrated in Figures 2 and 3 respectively.*

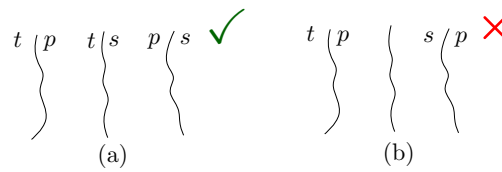


Figure 2 Non-intersecting bisectors; (a) is legal (✓); (b) is illegal (✗).

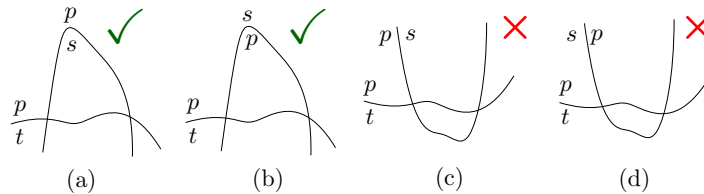


Figure 3 Bisectors intersecting twice; legal (✓) and illegal(✗).

Proof. In Figure 3(c) the pattern is illegal because of axiom A1, and in Figure 3(d) because of combining axioms A2 and A1: $J(s, t)$ must pass through the intersection points of $J(p, s)$ and $J(t, p)$, by A2. Then any possible configuration of $J(s, t)$ results in violating either axiom A1 or A2. If bisectors do not intersect, any pattern other than the one in Figure 2(a) can be shown illegal by combining axioms A1 and A2. ◀

► **Observation 2** ([8]). *In an admissible bisector system, no cycle in the arrangement of bisectors related to p can have the label p on the exterior of the cycle, for all of its arcs.*

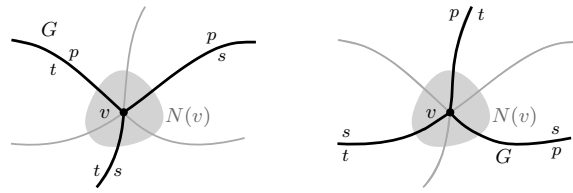
Any component α of a bisector curve $J(p, q)$ is called an *arc*. We use $s_\alpha \in S$ to denote the site such that $\alpha \subseteq J(p, s_\alpha)$. Any component of Γ is called a Γ -arc. The arrangement of a bisector set $\mathcal{J}_x \subseteq \mathcal{J}$ is denoted by $\mathcal{A}(\mathcal{J}_x)$.

3 Defining abstract Voronoi-like graphs and cycles

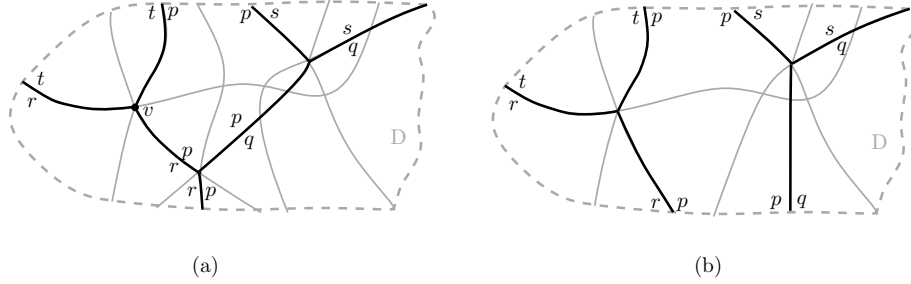
In order to define Voronoi-like graphs in a broader sense, we can relax axioms A1-A4 in this section. In particular, we drop axiom A1 to allow disconnected Voronoi regions and relax axiom A3 to allow disconnected (or even closed) bisecting curves. The bisector $J(p, q)$ of two sites $p, q \in S$ still subdivides the plane into two open domains: the *dominance region of p* , $D(p, q)$, and the *dominance region of q* , $D(q, p)$, however, $D(p, q)$ may be disconnected or bounded. Axioms A2 and A4 remain. Unless otherwise specified, we use the general term *abstract bisector system* to denote such a relaxed variant in the subsequent definitions and in Theorem 6. The term *admissible bisector system* always implies axioms A1-A4.

Let $G = (V, E)$ be a graph on the arrangement of an abstract bisector system \mathcal{J} , truncated within a simply connected domain $D \subseteq D_\Gamma$ (the leaves of G are on ∂D). The vertices of G are arrangement vertices and the edges are maximal bisector arcs connecting pairs of vertices. Figure 5 illustrates examples of such graphs on a bisector arrangement (shown in grey). Under the general position assumption, the vertices of G , except the leaves on ∂D , are of degree 3.

► **Definition 3.** *A vertex v in graph G is called locally Voronoi, if v and its incident graph edges, within a small neighborhood around v , $N(v)$, appear in the Voronoi diagram of the set of three sites defining v , denoted S_v , see Figure 4(a).*



■ **Figure 4** (a) Vertex v is locally Voronoi: $G \cap N(v) = \mathcal{V}(\{p, s, t\}) \cap N(v)$; $N(v)$ is shaded, G is bold, and bisectors are grey. (b) Vertex v is locally farthest Voronoi.



■ **Figure 5** Voronoi-like graphs shown in bold on an arrangement of bisectors shown in grey.

If instead we consider the farthest Voronoi diagram of S_v , then v is called locally Voronoi of the farthest-type, see Figure 4(b). An ordinary locally Voronoi vertex is of the nearest-type.

► **Definition 4.** A graph G on the arrangement of an abstract bisector system, enclosed within a simply connected domain D , is called Voronoi-like, if its vertices (other than its leaves on ∂D) are locally Voronoi vertices. If G is disconnected, we further require that consecutive leaves on ∂D have consistent labels, i.e., they are incident to the dominance region of the same site, as implied by the incident bisector edges in G , see Figure 5.

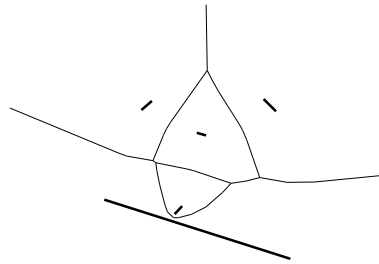
The graph G is actually called an *abstract Voronoi-like graph* but, for brevity, we usually skip the term *abstract*. We next consider the relation between a Voronoi-like graph G and the Voronoi diagram $\mathcal{V}(S) \cap D$, where S is the set of sites involved in the edges of G . Since the vertices of G are locally Voronoi, each face f in G must have the label of exactly one site s_f in its interior, which is called the *site of f* .

► **Definition 5.** Imagine we superimpose G and $\mathcal{V}(S) \cap D$. A face f of $\mathcal{V}(S) \cap D$ is said to be missing from G , if f is covered by faces of G that belong to sites that are different from the site of f , see Figure 7, which is derived from Figure 6.

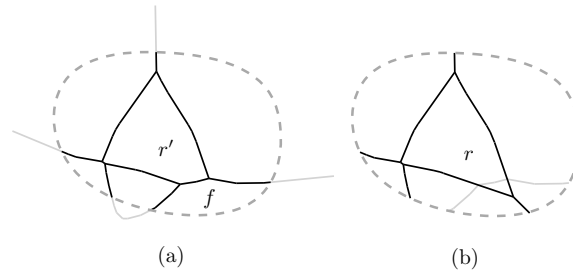
► **Theorem 6.** Let r be a face of an abstract Voronoi-like graph G and let s_r denote its site (the bisectors bounding r have the label s_r inside r). Then one of the following holds:

1. there is a Voronoi face r' in $\mathcal{V}(S) \cap D$, of the same site as r , $r' \subseteq \text{VR}(s_r, S)$, such that $r' \subseteq r$, see Figure 7.
2. face r is disjoint from the Voronoi region $\text{VR}(s_r, S)$. Further, it is entirely covered by Voronoi faces of $\mathcal{V}(S) \cap D$, which are missing from G , see Figure 8.

Proof. Imagine we superimpose G and $\mathcal{V}(S) \cap D$. Face r in G cannot partially overlap any face of the Voronoi region $\text{VR}(s_r, S)$ because if it did, some s_r -related bisector, which contributes to the boundary of r , would intersect the interior of $\text{VR}(s_r, S)$, which is not



■ **Figure 6** A Voronoi diagram of 4 segments.



■ **Figure 7** (a) Voronoi diagram $V(S) \cap D$; (b) Voronoi-like graph G ; $r' \subseteq r$; f is missing from (b).

possible by the definition of a Voronoi region. For the same reason, r cannot be contained in $VR(s_r, S)$. Since Voronoi regions cover the plane, the claim, except from the last sentence in item 2, follows.

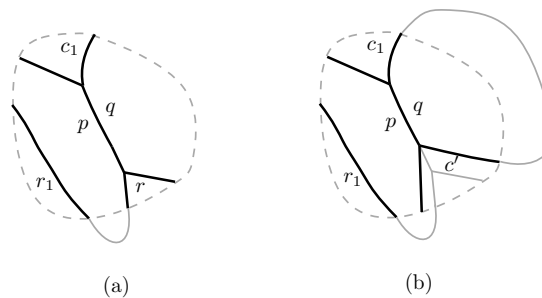
Consider a Voronoi face c' of $\mathcal{V}(S) \cap D$ that overlaps with face r of G in case 2, where the site of c' , s_c , is different from s_r . Since c' overlaps with r , it follows that c' cannot be entirely contained in any face of site s_c in G . Furthermore, c' cannot overlap partially with any face of s_c in G , by the proof in the previous paragraph. Thus, c' is disjoint from any face of G of site s_c , i.e., it must be missing from G . In Figure 8, face c' contains r . ◀

► **Corollary 7.** *If no Voronoi face of $\mathcal{V}(S) \cap D$ is missing from G , then $G = \mathcal{V}(S) \cap D$.*

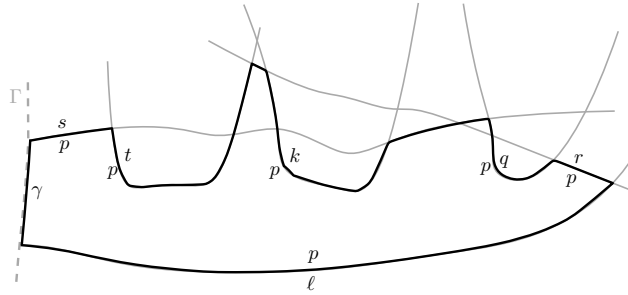
Let us now consider an admissible bisector system, satisfying axioms A1-A4.

► **Corollary 8.** *In an admissible bisector system \mathcal{J} , if D corresponds to the entire plane, then any Voronoi-like graph on \mathcal{J} equals the Voronoi diagram of the relevant set of sites.*

In an admissible bisector system, Voronoi regions are connected, thus, only faces incident to ∂D may be missing from $\mathcal{V}(S) \cap D$.



■ **Figure 8** (a) Voronoi-like graph G ; (b) $V(S) \cap D$; face c' , which covers r , is missing from G .



■ **Figure 9** A Voronoi-like cycle for site p . $S_C = \{s, t, k, q, r, l\}$.

► **Corollary 9.** *In an admissible bisector system, any face f of G that does not touch ∂D either coincides with or contains the Voronoi region $VR(s_f, S)$.*

Thus, in an admissible bisector system, we need to characterize the faces of a Voronoi-like graph that interact with the boundary of the domain D . That is, we are interested in Voronoi-like trees and forests.

Let p be a site in S and let \mathcal{J}_p denote the set of p -related bisectors in \mathcal{J} .

► **Definition 10.** *Let C be a cycle in the arrangement of p -related bisectors $\mathcal{A}(\mathcal{J}_p \cup \Gamma)$ such that the label p appears in the interior of C . A vertex v in C is called degree-2 locally Voronoi, if its two incident bisector arcs correspond to edges in the Voronoi diagram $\mathcal{V}(S_v)$ of the three sites that define v ($p \in S_v$). In particular, $C \cap N(v) \subseteq \mathcal{V}(S_v) \cap N(v)$, where $N(v)$ is a small neighborhood around v . The cycle C is called Voronoi-like, if its vertices are either degree-2 locally Voronoi or points on Γ . For brevity, C is also called a p -cycle, or site-cycle, if the site p is not specified. If C bounds a Voronoi region, then it is called a Voronoi cycle.*

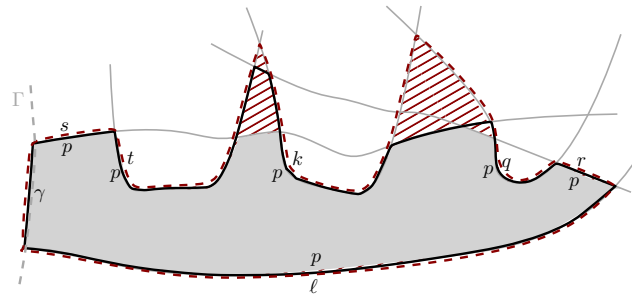
C is called bounded if it contains no Γ -arcs, otherwise, it is called unbounded.

The part of the plane enclosed by C is called the *domain of C* , denoted as D_C . Any Γ -arc of C indicates an opening of the domain to infinity. Figure 9 illustrates a Voronoi-like cycle for site p , which is unbounded (see the Γ -arc γ). It is easy to see in this figure that other p -cycles exist, on the same set of sites, which may enclose or be enclosed by C . The innermost such cycle is the boundary of a Voronoi region, see Figure 10.

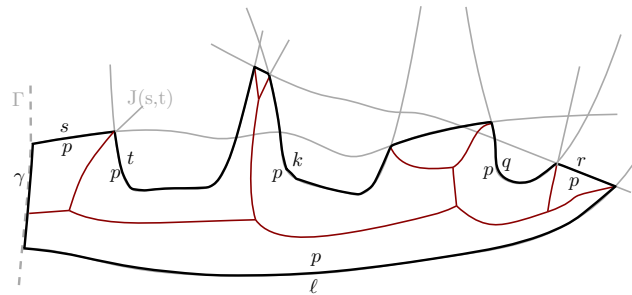
Let $S_C \subseteq S \setminus \{p\}$ denote the set of sites that (together with p) contribute the bisector arcs of C , $S_C = \{s_\alpha \in S \setminus \{p\} \mid \alpha \in C \setminus \Gamma\}$. We refer to S_C as the *set of sites relevant to C* . Let \hat{C} denote the Voronoi cycle $\hat{C} = \partial(VR(p, S_C \cup \{p\}) \cap D_\Gamma)$.

► **Observation 11.** *In an admissible bisector system, there can be many different Voronoi-like cycles involving the same set of sites. Any such cycle C must enclose the Voronoi cycle \hat{C} . Further, $S_{\hat{C}} \subseteq S_C$. In the special case of a line arrangement, e.g., bisectors of point-sites in the Euclidean plane, a site-cycle C is unique for S_C ; in particular, $C = \hat{C}$.*

A Voronoi-like cycle C must share several bisector arcs with its Voronoi cycle \hat{C} , at least one bisector arc for each site in $S_{\hat{C}}$. Let $C \cap \hat{C}$ denote the sequence of common arcs between C and \hat{C} . Several other p -cycles C' , where $S_{\hat{C}} \subseteq S_{C'} \subseteq S_C$, may lie between C and \hat{C} , all sharing Other p -cycles may enclose C . Figure 10 shows such cycles, where the innermost one is \hat{C} ; its domain (a Voronoi region) is shown in solid grey.



■ **Figure 10** Voronoi-like cycles for site p , $S_c = \{s, t, k, q, r, l\}$.



■ **Figure 11** The Voronoi-like graph $\mathcal{V}_l(C)$ (red tree) of the site-cycle C of Fig. 9.

4 The Voronoi-like graph of a cycle

Let \mathcal{J} be an admissible bisector system and let C be a Voronoi-like cycle for site p , which involves a set of sites S_C ($p \notin S_C$). Let $\mathcal{J}_C \subseteq \mathcal{J}$ be the subset of all bisectors that are related to the sites in S_C . The cycle C corresponds to a sequence of *site-occurrences* from S_C , which imply a Voronoi-like graph $\mathcal{V}_l(C)$ in the domain of C , defined as follows:

► **Definition 12.** *The Voronoi-like graph $\mathcal{V}_l(C)$, implied by a Voronoi-like cycle C , is a graph on the underlying arrangement of bisectors $\mathcal{A}(\mathcal{J}_C) \cap D_C$, whose leaves are the vertices of C , and its remaining (non-leaf) vertices are locally Voronoi vertices, see Figure 11. (The existence of such a graph on $\mathcal{A}(\mathcal{J}_C) \cap D_C$ remains to be established).*

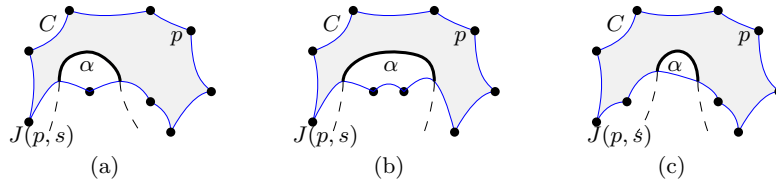
In this section we prove the following theorem for any Voronoi-like cycle C on $\mathcal{A}(\mathcal{J}_p \cup \Gamma)$.

► **Theorem 13.** *The Voronoi-like graph $\mathcal{V}_l(C)$ of a p -cycle C has the following properties:*

1. *it exists and is unique;*
2. *it is a tree if C is bounded, and a forest if C is unbounded;*
3. *it can be computed in expected linear time, if it is the boundary of a Voronoi region; otherwise, in expected linear time we can compute $\mathcal{V}_l(C')$ for some p -cycle C' that is enclosed by C (possibly, $C' = C$ or $C' = \hat{C}$).*

Recall that \hat{C} denotes the Voronoi-cycle enclosed by C , where $\hat{C} = \partial[\text{VR}(p, S_C \cup \{p\}) \cap D_\Gamma]$. Then $\mathcal{V}_l(\hat{C})$ is the Voronoi diagram $\mathcal{V}(S_C) \cap D_{\hat{C}}$. To derive Theorem 13 we show each item separately in subsequent lemmas.

► **Lemma 14.** *Assuming that it exists, $\mathcal{V}_l(C)$ is a forest, and if C is bounded, then $\mathcal{V}_l(C)$ is a tree. Each face of $\mathcal{V}_l(C)$ is incident to exactly one bisector arc α of C , which is called the face (or region) of α , denoted $R(\alpha, C)$.*



■ **Figure 12** Three cases of the arc insertion operation.

If C is the boundary of a Voronoi region, the tree property of the Voronoi diagram $\mathcal{V}(S) \cap D_C$ had been previously shown in [8, 4]. Lemma 14 generalizes it to Voronoi-like graphs for any Voronoi-like cycle C .

In [8], a *Voronoi-like diagram* was defined as a tree structure subdividing the domain of a so-called *boundary curve*, which was implied by a set of Voronoi edges. A boundary curve is a Voronoi-like cycle but not necessarily vice versa. That is, the tree structure of [8] was defined using some of the properties in Lemma 14 as its definition, and the question whether such a tree always existed had remained open. In this paper a Voronoi-like graph is defined entirely differently, but Lemma 14 implies that the two structures are equivalent within the domain of a boundary curve. As a result, we can use and extend the results of [8].

Given a p -cycle C , and a bisector $J(p, s)$ that intersects it, an *arc-insertion operation* can be defined as follows [8]. Let $\alpha \subseteq J(p, s)$ be a maximal component of $J(p, s)$ in the domain of C , see Figure 12. Let $C_\alpha = C \oplus \alpha$ denote the p -cycle obtained by substituting with α the superfluous portion of C between the endpoints of α . (Note that only one portion of C forms a p -cycle with α , thus, no ambiguity exists). There are three different main cases possible as a result, see Figure 12: 1) α may lie between two consecutive arcs of C , in which case $|C_\alpha| = |C| + 1$; 2) α may cause the deletion of one or more arcs in C , thus, $|C_\alpha| \leq |C|$; 3) the endpoints of α may lie on the same arc ω of C , in which case ω splits in two different arcs, thus, $|C_\alpha| = |C| + 2$. In all cases C_α is enclosed by C ($|\cdot|$ denotes cardinality).

The arc-insertion operation can be naturally extended to the Voronoi-like graph $\mathcal{V}_l(C)$ to insert arc α and obtain $\mathcal{V}_l(C_\alpha)$. We use the following lemma, which can be extracted from [8] (using Theorem 18, Theorem 20, and Lemma 21 of [8]).

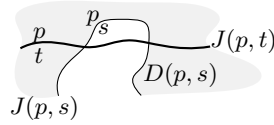
► **Lemma 15** ([8]). *Given $\mathcal{V}_l(C)$, arc $\alpha \in J(p, s) \cap D_C$, and the endpoints of α on C , we can compute the merge curve $J(\alpha) = \partial R(\alpha, C_\alpha)$, using standard techniques as in ordinary Voronoi diagrams. If the endpoints of α lie on different arcs of C , or Γ , the time complexity is $O(|J(\alpha)| + |C \setminus C_\alpha|)$. Otherwise, α splits a bisector arc ω , and its region $R(\omega, C)$, into $R(\omega_1, C_\alpha)$ and $R(\omega_2, C_\alpha)$; the time complexity increases to $O(|J(\alpha)| + \min\{|R(\omega_1, C_\alpha)|, |R(\omega_2, C_\alpha)|\})$.*

The correctness proofs from [8, 10], which are related to Lemma 15, remain intact if performed on a Voronoi-like cycle, as long as the arc α is contained in the cycle's domain; see also [10, Lemma 9]. Thus, Lemma 15 can be established.

Next we prove the existence of $\mathcal{V}_l(C)$ by construction. To this goal we use a *split relation* between bisectors in \mathcal{J}_p or sites in S_C , which had also been considered in [9], see Figure 13.

► **Definition 16.** *For any two sites $s, t \in S_C$, we say that $J(p, s)$ splits $J(p, t)$ (we also say that s splits t , with respect to p), if $J(p, t) \cap D(p, s)$ contains two connected components.*

From the fact that related bisectors in an admissible bisector system intersect at most twice, as shown in Figs. 2 and 3, we can infer that the split relation is asymmetric and transitive, thus, it is also acyclic. The split relation induces a strict partial order on S_C ,



■ **Figure 13** $J(p, s)$ splits $J(p, t)$. In the partial order, $s <_p t$.

where $s <_p t$, if $J(p, s)$ splits $J(p, t)$, see Figure 13. Let o_p be a topological order of the resulting directed acyclic graph, which underlies the split relation on S_C induced by p .

The following lemma shows that $\mathcal{V}_l(C)$ exists by construction, see [14]. It builds upon a more-restricted version regarding a boundary curve, which was considered in [9].

► **Lemma 17.** *Given the topological ordering of the split relation o_p , $\mathcal{V}_l(C)$ can be constructed in $O(|C|^2)$ time; thus, $\mathcal{V}_l(C)$ exists. Further, at the same time, we can construct $\mathcal{V}_l(C')$ for any other Voronoi-like cycle C' that is enclosed by C , $S_{C'} \subseteq S_C$.*

The following lemma can also be extracted from [8, 10]. It can be used to establish the uniqueness of $\mathcal{V}_l(C)$. Similarly to Lemma 15, its original statement does not refer to a p -cycle, however, nothing in its proof prevents its adaptation to a p -cycle, see [10, Lemma 29].

► **Lemma 18.** [10] *Let C be a p -cycle and let α, β be two bisector arcs in C , where $s_\alpha \neq s_\beta$. Suppose that a component e of $J(s_\alpha, s_\beta)$ intersects $R(\alpha, C)$. Then $J(p, s_\beta)$ must intersect D_C with a component $\beta' \subseteq J(p, s_\beta) \cap D_C$ such that e is a portion of $\partial R(\beta', C \oplus \beta')$.*

By Lemma 18, if $J(s_\alpha, s_\beta)$ intersects $R(\alpha, C)$, then a face of s_β must be missing from $\mathcal{V}_l(C)$ (compared to $\mathcal{V}_l(\hat{C})$) implying that an arc of $J(p, s_\beta)$ must be missing from C . Thus, $\mathcal{V}_l(C)$ must be unique.

We now use the randomized incremental construction of [8] to construct $\mathcal{V}_l(C)$, which in turn follows Chew [5], to establish the last claim of Theorem 13. Let $o = (\alpha_1, \dots, \alpha_n)$ be a random permutation of the bisector arcs of C , where each arc represents a different occurrence of a site in S_C . The incremental algorithm works in two phases. In phase 1, delete arcs from C in the reverse order o^{-1} , while registering their neighbors at the time of deletion. In phase 2, insert the arcs, following o , using their neighbors information from phase 1.

Let C_i denote the p -cycle constructed by considering the first i arcs in o in this order. C_1 is the p -cycle consisting of $J(s_{\alpha_1}, p)$ and the relevant Γ -arc. Given C_i , let α'_{i+1} denote the bisector component of $J(p, s_{\alpha_{i+1}}) \cap D_{C_i}$ that contains α_{i+1} (if any), see Figure 12 where α stands for α'_{i+1} . If α_{i+1} lies outside C_i , then $\alpha'_{i+1} = \emptyset$ (this is only possible if C_i is not a Voronoi cycle). Let cycle $C_{i+1} = C_i \oplus \alpha'_{i+1}$ (if $\alpha'_{i+1} = \emptyset$, then $C_{i+1} = C_i$). Given α'_{i+1} , and $\mathcal{V}_l(C_i)$, the graph $\mathcal{V}_l(C_{i+1})$ is obtained by applying Lemma 15.

Let us point out a critical case, which clearly differentiates from [5]: both endpoints of α'_{i+1} lie on the same arc ω of C_i , see Figure 12(c) where α stands for α'_{i+1} . That is, the insertion of α_{i+1} splits the arc ω in two arcs, ω_1 and ω_2 (note $s_{\alpha_{i+1}} <_p s_\omega$) Because of this split, C_i , and thus $\mathcal{V}_l(C_i)$, is order-dependent: if α_{i+1} were considered before ω , in some alternative ordering, then ω_1 or ω_2 would not exist in the resulting cycle, and similarly for their faces in $\mathcal{V}_l(C_{i+1})$. The time to split $R(\omega, C_i)$ is proportional to the minimum complexity of $R(\omega_1, C_{i+1})$ and $R(\omega_2, C_{i+1})$, which is added to the time complexity of step i . Another side effect of the split relation is that α_{i+1} might fall outside C_i , unless C is a Voronoi-cycle, in which case $C_{i+1} = C_i$. Then $C_n \neq C$, in particular, C_n is enclosed by C .

Because the computed cycles are order-dependent, standard backwards analysis cannot be directly applied to step i . In [10] an alternative technique was proposed, which can be applied

to the above construction. The main difference from [10] is the case $C_{i+1} = C_i$, however, such a case has no effect to time complexity, thus, the analysis of [10] can be applied.

► **Proposition 19.** *By the variant of backwards analysis in [10], the time complexity of step i is expected $O(1)$.*

4.1 Relations among the Voronoi-like graphs $\mathcal{V}_l(C)$, $\mathcal{V}_l(C')$, and $\mathcal{V}_l(\hat{C})$

In the following proposition, the first claim follows from Theorem 6 and the second follows from the proof of Lemma 17.

► **Proposition 20.** *Let C' be a Voronoi-like cycle between C and \hat{C} such that $S_{\hat{C}} \subseteq S_{C'} \subseteq S_C$.*

1. $R(\alpha, C') \supseteq R(\alpha, \hat{C})$, for any arc $\alpha \in C' \cap \hat{C}$.
2. $R(\alpha, C') \subseteq R(\alpha, C)$, for any arc $\alpha \in C \cap C'$.

Proposition 20 indicates that the faces of $\mathcal{V}_l(C')$ *shrink* as we move from the outer cycle C to an inner one, until we reach the Voronoi faces of $\mathcal{V}_l(\hat{C})$, which are contained in all others. It also indicates that $\mathcal{V}_l(C)$, $\mathcal{V}_l(C')$ and $\mathcal{V}_l(\hat{C})$ share common subgraphs, and that the adjacencies of the Voronoi diagram $\mathcal{V}_l(\hat{C})$ are preserved. More formally,

► **Definition 21.** *Let $\mathcal{V}_l(C', C \cap C')$ be the following subgraph of $\mathcal{V}_l(C')$: vertex $v \in \mathcal{V}_l(C')$ is included in $\mathcal{V}_l(C', C \cap C')$, if all three faces incident to v belong to arcs in $C \cap C'$; edge $e \in \mathcal{V}_l(C')$ is included to $\mathcal{V}_l(C', C \cap C')$ if both faces incident to e belong to arcs in $C \cap C'$.*

► **Proposition 22.** *For any Voronoi-like cycle C' , enclosed by C , where $S_{C'} \subseteq S_C$, it holds: $\mathcal{V}_l(C', C \cap C') \subseteq \mathcal{V}_l(C)$.*

Depending on the problem at hand, computing $\mathcal{V}_l(C')$ (instead of the more expensive task of computing $\mathcal{V}_l(C)$) may be sufficient. For an example see [14, Section 5].

Computing $\mathcal{V}_l(C)$ in linear expected time (instead of $\mathcal{V}_l(C')$) is possible if the faces of $\mathcal{V}_l(C)$ are Voronoi regions. This can be done by deleting the superfluous arcs in $C' \setminus C$, which are called *auxiliary arcs* (created by arc-splits). A concrete example is given in Section 6. During any step of the construction, if $R(\alpha', C_i)$ is a Voronoi region, but $\alpha' \cap C = \emptyset$, we can call the site-deletion procedure of [8] to eliminate α' and $R(\alpha', C_i)$ from $\mathcal{V}_l(C_i)$. In particular,

► **Proposition 23.** *Given $\mathcal{V}_l(C_i)$, $1 \leq i \leq n$, we can delete $R(\alpha, C_i)$, if $R(\alpha, C_i) \subseteq VR(s_\alpha, S_\alpha)$, where $S_\alpha \subseteq S_C$ is the set of sites that define $\partial R(\alpha, C_i)$, in expected time linear on $|S_\alpha|$.*

There are two ways to use Proposition 23, if applicable:

1. Use it when necessary to maintain the invariant that C_i encloses C (by deleting [8] any auxiliary arc in C_{i-1} that blocks the insertion of α_i , thus, eliminating the case $C_i = C_{i-1}$).
2. Eliminate any auxiliary arc at the time of its creation. If the insertion of α_i splits an arc $\omega \in C_{i-1}$ into ω_1 and ω_2 , but $\omega_2 \notin C$, then eliminate $R(\omega_2, C_i)$ by calling [8].

The advantage of the latter is that Voronoi-like cycles become order-independent, therefore, backwards analysis can be applied to establish the algorithm's time complexity. We give the backwards analysis argument on the concrete case of Section 6; the same type of argument, only more technical, can be derived for this abstract formulation as well.

5 Extending to Voronoi-like cycles of k sites

Theorem 13 can be extended to a *Voronoi-like k -cycle*, for brevity, a *k -cycle*, which involves a set P of k sites whose labels appear in the interior of the cycle. A k -cycle C_k lies in the arrangement $\mathcal{A}(\mathcal{J}_P \cup \Gamma)$ and its vertices are degree-2 locally Voronoi, where \mathcal{J}_P is the set of bisectors related to the sites in P . It implies a Voronoi-like graph $\mathcal{V}_l(C_k)$ involving the set of sites $S_C \subseteq S \setminus P$, which (together with the sites in P) define the bisector arcs of C_k . The definition of $\mathcal{V}_l(C_k)$ is analogous to Def. 12.

There are two types of k -cycles on $\mathcal{A}(\mathcal{J}_P \cup \Gamma)$ of interest: 1. *k -site Voronoi-like cycles* whose vertices are all of the nearest type, e.g., the boundary of the union of k neighboring Voronoi regions; and 2. *order- k Voronoi-like cycles* whose vertices are both of the nearest and the farthest type, e.g., the boundary of an order- k Voronoi face. In either case we partition a k -cycle C_k into *maximal compound arcs*, each induced by one site in S_C . Vertices in the interior of a compound arc are switches between sites in P , and the endpoints of compound arcs are switches between sites in S_C . For an order- k cycle, the former vertices are of the farthest type, whereas the latter (endpoints of compound arcs) are of the nearest type.

► **Lemma 24.** *Assuming that it exists, $\mathcal{V}_l(C_k)$ is a forest, and if C_k is bounded, then $\mathcal{V}_l(C_k)$ is a tree. Each face of $\mathcal{V}_l(C_k)$ is incident to exactly one compound arc α of C_k , which is denoted as $R(\alpha, C_k)$.*

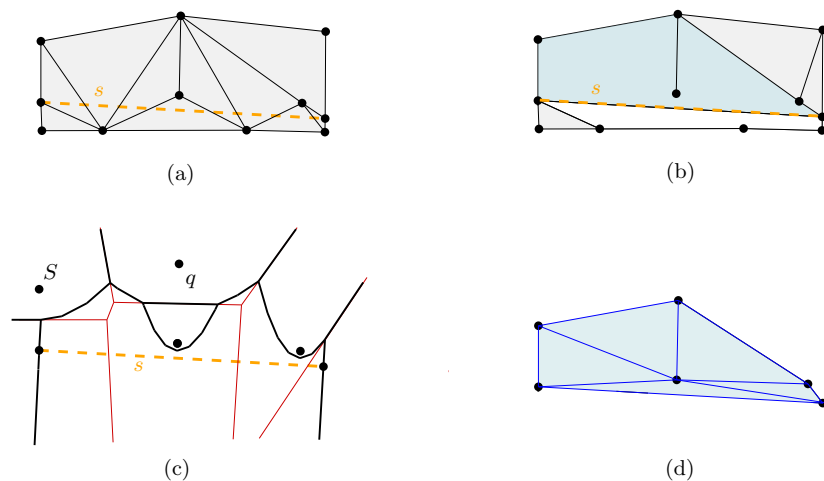
The remaining claims of Theorem 13 can be derived similarly to Section 4, see [14].

6 Updating a constraint Delaunay triangulation

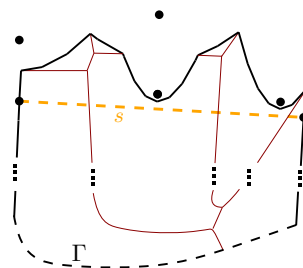
We give an example of a Voronoi-like cycle C , which does not correspond to a Voronoi region, but we need to compute the adjacencies of the Voronoi-like graph $\mathcal{V}_l(C)$. The problem appears in the incremental construction of a *constraint Delaunay triangulation* (CDT), a well-known variant of the Delaunay triangulation, in which a given set of segments is constrained to appear in the triangulation of a point set Q , which includes the endpoints of the segments, see [16] and references therein.

Every edge of the CDT is either an input segment or is *locally Delaunay* (see Section 1). The incremental construction to compute a CDT, first constructs an ordinary Delaunay triangulation of the points in Q , and then inserts segment constraints, one by one, updating the triangulation after each insertion. Shewchuk and Brown [16] gave an expected linear-time algorithm to perform each update. Although the algorithm is summarized in a pseudocode, which could then be directly implemented, the algorithmic description is quite technical having to make sense of self-intersecting polygons, their triangulations, and other exceptions. We show that the problem corresponds exactly to computing (in dual sense) the Voronoi-like graph of a Voronoi-like cycle. Thus, a very simple randomized incremental construction, with occasional calls to Chew's algorithm [5] to delete a Voronoi region of points, can be derived. Quoting from [16]: incremental segment insertion is likely to remain the most used CDT construction algorithm, so it is important to provide an understanding of its performance and how to make it run fast. We do exactly the latter in this section.

When a new constraint segment s is inserted in a CDT, the triangles, which get destroyed by that segment, are identified and deleted [16]. This creates two *cavities* that need to be re-triangulated using *constrained Delaunay triangles*, see Figure 14(a),(b), borrowed from [16], where one cavity is shown shaded (in light blue) and the other unshaded. The boundary of each cavity need not be a simple polygon. However, each cavity implies a Voronoi-like cycle, whose Voronoi-like graph re-triangulates the cavity, see Figure 14(c),(d).



■ **Figure 14** Point set from [16, Fig. 3]. (a) The given CDT and the segment s superimposed; (b) the cavity P in blue; (c) $\mathcal{V}_i(C)$ in red, $C = \partial\text{VR}(s, S \cup \{s\})$ in black; (d) the re-triangulated P .

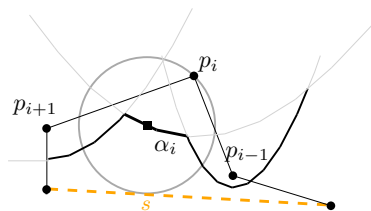


■ **Figure 15** The Voronoi-like cycle C and $\mathcal{V}_i(C)$ (in red) for the example of Fig. 14.

Let $P = (p_1, p_2, \dots, p_n)$ denote one of the cavities, where $p_1 \dots p_n$ is the sequence of cavity vertices in counterclockwise order, and p_1, p_n are the endpoints of s . Let S denote the corresponding set of points ($|S| \leq n$) and let \mathcal{J}_s denote the underlying bisector system involving the segment s and points in S . Let C be the s -cycle in $\mathcal{A}(\mathcal{J}_s \cup \Gamma)$ that has one s -bisector arc for each vertex in P , in the same order as P , see Figure 15. Note that one point in S may contribute more than one arc in C .

► **Lemma 25.** *The s -cycle C exists and can be derived from P in linear time.*

Proof. Let $p_i \in P, 1 < i < n$. The diagonal of the original CDT, which bounded the triangle incident to $p_i p_{i-1}$ (resp. $p_i p_{i+1}$) was a locally Delaunay edge intersected by s . Thus, there is a circle through p_i that is tangent to s that contains neither p_{i-1} nor p_{i+1} , see Figure 16.



■ **Figure 16** Proof of Lemma 25.

Hence, an arc of $J(p_i, s)$ must exist, which contains the center of this circle, and extends from an intersection point of $J(p_i, s) \cap J(p_{i-1}, s)$ to an intersection point of $J(p_i, s) \cap J(p_{i+1}, s)$. The portion of $J(p_i, s)$ between these two intersections corresponds to the arc of p_i on C , denoted α_i . Note that the s -bisectors are parabolas that share the same directrix (the line through s), thus, they may intersect twice. It is also possible that $p_{i-1} = p_{i+1}$. In each case, we can determine which intersection is relevant to arc α_i , given the order of P . Such questions can be reduced to *in-circle tests* involving the segment s and three points. ◀

Let $\text{CDT}(P)$ denote the constraint Delaunay triangulation of P . Its edges are either locally Delaunay or they are cavity edges on the boundary of P .

► **Lemma 26.** *The $\text{CDT}(P)$ is dual to $\mathcal{V}_l(C)$, where C is the s -cycle derived from P .*

Proof. The claim can be derived from the definitions, Lemma 25, which shows the existence of C , and the properties of Theorem 13. The dual of $\mathcal{V}_l(C)$ has one node for each s -bisector arc of C , thus, one node per vertex in P . An edge of $\mathcal{V}_l(C)$ incident to two locally Voronoi vertices v, u involves four different sites in S ; thus, its dual edge is locally Delaunay. The dual of an edge incident to a leaf of C , is an edge of the boundary of P . ◀

Next, we compute $\mathcal{V}_l(C)$ in expected linear time. Because C is not the complete boundary of a Voronoi-region, if we apply the construction of Theorem 13, the computed cycle C_n may be enclosed by C . This is because of occasional split operations, which may create *auxiliary arcs* that have no correspondence to vertices of P . However, we can use Proposition 23 to delete such auxiliary arcs and their faces. The sites in S are points, thus, any Voronoi-like cycle in their bisector arrangement coincides with a Voronoi region. By calling Chew's algorithm [5] we can delete any face of any auxiliary arc in expected time linear in the complexity of the face. The side effect of always deleting auxiliary arcs is that the computed s -cycles are order-independent, making it possible to use backwards analysis to analyse the time complexity of step i , which remains $O(1)$, despite the additional calls to Chew's procedure. We give a proof and additional algorithmic details in [14].

It is easy to dualize the technique to directly compute constraint Delaunay triangles. In fact, the cycle C can remain conceptual. The dual nodes are graph theoretic, each one corresponding to an s -bisector arc, which in turn corresponds to a cavity vertex. This explains the polygon self-crossings of [16] if we draw these graph-theoretic nodes on the cavity vertices during the intermediate steps of the construction.

7 Concluding remarks

We have also considered the variant of computing, in linear expected time, a Voronoi-like tree (or forest) within a simply connected domain D , of constant boundary complexity, given the ordering of some Voronoi faces along the boundary of D . In an extended version we will provide conditions under which the same essentially technique can be applied.

In future research, we are also interested in considering deterministic linear-time algorithms to compute abstract Voronoi-like trees and forests as inspired by [1].

References

- 1 Alok Aggarwal, Leonidas J. Guibas, James B. Saxe, and Peter W. Shor. A linear-time algorithm for computing the Voronoi diagram of a convex polygon. *Discrete and Computational Geometry*, 4:591–604, 1989. doi:10.1007/BF02187749.

- 2 Franz Aurenhammer, Rolf Klein, and Der-Tsai Lee. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013. URL: <http://www.worldscientific.com/worldscibooks/10.1142/8685>.
- 3 Cecilia Bohler, Rolf Klein, and Chih-Hung Liu. Abstract Voronoi diagrams from closed bisecting curves. *International Journal of Computational Geometry and Applications*, 27(3):221–240, 2017.
- 4 Cecilia Bohler, Rolf Klein, and Chih-Hung Liu. An efficient randomized algorithm for higher-order abstract Voronoi diagrams. *Algorithmica*, 81(6):2317–2345, 2019.
- 5 L. Paul Chew. Building Voronoi diagrams for convex polygons in linear expected time. Technical report, Dartmouth College, Hanover, USA, 1990.
- 6 B. Delaunay. Sur la sphère vide. A la memoire de Georges Voronoï. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et naturelles*, 6:793–800, 1934.
- 7 H. Edelsbrunner and R. Seidel. Voronoi diagrams and Arrangements. *Discrete and Computational Geometry*, 1:25–44, 1986.
- 8 Kolja Junginger and Evanthia Papadopoulou. Deletion in Abstract Voronoi Diagrams in Expected Linear Time. In *34th International Symposium on Computational Geometry (SoCG)*, volume 99 of *LIPICs*, pages 50:1–50:14, Dagstuhl, Germany, 2018.
- 9 Kolja Junginger and Evanthia Papadopoulou. On tree-like abstract Voronoi diagrams in expected linear time. In *CGWeek Young Researchers Forum (CG:YRF)*, 2019.
- 10 Kolja Junginger and Evanthia Papadopoulou. Deletion in abstract Voronoi diagrams in expected linear time and related problems. arXiv:1803.05372v2 [cs.CG], 2020. To appear in *Discrete and Computational Geometry*.
- 11 Rolf Klein. *Concrete and Abstract Voronoi Diagrams*, volume 400 of *Lecture Notes in Computer Science*. Springer-Verlag, 1989.
- 12 Rolf Klein, Elmar Langetepe, and Z. Nilforoushan. Abstract Voronoi diagrams revisited. *Computational Geometry: Theory and Applications*, 42(9):885–902, 2009.
- 13 Rolf Klein, Kurt Mehlhorn, and Stefan Meiser. Randomized incremental construction of abstract Voronoi diagrams. *Computational Geometry: Theory and Applications*, 3:157–184, 1993.
- 14 Evanthia Papadopoulou. Abstract Voronoi-like Graphs: Extending Delaunay's Theorem and Applications. arXiv:2303.06669 [cs.CG], March 2023. doi:10.48550/arXiv.2303.06669.
- 15 Micha Sharir and Pankaj K. Agarwal. *Davenport-Schinzel sequences and their geometric applications*. Cambridge university press, 1995.
- 16 Jonathan Richard Shewchuk and Brielin C. Brown. Fast segment insertion and incremental construction of constrained Delaunay triangulations. *Computational Geometry: Theory and Applications*, 48(8):554–574, 2015.

Random Projections for Curves in High Dimensions

Ioannis Psarros  

Athena Research Center, Marousi, Greece

Dennis Rohde  

Universität Bonn, Germany

Abstract

Modern time series analysis requires the ability to handle datasets that are inherently high-dimensional; examples include applications in climatology, where measurements from numerous sensors must be taken into account, or inventory tracking of large shops, where the dimension is defined by the number of tracked items. The standard way to mitigate computational issues arising from the high dimensionality of the data is by applying some dimension reduction technique that preserves the structural properties of the ambient space. The dissimilarity between two time series is often measured by “discrete” notions of distance, e.g. the dynamic time warping or the discrete Fréchet distance. Since all these distance functions are computed directly on the points of a time series, they are sensitive to different sampling rates or gaps. The continuous Fréchet distance offers a popular alternative which aims to alleviate this by taking into account all points on the polygonal curve obtained by linearly interpolating between any two consecutive points in a sequence.

We study the ability of random projections à la Johnson and Lindenstrauss to preserve the continuous Fréchet distance of polygonal curves by effectively reducing the dimension. In particular, we show that one can reduce the dimension to $O(\varepsilon^{-2} \log N)$, where N is the total number of input points while preserving the continuous Fréchet distance between any two determined polygonal curves within a factor of $1 \pm \varepsilon$. We conclude with applications on clustering.

2012 ACM Subject Classification Theory of computation → Computational geometry; Theory of computation → Random projections and metric embeddings

Keywords and phrases polygonal curves, time series, dimension reduction, Johnson-Lindenstrauss lemma, Fréchet distance

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.53

Related Version *Full Version:* <https://arxiv.org/abs/2207.07442>

Funding *Ioannis Psarros:* The author was partially supported by the EU’s Horizon 2020 Research and Innovation programme, under the grant agreement No. 957345: “MORE”.

Dennis Rohde: The author was partially supported by the Hausdorff Center for Mathematics.



© Ioannis Psarros and Dennis Rohde;

licensed under Creative Commons License CC-BY 4.0

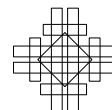
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 53; pp. 53:1–53:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Time series analysis lies in the core of various modern applications. Typically, a time series consists of various (physical) measurements over time. Formally, it is a finite sequence of points in \mathbb{R}^d . Depending on the use case, the ambient space may be extremely high-dimensional, for example $d \in 2^{\Omega(\log n)}$, or even $d \in 2^{\Omega(n)}$, where n is the number of given sequences. For example, large facilities nowadays supervise their production lines using a plethora of sensors. Another concrete example are climatology applications, where data consist of measurements from multiple sensors, each one corresponding to a different dimension.

Many analysis techniques are based on (dis-)similarity between time series, c.f. [27]. This is often measured by distance functions such as the Euclidean distance, which however requires the time series to be of same length and does not include any form of alignment between the sequences. This is of course less expressive than distances which are indeed defined over an optimal alignment, e.g. the dynamic time warping, or the discrete Fréchet distance, which are based on Euclidean distances between the points but enable compensation of differences in phase. A common downside of these distances is that they take into account solely the points of a time series. Hence, they are sensitive to differences in sampling rates or data gaps. Here, the continuous Fréchet distance offers a popular alternative which aims to alleviate this issue by assuming that time series are discretizations of continuous functions of time. It is an extension of the discrete Fréchet distance that takes into account all points on the polygonal curves obtained by linearly interpolating between any two consecutive points in a sequence (where the interpolation is carried out only implicitly).

Two main parameters typically govern computational tasks associated with the Fréchet distance: the lengths of the time series and the number of dimensions of the ambient space. In this paper, we study the problem of compressing the input with respect to the latter parameter using a dimension reducing linear transform that preserves Euclidean distances within a factor of $(1 \pm \varepsilon)$. These transforms, which are usually named Johnson-Lindenstrauss (JL) transforms or embeddings, are a popular tool in dimensionality reduction. The preservation of pairwise distances within a factor of $(1 \pm \varepsilon)$ is sometimes called JL guarantee. Recent work has provided various probability distributions over JL transforms [14, 21, 1, 28, 24], which are efficient to sample from and which yield the JL guarantee with at least constant positive probability while the target dimension is only $O(\varepsilon^{-2} \log n)$, where n is the size of the input point set. Towards applying this result on time series, one can easily guarantee that all Euclidean distances between points of the time series are preserved. While this has direct implications on “discrete” notions of distances between time series, the case of the continuous Fréchet distance is far more intriguing.

1.1 Related Work

In their seminal paper [23], Johnson and Lindenstrauss proved the following statement, which is commonly known as the Johnson-Lindenstrauss lemma and coined the term JL embedding.

► **Theorem 1** ([23]). *For any $n \in \mathbb{N}$ and $\varepsilon \in (0, 1)$ there exists a probability distribution over linear maps $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, where $d' \in O(\varepsilon^{-2} \log n)$, such that for any n -point set $X \subset \mathbb{R}^d$ the following holds with high probability over the choice of f :*

$$\forall p, q \in X : (1 - \varepsilon)\|p - q\| \leq \|f(p) - f(q)\| \leq (1 + \varepsilon)\|p - q\|.$$

In their proof, Johnson and Lindenstrauss [23] show that this can be achieved by orthogonally projecting the points onto a random linear subspace of dimension $O(\varepsilon^{-2} \log n)$ – and indeed

there are point sets that require $\Omega(\varepsilon^{-2} \log n)$ dimensions [5, 25, 26]. Several proofs of their statement followed, these however don't require a proper projection but only a multiplying the points with a certain random matrix, cf. [22, 14, 1, 24, 4].

The impact of a JL embedding on higher-dimensional objects other than points has already been studied. Magen [29, 30] shows that applying a (scaled) JL embedding not only to a given set $P \subset \mathbb{R}^d$ of points, but to $P \cup W$, where $W \subset \mathbb{R}^d$ is a well-chosen set of points determined by P , approximately preserves the height and angles of all triangles determined by any three points in P . Magen even extends this result and shows that by a clever choice of W , the volume (Lebesgue measure) of the convex hull of any $k - 1$ points from P is approximately preserved when the target dimension is in $\Theta(\varepsilon^{-2} k \log |P|)$. Furthermore, in this case the distance of any point from P to the affine hull of any $k - 1$ other points from P is also approximately preserved. Furthermore, JL embeddings can even be utilized to preserve all pairwise Euclidean and geodesic distances on a smooth manifold [8].

Fréchet distance preserving embeddings are a relatively unexplored topic. Recently Driemel and Krivosija [17] studied the first Fréchet distance preserving embedding for c -packed curves, which are curves whose intersections with any ball of radius r are of length at most cr . This class of curves was introduced by Driemel et al. [16] and has so far been considered a viable assumption for realistic curves, see e.g. [3, 9, 15]. Driemel and Krivosija consider projections on random lines, where curves are orthogonally projected on a vector which is sampled uniformly at random from the unit sphere. They observed that in any case (even if the curves are not c -packed), the discrete Fréchet distance between the curves decreases. Furthermore, they show that with high probability the discrete Fréchet distance between two curves σ and τ , of complexity (number of vertices of the curve) at most m , decreases by a factor in $O(m)$. Finally, they proved that there exist c -packed curves such that the discrete Fréchet distance decreases by a factor in $\Omega(m)$. The latter also holds for the continuous Fréchet distance and for the dynamic time warping distance.

More recently, Meintrup et al. [32] studied JL embeddings in the context of preserving the Fréchet distance to facilitate k -median clustering of curves in a high-dimensional ambient space. They show that when the dimension is reduced to $\Theta(\varepsilon^{-2} \log N)$, where N is the total number of vertices of the given curves, the Fréchet distances are preserved up to a combined multiplicative error of $(1 \pm \varepsilon)$ and additive error of $\pm \varepsilon L$, where L is the largest arclength of any input curve. For their proof, they only use the JL guarantee, i.e., the $(1 \pm \varepsilon)$ -preservation of Euclidean distances, and properties of the polygonal curves and the Fréchet distance, while linearity is not taken into account. In this setting, it seems that the additive error is possible – Meintrup et al. give a simple example where some vertex-to-vertex distances expand and others contract, which induces an additive error to the Fréchet distance. Meintrup et al. complement their results with experimental evaluation showing that in real world data and using a JL transform (which is a *linear* map), the Fréchet distance is preserved within the multiplicative error only in almost any case. The other cases can not be distinguished between a failed attempt to obtain a JL embedding (recall the probabilistic nature) and a successful attempt to obtain a JL embedding with the additive error occurring.

1.2 Our Contributions

We study the ability of random projections à la Johnson and Lindenstrauss to preserve the continuous Fréchet distances among a given set of n polygonal curves, each of complexity (number of vertices of the curve) at most m . We show that there exists a set X of vectors (in \mathbb{R}^d), of size polynomial in n and m and depending only on the given curves, such that any JL transform for the curves vertices and X also preserves the continuous Fréchet distance

between any two of the given polygonal curves within a factor of $(1 \pm \varepsilon)$, without additional additive error. This effectively extends the JL guarantee to pairwise Fréchet distances. By plugging in any known JL transform from one of [14, 21, 1, 28, 24] we obtain our main dimension reduction result which states that one can reduce the number of dimensions to $O(\varepsilon^{-2} \log(nm))$. We achieve our result using a completely different approach than Meintrup et al. [32]. Our approach relies on Fréchet distance predicates originating from [2]. These allow a reduction from deciding the continuous distance to a finite set of events occurring. Using only the predicates, it is relatively easy to prove that the Fréchet distance between two curves does not expand by more than a factor of $(1 + \varepsilon)$ under a (*linear*) JL transform. To prove that the Fréchet distance does not contract by more than a factor of $(1 - \varepsilon)$ is however much more challenging. We achieve this by proving that *all* distances between one fixed point and any point on a fixed line do not contract by more than a factor of $(1 - \varepsilon)$ when a JL transform is applied to a well-chosen set of four vectors determined by the point and the line, which is then applied to any vertex of any curve and any line determined by an edge of any curve. We note that this result is comparable to a result by Magen [29, 30], but our statement is stronger since it takes into account *all* distances between the fixed point and the line and not only the affine distance, i.e., the distance between the point and its orthogonal projection onto the line.

Our motivation is that distance preserving dimensionality reductions imply improved algorithms for various tasks. Best-known algorithms for many proximity problems under the continuous Fréchet distance have exponential dependency on the dimension, in at least one of their performance parameters. Such algorithms either directly employ the continuous Fréchet distance, e.g. the approximation algorithms for k -clustering problems [13], or approximate it with the discrete Fréchet distance by resampling the time series to a higher granularity. For example, to the best of our knowledge, the best solution for the approximate near neighbor (ANN) problem in general dimensions derives from building the data structure of Filtser et al. [20], which originally solves the problem for the discrete Fréchet distance, on a modified input. The idea is that a new dense set of vertices can be added to each input polygonal curve so that the discrete Fréchet distance of the resulting curves approximates the continuous Fréchet distance of the original curves. Under the somewhat restrictive assumption that the arclength of each curve is short, a small number of new vertices suffices. Even in this case though, the space and preprocessing time of the data structure depends exponentially on the number of dimensions. Obviously, polynomial-time algorithms (e.g. [10]) can also benefit from reducing the number of dimensions, especially when it comes to real applications.

Our embedding naturally inherits desired properties of the JL transforms like the fact that they are oblivious to the input. This makes it directly applicable to data structure problems like the above-mentioned ANN problem. Moreover, we show that our embedding is also applicable to estimating clustering costs. First, we show that one can approximate the optimal k -center cost within a constant factor, with an algorithm that has no dependency on the original dimensionality apart from an initial step of randomly projecting the input curves. Second, we show that one can use any algorithm for computing the k -median cost in the dimensionality-reduced space to get a constant factor approximation of the k -median cost in the original space.

1.3 Organization

The paper is organized as follows. In Section 2 we introduce the necessary notation, definitions and the concept of Fréchet distance predicates. In Section 3 we prove our main result in two steps. First, as a warm-up, we prove that an application of any JL transformation

for the given curves vertices and a polynomial-sized set X determined by these does not increase Fréchet distances by more than a factor of $(1 + \varepsilon)$. In Section 3.1 we prove the challenging part that this also does not decrease Fréchet distances by less than a factor of $(1 - \varepsilon)$. Interestingly, here a different polynomial-sized set X' is used. In Section 3.2 we combine both to our main result. Finally, in Section 4 we apply our main result to clustering of curves; we modify an existing approximation algorithm for the (k, ℓ) -center problem (see [10]) which has negligibly decreased approximation quality compared to the original and we prove that applying *any* algorithm for the (k, ℓ) -median problem (such as the one from [13]) on the embedded curves leads to a constant factor approximation in terms of clustering cost. Section 5 concludes the paper.

2 Preliminaries

For $n \in \mathbb{N}$ we define $[n] = \{1, \dots, n\}$. By $\|\cdot\|$ we denote the Euclidean norm, by $\langle \cdot, \cdot \rangle$ we denote the Euclidean dot product and by $\mathbb{S}^{d-1} = \{p \in \mathbb{R}^d \mid \|p\| = 1\}$ we denote the unit sphere in \mathbb{R}^d . We define line segments, the building blocks of polygonal curves.

► **Definition 2.** A line segment between two points $p_1, p_2 \in \mathbb{R}^d$, denoted by $\overline{p_1 p_2}$, is the set of points $\{(1 - \lambda)p_1 + \lambda p_2 \mid \lambda \in [0, 1]\}$. For $\lambda \in \mathbb{R}$ we denote by $\text{lp}(\overline{p_1 p_2}, \lambda)$ the point $(1 - \lambda)p_1 + \lambda p_2$, lying on the line supporting the segment $\overline{p_1 p_2}$.

We formally define polygonal curves.

► **Definition 3.** A (parameterized) curve is a continuous mapping $\tau: [0, 1] \rightarrow \mathbb{R}^d$. A curve τ is polygonal, if and only if, there exist $v_1, \dots, v_m \in \mathbb{R}^d$, no three consecutive on a line, called τ 's vertices and $t_1, \dots, t_m \in [0, 1]$ with $t_1 < \dots < t_m$, $t_1 = 0$ and $t_m = 1$, called τ 's instants, such that τ connects every two consecutive vertices $v_i = \tau(t_i)$, $v_{i+1} = \tau(t_{i+1})$ by a line segment.

We call the line segments $\overline{v_1 v_2}, \dots, \overline{v_{m-1} v_m}$ the edges of τ and m the complexity of τ , denoted by $|\tau|$. Sometimes we will argue about a sub-curve $\tau[i, j]$ of a given curve τ , which is the polygonal curve determined by the vertices v_i, \dots, v_j . We define two notions of continuous Fréchet distances. We note that the weak Fréchet distance is used only rarely.

► **Definition 4.** Let σ, τ be curves. The weak Fréchet distance between σ and τ is

$$d_{\text{wF}}(\sigma, \tau) = \inf_{\substack{f: [0,1] \rightarrow [0,1] \\ g: [0,1] \rightarrow [0,1]}} \max_{t \in [0,1]} \|\sigma(f(t)) - \tau(g(t))\|,$$

where f and g are continuous functions with $f(0) = g(0) = 0$ and $f(1) = g(1) = 1$. The Fréchet distance between σ and τ is

$$d_{\text{F}}(\sigma, \tau) = \inf_{\substack{f: [0,1] \rightarrow [0,1] \\ g: [0,1] \rightarrow [0,1]}} \max_{t \in [0,1]} \|\sigma(f(t)) - \tau(g(t))\|,$$

where f and g are continuous bijections with $f(0) = g(0) = 0$ and $f(1) = g(1) = 1$.

We define the type of embedding we are interested in. Since we want to keep our results general, we do not specify the target number of dimensions. As a consequence, we drop the JL-terminology and call these $(1 \pm \varepsilon)$ -embeddings.

► **Definition 5.** Given a set $P \subset \mathbb{R}^d$ of points and $\varepsilon \in (0, 1)$, a function $f: \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a $(1 \pm \varepsilon)$ -embedding for P , if it holds that

$$\forall p, q \in P : (1 - \varepsilon)\|p - q\| \leq \|f(p) - f(q)\| \leq (1 + \varepsilon)\|p - q\|.$$

We note that if f is linear and $0 \in P$, then $\forall p \in P: (1 - \varepsilon)\|p\| \leq \|f(p)\| \leq (1 + \varepsilon)\|p\|$.

We now define valid sequences with respect to two polygonal curves. Such a sequence can be seen as a discrete skeleton in deciding the continuous Fréchet distance and is derived from the free space diagram concept used in Alt and Godau's algorithm [6].

► **Definition 6.** Let σ, τ be polygonal curves with vertices $v_1^\sigma, \dots, v_{|\sigma|}^\sigma$, respectively $v_1^\tau, \dots, v_{|\tau|}^\tau$. A valid sequence with respect to σ and τ is a sequence $\mathcal{F} = (i_1, j_1), \dots, (i_k, j_k)$ with

- $i_1 = j_1 = 1, i_k = |\sigma| - 1, j_k = |\tau| - 1$,
- $(i_l, j_l) \in [|\sigma| - 1] \times [|\tau| - 1]$,
- $(i_l - i_{l-1}, j_l - j_{l-1}) \in \{(0, 1), (1, 0), (0, -1), (-1, 0)\}$ for all $1 < l < k$ and
- any pair $(i_l, j_l) \in [|\sigma| - 1] \times [|\tau| - 1]$ appears at most once in \mathcal{F} .

A valid sequence is said to be monotone if $(i_l - i_{l-1}, j_l - j_{l-1}) \in \{(0, 1), (1, 0)\}$ for all $1 < l < k$.

Further decomposing the free space diagram concept, any valid sequence for two curves σ, τ and any radius $r \geq 0$ induces a set of predicates which truth values in conjunction determine whether $d_F(\sigma, \tau) \leq r$, respectively $d_{wF}(\sigma, \tau) \leq r$.

► **Definition 7** ([2, 19]). Let σ, τ be polygonal curves with vertices $v_1^\sigma, \dots, v_{|\sigma|}^\sigma$, respectively $v_1^\tau, \dots, v_{|\tau|}^\tau$ and $r \in \mathbb{R}_{\geq 0}$. We define the Fréchet distance predicates for σ and τ with respect to r .

- $(P_1)^{\sigma, \tau, r}$: This predicate is true, iff $\|\sigma_1 - \tau_1\| \leq r$.
- $(P_2)^{\sigma, \tau, r}$: This predicate is true, iff $\|\sigma_{|\sigma|} - \tau_{|\tau|}\| \leq r$.
- $(P_3)_{(i,j)}^{\sigma, \tau, r}$: This predicate is true, iff there exists a point $p \in \overline{v_i^\sigma v_{i+1}^\sigma}$ with $\|p - v_j^\tau\| \leq r$.
- $(P_4)_{(i,j)}^{\sigma, \tau, r}$: This predicate is true, iff there exists a point $p \in \overline{v_j^\tau v_{j+1}^\tau}$ with $\|p - v_i^\sigma\| \leq r$.
- $(P_5)_{(i,j,k)}^{\sigma, \tau, r}$: This predicate is true, iff there exist $p_1 = \text{lp}(\overline{v_j^\sigma v_{j+1}^\sigma}, t_1)$ and $p_2 = \text{lp}(\overline{v_j^\tau v_{j+1}^\tau}, t_2)$ with $\|v_i^\sigma - p_1\| \leq r, \|v_k^\tau - p_2\| \leq r$ and $t_1 \leq t_2$.
- $(P_6)_{(i,j,k)}^{\sigma, \tau, r}$: This predicate is true, iff there exist $p_1 = \text{lp}(\overline{v_i^\sigma v_{i+1}^\sigma}, t_1)$ and $p_2 = \text{lp}(\overline{v_i^\tau v_{i+1}^\tau}, t_2)$ with $\|v_j^\sigma - p_1\| \leq r, \|v_k^\tau - p_2\| \leq r$ and $t_1 \leq t_2$.

The following two theorems state the aforementioned facts. These will be one of our main tools in obtaining our main results. We note that these are rephrased here to fit our needs.

► **Theorem 8** ([19]). Let σ, τ be polygonal curves and $r \in \mathbb{R}_{\geq 0}$. There exists a valid sequence \mathcal{F} with respect to σ and τ , such that $(P_1)^{\sigma, \tau, r} \wedge (P_2)^{\sigma, \tau, r} \wedge \Psi_w^{\sigma, \tau, r}(\mathcal{F})$ is true, where

$$\Psi_w^{\sigma, \tau, r}(\mathcal{F}) = \bigwedge_{\substack{(i,j) \in [|\sigma|] \times [|\tau|] \\ (i,j-1), (i,j) \in \mathcal{F}}} (P_3)_{(i,j)}^{\sigma, \tau, r} \bigwedge_{\substack{(i,j) \in [|\tau|] \times [|\sigma|] \\ (i-1,j), (i,j) \in \mathcal{F}}} (P_4)_{(i,j)}^{\sigma, \tau, r},$$

if, and only if, $d_{wF}(\sigma, \tau) \leq r$.

► **Theorem 9** ([2, 19]). Let σ, τ be polygonal curves and $r \in \mathbb{R}_{\geq 0}$. There exists a monotone valid sequence \mathcal{F} with respect to σ and τ , such that $(P_1)^{\sigma, \tau, r} \wedge (P_2)^{\sigma, \tau, r} \wedge \Psi^{\sigma, \tau, r}(\mathcal{F})$ is true, where

$$\Psi^{\sigma, \tau, r}(\mathcal{F}) = \bigwedge_{\substack{(i,j) \in [|\sigma|] \times [|\tau|] \\ (i,j-1), (i,j) \in \mathcal{F}}} (P_3)_{(i,j)}^{\sigma, \tau, r} \bigwedge_{\substack{(i,j) \in [|\tau|] \times [|\sigma|] \\ (i-1,j), (i,j) \in \mathcal{F}}} (P_4)_{(i,j)}^{\sigma, \tau, r} \bigwedge_{\substack{(i,j,k) \in [|\tau|] \times [|\sigma|] \times [|\tau|] \\ (i,j-1), (i,k) \in \mathcal{F} \\ j < k}} (P_5)_{(i,j,k)}^{\sigma, \tau, r} \bigwedge_{\substack{(i,j,k) \in [|\tau|] \times [|\sigma|] \times [|\sigma|] \\ (i-1,j), (k,j) \in \mathcal{F} \\ i < k}} (P_6)_{(i,j,k)}^{\sigma, \tau, r},$$

if, and only if, $d_F(\sigma, \tau) \leq r$.

3 Linear Embeddings preserve Fréchet Distances

In this section we prove our main results on embeddings of polygonal curves that approximately preserve the Fréchet distance. In the following Lemma 10, we show that linear $(1 \pm \varepsilon)$ -embeddings for a polynomial number of points determined by the input polygonal curves imply embeddings for the curves that are not expansive by a factor greater than $(1 + \varepsilon)$. Similarly, in Section 3.1, we show that linear $(1 \pm \varepsilon)$ -embeddings for a polynomial number of points determined by the curves, imply embeddings for the curves that are not contractive by a factor smaller than $(1 - \varepsilon)$. Combining these two bounds yields our main results in Section 3.2. Our main dimensionality reduction result states that one can embed a set of n polygonal curves of complexity at most m into a Euclidean space of dimensions $d' \in O(\varepsilon^{-2} \log(nm))$, so that all Fréchet distances are preserved within a factor of $(1 \pm \varepsilon)$. The embedding is implemented by mapping the vertices of each polygonal curve with a JL transform. The image of each input curve is a curve in $\mathbb{R}^{d'}$ having as vertices the images of the original vertices.

► **Lemma 10.** *Let σ, τ be polygonal curves with vertices $v_1^\sigma, \dots, v_{|\sigma|}^\sigma$, respectively $v_1^\tau, \dots, v_{|\tau|}^\tau$, and let f be a linear $(1 \pm \varepsilon)$ -embedding for $P = \{v_1^\sigma, \dots, v_{|\sigma|}^\sigma, v_1^\tau, \dots, v_{|\tau|}^\tau\} \cup P'$, where P' is a set of points determined by σ and τ with $|P'| \in O(|\sigma|^2 \cdot |\tau| + |\tau|^2 \cdot |\sigma|)$. Let σ' and τ' be polygonal curves with vertices $f(v_1^\sigma), \dots, f(v_{|\sigma|}^\sigma)$, respectively $f(v_1^\tau), \dots, f(v_{|\tau|}^\tau)$. It holds that*

- $d_{\text{wF}}(\sigma', \tau') \leq (1 + \varepsilon) d_{\text{wF}}(\sigma, \tau)$ and
- $d_{\text{F}}(\sigma', \tau') \leq (1 + \varepsilon) d_{\text{F}}(\sigma, \tau)$.

The proof follows by an application of the $(1 \pm \varepsilon)$ -embedding to all points determined by the (weak) Fréchet distance predicates. The proof can be found in the full paper [35].

3.1 Lower Bound

In this section, we show that we can use linear $(1 \pm \varepsilon)$ -embeddings for a polynomial number of points determined by the input polygonal curves to define embeddings for the curves that are not contractive with respect to their Fréchet distance by a factor smaller than $(1 - \varepsilon)$.

We first introduce a few necessary technical lemmas and then we proceed with the main result. We make use of the following lemma, which indicates that inner products are (weakly) concentrated in $(1 \pm \varepsilon)$ -embeddings. Slightly different versions of this lemma have been used before (see e.g. [7, 34, 36]). Our statement is a bit more generic, it holds for any linear $(1 \pm \varepsilon)$ -embedding, and we make use of the involved scaling factors, we include a proof in [35].

► **Lemma 11.** *Let f be a linear $(1 \pm \varepsilon)$ -embedding for a finite set $P \subset \mathbb{R}^d$ with $0 \in P$. For all $p, q \in P$ it holds that*

$$\langle p, q \rangle - 16\varepsilon(\|p\| \cdot \|q\|) \leq \langle f(p), f(q) \rangle \leq \langle p, q \rangle + 14\varepsilon(\|p\| \cdot \|q\|).$$

Next, we prove that $(1 \pm \varepsilon)$ -embeddings for a specific point set do not contract distances between any point on a fixed ray starting from the origin and a fixed point lying in a certain halfspace by a factor smaller than $(1 - 3\varepsilon)$.

► **Lemma 12.** *Let $x \in \mathbb{R}^d$ and $u \in \mathbb{S}^{d-1}$ such that $\langle x, u \rangle \leq 0$. Let f be a linear $(1 \pm \varepsilon/16)$ -embedding for $\{0, x, u\}$. For any $\lambda \geq 0$, we have*

$$\|f(x) - \lambda \cdot f(u)\| \geq (1 - 3\varepsilon)\|x - \lambda u\|.$$

53:8 Random Projections for Curves in High Dimensions

Proof. By Lemma 11 and Definition 5: i) $\langle f(x), f(u) \rangle \in \langle x, u \rangle \pm \varepsilon \|x\|$, ii) $\|f(x)\| \in (1 \pm \varepsilon)\|x\|$ and iii) $\|f(u)\| \in (1 \pm \varepsilon)\|u\|$. For any $\lambda \geq 0$ we have:

$$\begin{aligned} \|f(x) - \lambda \cdot f(u)\|^2 &= \|f(x)\|^2 + \lambda^2 \cdot \|f(u)\|^2 - 2\lambda \cdot \langle f(x), f(u) \rangle \\ &\geq (1 - \varepsilon)^2 \|x\|^2 + (1 - \varepsilon)^2 \lambda^2 - 2\lambda \langle x, u \rangle - 2\lambda \varepsilon \|x\| \end{aligned} \quad (1)$$

$$\geq (1 - \varepsilon)^2 \|x\|^2 + (1 - \varepsilon)^2 \lambda^2 - (1 - \varepsilon)^2 \cdot 2\lambda \cdot \langle x, u \rangle - 2\lambda \varepsilon \cdot \|x\| \quad (2)$$

$$\begin{aligned} &\geq (1 - \varepsilon)^2 \|x - \lambda u\|^2 - 2\varepsilon \lambda \|x\| \\ &\geq (1 - \varepsilon)^2 \|x - \lambda u\|^2 - 2\varepsilon \|x - \lambda u\|^2 \\ &\geq (1 - 3\varepsilon)^2 \|x - \lambda u\|^2, \end{aligned} \quad (3)$$

where the last inequality holds, since $\varepsilon/16 \in (0, 1/4]$. In Equation (1) we use events i), ii), iii), in Equation (2) we use the fact that $\langle x, u \rangle \leq 0$, and in Equation (3) we use the fact that $\langle x, u \rangle \leq 0$ and $\lambda \geq 0$ implies that $\|x - \lambda u\| \geq \lambda$ and $\|x - \lambda u\| \geq \|x\|$. \blacktriangleleft

We now prove our main technical lemma. This says that given a fixed line and a fixed point p , there is a set P of points such that any linear $(1 \pm \varepsilon)$ -embedding for P does not contract distances between p and any point on the line by a factor smaller than $(1 - 3\varepsilon)$. A somewhat similar statement appears in [29] which however focuses on the distortion of point-line distances, i.e., how the distance between a point and its orthogonal projection onto the line changes after the embedding.

► Lemma 13. *Let $x, y, z \in \mathbb{R}^d$ and $\ell = \{p(\overline{yz}, \lambda) \mid \lambda \in \mathbb{R}\}$ be the line supporting \overline{yz} . Let f be a linear $(1 \pm \varepsilon/16)$ -embedding for $\{0, u, -u, x - (t + \langle x, u \rangle \cdot u)\}$, where $u \in \mathbb{S}^{d-1}$ and $t \in \mathbb{R}^d$, such that $\langle u, t \rangle = 0$ and $\{t + \lambda u \mid \lambda \in \mathbb{R}\} = \ell$. For all $\lambda \in \mathbb{R}$ it holds that*

$$\|f(x) - f(t + \lambda u)\| \geq (1 - 3\varepsilon)\|x - (t + \lambda u)\|.$$

Proof. We first note that such an element t exists, namely the orthogonal projection of 0 onto ℓ . Let $p = t + \langle x - t, u \rangle \cdot u = t + \langle x, u \rangle \cdot u$ be the projection of x onto ℓ and let $x' = x - p$. Notice that

$$\langle x', u \rangle = \langle x, u \rangle - \langle t + \langle x, u \rangle \cdot u, u \rangle = \langle x, u \rangle - \langle t, u \rangle - \langle x, u \rangle = 0.$$

We apply Lemma 12 on the vectors x', u . This implies that for any $\lambda \geq 0$,

$$\begin{aligned} \|f(x') - \lambda f(u)\| &\geq (1 - 3\varepsilon)\|x' - \lambda u\| \\ \iff \|f(x - p) - \lambda f(u)\| &\geq (1 - 3\varepsilon)\|x - p - \lambda u\| \\ \iff \|f(x - (t + \langle x, u \rangle \cdot u)) - \lambda f(u)\| &\geq (1 - 3\varepsilon)\|x - (t + \langle x, u \rangle \cdot u) - \lambda u\| \\ \iff \|f(x) - f(t) - \langle x, u \rangle \cdot f(u) - \lambda f(u)\| &\geq (1 - 3\varepsilon)\|x - t - \langle x, u \rangle \cdot u - \lambda u\| \\ \iff \|f(x) - f(t + (\langle x, u \rangle + \lambda) \cdot u)\| &\geq (1 - 3\varepsilon)\|x - (t + (\langle x, u \rangle + \lambda) \cdot u)\|. \end{aligned}$$

Now by reparametrizing $\lambda' \leftarrow \langle x, u \rangle + \lambda$, we conclude that for any $\lambda' \geq \langle x, u \rangle$,

$$\|f(x) - f(t + \lambda' \cdot u)\| \geq (1 - 3\varepsilon)\|x - (t + \lambda' \cdot u)\|. \quad (4)$$

Finally, we apply Lemma 12 on the vectors $x', -u$. Notice that $\langle x', -u \rangle = -\langle x', u \rangle = 0$. This implies that for any $\lambda \geq 0$,

$$\begin{aligned} \|f(x') - \lambda f(-u)\| &\geq (1 - 3\varepsilon)\|x' - \lambda(-u)\| \\ \iff \|f(x - p) - \lambda f(-u)\| &\geq (1 - 3\varepsilon)\|x - p - \lambda(-u)\| \end{aligned}$$

$$\begin{aligned} \iff \|f(x - (t + \langle x, u \rangle \cdot u)) - \lambda f(-u)\| &\geq (1 - 3\varepsilon)\|x - (t + \langle x, u \rangle \cdot u) - \lambda(-u)\| \\ \iff \|f(x) - f(t) - \langle x, u \rangle \cdot f(u) - \lambda f(-u)\| &\geq (1 - 3\varepsilon)\|x - t - \langle x, u \rangle \cdot u - \lambda(-u)\| \\ \iff \|f(x) - f(t + (\langle x, u \rangle - \lambda) \cdot u)\| &\geq (1 - 3\varepsilon)\|x - (t + (\langle x, u \rangle - \lambda) \cdot u)\|. \end{aligned}$$

Now by reparametrizing $\lambda' \leftarrow \langle x, u \rangle - \lambda$, we conclude that for any $\lambda' \leq \langle x, u \rangle$,

$$\|f(x) - f(t + \lambda' \cdot u)\| \geq (1 - 3\varepsilon)\|x - (t + \lambda' \cdot u)\|. \tag{5}$$

Equation (4) and Equation (5) conclude the lemma. ◀

Using the lemma above we can finally prove the main result of this section.

► **Lemma 14.** *Let σ, τ be polygonal curves with vertices $v_1^\sigma, \dots, v_{|\sigma|}^\sigma$, respectively $v_1^\tau, \dots, v_{|\tau|}^\tau$, and let f be a linear $(1 \pm \varepsilon/48)$ -embedding for $P = \{v_1^\sigma, \dots, v_{|\sigma|}^\sigma, v_1^\tau, \dots, v_{|\tau|}^\tau\} \cup P'$, where P' is a set of points determined by σ and τ with $|P'| \in O(|\sigma| \cdot |\tau|)$. Let σ' and τ' be polygonal curves with vertices $f(v_1^\sigma), \dots, f(v_{|\sigma|}^\sigma)$, respectively $f(v_1^\tau), \dots, f(v_{|\tau|}^\tau)$. It holds that*

- $d_{\text{wF}}(\sigma', \tau') \geq (1 - \varepsilon) d_{\text{wF}}(\sigma, \tau)$ and
- $d_{\text{F}}(\sigma', \tau') \geq (1 - \varepsilon) d_{\text{F}}(\sigma, \tau)$.

Proof. For the first claim, let $r = d_{\text{wF}}(\sigma, \tau)$, for the second claim let $r = d_{\text{F}}(\sigma, \tau)$. In both cases, let $r' = (1 - \varepsilon)r$.

In the following, we prove that for any (monotone) valid sequence \mathcal{F} and any $\delta > 0$ we have that $(P_1)^{\sigma', \tau', r' - \delta} \wedge (P_2)^{\sigma', \tau', r' - \delta} \wedge \Psi_w^{\sigma', \tau', r' - \delta}(\mathcal{F})$, respectively $(P_1)^{\sigma', \tau', r' - \delta} \wedge (P_2)^{\sigma', \tau', r' - \delta} \wedge \Psi^{\sigma', \tau', r' - \delta}(\mathcal{F})$, is false and therefore $d_{\text{wF}}(\sigma', \tau') > r' - \delta$, respectively $d_{\text{F}}(\sigma', \tau') > r' - \delta$ by Theorem 8, respectively Theorem 9.

Now, let \mathcal{F} be an arbitrary (monotone) valid sequence. By definition of r and Theorem 8, respectively Theorem 9, we know that for any $\delta > 0$ it holds that $(P_1)^{\sigma, \tau, r - \delta} \wedge (P_2)^{\sigma, \tau, r - \delta} \wedge \Psi_w^{\sigma, \tau, r - \delta}(\mathcal{F})$, respectively $(P_1)^{\sigma, \tau, r - \delta} \wedge (P_2)^{\sigma, \tau, r - \delta} \wedge \Psi^{\sigma, \tau, r - \delta}(\mathcal{F})$, is false. If $(P_1)^{\sigma, \tau, r - \delta}$ or $(P_2)^{\sigma, \tau, r - \delta}$ is false then clearly $(P_1)^{\sigma', \tau', r' - \delta}$ or $(P_2)^{\sigma', \tau', r' - \delta}$ is also false by Definitions 5 and 7. In the following, we assume that $\Psi_w^{\sigma, \tau, r - \delta}(\mathcal{F})$, respectively $\Psi^{\sigma, \tau, r - \delta}(\mathcal{F})$ is false.

Since the arguments for predicates of type P_3 and P_4 are analogous, we focus on the former type. Assume that $\Psi_w^{\sigma, \tau, r - \delta}(\mathcal{F})$ is false because a predicate $(P_3)_{(i,j)}^{\sigma, \tau, r - \delta}$ is false. This means that there does not exist a point $p \in \overline{v_i^\sigma v_{i+1}^\sigma}$ with $\|p - v_j^\tau\| \leq r - \delta$. At this point, recall that since f is linear, any points $\text{lp}(\overline{pq}, t_1), \dots, \text{lp}(\overline{pq}, t_n)$, where $p, q \in \mathbb{R}^d$, are still collinear when f is applied and the relative order on the directed lines supporting \overline{pq} is preserved, which is immediate since $f(\text{lp}(\overline{pq}, t_i)) = \text{lp}(\overline{f(p)f(q)}, t_i)$. By Lemma 13 for any $t \in \mathbb{R}$ and the determined point $p = \text{lp}(\overline{v_i^\sigma v_{i+1}^\sigma}, t)$ on the line supporting $\overline{v_i^\sigma v_{i+1}^\sigma}$ it holds that $\|f(v_j^\tau) - f(p)\| \geq (1 - \varepsilon)\|p - v_j^\tau\|$. Thus, for any $f(p) \in \overline{f(v_i^\sigma)f(v_{i+1}^\sigma)}$ we have $p \in \overline{v_i^\sigma v_{i+1}^\sigma}$ and $\|f(v_j^\tau) - f(p)\| \geq (1 - \varepsilon)\|p - v_j^\tau\|$, which in conclusion is larger than $r' - \delta$, hence $(P_3)_{(i,j)}^{\sigma', \tau', r' - \delta}$ is false and therefore $\Psi_w^{\sigma', \tau', r' - \delta}(\mathcal{F})$ is false. The first claim follows by Theorem 8.

Now, since again the arguments for predicates of type P_5 and P_6 are also analogous, we focus on the former. Assume that $\Psi^{\sigma, \tau, r - \delta}(\mathcal{F})$ is false, because a predicate $(P_5)_{(i,j,k)}^{\sigma, \tau, r - \delta}$ is false. This means that for any two $t_1, t_2 \in \mathbb{R}$ with $t_1 \leq t_2$, the points $p_1 = \text{lp}(\overline{v_j^\sigma v_{j+1}^\sigma}, t_1)$ and $p_2 = \text{lp}(\overline{v_j^\sigma v_{j+1}^\sigma}, t_2)$ do not satisfy $\|v_i^\tau - p_1\| \leq r - \delta$ or $\|v_k^\tau - p_2\| \leq r - \delta$. Since by Lemma 13 we have $\|f(v_i^\tau) - f(p_1)\| \geq (1 - \varepsilon)\|v_i^\tau - p_1\|$ and $\|f(v_k^\tau) - f(p_2)\| \geq (1 - \varepsilon)\|v_k^\tau - p_2\|$, one of these distances must be larger than $r' - \delta$ and it follows that $(P_5)_{(i,j,k)}^{\sigma', \tau', r' - \delta}$ is false. Therefore, $\Psi^{\sigma', \tau', r' - \delta}(\mathcal{F})$ is false and the second claim follows by Theorem 9.

Finally, for the above statements to hold, the set P' contains 0, both directions $u, -u \in \mathbb{S}^{d-1}$ determined by an edge of σ or τ and all points $x - (t + \langle x, u \rangle \cdot u)$, where x is a vertex of a curve σ or τ , and t, u determine a line supporting an edge of τ or σ . ◀

3.2 Main Result

We now prove our main result which combines the upper and lower bounds on the distortion and Lemma 10.

► **Theorem 15.** *Let $T = \{\tau_1, \dots, \tau_n\}$ be a set of polygonal curves in \mathbb{R}^d , each of complexity at most m . There exists a probability distribution over linear maps $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, where $d' \in O(\varepsilon^{-2} \log(nm))$, such that with high probability over the choice of f , the following is true for all $\sigma, \tau \in T$:*

- $|\mathrm{d}_{\mathrm{wF}}(\sigma, \tau) - \mathrm{d}_{\mathrm{wF}}(F(\sigma), F(\tau))| \leq \varepsilon \cdot \mathrm{d}_{\mathrm{wF}}(\sigma, \tau)$ and
- $|\mathrm{d}_{\mathrm{F}}(\sigma, \tau) - \mathrm{d}_{\mathrm{F}}(F(\sigma), F(\tau))| \leq \varepsilon \cdot \mathrm{d}_{\mathrm{F}}(\sigma, \tau)$,

where for any $\tau \in T$ with vertices $v_1^\tau, \dots, v_{|\tau|}^\tau$ we let $F(\tau)$ be the curve with vertices $f(v_1^\tau), \dots, f(v_{|\tau|}^\tau)$.

Proof. We apply Theorem 1 on the set P of $O(n^2m^3)$ points determined by an application of Lemmas 10 and 14 on all pairs of curves in T . ◀

4 Application to Clustering

In this section, we study the effect of randomized $(1 \pm \varepsilon)$ -embeddings on the cost of k -clustering of polygonal curves. In particular, we show that a constant factor approximation of the cost of the optimal k -center solution can be computed with an algorithm, which, except for the time needed to embed the input curves, runs in time independent of the input dimensionality. Moreover, we show that the optimal cost of the k -median problem is preserved within a constant factor in the target space. This means that running any algorithm for the k -median problem in the target space, yields an algorithm for estimating the cost in the original space.

This effectively reduces the computational effort required for approximating the clustering cost, and it directly assists analytical tasks like estimating the optimal number of clusters – where cost estimations for multiple values of k are typically performed.

4.1 Clustering under the Fréchet Distance

In 2016, Driemel et al. [18] introduced clustering under the Fréchet distance, for the purpose of clustering (one-dimensional) time series. The objectives, named (k, ℓ) -center and (k, ℓ) -median, are derived from the well-known k -center and k -median objectives in Euclidean k -clustering. Both are NP-hard [18, 10, 11], even if $k = 1$ and $d = 1$, and the (k, ℓ) -center problem is even NP-hard to approximate within a factor of $(2.25 - \varepsilon)$ in general dimensions [10]. One particularity of these clustering approaches is that the obtained center curves should be of low complexity. In detail, while the given curves have complexity at most m each, the centers should be of complexity at most ℓ each, where $\ell \ll m$ is a constant. The idea behind is that due to the linear interpolation, a *compact* summary of the cluster members through an aggregate center curve is enabled. A nice side effect is that overfitting, which may occur without the complexity restriction, is suppressed. For further details see [18].

We now present a modification of the constant factor approximation algorithm for (k, ℓ) -center clustering from [10]. We note that due to its appealing complexity, this algorithm is used vastly in practice (c.f. [12]) and therefore constitutes a prime candidate to be combined with dimensionality reduction.

4.2 (k, ℓ) -Center Clustering

We formally define the (k, ℓ) -center clustering objective.

► **Definition 16.** *The (k, ℓ) -center clustering problem is to compute a set C of k polygonal curves in \mathbb{R}^d , of complexity at most ℓ each, which minimizes the cost $\max_{\tau \in T} \min_{c \in C} d_F(\tau, c)$, where $T = \{\tau_1, \dots, \tau_n\}$ is a given set of polygonal curves in \mathbb{R}^d of complexity at most m each, and $k \in \mathbb{N}, \ell \in \mathbb{N}_{\geq 2}$ are constant parameters of the problem.*

The following algorithm largely makes use of simplifications of input curves. We formally define this concept.

► **Definition 17.** *An α -approximate minimum-error ℓ -simplification of a curve τ in \mathbb{R}^d is a curve $\sigma = \text{simpl}(\tau)$ in \mathbb{R}^d with at most ℓ vertices, where $\ell \in \mathbb{N}_{\geq 2}$ and $\alpha \geq 1$ are given parameters, such that $d_F(\tau, \sigma) \leq \alpha \cdot d_F(\tau, \sigma')$ for all other curves σ' with ℓ vertices.*

A simplification $\sigma = \text{simpl}(\tau)$ is vertex-restricted if the sequence of its vertices is a subsequence of the sequence of τ 's vertices. Crucial in our modification of the algorithm by Buchin et al. [10] is that we want to compute simplifications in the dimensionality-reduced ambient space to spare running time. In the following, we give a thorough analysis of the effect of dimensionality reduction before simplification. The proof can be found in the full paper [35].

► **Theorem 18.** *Let F be the embedding of Theorem 15 with parameter $\varepsilon \in (0, 1/2]$, for a given set T of n polygonal curves in \mathbb{R}^d of complexity at most m each, all segments $\overline{v_i^\tau v_j^\tau}$, all subcurves $\tau[i, j]$ as well as all vertex-restricted ℓ -simplifications of all $\tau \in T$ (where $v_1^\tau, \dots, v_{|\tau|}^\tau$ are the vertices of τ and $i, j \in [|\tau|]$ with $i < j$). For each $\tau \in T$, a 4-approximate minimum-error ℓ -simplification $\text{simpl}(F(\tau))$ of $F(\tau)$ can be computed in time $O(d' \cdot |\tau|^3 \log |\tau|)$ and for all $\sigma \in T$ it holds that*

$$(1 - \varepsilon) d_F(\sigma, \text{simpl}(\tau)) \leq d_F(F(\sigma), \text{simpl}(F(\tau))) \leq (1 + \varepsilon) d_F(\sigma, \text{simpl}(\tau)),$$

where $\text{simpl}(\tau)$ denotes a $(4 + 16\varepsilon)$ -approximate minimum-error ℓ -simplification of τ .

We now present our modification of the algorithm. Let F denote the embedding from Theorem 15 for $T \cup T' \cup C^*$, where T' is the set of all segments $\overline{v_i^\tau v_j^\tau}$, all subcurves $\tau[i, j]$ as well as all vertex-restricted ℓ -simplifications of all $\tau \in T$ (where $v_1^\tau, \dots, v_{|\tau|}^\tau$ are the vertices of τ and $i, j \in [|\tau|]$ with $i < j$), and C^* is an optimal set of k centers for T .

The algorithm first sets $C = \{\text{simpl}(F(\tau))\}$ for an arbitrary $\tau \in T$. Then, until $|C| = k$ it computes a curve $\tau \in T$ that maximizes $\min_{c \in C} d_F(F(\tau), c)$ and sets $C = C \cup \{\text{simpl}(F(\tau))\}$. Finally, it returns C .

We now prove the approximation guarantee and analyse the running time of this algorithm, thereby we adapt parts of the analysis in [10]. The proof can be found in the full paper [35].

► **Theorem 19.** *Given a set T of n polygonal curves in \mathbb{R}^d of complexity at most m each, and a parameter $\varepsilon \in (0, 1/2]$, the above algorithm returns a solution C to the (k, ℓ) -center clustering problem, consisting of k curves in $\mathbb{R}^{O(\varepsilon^{-2} \ell \log(knm))}$ of complexity at most ℓ each, such that*

$$(1 - 3\varepsilon)r^* \leq \max_{\tau \in T} \min_{c \in C} d_F(F(\tau), c) \leq (6 + 38\varepsilon)r^*,$$

where r^* denotes the cost of an optimal solution. The algorithm has running time

$$O(\varepsilon^{-2} k \ell \log(nm + k) m^3 \log m + \varepsilon^{-2} \ell \log(nm + k) k^2 n m \log m).$$

4.3 (k, ℓ) -Median Clustering

In this section, we show that the cost of the optimal (k, ℓ) -median solution is preserved within a constant factor, when projecting the input curves as described in Section 3. We first define the (k, ℓ) -median clustering problem.

► **Definition 20.** *The (k, ℓ) -median clustering problem is to compute a set C of k polygonal curves in \mathbb{R}^d of complexity at most ℓ each, which minimizes the cost $\sum_{\tau \in T} \min_{c \in C} d_F(\tau, c)$, where $T = \{\tau_1, \dots, \tau_n\}$ is a given set of polygonal curves in \mathbb{R}^d of complexity at most m each, and $k \in \mathbb{N}, \ell \in \mathbb{N}_{\geq 2}$ are constant parameters of the problem.*

In Section 4.3.1, we focus on the case $\ell \geq m$, and we bound the distortion of the optimal cost by a factor of $2 + O(\varepsilon)$. In Section 4.3.2, we discuss case $\ell < m$, and we bound the distortion of the optimal cost by a factor of $6 + O(\varepsilon)$.

4.3.1 Unrestricted Medians

In this section, we present our results on the (k, ℓ) -median clustering problem, when $\ell \geq m$. Computing medians of complexity $\ell = m$ is a widely accepted scenario following, for example, from the wide acceptance of local search methods for clustering, which explore candidate solutions from the set of input curves. The proof follows a similar reasoning as in Section 4.3.2 and is diverted to the full paper [35]. Comparing to Section 4.3.2, we obtain an improved bound on the approximation factor. This is mainly because simplifications are no longer needed in order to obtain a meaningful bound.

► **Theorem 21.** *Let $T = \{\tau_1, \dots, \tau_n\}$ be a set of polygonal curves in \mathbb{R}^d of complexity at most m each and let $\ell \geq m$. There exists a probability distribution over linear maps $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, where $d' \in O(\varepsilon^{-2} \log(n\ell))$, such that with high probability over the choice of f , the following is true. For any polygonal curve τ with vertices $v_1^\tau, \dots, v_{|\tau|}^\tau$, we define $F(\tau)$ to be the curve with vertices $f(v_1^\tau), \dots, f(v_{|\tau|}^\tau)$. Then,*

$$\frac{1 - \varepsilon}{2} \cdot r^* \leq r_f^* \leq (1 + \varepsilon) \cdot r^*,$$

where r^* is the cost of an optimal solution to the (k, ℓ) -median problem on T , and r_f^* is the cost of an optimal solution to the (k, ℓ) -median problem on $F(T)$.

4.3.2 Restricted Medians

To bound the cost of the optimal (k, ℓ) -median in the projected space, we use the notion of simplifications which was introduced in Section 4.2. By an averaging argument, for each cluster, there exists an input curve σ_i which is within distance $\frac{1}{|T_i|} \cdot \sum_{\tau \in T_i} d_F(F(\tau), c_i^f)$ from the optimal median c_i^f , where T_i is the input curves associated with the i^{th} cluster in the projected space. To lower bound the optimal cost in the projected space, we repeatedly apply the triangle inequality on distances involving a vertex-restricted ℓ -simplification of σ_i and a vertex-restricted ℓ -simplification of $F(\sigma_i)$. The upper bound simply follows by the non-contraction guarantee of JL transforms, on distances between input curves and the optimal medians in the original space. The proof can be found in the full paper [35].

► **Theorem 22.** *Let $T = \{\tau_1, \dots, \tau_n\}$ be a set of polygonal curves in \mathbb{R}^d of complexity at most m each and let $\ell < m$. There exists a probability distribution over linear maps $f: \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, where $d' \in O(\varepsilon^{-2} \ell \log(nm))$, such that with high probability over the choice of f , the following*

is true. For any polygonal curve τ with vertices $v_1^\tau, \dots, v_{|\tau|}^\tau$, we define $F(\tau)$ to be the curve with vertices $f(v_1^\tau), \dots, f(v_{|\tau|}^\tau)$. Then,

$$\frac{1 - \varepsilon}{6 \cdot (1 + \varepsilon)} \cdot r^* \leq r_f^* \leq (1 + \varepsilon) \cdot r^*,$$

where r^* is the cost of an optimal solution to the (k, ℓ) -median problem on T , and r_f^* is the cost of an optimal solution to the (k, ℓ) -median problem on $F(T)$.

5 Conclusion

Our results are in line with the results by Magen [29, 30] in the sense that by increasing the constant hidden in the O -notation specifying the number of dimensions of the dimensionality-reduced space, JL transforms become more powerful and do not only preserve pairwise Euclidean distances but also affine distances, angles and volumes, and as we have proven, Fréchet distances.

Concerning JL transforms we have improved the work by Meintrup et al. [32] by proving that no additive error is involved in the resulting Fréchet distances. To facilitate this result, we had to incorporate the linearity of these transforms, which is not done in [32]. Interestingly, this shows that when one uses a terminal embedding instead (see e.g. [33]) – for example to handle a dynamic setting involving queries – this may induce an additive error to the Fréchet distance, as the results by Meintrup et al. [32] can still be applied but ours can not since terminal embeddings are non-linear. Consequently, in contrast to Euclidean distances where a terminal embedding constitutes a proper extension of a JL embedding, this may not be the case when it comes to Fréchet distances.

One open question of practical importance is whether one can improve our result for polygonal curves that satisfy some realistic structural assumption, e.g., c -packness [16]. Moreover, it is possible that our implications on clustering can be improved. One question there is whether one can reduce (or eliminate) the dependence on n from the target dimension, in the same spirit as with the analogous results for the Euclidean distance [31].



References

- 1 Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.
- 2 Peyman Afshani and Anne Driemel. On the complexity of range searching among curves. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 898–917. SIAM, 2018.
- 3 Pankaj K. Agarwal, Kyle Fox, Jiangwei Pan, and Rex Ying. Approximating Dynamic Time Warping and Edit Distance for a Pair of Point Sequences. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG, June 14-18, Boston, MA, USA*, volume 51 of *LIPICs*, pages 6:1–6:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
- 4 Nir Ailon and Bernard Chazelle. The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.
- 5 Noga Alon. Problems and results in extremal combinatorics–I. *Discrete Mathematics*, 273(1-3):31–53, 2003.
- 6 Helmut Alt and Michael Godau. Computing the Fréchet Distance between two Polygonal Curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995.

- 7 Rosa I. Arriaga and Santosh S. Vempala. An Algorithmic Theory of Learning: Robust Concepts and Random Projection. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 616–623. IEEE Computer Society, 1999.
- 8 Richard G. Baraniuk and Michael B. Wakin. Random projections of smooth manifolds. *Found. Comput. Math.*, 9(1):51–77, 2009.
- 9 Karl Bringmann. Why Walking the Dog Takes Time: Fréchet Distance Has No Strongly Subquadratic Algorithms Unless SETH Fails. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS, Philadelphia, PA, USA, October 18-21*, pages 661–670. IEEE Computer Society, 2014.
- 10 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ) -center clustering for curves. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, San Diego, California, USA, January 6-9, 2019*, pages 2922–2938. SIAM, 2019.
- 11 Kevin Buchin, Anne Driemel, and Martijn Struijs. On the Hardness of Computing an Average Curve. In Susanne Albers, editor, *17th Scandinavian Symposium and Workshops on Algorithm Theory*, volume 162 of *LIPICs*, pages 19:1–19:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- 12 Kevin Buchin, Anne Driemel, Natasja van de L’Isle, and André Nusser. klcluster: Center-based Clustering of Trajectories. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 496–499, 2019.
- 13 Maike Buchin, Anne Driemel, and Dennis Rohde. Approximating (k, ℓ) -Median Clustering for Polygonal Curves. In Daniel Marx, editor, *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA, Virtual Conference, January 10 - 131*, pages 2697–2717. SIAM, 2021.
- 14 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, January 2003.
- 15 Anne Driemel and Sariel Har-Peled. Jaywalking Your Dog: Computing the Fréchet Distance with Shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.
- 16 Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet Distance for Realistic Curves in Near Linear Time. *Discrete & Computational Geometry*, 48(1):94–127, 2012.
- 17 Anne Driemel and Amer Krivosija. Probabilistic Embeddings of the Fréchet Distance. In *Proceedings of the 16th International Workshop on Approximation and Online Algorithms (WAOA)*, pages 218–237, 2018.
- 18 Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 766–785, 2016.
- 19 Anne Driemel, André Nusser, Jeff M. Phillips, and Ioannis Psarros. The VC Dimension of Metric Balls under Fréchet and Hausdorff Distances. *Discrete & Computational Geometry*, 66(4):1351–1381, 2021.
- 20 Arnold Filtser, Omrit Filtser, and Matthew J. Katz. Approximate Nearest Neighbor for Curves - Simple, Efficient, and Deterministic. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 168 of *LIPICs*, pages 48:1–48:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.
- 21 Péter Frankl and Hiroshi Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988.
- 22 Piotr Indyk and Rajeev Motwani. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In Jeffrey Scott Vitter, editor, *Proceedings of the Thirtieth Annual*

- ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 604–613. ACM, 1998.
- 23 William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(1):189–206, 1984.
 - 24 Daniel M. Kane and Jelani Nelson. Sparsers Johnson-Lindenstrauss Transforms. *Journal of the ACM*, 61(1):4:1–4:23, 2014.
 - 25 Kasper Green Larsen and Jelani Nelson. The Johnson-Lindenstrauss Lemma Is Optimal for Linear Dimensionality Reduction. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP, July 11-15, Rome, Italy*, volume 55 of *LIPICs*, pages 82:1–82:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.
 - 26 Kasper Green Larsen and Jelani Nelson. Optimality of the Johnson-Lindenstrauss Lemma. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS, Berkeley, CA, USA, October 15-17*, pages 633–638. IEEE Computer Society, 2017.
 - 27 T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.
 - 28 Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, June 1995.
 - 29 Avner Magen. Dimensionality Reductions That Preserve Volumes and Distance to Affine Spaces, and Their Algorithmic Applications. In José D. P. Rolim and Salil P. Vadhan, editors, *Randomization and Approximation Techniques, 6th International Workshop, RANDOM, Cambridge, MA, USA, September 13-15, Proceedings*, volume 2483 of *Lecture Notes in Computer Science*, pages 239–253. Springer, 2002.
 - 30 Avner Magen. Dimensionality Reductions in l_2 that Preserve Volumes and Distance to Affine Spaces. *Discrete & Computational Geometry*, 38(1):139–153, 2007.
 - 31 Konstantin Makarychev, Yury Makarychev, and Ilya P. Razenshteyn. Performance of Johnson-Lindenstrauss transform for k -means and k -medians clustering. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1027–1038. ACM, 2019.
 - 32 Stefan Meintrup, Alexander Munteanu, and Dennis Rohde. Random Projections and Sampling Algorithms for Clustering of High-Dimensional Polygonal Curves. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS, December 8-14, Vancouver, BC, Canada*, pages 12807–12817, 2019.
 - 33 Shyam Narayanan and Jelani Nelson. Optimal terminal dimensionality reduction in Euclidean space. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 1064–1069. ACM, 2019.
 - 34 Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh S. Vempala. Latent Semantic Indexing: A Probabilistic Analysis. *Journal of Computer and System Sciences*, 61(2):217–235, 2000.
 - 35 Ioannis Psarros and Dennis Rohde. Random projections for curves in high dimensions. *CoRR*, abs/2207.07442, 2022. [arXiv:2207.07442](https://arxiv.org/abs/2207.07442).
 - 36 Tamás Sarlós. Improved Approximation Algorithms for Large Matrices via Random Projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

New Approximation Algorithms for Touring Regions

Benjamin Qi  

Massachusetts Institute of Technology, Cambridge, MA, USA

Richard Qi 

Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

We analyze the *touring regions problem*: find a $(1 + \epsilon)$ -approximate Euclidean shortest path in d -dimensional space that starts at a given starting point, ends at a given ending point, and visits given regions $R_1, R_2, R_3, \dots, R_n$ in that order.

Our main result is an $\mathcal{O}\left(\frac{n}{\sqrt{\epsilon}} \log \frac{1}{\epsilon} + \frac{1}{\epsilon}\right)$ -time algorithm for touring disjoint disks. We also give an $\mathcal{O}\left(\min\left(\frac{n}{\epsilon}, \frac{n^2}{\sqrt{\epsilon}}\right)\right)$ -time algorithm for touring disjoint two-dimensional convex fat bodies. Both of these results naturally generalize to larger dimensions; we obtain $\mathcal{O}\left(\frac{n}{\epsilon^{d-1}} \log^2 \frac{1}{\epsilon} + \frac{1}{\epsilon^{2d-2}}\right)$ and $\mathcal{O}\left(\frac{n}{\epsilon^{2d-2}}\right)$ -time algorithms for touring disjoint d -dimensional balls and convex fat bodies, respectively.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry; Theory of computation \rightarrow Approximation algorithms analysis

Keywords and phrases shortest paths, convex bodies, fat objects, disks

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.54

Related Version *Full Version*: <https://arxiv.org/abs/2303.06759>

Acknowledgements We thank Dhruv Rohatgi, Quanquan Liu, Erik Demaine, Danny Mittal, Spencer Compton, William Kuszmaul, Timothy Qian, Jonathan Kelner, and Chris Zhang for helpful discussions. Special thanks to Xinyang Chen for writing one of the proofs in the full version of this paper and providing inspiration.

1 Introduction

We analyze the *touring regions problem*: find a $(1 + \epsilon)$ -approximate Euclidean shortest path in d -dimensional space that starts at a given starting point, ends at a given ending point, and visits given regions $R_1, R_2, R_3, \dots, R_n$ in that order. We primarily present algorithms for the cases where the regions R_i are constrained to be convex fat bodies or balls.¹ To the best of our knowledge, we are the first to consider the cases where regions are disjoint convex fat bodies or balls in arbitrary dimensions. Consequently, our algorithms use techniques not previously considered in the touring regions literature (Section 1.4). Our algorithms work under the assumption that a closest point oracle is provided; closest point projection has been extensively used and studied in convex optimization and mathematics [5, 16].

Most prior work focuses on $d = 2$ or significantly restricts the convex bodies. The special case where $d = 2$ and all regions are constrained to be polygons is known as the *touring polygons problem*. Dror et al. [9] solved the case where every region is a convex polygon exactly, presenting an $\mathcal{O}\left(|V|n \log \frac{|V|}{n}\right)$ -time algorithm when the regions are disjoint as well as an $\mathcal{O}\left(|V|n^2 \log |V|\right)$ -time algorithm when the regions are possibly non-disjoint and the

¹ The full version also contains results for the case where the regions R_i are unions of general convex bodies.



© Benjamin Qi and Richard Qi;

licensed under Creative Commons License CC-BY 4.0

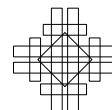
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 54; pp. 54:1–54:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



subpath between every two consecutive polygons in the tour is constrained to lie within a simply connected region called a *fence*. Here, $|V|$ is the total number of vertices over all polygons. Tan and Jiang [19] improved these bounds to $\mathcal{O}(|V|n)$ and $\mathcal{O}(|V|n^2)$ -time, respectively, without considering subpath constraints.

For touring nonconvex polygons, Ahadi et al. [3] proved that finding an optimal path is NP-hard even when polygons are disjoint and constrained to be two line segments each. Dror et al. [9] showed that approximately touring nonconvex polygons with constraining fences is a special case of 3D shortest path with obstacle polyhedra, which can be solved in $\tilde{\mathcal{O}}\left(\frac{e^4}{\epsilon^2}\right)$ time by applying results of Asano et al. [4], where e is the total number of edges over all polyhedra. Mozafari and Zarei [13] improved the bound for the case of nonconvex polygons with constraining fences to $\tilde{\mathcal{O}}\left(\frac{|V|^2 n^2}{\epsilon^2}\right)$ time. Ahadi et al. [3] also solve the *touring objects problem* exactly in polynomial time, in which the R_i are disjoint, nonconvex polygons and the objective is to visit the border of every region without entering the interior of any region.

For touring disjoint disks, a heuristic algorithm with experimental results was demonstrated by Chou [7]. Touring disjoint *unit* disks was given in a programming contest and was a source of inspiration for this paper; an $\mathcal{O}\left(\frac{n}{\epsilon}\right)$ -time algorithm was given [1]. The main result that we show for disks is superior to both of these algorithms.

Polishchuk and Mitchell [17] showed the case where regions are constrained to be intersections of balls or halfspaces in d dimensions to be a special instance of a second-order cone program (SOCP), which runs in $\mathcal{O}\left(d^3 c^{1.5} n^2 \log \frac{1}{\epsilon}\right)$ time using SOCP time bounds as a black box. Here, c is the number of halfspace or ball constraints.

1.1 Formal problem description

► **Definition 1** (Approximate touring regions problem). *Given n sets of points (regions) R_1, R_2, \dots, R_n each a subset of \mathbb{R}^d , a starting point p_0 , and an ending point p_{n+1} ,² define the function $D: (\mathbb{R}^d)^n \rightarrow \mathbb{R}$ as $D(p_1, p_2, \dots, p_n) \triangleq \sum_{i=0}^n \|p_i - p_{i+1}\|_2$.*

Let $\mathcal{A} \triangleq \{(p_1, p_2, \dots, p_n) \mid \forall i, p_i \in R_i\} \subseteq (\mathbb{R}^d)^n$. Find a tuple of points (tour) $(p'_1, p'_2, \dots, p'_n) \in \mathcal{A}$ such that $D(p'_1, p'_2, \dots, p'_n) \leq (1 + \epsilon) \min_{x \in \mathcal{A}} D(x)$.

We primarily consider two types of regions: convex fat bodies with constant bounded fatness and balls. Fat objects have been previously considered in a variety of computational geometry settings [12, 10, 15, 14].

► **Definition 2** (Bounded fatness). *We say that a convex region $R \subset \mathbb{R}^d$ is **fat** if there exist balls h, H with radii $0 < r_h \leq r_H$, respectively, that satisfy $h \subseteq R \subseteq H \subset \mathbb{R}^d$ and $\frac{r_H}{r_h} = \mathcal{O}(1)$.*

One element of the problem that has not yet been determined is how we represent the sets of points R_1, R_2, \dots, R_n ; this depends on what we restrict the regions to be:

- **Convex fat bodies:** We have access to each of the convex bodies R_i via a *closest point oracle*. This oracle allows us to call the function $closest_i(p)$ on some point p , which returns the point $p' \in R_i$ such that $\|p - p'\|$ is minimized in $\mathcal{O}(1)$ time (note that p' is unique due to convexity). Additionally, for each region, we are given the radius r_h of the inscribed ball (as described in Definition 2), and a constant upper bound on the quantity $\frac{r_H}{r_h}$ over all regions.
- **Balls:** For each ball in the input we are given its center $c \in \mathbb{R}^d$ and its radius $r \in \mathbb{R}_{>0}$.

² For convenience, some of our results define the degenerate regions $R_0 \triangleq \{p_0\}$ and $R_{n+1} \triangleq \{p_{n+1}\}$.

We consider the 2-dimensional and general d -dimensional cases separately. In the d -dimensional case, we assume d is a constant (for example, we say $2^d = \mathcal{O}(1)$). We also consider the possibly non-disjoint versus disjoint cases separately, where the latter is defined by the restriction $R_i \cap R_j = \emptyset$ for all $0 \leq i < j \leq n + 1$.

Motivation for our model

When considering general convex bodies, it is natural to augment the model of computation with oracle access to the bodies, including membership, separation, and optimization oracles [11]. In fact, when solving the touring regions problem for general convex bodies, a closest point oracle is necessary even for the case of a single region, where the starting point is the same as the ending point and the optimal solution must visit the closest point in the region to the starting point. Closest point oracles can be constructed trivially when the bodies are constant sized polytopes or balls. Closest point oracles have been used in the field of convex optimization [8, 5].

Our representations for convex fat bodies and balls have the nice structure that the former “contains” the latter: a ball is a specific type of convex fat body, and we can trivially construct a closest point oracle for balls. We justify considering convex fat bodies as they are in some sense “between” balls and general convex bodies: they obey some of the packing constraints of balls.

1.2 Summary of results

Our results and relevant previous results are summarized in Tables 1 and 2. We obtain a $\mathcal{O}\left(\frac{n}{\epsilon^{2d-2}}\right)$ time algorithm for touring disjoint convex fat bodies. Notice that this bound is linear in n ; in fact, we show that *any* FPTAS for touring convex fat bodies can be transformed into one that is linear in n (Lemma 14). If the regions are further restricted to be balls, we can apply our new technique of placing points nonuniformly, and the time complexity improves to $\mathcal{O}\left(\frac{n}{\epsilon^{d-1}} \log^2 \frac{1}{\epsilon} + \frac{1}{\epsilon^{2d-2}}\right)$, which roughly halves the exponent of $\frac{1}{\epsilon}$ compared to the convex fat bodies algorithm while retaining an additive $\frac{1}{\epsilon^{2d-2}}$ term.

Our 2D-specific optimizations allow us to obtain superior time bounds compared to if we substituted $d = 2$ into our general dimension algorithms. For convex fat bodies, we obtain an algorithm with linear time dependence on both n and $\frac{1}{\epsilon}$. For our main result of touring disjoint disks, we combine our optimizations for convex fat bodies and balls with 2D-specific optimizations.

► **Theorem 18.** *There is an $\mathcal{O}\left(\frac{n}{\sqrt{\epsilon}} \log \frac{1}{\epsilon} + \frac{1}{\epsilon}\right)$ -time algorithm for touring disjoint disks.*

With a new *polygonal approximation* technique, we use the result of [19] for touring polygons as a black box to obtain algorithms with a square root dependence on $\frac{1}{\epsilon}$, most notably an $\mathcal{O}\left(\frac{n^{3.5}}{\sqrt{\epsilon}}\right)$ -time algorithm for touring 2D convex bodies and an $\mathcal{O}\left(\frac{n^2}{\sqrt{\epsilon}}\right)$ -time algorithm for touring 2D disjoint convex fat bodies.

The $\mathcal{O}\left(c^{1.5}n^2 \log \frac{1}{\epsilon}\right)$ -time result for touring d dimensional convex bodies given by [17], where each body is an intersection of balls and half spaces (with a total of c constraints) can be applied specifically to balls to yield an $\mathcal{O}\left(n^{3.5} \log \frac{1}{\epsilon}\right)$ -time algorithm. Our algorithms for touring disjoint disks and balls all take time linear in n and are thus superior when ϵ is not too small.

Representation	Runtime	Intersecting?	Source
Convex Polygons (Exact)	$\mathcal{O}(V n), \mathcal{O}(V n^2)$	No, Yes	Touring Polygons [9], [19]
Convex (Oracle Access)	$\mathcal{O}\left(\frac{n^{2.5}}{\sqrt{\epsilon}}\right), \mathcal{O}\left(\frac{n^{3.5}}{\sqrt{\epsilon}}\right)$	No, Yes	Theorem 10
Convex Fat (Oracle Access)	$\mathcal{O}\left(\frac{n}{\epsilon}\right), \mathcal{O}\left(\frac{n^2}{\sqrt{\epsilon}}\right)$	No	Theorems 16, 17
Disks	$\mathcal{O}\left(\frac{n}{\sqrt{\epsilon}} \log \frac{1}{\epsilon} + \frac{1}{\epsilon}\right)$	No	Theorem 18

■ **Table 1** Previous and new bounds on touring n regions in two dimensions up to multiplicative error $1 + \epsilon$, where $\epsilon \leq \mathcal{O}(1)$. For polygons, $|V|$ is the total number of vertices over all polygons.

Representation	Runtime	Intersecting?	Source
Convex Bodies, each an intersection of balls or halfspaces	$\mathcal{O}\left(c^{1.5} n^2 \log \frac{1}{\epsilon}\right)$	Yes	SOCP [17]
Convex Fat (Oracle Access)	$\mathcal{O}\left(\frac{n}{\epsilon^{2d-2}}\right)$	No	Theorem 15
Balls	$\mathcal{O}\left(\frac{n}{\epsilon^{d-1}} \log^2 \frac{1}{\epsilon} + \frac{1}{\epsilon^{2d-2}}\right)$	No	Theorem 19

■ **Table 2** Previous and new bounds on touring n regions in $d \geq 2$ dimensions up to multiplicative error $1 + \epsilon$, where $\epsilon \leq \mathcal{O}(1)$. Note that d is treated as a constant. For polyhedra, c is the total number of constraints.

1.3 Organization of the paper

We start in Section 2 by introducing the general techniques used by all of our algorithms, including the closest point projection and 2D-specific optimizations. We then use the ideas of packing and grouping to obtain algorithms for convex fat bodies in Section 3. Finally, we optimize specifically for balls in Section 4 by placing points non-uniformly.

1.4 Summary of techniques

Here, we introduce the techniques mentioned in the previous subsection.

Placing points uniformly (Section 2)

A general idea that we use in our approximation algorithms is to approximate a convex body well using a set of points on its boundary. For previous results involving polygons or polyhedra [4, 13], this step of the process was trivial, as points were equally spaced along edges. In order to generalize to convex bodies in arbitrary dimensions, we equally space points on boundaries using the closest point projection oracle with a bounding hypercube (Lemma 4). After discretizing each body into a set of points, we can solve the problem in polynomial time using dynamic programming (DP): for each point, we find and store the optimal path ending at it by considering transitions from all points on the previous region.

2D-specific optimizations (Section 2)

When the input shapes are convex and disjoint, we use properties of Monge matrices to optimize dynamic programming transitions from quadratic to expected linear time (Lemma 5). Previous approximation algorithms for related problems discretize the boundary of each convex region using $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ points. We present a new approach to approximate each boundary

using a convex polygon with $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}}\right)$ vertices (Lemma 9). This allows us to use previous exact algorithms for touring convex polygons as black boxes.

Packing and grouping (Section 3)

The key ideas behind our improvements for disjoint convex fat bodies are *packing* and *grouping*. We use a simple *packing* argument to show that the path length for visiting n disjoint convex fat bodies with radius r must have length at least $\Omega(r \cdot n)$ for sufficiently large n (Lemma 11). This was used by [1] for the case of unit disks. However, it is not immediately clear how to use this observation to obtain improved time bounds when convex fat regions are not all restricted to be the same size. The idea of *grouping* is to split the sequence of regions into smaller contiguous subsequences of regions (groups). In each group, we find the minimum-sized region, called a *representative* region, which allows us to break up the global path into smaller subpaths between consecutive representatives. The earlier packing argument now becomes relevant here, as we can show a lower bound on the total length of the optimal path in terms of the sizes of the representatives.

Placing points non-uniformly (Section 4)

Previous approximation methods rely on discretizing the surfaces of bodies into evenly spaced points. For balls, we use the intuition that the portion of the optimal path from one ball to the next is “long” if the optimal path does not visit the parts of the surfaces that are closest together. This allows us to place points at a lower density on most of the surface area of each ball, leading to improved time bounds. We use this technique in conjunction with packing and grouping. For disks, we additionally apply the aforementioned 2D-specific optimizations.

2 General Techniques

First, we describe the general techniques used by all of our algorithms. We split the discussion into the general d -dimensional case and the 2-dimensional case.

2.1 General dimensions

The first main ingredient is the closest point projection, which allows us to equally space points on each convex body.

► **Lemma 3.** *For a convex region C , define $\text{closest}_C(p) \triangleq \operatorname{argmin}_{c \in C} \|c - p\|$. For any two points p_1 and p_2 , $\|\text{closest}_C(p_1) - \text{closest}_C(p_2)\| \leq \|p_1 - p_2\|$.*

For any closed set X , let ∂X denote the boundary of X .

► **Lemma 4** (Equal spacing via closest point projection). *Given a convex body C for which we have a closest point oracle and a hypercube \mathcal{H} with side length r , we can construct a set $S \subset C$ of $\mathcal{O}\left(\frac{1}{\epsilon^{d-1}}\right)$ points such that for all $p \in (\partial C) \cap \mathcal{H}$, there exists $p' \in S$ such that $\|p - p'\| \leq r\epsilon$.*

Proof Sketch. First, we prove the statement for $C = \mathcal{H}$. For this case, it suffices to equally space points on each face of an axis-aligned hypercube defined by $[0, r]^d$. For example, for the face defined by $x_d = 0$, we place points in a lattice at all coordinates $(x_1, x_2, \dots, x_{d-1}, x_d)$ that satisfy $x_d = 0$ and $x_i = k_i \cdot r\epsilon$ for all integers $k_i \in [0, \frac{1}{\epsilon}]$. For $C \neq \mathcal{H}$, equally space points on \mathcal{H} as we stated to create a set $S_{\mathcal{H}}$. Then define $S \triangleq \{\text{closest}_C(s) \mid s \in S_{\mathcal{H}}\}$. ◀

The proof of Lemma 3 and the remainder of the proof of Lemma 4 are deferred to the full version of this paper.

2.2 Two dimensions

When the convex bodies are constrained to lie in 2D, there are two main avenues for further improvements: first, by speeding up the dynamic programming (DP) transitions when all regions have been discretized into point sets, and second, by approximating convex bodies by convex polygons instead of sets of points.

2.2.1 Dynamic programming speedup

► **Lemma 5.** *Given are the vertices of two disjoint convex polygons $B = [b_1, \dots, b_m]$ and $A = [a_1, a_2, \dots, a_n]$ in counterclockwise order and real weights $[w_1, \dots, w_n]$, one for each vertex of A . Define $d(i, j) \triangleq w_j + \|a_j - b_i\|$. Then $\min_{1 \leq j \leq n} d(i, j)$ may be computed for all $i \in [1, m]$ in $\mathcal{O}(m+n)$ expected time.*

Proof. We first discuss the case where all $w_i = 0$. Aggarwal and Klawe [2] showed how to reduce the computation of $\min_{1 \leq j \leq n, a_j \text{ visible from } b_i} d(i, j)$ and $\min_{1 \leq j \leq n, a_j \text{ not visible from } b_i} d(i, j)$ for all $i \in [1, m]$ to computing the row minima of several *Monge partial matrices* with dimensions $m_1 \times n_1, m_2 \times n_2, \dots, m_k \times n_k$ such that $\sum(m_i + n_i) \leq \mathcal{O}(m+n)$ in $\mathcal{O}(m+n)$ time. Here, a_j is said to be visible from b_i if the segment $\overline{a_j b_i}$ intersects neither the interiors of polygons A nor B . The definition of Monge partial matrix can be found in [6]. Chan [6] recently introduced an $\mathcal{O}(m+n)$ expected time randomized algorithm for computing the row minima of an $m \times n$ Monge partial matrix.³ Thus, the case of $w_i = 0$ can be solved in $\mathcal{O}(m+n)$ expected time.

The key claim that Aggarwal and Klawe [2] use to show that all the matrices they construct are Monge partial is as follows:

▷ **Claim 6 (Lemma 2.1 of [2], adapted).** Assume all $w_j = 0$. Suppose $j \neq j'$ and $i \neq i'$. If $a_j a_{j'} b_i b_{i'}$ form a convex quadrilateral in that order then $d(i, j) + d(i', j') \leq d(i, j') + d(i', j)$.

The claim above holds by the triangle inequality, and it is easy to check that it still holds without the assumption $w_j = 0$. Thus the algorithm from [2] generalizes to the case of nonzero w_j with minor modifications. ◀

► **Corollary 7.** *The Touring Regions Problem in 2D, where all R_i are sets of finitely many points S_i that each form a convex polygon in counterclockwise order and the convex hulls of all S_i are disjoint, can be solved exactly in $\mathcal{O}(\sum_{i=1}^n |S_i|)$ expected time.*

Using the above techniques, the following result is proven in the full version and is used multiple times in Section 3.

► **Theorem 8.** *There is an $\mathcal{O}(n^2 (\log \log n + \frac{1}{\epsilon}))$ -time algorithm for touring disjoint convex bodies in two dimensions. When the bodies are possibly non-disjoint, the bound is $\mathcal{O}(n^3 (\log \log n + \frac{1}{\epsilon} + \frac{\log 1/\epsilon}{n\epsilon}))$ time.*

³ The Monge partial matrix does not have to be given explicitly; it suffices to provide an oracle that returns the value of any entry of the matrix in $\mathcal{O}(1)$ time.

2.2.2 Polygonal approximation algorithms

Up until now, we have approximated the perimeter of a convex region using points. We can alternatively approximate the perimeter using a convex polygon with fewer vertices. The proof is deferred to the full version.

► **Lemma 9** (Polygonal approximation). *Given a closest point oracle for a convex region C and a unit square U , we may select $\mathcal{O}(\epsilon^{-1/2})$ points in C such that every point within $C \cap U$ is within distance ϵ of the convex hull of the selected points.*

The polygonal approximation allows us to immediately obtain the following result.

► **Theorem 10.** *There is a $\mathcal{O}\left(\frac{n^{2.5}}{\sqrt{\epsilon}}\right)$ -time algorithm for touring 2D disjoint convex bodies. When the convex bodies are possibly non-disjoint, the bound is $\mathcal{O}\left(\frac{n^{3.5}}{\sqrt{\epsilon}}\right)$ time.*

Proof Sketch. After using Theorem 8 to find a constant approximation of the optimal path length, we draw a square of this side length around the starting point, and we know the optimal path must lie within the square. Then, we apply Lemma 9 to approximate each region with a convex polygon and use previous exact algorithms for touring polygons [19] to finish. ◀

3 Disjoint convex fat bodies

In this section, we present packing and grouping techniques for touring disjoint convex fat bodies and show how they can be applied to obtain $\mathcal{O}\left(\min\left(\frac{n}{\epsilon}, \frac{n^2}{\sqrt{\epsilon}}\right)\right)$ -time algorithms for touring 2D disjoint convex fat bodies.

3.1 Techniques

3.1.1 Packing

A packing argument shows that the length of the optimal path length is at least linear in the number of bodies and the minimum r_h (that is, the minimum radius of any inscribed ball). Intuitively, if we place n disjoint objects of radius at least 1 that are close to being disks on the plane, the length of the optimal tour that visits all of them should be at least linear in n for sufficiently large n . The proof is deferred to the full version.

► **Lemma 11** (Packing Lemma). *Assume a fixed upper bound on $\frac{r_H}{r_h}$. Then there exists $n_0 = \mathcal{O}(1)$ such that the optimal path length OPT for touring any $n \geq n_0$ disjoint convex fat objects is $\Omega(n \cdot \min r_h)$. For balls, $n_0 = 3$.*

The packing lemma allows us to obtain a strong lower bound on the length of the optimal tour in terms of the size of the regions, which will be crucial in proving that our algorithms have low relative error.

► **Corollary 12.** *Let r_i denote the i th largest r_h . For all $i \geq n_0$, $r_i \leq \mathcal{O}\left(\frac{OPT}{i}\right)$.*

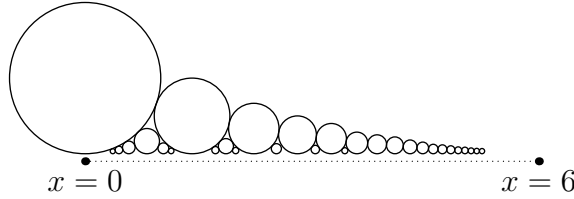
Proof. Consider dropping all regions except those with the i largest inner radii and let OPT_i be the optimal length of a tour that visits the remaining disks in the original order. By Lemma 11, for $i \geq n_0$, $OPT \geq OPT_i \geq \Omega(i \cdot r_i) \implies r_i \leq \mathcal{O}\left(\frac{OPT}{i}\right)$. ◀

► **Lemma 13.** *The optimal path length for touring n disjoint convex fat bodies is $\Omega\left(\sum_{i \geq n_0} r_i / \log n\right)$, and there exists a construction for which this bound is tight.*

Proof Sketch. Using Corollary 12,

$$\frac{\sum_{i \geq n_0} r_i}{\log n} \leq \sum_{i \geq n_0} \frac{\mathcal{O}\left(\frac{OPT}{i}\right)}{\log n} \leq \mathcal{O}\left(\frac{OPT}{\log n} \sum_{i=n_0}^n \frac{1}{i}\right) \leq \mathcal{O}(OPT).$$

We display the construction in Figure 1; we defer the full description to the full version. The idea is to place disjoint disks of radii $1/1, 1/2, 1/3, \dots$ such that they are all tangent to a segment of the x -axis of length $\mathcal{O}(1)$. ◀



■ **Figure 1** Construction from Lemma 13: placement of the first 30 disks.

3.1.2 Grouping

We now show that we can split up the optimal path into smaller subpaths by splitting the sequence of bodies into groups of consecutive bodies, finding the minimum-sized body in each group, and considering the subpaths between these small bodies. By the packing lemma, the sum of the radii of the representatives is small compared to the total path length.

In particular, using groups of size $\frac{1}{\epsilon}$, we can compress the smallest sized region into a single point, meaning that we can consider touring regions between these points independently from each other. This allows us to turn *any* polynomial time approximation scheme for touring disjoint convex fat bodies into one that is linear in n .

► **Lemma 14 (Grouping Lemma).** *Given an algorithm for touring disjoint convex fat bodies in d dimensions that runs in $f(n, \epsilon)$ time, where f is a polynomial, we can construct an algorithm that runs in $\mathcal{O}(n\epsilon + 1) \cdot f\left(\frac{1}{\epsilon}, \epsilon\right)$ time (for $\epsilon \leq \mathcal{O}(1)$).*

Proof. We describe an algorithm achieving a $(1 + \mathcal{O}(\epsilon))$ -approximation. To achieve a $(1 + \epsilon)$ -approximation, scale down ϵ by the appropriate factor.

Define $s \triangleq \lceil \frac{1}{\epsilon} \rceil$ and let n_0 be the constant defined in the statement of Lemma 11. We will prove the statement for all ϵ satisfying $\frac{1}{\epsilon} \geq n_0$. First, we divide the $n+2$ regions (including R_0 and R_{n+1}) into $k = \max\left(\lceil \frac{n+2}{s} \rceil, 2\right) \leq \mathcal{O}(n\epsilon + 1)$ consecutive subsequences, each with exactly s regions (except the starting and ending subsequences, which are allowed to have fewer). Let M_i be the region with minimum inscribed radius r_h in the i th subsequence; note that $M_1 = R_0$ and $M_k = R_{n+1}$. For each $i \in [1, k]$, pick an arbitrary point $p_i \in M_i$. Let OPT' be the length of the shortest tour of R_0, \dots, R_{n+1} that passes through all of the p_i . The p_1, \dots, p_k form $k - 1$ subproblems, each with at most $2s$ regions. Therefore, we can $(1 + \epsilon)$ -approximate OPT' by $(1 + \epsilon)$ -approximating each subproblem in $(k - 1) \cdot f(2s, \epsilon) \leq \mathcal{O}(n\epsilon + 1) \cdot f\left(\frac{1}{\epsilon}, \epsilon\right)$ time.

It remains to show that OPT' is a $(1 + \mathcal{O}(\epsilon))$ -approximation for OPT . Let r_i be shorthand for the radius r_h of M_i ($r_1 = r_k = 0$). By the definition of fatness, the distance between any two points in M_i is at most $\mathcal{O}(r_i)$. By following through OPT and detouring to each point p_i , we get a path through points p_i with length at most $OPT + \mathcal{O}\left(\sum r_i\right)$, and OPT' is at most this amount.

The last remaining step is to show $\sum r_i \leq \mathcal{O}(\epsilon \cdot OPT)$. We apply Lemma 11 to each subsequence, and obtain that $r_i s \leq \mathcal{O}(OPT_i)$, where OPT_i is the optimal distance to tour regions in subsequence i . Note that although the starting and ending subsequences can have sizes less than s , they satisfy $r_i = 0$, so this bound holds for all subsequences. Therefore, $\sum r_i \leq \mathcal{O}(\epsilon \cdot \sum OPT_i) \leq \mathcal{O}(\epsilon \cdot OPT)$. ◀

3.2 Algorithms for convex fat bodies

Using a similar grouping argument, but using constant sized instead of $\frac{1}{\epsilon}$ sized groups, along with earlier methods of using estimates of the path length to place points on the boundaries of the convex fat bodies yields the following results.

► **Theorem 15.** *There is an $\mathcal{O}\left(\frac{n}{\epsilon^{2d-2}}\right)$ -time algorithm for touring disjoint convex fat bodies in d dimensions.*

Proof. We proceed in a similar fashion as Lemma 14, except we define $s \triangleq n_0$, i.e., using constant sized groups instead of $\lceil \frac{1}{\epsilon} \rceil$ sized groups. Let the M_i be defined as in the proof of Lemma 14, and define m_i to be the outer radius of M_i .

For each pair of regions M_i, M_{i+1} , pick arbitrary points $a \in M_i, b \in M_{i+1}$, and use the d -dimensional analog of Theorem 8⁴ to obtain a 4-approximation D_{approx} of the length of the shortest path from a to b in $\mathcal{O}(1)$ time. Suppose that the optimal path uses $p \in M_i, q \in M_{i+1}$ and the shortest path from a to b has distance $OPT_{a,b}$; by the triangle inequality, we must have

$$\frac{1}{4} D_{approx} \leq OPT_{a,b} \leq OPT_i + 2m_i + 2m_{i+1}.$$

Now, consider the path where we start at p and then travel along the line segment from p to a , the approximate path of length D_{approx} from a to b (visiting the regions in between M_i and M_{i+1}), and the line segment from b to q . This path has length at most $D_{approx} + 2m_i + 2m_{i+1}$, and upper bounds the length of the optimal path between p and q . So, the entire path between p and q lies within a ball of radius $D_{approx} + 4m_i + 2m_{i+1}$ centered at a ; call this ball L . Note that L has radius $l = D_{approx} + 4m_i + 2m_{i+1} \leq \mathcal{O}(OPT_i + m_i + m_{i+1})$.

For each region R_j between M_i and M_{i+1} inclusive, we apply Lemma 4 with the region and a hypercube containing L , which has side length $2l$. Note that points are placed twice on each M_i ; this is fine. Lemma 4 guarantees the existence of a point in R_j that is $2l\epsilon$ close to the point OPT uses by placing $\mathcal{O}\left(\frac{1}{\epsilon^{d-1}}\right)$ points on each region.

We now bound the difference between the optimal and the shortest paths using only the points we placed. The difference is at most

$$\sum_{i=1}^k (2l_i \epsilon \cdot n_0) = \epsilon \cdot \mathcal{O}\left(\sum_{i=1}^k l_i\right) = \epsilon \cdot \mathcal{O}\left(OPT + \sum_{i=1}^k m_i\right) = \mathcal{O}(\epsilon \cdot OPT),$$

where the last step is due to Corollary 12 applied on each subsequence: in particular, the optimal path length visiting all the regions in subsequence i has length at least $\Omega(m_i)$, so summing this inequality over all subsequences, we have $\sum_{i=1}^k m_i \leq \mathcal{O}(OPT)$.

We have now reduced the problem to the case where each region has only finitely many points. We finish with dynamic programming. Since we have $\mathcal{O}\left(\frac{1}{\epsilon^{d-1}}\right)$ points on each of the n regions, the runtime is $\mathcal{O}\left(\frac{n}{\epsilon^{d-2}}\right)$, as desired. ◀

⁴ This theorem may be found in Table 2 of the full version.

► **Theorem 16.** *There is an $\mathcal{O}\left(\frac{n}{\epsilon}\right)$ -time algorithm for touring 2D disjoint convex fat bodies.*

Proof. This is almost the same as Theorem 15, where $\mathcal{O}\left(\frac{1}{\epsilon^{d-1}}\right) = \mathcal{O}\left(\frac{1}{\epsilon}\right)$ points are placed on each body, except that we use Corollary 7 to more efficiently solve the case where each region is a finite point set. ◀

► **Theorem 17.** *There is an $\mathcal{O}\left(\frac{n^2}{\sqrt{\epsilon}}\right)$ -time algorithm for touring 2D disjoint convex fat bodies.*

Proof. Theorem 16 through the construction of Theorem 15 places $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ points on an arc of length R on each convex fat body to guarantee additive error $\leq \epsilon R$. We can achieve the same additive error using a convex polygon with $\mathcal{O}\left(\epsilon^{-1/2}\right)$ vertices using Lemma 9. Then, recall that [19] gives an $\mathcal{O}(|V|n)$ -time exact algorithm for touring convex polygons, so we can recover a solution in $\mathcal{O}(|V|n) = \mathcal{O}\left((n \cdot \epsilon^{-1/2}) \cdot n\right)$ time. ◀

4 Balls

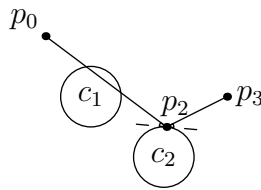
We can improve the results in previous sections by discretizing the surfaces non-uniformly, placing fewer points on areas of each hypersphere that are farther away from the previous and next ball in the sequence. This reduces the dependence on ϵ by a square root compared to Theorem 15 and Theorem 16. We first state the results:

► **Theorem 18.** *There is an $\mathcal{O}\left(\frac{n}{\sqrt{\epsilon}} \log \frac{1}{\epsilon} + \frac{1}{\epsilon}\right)$ -time algorithm for touring disjoint disks.*

► **Theorem 19.** *There is an $\mathcal{O}\left(\frac{n}{\epsilon^{d-1}} \log^2 \frac{1}{\epsilon} + \frac{1}{\epsilon^{2d-2}}\right)$ -time algorithm for touring disjoint balls in d dimensions.*

The crucial lemma we use for these results follows. We defer its proof to the full version.

► **Lemma 20.** *A tour of disjoint balls is globally optimal if and only if for each intermediate ball, the tour either passes straight through the ball or perfectly reflects off its border (see Figure 2 for an example).*



■ **Figure 2** Lemma 20: An optimal tour of two unit disks. The tour starts at p_0 , passes through c_1 , reflects off c_2 at p_2 , and ends at p_3 .

We start with the special case of unit disks and then generalize to non-unit disks (Theorem 18). First, we provide intuition through a simple example where $n = 1$ and R_1 is a line.

► **Example 21.** Given start and endpoints $p_0 = (-1, 1)$ and $p_2 = (1, 1)$, select p_1 from the x -axis such that $OPT = \|p_0 - p_1\| + \|p_1 - p_2\|$ is minimized.

Solution. To solve this exactly, choose $p_1 = (0, 0)$ such that the path perfectly reflects off the x -axis. This gives $OPT = 2\sqrt{2}$.

Now suppose that we are only interested in an approximate solution. Tile the x -axis with points at regular intervals such that every two consecutive points are separated by distance d , and round p_1 to the closest such point p'_1 . Since $\|p_1 - p'_1\| \leq d$,

$$\begin{aligned} OPT' &\triangleq \|p_0 - p'_1\| + \|p'_1 - p_2\| \\ &\leq \sqrt{1 + (1-d)^2} + \sqrt{1 + (1+d)^2} \leq \sqrt{2-2d+d^2} + \sqrt{2+2d+d^2} \\ &\leq \sqrt{2}(1-d/2+1+d/2 + \mathcal{O}(d^2)) \leq 2\sqrt{2}(1 + \mathcal{O}(d^2)). \end{aligned}$$

So, to attain $OPT' \leq (1 + \epsilon)OPT$, it suffices to take $d = \Theta(\sqrt{\epsilon})$ rather than $d = \Theta(\epsilon)$ because $p'_1 - p_1$ is parallel to the x -axis. We can apply a similar idea to replace the middle region with a point set when R_1 is a circle rather than a line since circles are locally linear. However, this doesn't quite work when either $\|p_0 - p_1\|$ or $\|p_1 - p_2\|$ is small. For example, if p_0 was very close to the x -axis (say, $p_0 = (-d, d)$) then rounding p_1 to the nearest p'_1 could cause OPT' to increase by $\Theta(d) \gg d^2$. So when we replace each circle with a point set, we need to be careful about how we handle two circles that are close to touching; the solution is to space points more densely near where they touch. ◀

► **Theorem 22.** *There is an $\mathcal{O}\left(\frac{n}{\sqrt{\epsilon}} \log \frac{1}{\epsilon}\right)$ -time algorithm for touring disjoint unit disks.*

Proof. We describe how to place a set of $\mathcal{O}\left(\frac{1}{\sqrt{\epsilon}} \log \frac{1}{\epsilon}\right)$ points S_i on each unit circle c_i so that the length of an optimal path increases by at most $\mathcal{O}(n\epsilon)$ after rounding each p_i to the nearest $p'_i \in S_i$. Define $\text{unit}(x) = \frac{x}{\|x\|}$. Let $o_i \triangleq p'_i - p_i$ for all $i \in [0, n + 1]$ (note that $o_0 = o_{n+1} = 0$), where o stands for offset. Also, define vectors

$$d_i \triangleq p'_{i+1} - p'_i = p_{i+1} + o_{i+1} - p_i - o_i$$

and scalars

$$a_i \triangleq d_i \cdot \text{unit}(p_{i+1} - p_i) = \|p_{i+1} - p_i\| + (o_{i+1} - o_i) \cdot \text{unit}(p_{i+1} - p_i),$$

where a_i is the component of d_i along the direction of $p_{i+1} - p_i$. Then the total path length after rounding each p_i to p'_i is:

$$\begin{aligned} \sum_{i=0}^n \|d_i\| &= \sum_{i=0}^n \sqrt{[d_i \cdot \text{unit}(p_{i+1} - p_i)]^2 + [d_i \cdot \text{unit}(p_{i+1} - p_i)^\perp]^2} \\ &= \sum_{i=0}^n \sqrt{a_i^2 + [(o_{i+1} - o_i) \cdot \text{unit}(p_{i+1} - p_i)^\perp]^2} \\ &= \sum_{i=0}^n \left[a_i + \left(\sqrt{a_i^2 + [(o_{i+1} - o_i) \cdot \text{unit}(p_{i+1} - p_i)^\perp]^2} - a_i \right) \right] \\ &= OPT + \sum_{i=1}^n \overbrace{o_i \cdot (\text{unit}(p_i - p_{i-1}) - \text{unit}(p_{i+1} - p_i))}^{\text{extra}_1(i)} \\ &\quad + \sum_{i=0}^n \overbrace{\left(\sqrt{a_i^2 + [(o_{i+1} - o_i) \cdot \text{unit}(p_{i+1} - p_i)^\perp]^2} - a_i \right)}^{\text{extra}_2(i)} \\ &= OPT + \sum_{i=1}^n \text{extra}_1(i) + \sum_{i=0}^n \text{extra}_2(i). \end{aligned}$$

We defer the construction of the sets S_i so that both extra terms are small to Lemma 23. Then we can finish with dynamic programming (Corollary 7). ◀

54:12 New Approximation Algorithms for Touring Regions

► **Lemma 23.** *It is possible to choose S_i in the proof of Theorem 22 such that $|S_i| \leq \mathcal{O}\left(\frac{1}{\sqrt{\epsilon}} \log \frac{1}{\epsilon}\right)$, $\text{extra}_1(i) \leq \mathcal{O}(\epsilon)$, and $\text{extra}_2(i) \leq \mathcal{O}(\epsilon)$ for all i .*

Proof. First, we present the construction. For every pair of adjacent disks i and $i + 1$ we describe a procedure to generate points on their borders. Then we set S_i to be the union of the generated points on the border of disk i when running the procedure on disks $(i, i + 1)$ and on disks $(i - 1, i)$. Finally, we show that $\text{extra}_1(i)$ and $\text{extra}_2(i)$ are sufficiently small for all i for our choice of S_i .

Procedure. Reorient the plane that $c_i = (0, y)$ and $c_{i+1} = (0, -y)$ for some $y > 1$. Let $\text{spacing}: \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{> 0}$ be a function that is nonincreasing in $|\phi|$ that we will define later. Given spacing , we use the following process to add points to S_i (and symmetrically for S_{i+1}):

1. Set $\phi = 0$.
2. While $\phi \leq \pi$:
 - Add $(\sin \phi, y - \cos \phi)$ to S_i .
 - $\phi += \text{spacing}(\phi)$.
3. Repeat steps 1-2 but for ϕ from 0 to $-\pi$.

This procedure has the property that for any $\phi \in [-\pi, \pi]$, the point $(\sin \phi, y - \cos \phi)$ is within distance $\text{spacing}(|\phi|)$ of some point in S_i . In particular, if the optimal path has $p_i = (\sin \phi_i, y - \cos \phi_i)$ then it is guaranteed that $\|o_i\| \leq \text{spacing}(\phi_i)$. To compute $|S_i|$, note that as long as $\text{spacing}(\phi)$ is sufficiently smooth that $\frac{\text{spacing}(\phi)}{\text{spacing}(\phi + \text{spacing}(\phi))} = \Theta(1)$ for all ϕ , the number of points added to S_i will be at most a constant factor larger than the value of the definite integral $\int_{-\pi}^{\pi} \frac{1}{\text{spacing}(\phi)} d\phi$.

Next, we construct spacing so that $|S_i| = \mathcal{O}\left(\frac{1}{\sqrt{\epsilon}} \log \frac{1}{\epsilon}\right)$. Intuitively, by Example 21, we should have $\text{spacing}(\phi) = \Theta(\epsilon)$ closer to circle $i + 1$ (when $\phi \approx 0$) and $\text{spacing}(\phi) = \Theta(\sqrt{\epsilon})$ farther from circle $i + 1$ (when $\phi = \Theta(1)$). Thus, we set $\text{spacing}(\phi) = \max(\epsilon, \sqrt{\epsilon}\phi)$. The total number of added points is on the order of:

$$\begin{aligned} \int_0^{\pi} \frac{1}{\text{spacing}(\phi)} d\phi &= \frac{1}{\sqrt{\epsilon}} \left(\int_0^{\sqrt{\epsilon}} \frac{1}{\sqrt{\epsilon}} d\phi + \int_{\sqrt{\epsilon}}^{\pi} \frac{1}{\phi} d\phi \right) \\ &= \frac{1}{\sqrt{\epsilon}} \left(1 + \log \left(\frac{\pi}{\sqrt{\epsilon}} \right) \right) \leq \mathcal{O} \left(\frac{1}{\sqrt{\epsilon}} \log \frac{1}{\epsilon} \right). \end{aligned}$$

Finally, we show that both extra terms are small for our choice of S_i .

Part 1: $\text{extra}_1(i)$. We note that $\text{unit}(p_i - p_{i-1}) - \text{unit}(p_{i+1} - p_i)$ must be parallel to $p_i - c_i$ for an optimal solution p . To verify this, it suffices to check the two cases from Lemma 20:

1. The points p_{i-1}, p_i, p_{i+1} are collinear, in which case $\text{unit}(p_i - p_{i-1}) - \text{unit}(p_{i+1} - p_i) = 0$.
2. The path reflects perfectly off circle i , in which case $\text{unit}(p_i - p_{i-1}) - \text{unit}(p_{i+1} - p_i)$ is parallel to $p_i - c_i$.

If we ensure that $\text{spacing}(\phi) \leq \sqrt{\epsilon}$ for all ϕ , then $|o_i \cdot \text{unit}(p_i - c_i)| \leq \epsilon$ because o_i is always nearly tangent to the circle centered at c_i at point p_i . The conclusion follows because $\text{extra}_1(i) \leq 2|o_i \cdot \text{unit}(p_i - c_i)| \leq 2\epsilon$.

Part 2: $\text{extra}_2(i)$. We upper bound $\text{extra}_2(i)$ by the sum of two summands, the first associated only with o_i and the second associated only with o_{i+1} .

▷ **Claim 24.** Letting $\text{ycoord}(\cdot)$ denote the y -coordinate of a point,

$$\text{extra}_2(i) \leq 2 \cdot \left(\min \left(\|o_i\|, \frac{4 \|o_i\|^2}{\text{ycoord}(p_i)} \right) + \min \left(\|o_{i+1}\|, \frac{4 \|o_{i+1}\|^2}{-\text{ycoord}(p_{i+1})} \right) \right).$$

Proof. We do casework based on which term is smaller on each of the mins.

1. $\|o_i\| \geq \frac{\text{ycoord}(p_i)}{4}, \|o_{i+1}\| \geq \frac{-\text{ycoord}(p_{i+1})}{4}$

The result, $\text{extra}_2(i) \leq 2(\|o_i\| + \|o_{i+1}\|)$, follows by summing the following two inequalities:

$$\begin{aligned} & \sqrt{a_i^2 + [(o_{i+1} - o_i) \cdot \text{unit}(p_{i+1} - p_i)^\perp]^2} - \|p_{i+1} - p_i\| \\ &= \|p_{i+1} - p_i + o_{i+1} - o_i\| - \|p_{i+1} - p_i\| \leq \|o_i\| + \|o_{i+1}\| \end{aligned}$$

and $\|p_{i+1} - p_i\| - a_i \leq \|o_i\| + \|o_{i+1}\|$.

2. $\|o_i\| \leq \frac{\text{ycoord}(p_i)}{4}, \|o_{i+1}\| \leq \frac{-\text{ycoord}(p_{i+1})}{4}$

Then $\|o_i\|, \|o_{i+1}\| \leq \frac{\|p_{i+1} - p_i\|}{4}$ so $a_i \geq \frac{\|p_{i+1} - p_i\|}{2}$, and

$$\begin{aligned} \text{extra}_2(i) &\leq \frac{\|o_{i+1} - o_i\|^2}{2a_i} \leq \frac{2(\|o_{i+1}\|^2 + \|o_i\|^2)}{2a_i} \\ &\leq 2 \cdot \frac{\|o_{i+1}\|^2 + \|o_i\|^2}{\|p_i - p_{i+1}\|} \leq 2 \cdot \left(\frac{\|o_i\|^2}{\text{ycoord}(p_i)} + \frac{\|o_{i+1}\|^2}{-\text{ycoord}(p_{i+1})} \right). \end{aligned}$$

3. $\|o_i\| \leq \frac{\text{ycoord}(p_i)}{4}, \|o_{i+1}\| \geq \frac{-\text{ycoord}(p_{i+1})}{4}$

Define $\text{extra}'(i)$ to be the same as $\text{extra}_2(i)$ with o_{i+1} set to 0. Then

$$\begin{aligned} \text{extra}'(i) &\triangleq \|p_{i+1} - p_i - o_i\| - (\|p_{i+1} - p_i\| - o_i \cdot \text{unit}(p_{i+1} - p_i)) \\ &= \sqrt{(\|p_{i+1} - p_i\| - o_i \cdot \text{unit}(p_{i+1} - p_i))^2 + [o_i \cdot \text{unit}(p_{i+1} - p_i)^\perp]^2} \\ &\quad - (\|p_{i+1} - p_i\| - o_i \cdot \text{unit}(p_{i+1} - p_i)) \\ &\leq \frac{\|o_i\|^2}{2 \cdot \frac{3}{4} \|p_i - p_{i+1}\|} \leq \frac{\|o_i\|^2}{2 \cdot \frac{3}{4} \cdot \text{ycoord}(p_i)} \end{aligned}$$

and by similar reasoning as case 1, $\text{extra}_2(i) - \text{extra}'(i) \leq 2 \|o_{i+1}\|$.

4. $\|o_i\| \geq \frac{\text{ycoord}(p_i)}{4}, \|o_{i+1}\| \leq \frac{-\text{ycoord}(p_{i+1})}{4}$

Similar to case 3. ◀

Now that we have a claim showing an upper bound on $\text{extra}_2(i)$, it remains to show that $\min \left(\|o_i\|, \frac{\|o_i\|^2}{\text{ycoord}(p_i)} \right) \leq \mathcal{O}(\epsilon)$ for our choice of *spacing*. Indeed, when $\phi \leq \sqrt{\epsilon}$ we have

$$\|o_i\| \leq \text{spacing}(\phi) \leq \epsilon, \text{ while for } \phi > \sqrt{\epsilon} \text{ we have } \frac{\|o_i\|^2}{\text{ycoord}(p_i)} \leq \mathcal{O} \left(\frac{\text{spacing}(\phi)}{\phi^2} \right) \leq \mathcal{O}(\epsilon). \quad \blacktriangleleft$$

With small modifications to the proof of Lemma 23, we have the following corollary:

▶ **Corollary 25.** *Consider the case of non-unit disks. If the i th disk has radius r_i , then we can place $\mathcal{O} \left(\frac{1}{\sqrt{\epsilon_i}} \log \frac{1}{\epsilon_i} \right)$ points on its border such that the additive error associated with c_i —specifically, $\text{extra}_1(i)$ plus the components of $\text{extra}_2(i-1)$ and $\text{extra}_2(i)$ associated with $\|o_i\|$ — is $\mathcal{O}(r_i \epsilon_i)$. Consequently, $\text{OPT} + \sum_{i=1}^n \text{extra}_1(i) + \sum_{i=0}^n \text{extra}_2(i) \leq \text{OPT} + \sum_{i=1}^n r_i \epsilon_i$.*

54:14 New Approximation Algorithms for Touring Regions

Now, we finally prove Theorems 18 and 19.

Proof of Theorem 18 (Non-Unit Disks). We first present a slightly weaker result, and then show how to improve it. Recall that by Corollary 12, the i th largest disk has radius $\mathcal{O}\left(\frac{OPT}{i}\right)$ for $i \geq 3$. So if we set $\epsilon_i = \epsilon' = \frac{\epsilon}{\log n}$ for each of the i th largest disks for $i \geq 3$, the total additive error contributed by these disks becomes

$$\mathcal{O}\left(\sum_{i=3}^n \frac{OPT}{i} \cdot \epsilon_i\right) \leq \mathcal{O}\left(OPT \cdot \epsilon' \cdot \sum_{i=3}^n \frac{1}{i}\right) \leq \mathcal{O}(\epsilon OPT)$$

by Corollary 25. For the two largest disks, we use the previous naive discretization (placing $\mathcal{O}\left(\frac{1}{\epsilon}\right)$ points uniformly on the intersection of the circles with a square of side length $\mathcal{O}(OPT)$ centered about the starting point). We may assume we have already computed a constant approximation to OPT in $\mathcal{O}(n)$ time by applying Theorem 16 with $\epsilon = 1$. After selecting the point sets, we can finish with Corollary 7. The overall time complexity is $\mathcal{O}\left(\frac{n}{\sqrt{\epsilon}} \log \frac{1}{\epsilon'} + \frac{1}{\epsilon}\right) \leq \mathcal{O}\left(\frac{n\sqrt{\log n}}{\sqrt{\epsilon}} \log\left(\frac{\log n}{\epsilon}\right) + \frac{1}{\epsilon}\right)$.

We can remove the factors of $\log n$ by selecting the ϵ_i to be an increasing sequence. Set $\epsilon_i = \Theta\left(\frac{\epsilon_i^{2/3}}{n^{2/3}}\right)$ for each $i \in [3, n]$ such that more points are placed on larger disks. Then the total added error remains

$$\begin{aligned} \mathcal{O}\left(OPT \cdot \left(\epsilon + \sum_{i=3}^n \frac{\epsilon_i}{i}\right)\right) &= \mathcal{O}\left(OPT \cdot \left(\epsilon + \sum_{i=3}^n \frac{1}{i} \cdot \frac{\epsilon_i^{2/3}}{n^{2/3}}\right)\right) \\ &= \mathcal{O}\left(OPT \epsilon \cdot \left(1 + n^{-2/3} \cdot \sum_{i=3}^n i^{-1/3}\right)\right) \leq \mathcal{O}(OPT \epsilon), \end{aligned}$$

and the factors involving $\log n$ drop out from the time complexity:

$$\begin{aligned} \mathcal{O}\left(\sum_{i=3}^n \frac{1}{\sqrt{\epsilon_i}} \log\left(\frac{1}{\epsilon_i}\right) + \frac{1}{\epsilon}\right) &\leq \mathcal{O}\left(\int_{i=3}^n \frac{1}{\sqrt{\epsilon}} n^{1/3} i^{-1/3} \log\left(\frac{n^{2/3}}{i^{2/3} \epsilon}\right) di + \frac{1}{\epsilon}\right) \\ &\leq \mathcal{O}\left(\frac{3n^{1/3}}{2\sqrt{\epsilon}} i^{2/3} \left(\log \frac{n^{2/3}}{i^{2/3} \epsilon} + 1\right) \Big|_3^n + \frac{1}{\epsilon}\right) \\ &\leq \mathcal{O}\left(\frac{n}{\sqrt{\epsilon}} \log\left(\frac{1}{\epsilon}\right) + \frac{1}{\epsilon}\right). \quad \blacktriangleleft \end{aligned}$$

To extend to multiple dimensions, we generalize the construction from Lemma 23.

Proof of Theorem 19 (Balls). As in Lemma 23, set $spacing(\phi) = \max(\epsilon, \sqrt{\epsilon}\phi)$ for a point p_i satisfying $m\angle p_i c_i c_{i+1} = \phi$, meaning that there must exist $p'_i \in S_i$ satisfying $\|p_i - p'_i\| \leq r_i \cdot spacing(\phi)$. The total number of points $|S_i|$ placed on the surface of a d -dimensional sphere is proportional to

$$\begin{aligned} \int_0^\pi \frac{\sin^{d-2}(\phi)}{spacing(\phi)^{d-1}} d\phi &\leq \frac{1}{(\sqrt{\epsilon})^{d-1}} \int_0^\pi \frac{\phi^{d-2}}{\max(\sqrt{\epsilon}, \phi)^{d-1}} d\phi \\ &= \frac{1}{\epsilon^{(d-1)/2}} \left(\int_0^{\sqrt{\epsilon}} \frac{\phi^{d-2}}{(\sqrt{\epsilon})^{d-1}} d\phi + \int_0^{\sqrt{\epsilon}} \frac{1}{\phi} d\phi \right) \leq \mathcal{O}\left(\frac{1}{\epsilon^{(d-1)/2}} \log \frac{1}{\epsilon}\right), \end{aligned}$$

where the derivation of the integration factor $\sin^{d-2}(\phi)$ can be found in [18].

It remains to describe how to space points so that they satisfy the given spacing function. For each spacing $s = \epsilon, 2\epsilon, 4\epsilon, \dots, \sqrt{\epsilon}$, we can find a d -dimensional hypercube of side length

$O(s/\sqrt{\epsilon})$ that encloses all points on the hypersphere with required spacing at most $2s$. Evenly space points with spacing s across the surface of this hypercube according to Lemma 4, and project each of these points onto the hypersphere. There are a total of $\mathcal{O}(\log \frac{1}{\epsilon})$ values of s , and each s results in $\mathcal{O}(\frac{1}{\epsilon^{(d-1)/2}})$ points being projected onto the hypersphere, for a total of $\mathcal{O}(\frac{1}{\epsilon^{(d-1)/2}} \log \frac{1}{\epsilon})$ points. ◀

References

- 1 Xxii open cup, grand prix of siberia, 2022. Last accessed 9 April 2022. URL: <https://codeforces.com/blog/entry/96710>.
- 2 Alok Aggarwal and Maria Klawe. Applications of generalized matrix searching to geometric algorithms. *Discrete Applied Mathematics*, 27(1-2):3–23, 1990.
- 3 Arash Ahadi, Amirhossein Mozafari, and Alireza Zarei. Touring a sequence of disjoint polygons: Complexity and extension. *Theoretical Computer Science*, 556:45–54, 2014.
- 4 Tetsuo Asano, David Kirkpatrick, and Chee Yap. Pseudo approximation algorithms with applications to optimal motion planning. *Discrete & Computational Geometry*, 31(1):139–171, 2004.
- 5 Heinz H. Bauschke and Jonathan M. Borwein. On projection algorithms for solving convex feasibility problems. *SIAM Review*, 38(3):367–426, 1996. doi:10.1137/S0036144593251710.
- 6 Timothy M Chan. (near-) linear-time randomized algorithms for row minima in monge partial matrices and related problems. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1465–1482. SIAM, 2021.
- 7 Chang-Chien Chou. An exact algorithm for computing the shortest path touring n circles in 2d. In *International Conference on Computational Problem-Solving*, pages 98–103, 2010.
- 8 P. L. Combettes and H. J. Trussell. Method of successive projections for finding a common point of sets in metric spaces, December 1990. doi:10.1007/bf00939646.
- 9 Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph S. B. Mitchell. Touring a sequence of polygons. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC '03*, pages 473–482, New York, NY, USA, 2003. Association for Computing Machinery. doi:10.1145/780542.780612.
- 10 Alon Efrat, Günter Rote, and Micha Sharir. On the union of fat wedges and separating a collection of segments by a line. *Comput. Geom. Theory Appl.*, 3(5):277–288, November 1993. doi:10.1016/0925-7721(93)90018-2.
- 11 Peter Gritzmann and Victor Klee. *Computational convexity*. CRC Press, Taylor; Francis Group, a Chapman; Hall Book, 2017.
- 12 Matthew J. Katz. 3-d vertical ray shooting and 2-d point enclosure, range searching, and arc shooting amidst convex fat objects. *Computational Geometry*, 8(6):299–316, 1997. doi:10.1016/S0925-7721(96)00027-2.
- 13 Amirhossein Mozafari and Alireza Zarei. Touring polygons: An approximation algorithm. In *IWOCA*, 2012.
- 14 Mark H. Overmars. Point location in fat subdivisions. *Information Processing Letters*, 44(5):261–265, 1992. doi:10.1016/0020-0190(92)90211-D.
- 15 Mark H. Overmars and A.F. van der Stappen. Range searching and point location among fat objects. *J. Algorithms*, 21:629–656, 1996.
- 16 R. R. Phelps. Convex sets and nearest points. *Proceedings of the American Mathematical Society*, 8(4):790–797, 1957. URL: <http://www.jstor.org/stable/2033300>.
- 17 Valentin Polishchuk and Joseph SB Mitchell. Touring convex bodies—a conic programming solution. In *CCCG*, pages 290–293, 2005.
- 18 David J. Smith and Mavina K. Vamanamurthy. How small is a unit ball? *Mathematics Magazine*, 62(2):101–107, 1989. URL: <http://www.jstor.org/stable/2690391>.

54:16 New Approximation Algorithms for Touring Regions

- 19 Xuehou Tan and Bo Jiang. Efficient algorithms for touring a sequence of convex polygons and related problems. In T.V. Gopal, Gerhard Jäger, and Silvia Steila, editors, *Theory and Applications of Models of Computation*, pages 614–627, Cham, 2017. Springer International Publishing.

Combinatorial Depth Measures for Hyperplane Arrangements

Patrick Schneider ✉ 🏠 

Department of Computer Science, ETH Zürich, Switzerland

Pablo Soberón ✉ 🏠 

Department of Mathematics, Baruch College, City University of New York, NY, USA

Department of Mathematics, The Graudate Center, City University of New York, NY, USA

Abstract

Regression depth, introduced by Rousseeuw and Hubert in 1999, is a notion that measures how good of a regression hyperplane a given query hyperplane is with respect to a set of data points. Under projective duality, this can be interpreted as a depth measure for query points with respect to an arrangement of data hyperplanes. The study of depth measures for query points with respect to a set of data points has a long history, and many such depth measures have natural counterparts in the setting of hyperplane arrangements. For example, regression depth is the counterpart of Tukey depth. Motivated by this, we study general families of depth measures for hyperplane arrangements and show that all of them must have a deep point. Along the way we prove a Tverberg-type theorem for hyperplane arrangements, giving a positive answer to a conjecture by Rousseeuw and Hubert from 1999. We also get three new proofs of the centerpoint theorem for regression depth, all of which are either stronger or more general than the original proof by Amenta, Bern, Eppstein, and Teng. Finally, we prove a version of the center transversal theorem for regression depth.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Combinatorics

Keywords and phrases Depth measures, Hyperplane arrangements, Regression depth, Tverberg theorem

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.55

Related Version *Full Version:* <https://arxiv.org/abs/2302.07768>

Funding *Pablo Soberón:* Supported by NSF DMS grant 2054419.

1 Introduction

A central topic in combinatorial geometry and computational geometry is the study of structural properties of finite families of points in Euclidean spaces. Studying which sets can be separated from others by hyperplanes is a natural question, which leads us to study combinatorial properties of convex sets. Classic results, such as Tverberg's theorem [26] and Rado's centerpoint theorem [17] follow from this line of thought.

In some cases, instead of being provided our data as a finite set of points in \mathbb{R}^d , we might receive it as a set of hyperplanes. Understanding which results for families of points transfer to families of hyperplanes is a natural question.

Given a hyperplane arrangement A in \mathbb{R}^d and a point q , we first consider the depth of q with respect to A as follows.

▶ **Definition 1.** *The regression depth of a query point q with respect to hyperplane arrangement A , denoted by $RD(A, q)$, is the minimum number of hyperplanes in A intersected by or parallel to any ray emanating from q .*

Note that if q lies on a hyperplane H , then any ray emanating from q intersects H . Regression depth has been widely studied [1, 7, 21, 22, 28]. In this manuscript we provide



© Patrick Schneider and Pablo Soberón;

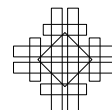
licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 55; pp. 55:1–55:14

Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



new structural results for regression depth, related to Tverberg's theorem and *enclosing depth*. In particular, given a finite arrangement A of hyperplanes in \mathbb{R}^d , we might measure the depth of a point q in A in several different ways, so we study general properties of depth measures with respect to arrangements of hyperplanes. This follows a similar approach recently taken for depth measures with respect to finite families of points [23].

Given an arrangement A of n hyperplanes, the existence of points with regression depth at least $n/(d+1)$ has been established by Amenta, Bern, Eppstein, and Teng [1], and later by Mizera [15] as well as Karasev [11]. This can be considered a hyperplane version of Rado's centerpoint theorem [17]. We give three new proofs of the existence of points with large regression depth. First, we prove a Tverberg-type theorem for hyperplanes, confirming a conjecture of Rousseeuw and Hubert [21].

► **Theorem 2.** *Let r, d be positive integers and A be an arrangement of at least $(r-1)(d+1)+1$ hyperplanes in \mathbb{R}^d . Then, there exists a point q in \mathbb{R}^d and a partition of A into r parts such that q has positive regression depth with respect to each of the r parts.*

This was previously known when $d = 2$ [21] or when r is a prime power [12, 13]. The version for prime powers by Karasev holds with a slightly more restrictive version of regression depth. Based on this result, we define the *hyperplane Tverberg depth* of a point.

► **Definition 3.** *The hyperplane Tverberg depth of a query point q with respect to hyperplane arrangement A , denoted by $HTvD(A, q)$, is the maximum r such that there is a partition of A into r parts such that q has positive regression depth with respect to each part.*

Our other two proofs are topological, and each also has stronger consequences. One proof based on a topological version of Helly's theorem shows the existence of points of high *open regression depth*, which is a slightly weaker measure of depth introduced in Section 5. The last proof, based on properties of vector bundles, works for regression depth in families of weighted arrangements.

Another way to measure the depth of a point with respect to a hyperplane arrangement is via k -enclosures. We say that an arrangement A k -encloses a query point q if A can be partitioned into $d+1$ pairwise disjoint subsets A_1, \dots, A_{d+1} , each of size k , such that for every choice $h_1 \in A_1, \dots, h_{d+1} \in A_{d+1}$ we have that $RD(\{h_1, \dots, h_{d+1}\}, q) \geq 1$.

► **Definition 4.** *The hyperplane enclosing depth of a query point q with respect to a hyperplane arrangement A , denoted by $HED(A, q)$, is the maximum k such that there is a sub-arrangement of A which k -encloses q .*

Given a finite hyperplane arrangement A , we prove the existence of points with high hyperplane enclosing depth with respect to A . In particular, our lower bound is linear in $|A|$. The existence of points with large enclosing depth for families of points has been established by Pach [16] and by Fabila-Monroy and Huemer [6] (see [23] for improved constants).

One striking generalization of Rado's centerpoint theorem is the central transversal theorem, proven independently by Dolnikov and by Živaljević and Vrećica [5, 29]. In Section 7 we prove an analogue for hyperplane arrangements. Given a hyperplane arrangement A in \mathbb{R}^d and a linear subspace L in \mathbb{R}^d , we denote by $A \cap L$ the restriction of A to L . In Theorem 28, we show that given $d-k+1$ different arrangements of hyperplanes in \mathbb{R}^d , there exists a k -dimensional linear subspace L such that the restrictions of each arrangement to L share a point with high regression depth.

In particular, just as the central transversal theorem generalizes the ham sandwich theorem, Theorem 28 has the following corollary.

► **Corollary 5.** *Let A_1, \dots, A_d be d hyperplane arrangements in \mathbb{R}^d . There exists a line ℓ through the origin in \mathbb{R}^d and a point $q \in \ell$ such that each of the two rays in ℓ starting from q intersects at least $|A_i|/2$ hyperplanes of A_i , for each $i = 1, \dots, d$.*

The corollary above is similar to mass partition results for families of hyperplanes with segments [3, 18], and to projective versions of the central transversal theorem [13].

2 Correspondence to depth measures for point sets

For an arrangement A and a query point q , we define the *dual of A at q* , denoted by A_q^* , as follows. For each hyperplane $h \in A$, let $p(h)$ be the unique point on h that is closest to q . We define A_q^* as the set formed by all these points, that is, $A_q^* := \{p(h) \mid h \in A\}$. Note that if q lies on k hyperplanes, then those k dual points coincide with q in A_q^* .

Using this duality, for every depth measure ρ on point sets we can define a corresponding depth measure ρ^* on hyperplane arrangements and vice versa, by setting $\rho^*(A, q) = \rho(A_q^*, q)$. We have the following observation.

- **Observation 6.**
1. *a ray r emanating from q intersects a hyperplane h if and only if the half-space r^\perp defined by the hyperplane through q orthogonal to r , oriented such that it contains r , contains $p(h)$;*
 2. *the point q has positive regression depth with respect to h_1, \dots, h_n if and only if it is in the convex hull of $p(h_1), \dots, p(h_n)$.*
 3. *the point q lies in the simplex defined by h_1, \dots, h_{d+1} if and only if it is in the interior of the convex hull of $p(h_1), \dots, p(h_{d+1})$.*

The three depth measures for hyperplane arrangements defined in Section 1 all have natural corresponding depth measures for point sets that follow immediately from Observation 6. For regression depth, the corresponding depth measure is *Tukey depth* (TD), which is defined as the minimum number of data points contained in any closed half-space containing the query point q [25]. For hyperplane Tverberg depth we get Tverberg depth (TvD), which is defined as the maximum r for which there exists an r -partition of the data points containing the query point q in their intersection. Finally, for hyperplane enclosing depth, we get *enclosing depth* (ED), which is defined as the maximum k for which there exists a subset of the data points that k -encloses the query point q [23].

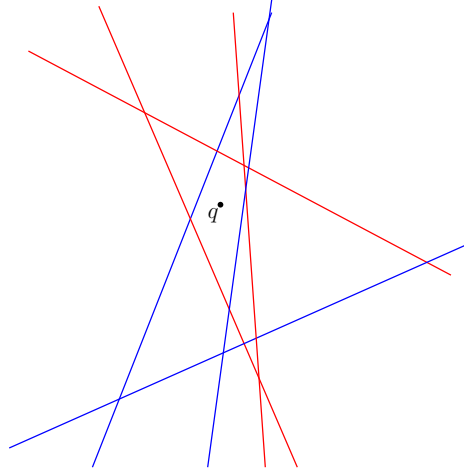
► **Corollary 7.** *Let A be an arrangement of hyperplanes in general position in \mathbb{R}^d and let q be a query point. Then*

1. $RD(A, q) = TD(A_q^*, q)$;
2. $HTvD(A, q) = TvD(A_q^*, q)$;
3. $HED(A, q) = ED(A_q^*, q)$.

3 Axioms for hyperplane depth

Let $A^{\mathbb{R}^d}$ denote the family of all finite arrangements of hyperplanes in \mathbb{R}^d . A depth measure for hyperplanes is a function $\rho : (A^{\mathbb{R}^d}, \mathbb{R}^d) \rightarrow \mathbb{R}_{\geq 0}$ which assigns to each pair (A, q) consisting of a hyperplane arrangement A and a query point q a value, which describes how deep the query point q lies within the arrangement A . A depth measure is called *combinatorial* if it is the same for all points in a face of A . Similar to [23], we introduce some axioms, that reasonable depth measures for hyperplane arrangements should satisfy.

We say that a combinatorial depth measure for hyperplanes is *super-additive* if it satisfies the following four conditions.



■ **Figure 1** Hyperplane enclosing depth does not satisfy condition (iv): the point q has hyperplane enclosing depth 1 with respect to both the blue and the red lines, but its hyperplane enclosing depth with respect to the union of the two sets is still 1.

- (i) for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ and any hyperplane h we have $|\rho(A, q) - \rho(A \cup \{h\}, q)| \leq 1$,
- (ii) for all $A \in A^{\mathbb{R}^d}$ we have $\rho(A, q) = 0$ if q is in an unbounded cell of A ,
- (iii) for all $A \in A^{\mathbb{R}^d}$ we have $\rho(A, q) \geq 1$ if q is in a bounded cell or if q lies on a hyperplane of A ,
- (iv) for any disjoint subsets $A_1, A_2 \subseteq A$ and $q \in \mathbb{R}^d$ we have $\rho(A, q) \geq \rho(A_1, q) + \rho(A_2, q)$.

► **Observation 8.** *Regression depth and hyperplane Tverberg depth are super-additive, but hyperplane enclosing depth is not.*

For hyperplane enclosing depth, an example with $\text{HED}(A_1, q) = \text{HED}(A_2, q) = \text{HED}(A, q) = 1$ can be found in Figure 1.

► **Lemma 9.** *Let ρ be any combinatorial depth measure that satisfies conditions (i) and (ii). Then for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ we have $\rho(A, q) \leq \text{RD}(A, q)$.*

Proof. Let $\text{RD}(A, q) = k$. This means that there is a ray r which intersects or is parallel to some k hyperplanes of A . Removing these k hyperplanes, we get a new arrangement A' and we have $\text{RD}(A', q) = 0$. In particular, q is in an unbounded cell of A' and thus also $\rho(A', q) = 0$ by condition (ii). By condition (i) we have $\rho(A, q) \leq \rho(A', q) + k = k$. ◀

► **Lemma 10.** *Let ρ be any combinatorial depth measure that satisfies conditions (iii) and (iv). Then for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ we have $\rho(A, q) \geq \text{HTvD}(A, q)$.*

Proof. Let $\text{HTvD}(A, q) = k$. This means that there is a k -partition A_1, \dots, A_k such that q has regression depth ≥ 1 with respect to each part. By condition (iii) we have $\rho(A_i, q) \geq 1$ for each A_i . By condition (iv) we get $\rho(A, q) \geq \rho(A_1, q) + \dots + \rho(A_k, q) \geq k$. ◀

► **Lemma 11.** *For all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ we have $\text{HTvD}(A, q) \geq \frac{1}{d} \text{RD}(A, q)$.*

Proof. By Corollary 7 we have $\text{HTvD}(A, q) = \text{TvD}(A_q^*, q)$ and $\text{RD}(A, q) = \text{TD}(A_q^*, q)$. It is well known that for any point set S in \mathbb{R}^d and any query point q we have $\text{TvD}(S, q) \geq \frac{1}{d} \text{TD}(S, q)$, see e.g. [1, 8, 19]. ◀

Combining all of the above, we get

► **Theorem 12.** *Let ρ be a super-additive depth measure for hyperplanes. Then for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ we have $RD(A, q) \geq \rho(A, q) \geq HTvD(A, q) \geq \frac{1}{d}RD(A, q)$.*

As we have seen above, not all depth measures are super-additive: hyperplane enclosing depth is an example of a measure that is not. To include more general depth measures, we define a second family of measures, defined by a weaker set of axioms. We call a combinatorial depth measure for hyperplanes *enclosable* if it satisfies the following conditions.

- (i) for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ and any hyperplane h we have $|\rho(A, q) - \rho(A \cup \{h\}, q)| \leq 1$,
- (ii) for all $A \in A^{\mathbb{R}^d}$ we have $\rho(A, q) = 0$ if q is in an unbounded cell of A ,
- (iii') for all $A \in A^{\mathbb{R}^d}$ we have $\rho(A, q) \geq k$ if A k -encloses q ,
- (iv') for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ and any hyperplane h we have $\rho(A \cup \{h\}, q) \geq \rho(A, q)$.

► **Observation 13.** *Regression depth, hyperplane Tverberg depth and hyperplane enclosing depth are all enclosable.*

By Lemma 9, any enclosable depth measure is bounded from above by regression depth. On the other hand, it follows immediately from conditions (iii') and (iv') that any enclosable depth measure is bounded from below by hyperplane enclosing depth. We finish this section by showing a lower bound for hyperplane enclosing depth. In Theorem 17 in [23] it was shown that there is a constant $c(d)$ such that for any point set S in \mathbb{R}^d and any query point q we have $ED(S, q) \geq c \cdot TD(S, q)$. Let now q be a point of largest regression depth for a hyperplane arrangement A . We will see in Theorem 24 that q has regression depth at least $\frac{|A|}{d+1}$. By Observation 6, this means $TD(A_q^*, q) \geq \frac{|A|}{d+1}$. By Theorem 17 in [23], it follows that $ED(A_q^*, q) \geq \frac{c|A|}{d+1}$. Using Observation 6 again, we deduce the following:

► **Theorem 14.** *Let A be an arrangement of hyperplanes in \mathbb{R}^d . There is a constant $c = c(d)$ such that there is a query point q with hyperplane enclosing depth $HED(A, q) \geq \frac{c|A|}{d+1}$.*

Combining all of the above, we get an analogue to Theorem 12.

► **Theorem 15.** *Let ρ be an enclosable depth measure for hyperplanes. Then for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ we have $RD(A, q) \geq \rho(A, q) \geq HED(A, q) \geq c \cdot RD(A, q)$.*

In particular, all combinatorial depth measures for hyperplanes that we consider in this paper are constant factor approximations of regression depth. In the next three sections, we give three lower bounds for the depth of a deepest point. In Section 4 we give a lower bound for hyperplane Tverberg depth, in Section 5 a slightly stronger bound for regression depth, and in Section 6 we give a lower bound for super-additive depth measures with contractible depth regions in the more general setting of weighted arrangements.

4 A first lower bound: Hyperplane Tverberg Depth

In this section we prove an analogue of Tverberg's theorem for hyperplane arrangements, resolving a conjecture by Rousseeuw and Hubert from 1999 [21]. Our proof is inspired by the proof of Tverberg's theorem by Roudneff [20], see also [2].

► **Theorem 2.** *Let r, d be positive integers and A be an arrangement of at least $(r-1)(d+1)+1$ hyperplanes in \mathbb{R}^d . Then, there exists a point q in \mathbb{R}^d and a partition of A into r parts such that q has positive regression depth with respect to each of the r parts.*

Proof. Let π be a partition of A into r parts, each of size at most $d + 1$. Note that π can have at most d parts of size $\leq d$. Define the following function $f_\pi : \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$: for each point $q \in \mathbb{R}^d$, consider the point set A_q^* . The partition π induces a partition of this point set into parts $X_1(q), \dots, X_r(q)$. Let $B(q)$ be the smallest ball centered at q which for every part intersects the convex hull, and define $f_\pi(q)$ as the radius of this ball. As the map which for a hyperplane h assigns to a point in \mathbb{R}^d the closest point on h is continuous as a function of q , the function f_π is also continuous. Further, the function goes to infinity along any ray, so it attains a minimum. Denote by $C(q)$ the set of parts whose convex hulls $B(q)$ is tangent to. By general position, we may assume that $|C(q)| \leq d + 1$.

Let now π be a partition which minimizes $\min f_\pi$ and let p be a point where f_π attains its minimum. If $f_\pi(p) = 0$, then by Observation 6, p is the desired point. So, assume that $f_\pi(p) > 0$. For each $X_j(q)$ let $y_j(q)$ denote the unique point in $\text{conv}X_j(q)$ that minimizes the distance to q , i.e., $d(q, \text{conv}X_j) = \|q - y_j(q)\|$, and define $Y_j(q) \subset X_j(q)$ as the unique subset for which $y_j(q)$ lies in the relative interior of $\text{conv}(Y_j(q))$. In particular we can write $f_\pi(q) = \frac{1}{|C(q)|} \sum_{X_j \in C(q)} \|q - y_j(q)\|$, and its gradient as $\nabla f_\pi(q) = \frac{1}{|C(q)|} \sum_{X_j \in C(q)} (q - y_j(q))$. As f_π is minimized at p , we have $\nabla f_\pi(p) = 0$.

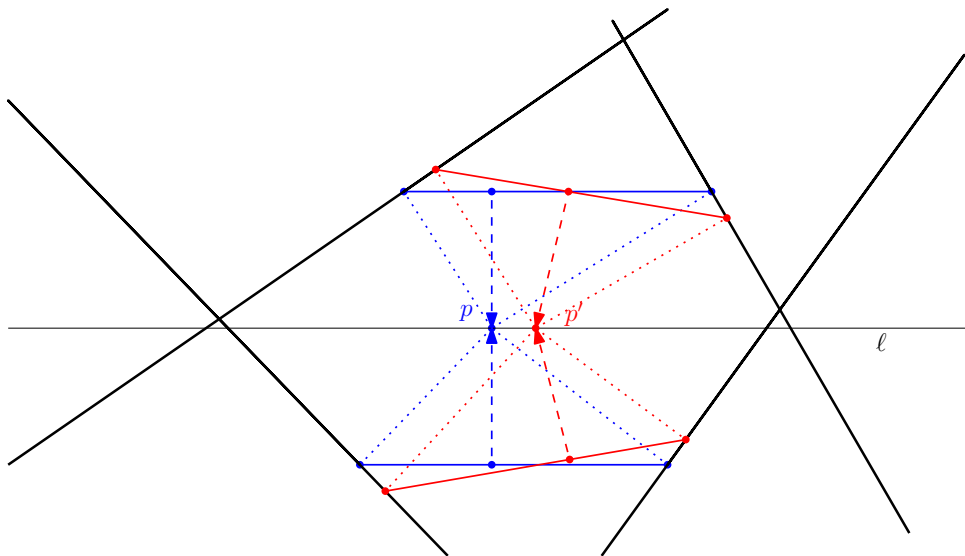
We claim that $C(p)$ consists of exactly $d + 1$ parts and that no d of the corresponding vectors $(p - y_j(p))$ lie in a common hyperplane with p . Assume for the sake of contradiction that the latter is not the case, that is, that there is a hyperplane h containing all except possibly one of the vectors $(p - y_j(p))$. Let ℓ be a line through p that is orthogonal to h . Note that all except possibly one of the affine hulls $\text{aff}Y_j(p)$ for $X_j(p) \in C(p)$ are parallel to ℓ . If there is a single vector not in h , then this vector induces a direction on ℓ . Move p a distance ε in the opposite direction. If all vectors are in h , then move p along ℓ in any direction. Call the resulting point p' . We can choose ε small enough that $C(p') = C(p)$. Let h' be the hyperplane through p' that is parallel to h and let h^+ be its side containing p . Consider now the point $y_j(p')$ for some $X_j(p) \in C(p)$. This point is in the relative interior of the points in $Y_j(p')$. Let $a(p') \in Y_j(p')$ and let $a(p)$ be the corresponding point in $Y_j(p)$. If $a(p)$ is on the same side of h as p' , then $d(a(p'), p') < d(a(p), p)$ and if $a(p)$ is on the other side then $d(a(p'), p') > d(a(p), p)$, see Figure 2. In particular, The affine subspace $Y_j(p')$ is not parallel to ℓ and the vector $(p' - y_j(p'))$ points into h^+ . As this holds for any $X_j \in C(p)$, then also the gradient $\nabla f_\pi(p') = \frac{1}{|C(p)|} \sum_{X_j \in C(p)} (p' - y_j(p'))$ points into h^+ , and as this is the side that contains p , this means that p cannot be a local minimum, which is a contradiction to the choice of p . It follows that any d of the vectors $(p - y_j(p))$ are linearly independent, and thus we need at least $d + 1$ of them to have $\nabla f_\pi(p) = 0$.

Thus, the ball $B(p)$ is tangent to exactly $d + 1$ convex hulls, and the $d + 1$ tangent hyperplanes form a simplex containing p in its interior. As there are at most d parts of size $\leq d$, there must be a point v in some X_j such that $B(p)$ still intersects the convex hull of $X_j \setminus \{v\}$. This point must lie on the same side as p of some other tangent hyperplane, say of X_i . Then adding v to X_i gives a new partition π' in which $B(p)$ intersects the interior of the convex hull of X_i . In particular, due to the arguments above, p is not a minimum of $f_{\pi'}$, and thus $\min f_{\pi'} < \min f_\pi$. This is a contradiction to the choice of π , showing that $\min f_\pi = 0$. ◀

From Theorem 2, for any super-additive depth measure the existence of a point with depth at least $\frac{|A|}{d+1}$ follows using Theorem 12.

The existence of a Tverberg theorem for regression depth naturally leads us to ask about a colorful version of such a result.

► **Conjecture 16.** *Let r, d be positive integers and A_1, \dots, A_{d+1} be sets of r hyperplanes each*



■ **Figure 2** Moving p to p' the affine hulls of Y_j are not parallel to ℓ anymore.

in \mathbb{R}^d . There is a partition of their union into r sets B_1, \dots, B_r such that $|A_i \cap B_j| = 1$ for every $i \in [d + 1], j \in [r]$ and a point q such that q has positive regression depth for each B_j .

In the plane, Karasev conjectured, provided the hyperplanes are in general position, such a partition could be found so that q was in the simplex determined by each B_j , since his Tverberg-type results for hyperplanes hold in that setting [11]. However, his conjecture and its natural extensions to \mathbb{R}^d have been disproved [4, 14]. Yet, those counterexamples do not disprove the regression depth version, in which the q can be in the simplex determined by each B_j or the union of the hyperplanes making B_j .

5 A second lower bound: topological Helly theorem

In this section, we give a proof for the centerpoint theorem for regression depth based on one of the first topological versions of Helly’s theorem, which states that given a finite family F of subsets of \mathbb{R}^d with the property that for any $d + 1$ or fewer of them their intersection is non-empty and contractible, there is a point in the intersection of all families [9]. In fact, this method proves a stronger statement: we will show that for an arrangement A in general position, there is always as point in a cell of A of regression depth $\lceil \frac{|A|-d}{d+1} \rceil$. As we will see, this implies that there is always a point of regression depth $\lfloor \frac{|A|}{d+1} \rfloor + 1$.

The basic idea is the following: given an arrangement A of hyperplanes, consider some direction ℓ , and for every point q in \mathbb{R}^d compute how many hyperplanes of A the open ray with direction ℓ emanating from q intersects. Denote this number by $\ell(q)$. Define $R_A(k, \ell)$ as the set of points where $\ell(q) \geq k$. As A is finite, there are only finitely many different such regions. If we can show that for $k = \lceil \frac{|A|-d}{d+1} \rceil$ the intersection of any $d + 1$ or fewer such regions is contractible, then the existence of a deep point as claimed above follows from the topological Helly theorem. In fact, our arguments will show that any non-empty depth region is contractible.

There is however one technical issue: the depth regions of regression depth are in general not contractible. Consider three lines in the plane that form a triangle. The regression depth is 1 on any line or in the interior of the triangle, but it is 2 on the three corners, where two

of the lines intersect. So, the region of depth 2 consists of three isolated points and is thus not contractible.

If we however look only at the 2-dimensional cells of a planar line arrangement, then it is easy to show that the closure of the union of cells of depth at least k is contractible: no cell can be completely surrounded by cells of larger depth, as any ray witnessing depth k , that is, intersecting exactly k lines, also witnesses that the other cells it intersects all have depth smaller than k .

To overcome this issue, we define a new measure, which we call *open regression depth*, denoted by RD' as follows: let A be an arrangement of hyperplanes in \mathbb{R}^d . We first slightly perturb A to get an arrangement A' in general position. In particular, in any k -dimensional affine subspace at most $d - k$ of the hyperplanes intersect. Reversing this perturbation induces a surjective map π of the faces of A' to the faces of A . For any face F of A , we call $\pi^{-1}(F)$ the faces *perturbed* from F . Note that if A was already in general position, then π is a bijection.

Consider now the perturbed arrangement A' . For any point $q \in \mathbb{R}^d$, define the open regression depth with respect to the perturbed arrangement as the minimum number of hyperplanes of A that any ray emanating from q crosses or is parallel to, where a ray crosses a hyperplane if there is a point in the relative interior of the ray that is also on the hyperplane. In other words, the open regression depth for perturbed arrangement is just the regression depth, where we do not count the hyperplanes that q lies on. The depth regions of open regression depth in a perturbed arrangement are the unions of cells with large enough depth, with lower-dimensional faces added whenever they are incident to only deep enough cells.

In order to extend the definition to the original arrangement, we define the open regression depth of a query point q in some face F_q of the arrangement A as $\text{RD}'(A, q) := \max_{q' \in F \in F_q} \{\text{RD}'(A', q')\}$, that is, as the maximum open regression depth of any point in one of the faces perturbed from F_q . Note that we can perturb the arrangement in a deterministic way, ensuring that the open regression depth is well defined. The following lemma follows immediately from the definition:

► **Lemma 17.** *For any arrangement of hyperplanes A and any query point q , we have $\text{RD}'(A, q) \leq \text{RD}(A, q)$.*

In particular, proving the existence of deep points for open regression depth implies the existence of deep points for regression depth. Note, however, that open regression depth is not super-additive: it does not satisfy condition (iii). We will now prove the existence of deep points for open regression depth using the approach sketched above. We show that we have the necessary ingredients to apply the topological Helly theorem, starting with the contractability of the relevant regions. Recall that we defined the regions $R_A(k, \ell)$ as the set of points where $\ell(q) \geq k$ for a hyperplane arrangement A and a direction ℓ , where we considered the relevant ray to be open, that is, not containing q . Also recall that as A is finite, it is sufficient to restrict our attention to finitely many directions, and we may assume that these directions are d -wise linearly independent, that is, any d of them span a d -dimensional cone.

Our proof of contractability requires some algebraic topology and is independent of the rest of the manuscript, so we defer its proof to the full version [24]. For our purposes, a *homology cell* is the same as a contractible space.

► **Lemma 18.** *Let A_1, \dots, A_m be open subsets of \mathbb{R}^d . Assume that each set is a homology cell and that the union of any d of them is a homology cell. Then $\bigcap A := \bigcap_{i=1}^m A_i$ is either empty or a homology cell.*

We can apply this result to our setting.

► **Lemma 19.** *Let ℓ_1, \dots, ℓ_m be directions in d -wise general position in \mathbb{R}^d , let A be a hyperplane arrangement and let $R(k) := \bigcap_{i=1}^m R_A(k, \ell_i)$. If $R(k) \neq \emptyset$ then $R(k)$ is contractible.*

In particular, the depth regions, that is, the intersections of $R_A(k, \ell)$ over all considered directions ℓ is contractible.

Proof. As in our setting homology cells are contractible, by Lemma 18 it suffices to show that the union of any d regions $R_A(k, \ell_1), \dots, R_A(k, \ell_d)$ is contractible. Denote this union by $U(k)$ and let $n = |A|$. Let $C \subsetneq \mathbb{R}^d$ be the cone spanned by the d directions and let ℓ_0 be a direction in $-C$. In particular, moving from any point in $U(k)$ in direction ℓ_0 we never leave $U(k)$. Thus, $U(k)$ is contractible. ◀

The final property that we need in order to apply the topological Helly theorem is that the intersection of any $d + 1$ or fewer regions is non-empty.

► **Lemma 20.** *Let ℓ_1, \dots, ℓ_m be directions in \mathbb{R}^d , $m \leq d+1$, let A be a hyperplane arrangement and let $R := \bigcap_{i=1}^m R_A(k, \ell_i)$ for $k \leq \lceil \frac{|A|-d}{d+1} \rceil$. Then $R \neq \emptyset$.*

Proof. For every direction ℓ_i let h_i be a hyperplane orthogonal to ℓ_i which bounds a half-space h_i^+ that contains $R_A(k, \ell_i)$. In particular, for any point p in h_i^+ , moving p in direction ℓ_i , we eventually enter $R_A(k, \ell_i)$ and never leave it again. Thus, if all these half-spaces have a common intersection, then this intersection can be translated to lie in R , showing $R \neq \emptyset$. So, assume that these half-spaces have an empty intersection. As we assumed that any d of our directions are linearly independent, this can only happen for $m = d + 1$. In this case, we find a point $q \in \mathbb{R}^d$ such that the $d + 1$ (closed) rays emanating from q with directions $-\ell_i$ all intersect strictly more than $|A| - k$ hyperplanes of A . Each hyperplane that does not contain q can intersect at most d of the rays, and by the general position assumption, at most d hyperplanes contain q . Thus, if x denotes the number of intersections between rays and hyperplanes, we have

$$(|A| - k)(d + 1) < x \leq (|A| - d)d + d(d + 1) = (|A| + 1)d.$$

Rearranging this and using that all numbers are integers gives $k > \lceil \frac{|A|-d}{d+1} \rceil$, which is a contradiction to the assumption, showing that $R \neq \emptyset$. ◀

Now we have all the ingredients that are necessary for the topological Helly theorem, and we deduce the following

► **Corollary 21.** *For every hyperplane arrangement A in \mathbb{R}^d there is a point $q \in \mathbb{R}^d$ for which the open regression depth is $RD^+(A, q) \geq \lceil \frac{|A|-d}{d+1} \rceil$.*

In particular, by the definition of open regression depth, if the arrangement A is in general position, such a point can be chosen in a cell of A . It remains to show that we can get even deeper points for regression depth.

► **Lemma 22.** *For every hyperplane arrangement A in \mathbb{R}^d there is a point $q \in \mathbb{R}^d$ for which the regression depth is $RD(A, q) \geq \lfloor \frac{|A|}{d+1} \rfloor + 1$.*

Proof. Consider a point q in a cell C of maximum open regression depth k , and let ∂C be the boundary of the cell C . If there is a point on ∂C with open regression depth k , then this point has regression depth $k + 1$, and the claim follows. So assume that the open regression depth is strictly smaller everywhere on ∂C . Then we again find $d + 1$ directions such that

the rays emanating from q with these directions intersect exactly k hyperplanes. Looking at the opposite directions, the rays thus intersect exactly $|A| - k$ hyperplanes, and as q lies in the interior of a cell every hyperplane intersects at most d rays. Analogous to the proof of Lemma 20 we thus get $k > \lceil \frac{|A|}{d+1} \rceil$. This proves the claim for all cases where $d+1$ does not divide $|A|$. If $d+1$ divides $|A|$, note that as soon as one of the hyperplanes only intersects $d-1$ of the considered rays, then we get $k > \lceil \frac{|A|+1}{d+1} \rceil$, and the claim follows again. So, assume that each hyperplane intersects exactly d rays. This gives a partition of the set of hyperplanes into $d+1$ parts, each of size $\frac{|A|}{d+1}$ defined by the ray they do not intersect. The boundary ∂C inherits this partition, and each of the parts is contractible. In particular, ∂C contains a vertex q that is the intersection of d hyperplanes of d different parts. Now every ray emanating from q must intersect all hyperplanes of some part, but also lies on at least $d-1$ other hyperplanes, showing that the regression depth of q is at least $k+d-1$, which is a contradiction to the assumption that the open regression depth is strictly smaller everywhere on ∂C . ◀

Using the above insights, we can also conclude the contractability of many regions of regression depth.

► **Lemma 23.** *Let $k \leq \lceil \frac{|A|}{d+1} \rceil$. Then the region R of points p whose regression depth is $RD(A, q) \geq k$ is contractible.*

Proof. If there is a point with open regression depth k , then R is just the closure of the region of points with open regression depth at least k , which is contractible by Lemma 19. Otherwise, by the proof of Lemma 22, R is the union of faces incident only to cells of maximum open regression depth. As no cell is completely surrounded by deeper faces there is a contraction from a cell of maximum open regression depth to the deeper faces incident to it. Thus, as the region of maximum open regression depth is contractible, so is R . ◀

6 A third lower bound: weighted arrangements

In this section we give yet another proof for the existence of points with large regression depth. The proof we give here works for (and actually requires) the more general case of *weighted arrangements* of hyperplanes. A weighted arrangement of hyperplanes is a tuple (A, w) consisting of a finite arrangement A of hyperplanes and a weight function $w : A \rightarrow \mathbb{R}_{\geq 0}$ which assigns to each hyperplane a weight. By a slight abuse of notation we will often just write A for a weighted arrangement. For a subarrangement $A' \subseteq A$ we have $w'(h) \leq w(h)$, where w' is the weight function on A' , and we write $w'(A') := \sum_{h \in A'} w'(h)$. We say that $A' \subsetneq A$ is a strict subset of A if the underlying hyperplane arrangement of A' is a strict subset of that of A . The definition of regression depth extends to weighted arrangements: for any ray r emanating from a query point q , let $A(r)$ be the hyperplanes intersected by r . Then, the regression depth $RD(A, q)$ of q is the minimum of $w(A(r))$ taken over all rays emanating from q . This definition is similar to, but more restrictive than a measure-theoretic generalization of regression depth considered by Mizera [15].

Our proof also works for more general families of depth measures on weighted hyperplane arrangements. We extend the definition of super-additive depth measures above to weighted hyperplane arrangements as follows:

- (i) for all $A \in A^{\mathbb{R}^d}$ and $q \in \mathbb{R}^d$ and any hyperplane h we have $|\rho(A, q) - \rho(A \cup \{h\}, q)| \leq w(h)$,
- (ii) for all $A \in A^{\mathbb{R}^d}$ we have $\rho(A, q) = 0$ if q is in an unbounded cell of A ,

- (iii) for all $A \in \mathbb{R}^d$ we have $\rho(A, q) \geq \min\{w(h) \mid h \in A\}$ if q is in a bounded cell or if q lies on a hyperplane of A ,
- (iv) for any disjoint subsets $A_1, A_2 \subseteq A$ and $q \in \mathbb{R}^d$ we have $\rho(A, q) \geq \rho(A_1, q) + \rho(A_2, q)$.

Note that any hyperplane arrangement can be considered as a weighted hyperplane arrangement by assigning weight 1 to each hyperplane. On the other hand, each depth measure for hyperplane arrangement can be extended to a depth measure on weighted hyperplanes: using the fact that \mathbb{Q} is dense in \mathbb{R} , we can place multiple hyperplanes in the same position and the normalize to get a weighted arrangement.

For a weighted arrangement of hyperplanes A and a depth measure ρ denote by $R_\rho^A(\alpha) := \{q \in \mathbb{R}^d \mid \rho(A, q) \geq \alpha\}$ the α -depth region. The median region, which is the deepest non-empty depth region, is denoted by M_ρ^A .

► **Theorem 24.** *Let A be a weighted arrangement of hyperplanes in \mathbb{R}^d and let ρ be a super-additive depth measure on weighted hyperplanes whose depth regions are compact and contractible. Then there exists a point $q \in \mathbb{R}^d$ for which $\rho(A, q) \geq \frac{w(A)}{d+1}$.*

Before we prove Theorem 24, we give some lemmata that we will need in the proof. The first lemma concerns a generalization of a section in a vector bundle. Let $\pi : E \rightarrow B$ be a real vector bundle over a compact manifold B . Following [29] we say that $\phi : B \rightarrow E$ is a *multisection* if for every $x \in B$ we have that $\phi(x) \subseteq F_x := \pi^{-1}(x)$. We further say that ϕ is *contractible* if it is contractible in each fiber, that is, for every $x \in B$ the set $\phi(x)$ is contractible. Finally, we say that ϕ is compact if $\Gamma(\phi) := \{(x, v) \mid v \in \phi(x)\} \subseteq B \times E$ is compact. For any multisection ϕ , denote by $Z(\phi)$ its intersection with the zero section. In the full version [24] we show the following

► **Lemma 25.** *Let $\pi : E \rightarrow B$ be a real vector bundle over a compact manifold B . Let ϕ be a compact contractible multisection. Then there is a section s with $Z(s) = Z(\phi)$. In particular, if π has no nowhere zero section, then ϕ must intersect the zero section.*

The second lemma is about partitions of hyperplane arrangements.

► **Lemma 26.** *Let ρ be a depth measure for weighted hyperplanes whose depth regions are compact and contractible and let A be a weighted hyperplane arrangement in \mathbb{R}^d with $|A| \geq d + 2$. Then there exists a partition of A into strict subarrangements A_1 and A_2 whose median regions intersect.*

The proof is analogous to the proof of Lemma 9 in [23], replacing Proposition 1 from [29] with our Lemma 25. For a full proof we refer to the full version [24].

Proof of Theorem 24. Let A be a weighted arrangement of hyperplanes in \mathbb{R}^d . We prove the statement by induction on the number of hyperplanes in A . If A consists of at most $d + 1$ hyperplanes, it follows from condition (iii) that $\rho(A, q) \geq \frac{w(A)}{d+1}$ for some $q \in \mathbb{R}^d$: just take q as any point on a hyperplane of maximum weight. So assume that A consists of at least $d + 2$ hyperplanes. By assumption the depth regions are compact and contractible. Thus, by Lemma 26, we can partition A into strict subarrangements A_1 and A_2 whose median regions intersect. As both A_1 and A_2 are strict subarrangements, by the induction hypothesis for any point q in the intersection of their median regions we have $\rho(A_1, q) \geq \frac{w_1(A_1)}{d+1}$ and $\rho(A_2, q) \geq \frac{w_2(A_2)}{d+1}$. As ρ satisfies condition (iv), for any such point we thus have

$$\rho(A, q) \geq \rho(A_1, q) + \rho(A_2, q) \geq \frac{w_1(A_1) + w_2(A_2)}{d + 1} = \frac{w(A)}{d + 1}.$$



At this point, it is not clear how we can use Theorem 24 to prove the existence of centerpoints for regression depth. If we look at the depth regions of regression depth, we have seen in Section 5 that they are in general not contractible. To overcome this issue, we have introduced open regression depth and argued that the depth regions of open regression depth are contractible, and these arguments go through even if the arrangement is weighted. However, for a hyperplane arrangement in general position, these regions are by definition open, and thus not compact. Further, open regression depth is not a super-additive depth measure, as it does not satisfy condition (iii). In particular, if A consists of a single hyperplane, then the open regression depth is 0 everywhere, and so the base case of the proof of Theorem 24 fails. However, as we have seen in Lemma 23, if $k \leq \lceil \frac{|A|}{d+1} \rceil$ the region of regression depth at least k is contractible. Again, the involved arguments go through if the arrangement is weighted, implying that if $k \leq \frac{w(A)}{d+1}$, then the region of regression depth at least k is contractible. Thus, defining a new measure *truncated regression depth* by

$$\text{TRD}(A, q) := \min \left(\frac{w(A)}{d+1}, \text{RD}(A, q) \right),$$

we get a measure whose depth regions are closed and contractible. Clearly, the only unbounded regions are the ones containing an unbounded face of the arrangement, and we can make those compact by intersecting with a sufficiently large ball. Finally, as regression depth is super-additive, so is truncated regression depth, and by definition, truncated regression depth is bounded from above by regression depth. We thus have the following:

► **Lemma 27.** *Truncated regression depth is a super-additive depth measure for hyperplane arrangements which has compact and contractible depth regions. Further, for every arrangement A and every point q we have $\text{TRD}(A, q) \leq \text{RD}(A, q)$.*

It now follows from Theorem 24 that there is always a point of truncated regression depth $\text{TRD}(A, q) \geq \frac{w(A)}{d+1}$ and such a point also has regression depth $\text{RD}(A, q) \geq \frac{w(A)}{d+1}$.

7 A regression depth version of the center transversal theorem

Let A be an arrangement of hyperplanes in \mathbb{R}^d . Assume that the origin is not contained in any hyperplane in A . Let L be a k -dimensional linear subspace of \mathbb{R}^d . Then $A \cap L$ is a hyperplane arrangement in L . In particular, we can again study the depth of points $q \in L$ within the Euclidean space L with respect to the arrangement $A \cap L$. Note however that $A \cap L$ might have smaller cardinality than A , as some hyperplanes of A might be parallel to L . In fact, if all of them are parallel to L , then $A \cap L$ is empty. We define the regression depth of $q \in L$ with respect to $A \cap L$ as the minimum number of hyperplanes in A intersected by or parallel to any ray in L emanating from q , and denote it by $\text{RD}(A, q, L)$. In particular, if all hyperplanes in A are parallel to L , then $\text{RD}(A, q, L) = |A|$ for all $q \in L$. This definition extends to open regression depth and truncated regression depth, where we truncate at $\frac{|A_i|}{k+1}$.

► **Theorem 28.** *Let $1 \leq k \leq d$ be integers and A_1, \dots, A_{d-k+1} be $d-k+1$ finite arrangements of hyperplanes in \mathbb{R}^d . Then there exists a k -dimensional linear subspace L and a point $q \in L$ such that q has regression depth $\text{RD}(A_i, q, L) \geq \frac{|A_i|}{k+1}$ in L for every $i \in \{1, \dots, d-k+1\}$.*

Proof. We will prove the statement for truncated regression depth, which will imply the theorem as regression depth is bounded from below by truncated regression depth. Consider the Grassmann manifold $Gr_k(\mathbb{R}^d)$ of all k -dimensional subspaces of \mathbb{R}^d . Let γ_k^d be the canonical bundle over $Gr_k(\mathbb{R}^d)$, which has total space $E := \{(L, v) \mid v \in L\}$ and whose

projection $\pi : E \rightarrow Gr_k(\mathbb{R}^d)$ is given by $\pi((L, v)) = L$. For an arrangement A_i , let $R_i(L)$ be the set of points in L that have large depth, that is, $R_i(L) := \{v \in L \mid \text{TRD}(A_i, v, L) \geq \frac{|A_i|}{k+1}\}$. By Lemma 27, each $R_i(L)$ is compact and contractible. Further, when a hyperplane $h \in A_i$ becomes parallel to L , the depth of any point can only increase, thus $R_i(L') \subseteq R_i(L)$ for any L' in a small neighborhood of L . Thus, R_i is a compact contractible multisection. Define the negative multisection $-R_i$ by reflecting $R_i(L)$ at the origin for each L , and for each $i \in \{1, \dots, d-k\}$ consider $Q_i := R_{d-k+1} - R_i$, defined by taking the Minkowski sum of $R_{d-k+1}(L)$ and $-R_i(L)$ on each L . As Minkowski sums of compact and contractible sets are again compact and contractible, Q_i is again a compact contractible multisection. In particular, by Lemma 25, there are sections s_i whose zeroes coincide with the zeroes of Q_i . It was shown in [29], Prop. 2 (see also [5], Lem. 1), that any $d-k$ sections on γ_k^d must have a common zero, that is, there is a subspace L such that $s_1(L) = \dots = s_{d-k}(L) = 0$. By the definition of the sections s_i , this implies that there is a point $q \in L$ such that $q \in R_i(L)$ for all $i \in \{1, \dots, d-k+1\}$. In particular, $\text{TRD}(A_i, q, L) \geq \frac{|A_i|}{k+1}$ in L for every $i \in \{1, \dots, d-k+1\}$. ◀

Since there is a regression depth version of the center transversal theorem and of Tverberg's theorem, a natural question is if there is a generalization of both. This is still open in the case of finite families of points, since it was conjectured by Tverberg and Vrećica in 1993 [27].

► **Conjecture 29.** *Let $1 \leq k \leq d$ be integers and A_1, \dots, A_{d-k+1} be $d-k+1$ finite arrangements of hyperplanes in \mathbb{R}^d . Assume that $|A_i| = (k+1)(r_i-1) + 1$ for some positive integer r_i , for each $i = 1, \dots, d-k+1$. Then, there exists a k -dimensional subspace L , a point $q \in L$, and a partition of each A_i into r_i parts $A_i^{(1)}, \dots, A_i^{(r_i)}$ such that $\text{RD}(A_i^{(j)}, q, L) \geq 1$ for each $i = 1, \dots, d-k+1, j = 1, \dots, r_i$.*

The classic conjecture for families of points, which has similar parameters, has only been confirmed when all r_i are powers of the same prime p and pk is even [10].

References

- 1 Nina Amenta, Marshall Bern, David Eppstein, and S H Teng. Regression depth and center points. *Discrete & Computational Geometry*, 23(3):305–323, 2000.
- 2 Imre Bárány and Pablo Soberón. Tverberg's theorem is 50 years old: a survey. *Bulletin of the American Mathematical Society*, 55(4):459–492, 2018.
- 3 Sergey Bereg, Ferran Hurtado, Mikio Kano, Matias Korman, Dolores Lara, Carlos Seara, Rodrigo I Silveira, Jorge Urrutia, and Kevin Verbeek. Balanced partitions of 3-colored geometric sets in the plane. *Discrete Applied Mathematics*, 181:21–32, January 2015. doi:10.1016/j.dam.2014.10.015.
- 4 J. P. Carvalho and P. Soberón. Counterexamples to the colorful Tverberg conjecture for hyperplanes. *Acta Math. Hungar.*, 167(2):385–392, 2022. doi:10.1007/s10474-022-01249-8.
- 5 VL Dol'nikov. Transversals of families of sets in and a connection between the Helly and Borsuk theorems. *Russian Academy of Sciences. Sbornik Mathematics*, 79(1):93, 1994.
- 6 Ruy Fabila-Monroy and Clemens Huemer. Carathéodory's theorem in depth. *Discrete Comput. Geom.*, 58(1):51–66, 2017. doi:10.1007/s00454-017-9893-8.
- 7 Radoslav Fulek, Andreas F. Holmsen, and János Pach. Intersecting Convex Sets by Rays. *Discrete & Computational Geometry*, 42(3):343–358, 2009. doi:10.1007/s00454-009-9163-5.
- 8 Sarel Har-Peled and Timothy Zhou. Improved Approximation Algorithms for Tverberg Partitions. *arXiv preprint arXiv:2007.08717*, 2020.
- 9 Eduard Helly. Über Systeme abgeschlossener Mengen mit gemeinschaftlichen Punkten. *Monatshfte d. Mathematik*, 37:281–302, 1930.

- 10 Roman N. Karasev. Tverberg's Transversal Conjecture and Analogues of Nonembeddability Theorems for Transversals. *Discrete & Computational Geometry*, 38(3):513–525, December 2007. doi:10.1007/s00454-007-1355-2.
- 11 Roman N. Karasev. Dual theorems on central points and their generalizations. *Sbornik: Mathematics*, 199(10):1459–1479, 2008. doi:10.1070/sm2008v199n10abeh003968.
- 12 Roman N. Karasev. Tverberg-Type Theorems for Intersecting by Rays. *Discrete & Computational Geometry*, 45(2):340–347, 2011. doi:10.1007/s00454-010-9294-8.
- 13 Roman N. Karasev and Benjamin Matschke. Projective Center Point and Tverberg Theorems. *Discrete & Computational Geometry*, 52(1):88–101, 2014. doi:10.1007/s00454-014-9602-9.
- 14 Seunghun Lee and Kangmin Yoo. On a conjecture of Karasev. *Comput. Geom.*, 75:1–10, 2018. doi:10.1016/j.comgeo.2018.06.003.
- 15 Ivan Mizera. On depth and deep points: a calculus. *The Annals of Statistics*, 30(6):1681–1736, 2002.
- 16 János Pach. A Tverberg-type result on multicolored simplices. *Computational Geometry*, 10(2):71–76, 1998. doi:10.1016/s0925-7721(97)00022-9.
- 17 Richard Rado. A Theorem on General Measure. *Journal of the London Mathematical Society*, s1-21(4):291–300, 1946. doi:10.1112/jlms/s1-21.4.291.
- 18 Edgardo Roldán-Pensado and Pablo Soberón. A survey of mass partitions. *Bull. Amer. Math. Soc. (N.S.)*, 59(2):227–267, 2022. doi:10.1090/bull1/1725.
- 19 David Rolnick and Pablo Soberón. Algorithms for Tverberg's theorem via centerpoint theorems. *arXiv preprint arXiv:1601.03083*, 2016.
- 20 Jean-Pierre Roudneff. Partitions of points into simplices with k -dimensional intersection. part I: The conic Tverberg's theorem. *European Journal of Combinatorics*, 22(5):733–743, 2001.
- 21 Peter J. Rousseeuw and Mia Hubert. Depth in an arrangement of hyperplanes. *Discrete & Computational Geometry*, 22(2):167–176, 1999.
- 22 Peter J. Rousseeuw and Mia Hubert. Regression depth. *J. Amer. Statist. Assoc.*, 94(446):388–433, 1999. With discussion and a reply by the authors and Stefan Van Aelst. doi:10.2307/2670155.
- 23 Patrick Schnider. Enclosing depth and other depth measures. *arXiv preprint arXiv:2103.08421*, 2021.
- 24 Patrick Schnider and Pablo Soberón. Combinatorial depth measures for hyperplane arrangements, 2023. doi:10.48550/ARXIV.2302.07768.
- 25 John W. Tukey. Mathematics and the picturing of data. In *Proc. International Congress of Mathematicians*, pages 523–531, 1975.
- 26 Helge Tverberg. A generalization of Radon's theorem. *J. London Math. Soc.*, 41(1):123–128, 1966.
- 27 Helge Tverberg and Siniša T. Vrećica. On Generalizations of Radon's Theorem and the Ham Sandwich Theorem. *European Journal of Combinatorics*, 14(3):259–264, 1993. doi:10.1006/eujc.1993.1029.
- 28 Marc van Kreveld, Joseph S. B. Mitchell, Peter Rousseeuw, Micha Sharir, Jack Snoeyink, and Bettina Speckmann. Efficient algorithms for maximum regression depth. *Discrete Comput. Geom.*, 39(4):656–677, 2008. doi:10.1007/s00454-007-9046-6.
- 29 Rade T. Živaljević and Siniša T. Vrećica. An extension of the ham sandwich theorem. *Bulletin of the London Mathematical Society*, 22(2):183–186, 1990.

FibeRed: Fiberwise Dimensionality Reduction of Topologically Complex Data with Vector Bundles

Luis Scoccola   

Department of Mathematics, Northeastern University, Boston, MA, USA

Jose A. Perea   

Department of Mathematics and Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA

Abstract

Datasets with non-trivial large scale topology can be hard to embed in low-dimensional Euclidean space with existing dimensionality reduction algorithms. We propose to model topologically complex datasets using vector bundles, in such a way that the base space accounts for the large scale topology, while the fibers account for the local geometry. This allows one to reduce the dimensionality of the fibers, while preserving the large scale topology. We formalize this point of view and, as an application, we describe a dimensionality reduction algorithm based on topological inference for vector bundles. The algorithm takes as input a dataset together with an initial representation in Euclidean space, assumed to recover part of its large scale topology, and outputs a new representation that integrates local representations obtained through local linear dimensionality reduction. We demonstrate this algorithm on examples coming from dynamical systems and chemistry. In these examples, our algorithm is able to learn topologically faithful embeddings of the data in lower target dimension than various well known metric-based dimensionality reduction algorithms.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases topological inference, dimensionality reduction, vector bundle, cocycle

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.56

Related Version *Full Version*: <https://arxiv.org/abs/2206.06513>

Supplementary Material Proof-of-concept implementation [44]

Funding *Luis Scoccola*: supported by the NSF through grants CCF-2006661 and CAREER award DMS-1943758.

Jose A. Perea: supported by the NSF through grants CCF-2006661 and CAREER award DMS-1943758.

Acknowledgements The authors thank Matt Piekenbrock for various fruitful conversations.

1 Introduction

Motivation. We take the manifold hypothesis at face value and consider data consisting of a finite sample of a Riemannian manifold. We take the goal of *dimensionality reduction* to be that of learning an embedding of the input data in low dimension, in such a way that the differentiable structure of the underlying manifold is preserved. This is different from *charting*, whose objective we take to be that of producing local parametrizations of the data that, together, cover the entire manifold.

We refer to dimensionality reduction algorithms which aim to preserve metric relationships and do not explicitly incorporate large scale topology in their objective function as *metric-based*. Metric-based algorithms work best when the Riemannian manifold underlying the data can be isometrically embedded in the target dimension. For example, algorithms such as Isomap [53], Local Tangent Space Alignment (LTSA) [59], and Hessian Eigenmaps (HLE)



© Luis Scoccola and Jose A. Perea;

licensed under Creative Commons License CC-BY 4.0

39th International Symposium on Computational Geometry (SoCG 2023).

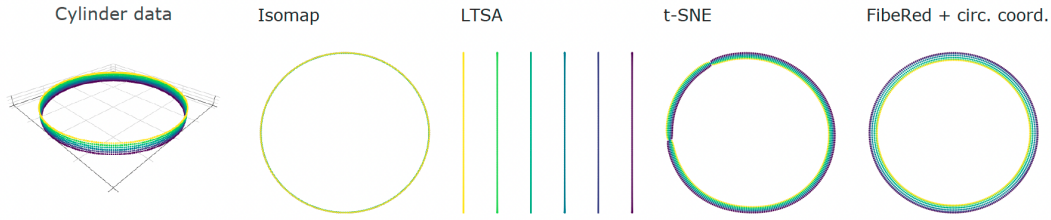
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 56; pp. 56:1–56:18

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** *Left*: A sample from a cylinder with height equal to 0.15 times its radius, colored by height. The cylinder is developable, since it is diffeomorphic to an annulus in \mathbb{R}^2 , and is also flat, but it is not isometric to the annulus, which also has a flat, yet distinct, Riemannian metric. *Center three*: Well known dimensionality reduction algorithms run on the cylinder data. The outputs are representative of other parameter choices and of Laplacian Eigenmaps (LE) [8], Diffusion Maps (DM) [13], LLE [42], HLLE, t-SNE [55], and UMAP [28]. Some algorithms only capture the circularity, others only the local 2D structure, while others capture both, but they are not able to consistently align the local 2D structure. *Right*: The output of fiberwise dimensionality reduction.

[15] assume that the manifold \mathcal{X} underlying the data is *isometrically developable*, in the sense that \mathcal{X} is a d -dimensional Riemannian manifold for which there exists an embedded d -dimensional manifold $\mathcal{X}' \subseteq \mathbb{R}^d$ and a diffeomorphism $\mathcal{X}' \rightarrow \mathcal{X}$ which is a Riemannian isometry. An isometrically developable manifold \mathcal{X} is necessarily *flat* (i.e., locally isometric to Euclidean space) and *developable* (i.e., diffeomorphic to an embedded d -dimensional manifold $\mathcal{X}' \subseteq \mathbb{R}^d$). But a manifold can be flat and developable without it being isometrically developable: a simple example is that of a straight cylinder in \mathbb{R}^3 (Figure 1). As observed in [24], and shown in Figure 1, already in the setting of a flat and developable d -dimensional manifold, metric-based dimensionality reduction algorithms can fail to find an embedding of the data in \mathbb{R}^d . On the mathematical side, while Whitney’s embedding theorem [57] guarantees that any closed d -dimensional manifold admits a smooth (C^∞) embedding in $2d$ dimensions, a smooth, Riemannian isometric embedding of a closed d -dimensional Riemannian manifold can require in the order of d^2 dimensions [11]. Thus, the preservation of distances requires more complicated embeddings than the preservation of topology.

If we remove a small portion of the cylinder of Figure 1, in order to make it a curved rectangle, most metric-based dimensionality reduction algorithms have no problem finding an embedding in \mathbb{R}^2 . It is thus the non-trivial topology of the cylinder – its circularity – that causes difficulties. This suggests that embeddings of topologically non-trivial manifolds can be built by gluing local representations along a representation of the global topological structure: in the case of the cylinder, one would try to glue 2D patches around a circle in a globally consistent manner. This leads to the following problem, formalized as the vector bundle embedding problem (Problem 1):

Given a dataset X and an initial map $X \rightarrow \mathbb{R}^D$ capturing the large scale topology of X , find a new representation $X \rightarrow \mathbb{R}^D$ that captures the large scale topology as well as the local geometry.

We call our approach to the above problem *fiberwise dimensionality reduction* (FIBERED). In the examples of Section 4, we focus on manifolds with an essential loop, and, as initial map, we use circular coordinates based on persistent cohomology [14, 37], a technique from Topological Data Analysis [32, 18]. Nevertheless, the approach is not restricted to the case of a circular initial embedding and one could use as initial map one constructed by, e.g., other cohomological coordinates [35, 38, 43], standard non-linear dimensionality reduction methods [23, 13], or lens functions as in [50, Section 4].

Contributions. We show that the theory of vector bundles is useful in abstracting (Section 2.1), devising solutions to (Section 3), and computing obstructions to solving (Section 2.3) the problem of extending an initial coarse representation of data to a new, more descriptive representation. We demonstrate with computational examples (Section 4) that topological inference for vector bundles can be carried out in practice. In particular, we show that efficient embeddings and chartings of topologically non-trivial data can be learned with this approach and give examples supporting the claim that metric-based dimensionality reduction algorithms are often not able to find such representations. We implement our main algorithm in [44].

Related work. Various dimensionality reduction schemes [52, 41, 10] learn a global alignment of local linear models from the local interactions of the models, which can be challenging in the presence of non-trivial topology. In contrast, our approach assumes a global topological representation is given and builds and aligns the local linear models along this representation.

There has been recent interest in designing topology-preserving dimensionality reduction schemes [26, 58, 31, 56]. Our approach is different from previous approaches we are aware of, as it builds a new representation around an initial topological representation, instead of using topology to regularize an essentially metric objective.

Our cut-unfold technique of [45, Appendix C.1] has a similar goal to that of [24, 58], which propose to tear a data manifold in order to find efficient representations of it. A main difference is that our technique allows the user to select a specific hole to cut and to use topological persistence to guide this choice.

2 The vector bundle embedding problem

For background, please refer to [45, Appendix A.1]. In Section 2.1 we describe the Vector Bundle Embedding problem; in Section 2.2, we recall the notion of discrete vector bundle that we use to estimate vector bundles from finite samples; and in Section 2.3 we explain how characteristic classes of vector bundles give computable obstructions to solving the vector bundle embedding problem and can thus be used for parameter selection.

2.1 Main problem

Let \mathcal{B} be a closed differentiable manifold and let $\pi : \mathcal{X} \rightarrow \mathcal{B}$ be a rank r Euclidean vector bundle with zero-section $s_0 : \mathcal{B} \rightarrow \mathcal{X}$, where by *Euclidean* we mean that π is endowed with a scalar product on each fiber $\pi^{-1}(b) \subseteq \mathcal{X}$, which varies smoothly with $b \in \mathcal{B}$. The main problem we seek to solve is that of extending an embedding $\mathcal{B} \rightarrow \mathbb{R}^D$ to a fiberwise isometric embedding of \mathcal{X} , as follows:

► **Problem 1.** *Given an embedding $\iota : \mathcal{B} \rightarrow \mathbb{R}^D$, find a fiberwise isometric embedding $\bar{\iota} : \mathcal{X} \rightarrow \mathbb{R}^D$ that extends ι in the sense that $\bar{\iota} \circ s_0 = \iota$, and that is orthogonal to \mathcal{B} , in the sense that $\bar{\iota}(\pi^{-1}(b)) \perp \iota(T_b\mathcal{B})$ for all $b \in \mathcal{B}$.*

By *fiberwise isometric embedding* $\mathcal{X} \rightarrow \mathbb{R}^D$ we mean a map that is a linear isometry when restricted to each fiber $\pi^{-1}(b) \subseteq \mathcal{X}$, where $b \in \mathcal{B}$.

Let $\nu : N \rightarrow \mathcal{B}$ be the normal bundle of the embedding $\iota : \mathcal{B} \rightarrow \mathbb{R}^D$ and endow ν with the Euclidean structure inherited from \mathbb{R}^D . The following result reduces Problem 1 to a problem only involving vector bundles.

► **Lemma 2.** *Problem 1 admits a solution if and only if there exists a morphism $\mathcal{X} \rightarrow N$ of vector bundles over \mathcal{B} that is an isometry in each fiber.*

In order to do this, we trivialize the bundles \mathcal{X} and N over a common cover of the base \mathcal{B} and construct the embedding $\mathcal{X} \rightarrow N$ by restricting to each element of the cover. Formally, we proceed as follows.

Let e be the dimension of \mathcal{B} , so that the rank of ν is $D - e$. Let $\mathcal{U} = \{U_i\}$ be a cover of \mathcal{B} such that both π and ν can be trivialized over \mathcal{U} and let $\mathcal{X}_i := \pi^{-1}(U_i)$. Recall that $\mathbb{V}(n, m)$ denotes the Stiefel manifold, which consist of m -by- n matrices with orthonormal columns and that $O(n) = \mathbb{V}(n, n)$ denotes the orthogonal group. Let $\alpha = \{\alpha_i : U_i \rightarrow \mathbb{V}(D - e, D)\}$ be local bases for N , and let $\Theta = \{\Theta_{ij} : U_i \cap U_j \rightarrow O(D - e)\}$ be defined by $\Theta_{ij}(b) = \alpha_i(b)\alpha_j(b)^T$ for all $b \in U_i \cap U_j$, so that Θ is a cocycle with associated vector bundle ν . Finally, let $\{(\pi|_{\mathcal{X}_i}, f_i) : \mathcal{X}_i \rightarrow U_i \times \mathbb{R}^r\}$ be a metric trivialization of \mathcal{X} over \mathcal{U} and let $\Omega = \{\Omega_{ij} : U_i \cap U_j \rightarrow O(r)\}$ be defined as the unique set of maps satisfying

$$\Omega_{ij}(\pi(x)) f_j(x) = f_i(x), \text{ for all } x \in \mathcal{X}_i \cap \mathcal{X}_j, \quad (1)$$

so that Ω is a cocycle with associated vector bundle π . We refer to the maps $\{f_i : \mathcal{X}_i \rightarrow \mathbb{R}^r\}$ as the *fiber coordinates*. With these definitions, one can use Lemma 2 to prove the following.

► **Proposition 3.** *There exists a fiberwise isometric embedding $\mathcal{X} \rightarrow N$ if and only if there exist maps $\Phi = \{\Phi_i : U_i \rightarrow \mathbb{V}(r, D - e)\}$ such that*

$$\Phi_i(b)\Omega_{ij}(b) = \Theta_{ij}(b)\Phi_j(b), \text{ for all } i \text{ and } j \text{ and } b \in U_i \cap U_j. \quad (2)$$

Given the maps $\Phi = \{\Phi_i : U_i \rightarrow \mathbb{V}(r, D - e)\}$ of Proposition 3, one obtains the fiberwise isometric embedding $\bar{\iota} : \mathcal{X} \rightarrow \mathbb{R}^D$ by $\bar{\iota}(x) = \alpha_i(b)\Phi_i(b)f_i(x) + \iota(b)$, where $b = \pi(x)$.

In general, the fiberwise isometric embedding $\bar{\iota} : \mathcal{X} \rightarrow \mathbb{R}^D$ is not an embedding of \mathcal{X} as a manifold, since different fibers may intersect. Nonetheless, if $\tau > 0$ is the *reach* [1, Definition 2.1] of $\iota(\mathcal{B}) \subseteq \mathbb{R}^D$, i.e., the largest possible radius of a uniform tubular neighborhood around $\iota(\mathcal{B})$, one can find an embedding of a full-dimensional compact subset of \mathcal{X} by scaling the fibers by a fraction of τ , as follows. Let $\text{disk}(\pi) \subseteq \mathcal{X}$ be the unit disk of the bundle π , namely, the subspace of points $x \in \mathcal{X}$ such that $\|x - s_0(\pi(x))\| \leq 1$, where $\|\cdot\|$ denotes the norm of the fiber $\pi^{-1}(\pi(x))$ induced by the Euclidean structure of π . Then, the following formula gives an embedding $\text{disk}(\pi) \rightarrow \mathbb{R}^D$:

$$x \mapsto c\tau \cdot \alpha_i(\pi(x))\Phi_i(\pi(x))f_i(x) + \iota(\pi(x)), \text{ for } \pi(x) \in U_i, \quad (3)$$

where $0 < c < 1$ is any fixed constant.

2.2 Vector bundles from finite samples

In practice, continuous maps to a Stiefel manifold or orthogonal group – such as the maps $\{\alpha_i : U_i \rightarrow \mathbb{V}(D - e, D)\}$ or the cocycle $\{\Omega_{ij} : U_i \cap U_j \rightarrow O(r)\}$ of Section 2.1 – are hard to work with, as they are potentially determined by an infinite amount of data. One of the main takeaways of [46] is that one can work with Euclidean vector bundles in practice by considering only constant maps into Stiefel manifolds or orthogonal groups. In order to accomplish this, one relaxes the notion of Euclidean vector bundle as follows.

Given a simplicial complex S , a rank r *discrete approximate* cocycle on S ([46, Definition 5.1]) consists of a family of matrices $\{\Omega_{ij} \in O(r)\}$ indexed by the oriented 1-simplices of S , which satisfies $\Omega_{ij} = \Omega_{ji}^T$. There is a similar way of discretizing maps into a Stiefel manifold ([46, Definition 5.4]). These discretizations can be used to represent usual vector bundles [46, Theorem A] and any vector bundle can be represented in this way [46, Proposition 5.7].

This justifies the fact that, in Section 3, we discretize the base \mathcal{B} by considering the simplicial complex given by the nerve of a cover $\mathcal{U} = \{U_i\}$ and we consider constant maps from U_i into a Stiefel manifold and from $U_i \cap U_j$ into an orthogonal group.

2.3 Computable obstructions to vector bundle embedding

The theory of vector bundles provides us with algebraic obstructions to solving Problem 1, namely, characteristic classes. We now give a few details about the subject; we refer the reader to [30] for a detailed account of the theory of characteristic classes.

To a vector bundle $\pi : \mathcal{X} \rightarrow \mathcal{B}$ and number $i \in \mathbb{N}$, one can associate an element $w_i(\pi) \in H^i(\mathcal{B}; \mathbb{Z}/2)$ of the i th cohomology group of \mathcal{B} with coefficients in the group $\mathbb{Z}/2$, called the i th *Stiefel–Whitney class* of π . This procedure is such that, if π and π' are isomorphic vector bundles over the same base \mathcal{B} , then $w_i(\pi) = w_i(\pi')$.

If Problem 1 admits a solution, then there exists a complement of π in ν , that is there exists a vector bundle κ over \mathcal{B} such that $\pi \oplus \kappa \cong \nu$, where \oplus denotes the direct sum of vector bundles. It follows from the Whitney product formula [30, Section 4, Axiom 3] that $w(\pi) \smile w(\kappa) = w(\nu)$, where \smile denotes the cup-product in cohomology [21, Section 3.2]. In particular, when Problem 1 admits a solution, we have the following:

- If $D = r + e$, then $w_1(\pi) = w_1(\nu) \in H^1(\mathcal{B}; \mathbb{Z}/2)$.
- If $D = r + e + 1$, then $w_2(\pi) - w_1(\pi)^2 + w_1(\pi) \smile w_1(\nu) = w_2(\nu) \in H^2(\mathcal{B}; \mathbb{Z}/2)$.

Thus, if any of these equalities is not satisfied, then Problem 1 does not admit a solution. These obstructions can be computed from finite samples using [46, Theorem C].

3 The fiberwise dimensionality reduction scheme

We describe the FIBERED algorithm in Sections 3.1–3.4. In Section 3.5 we justify a main subroutine of the algorithm. In Section 3.6, we explain how we choose parameters. We represent vector bundles using discrete approximate cocycles as in [46] (see Section 2.2).

To facilitate the interpretation of the different steps of the algorithm, the notation is kept as in Section 2.1, except for the spaces \mathcal{X} and \mathcal{B} , which we denote here by X and B to emphasize the fact that we are working with finite samples $X \subseteq \mathcal{X}$ and $B \subseteq \mathcal{B}$. See also Figure 2 for a schematic representation of some of the main steps (4,5,6) of the algorithm.

Precise assumptions about the input of the algorithm are in [45, Appendix B.1]. Our algorithm can be efficiently implemented; we give more details in [45, Appendix B.2].

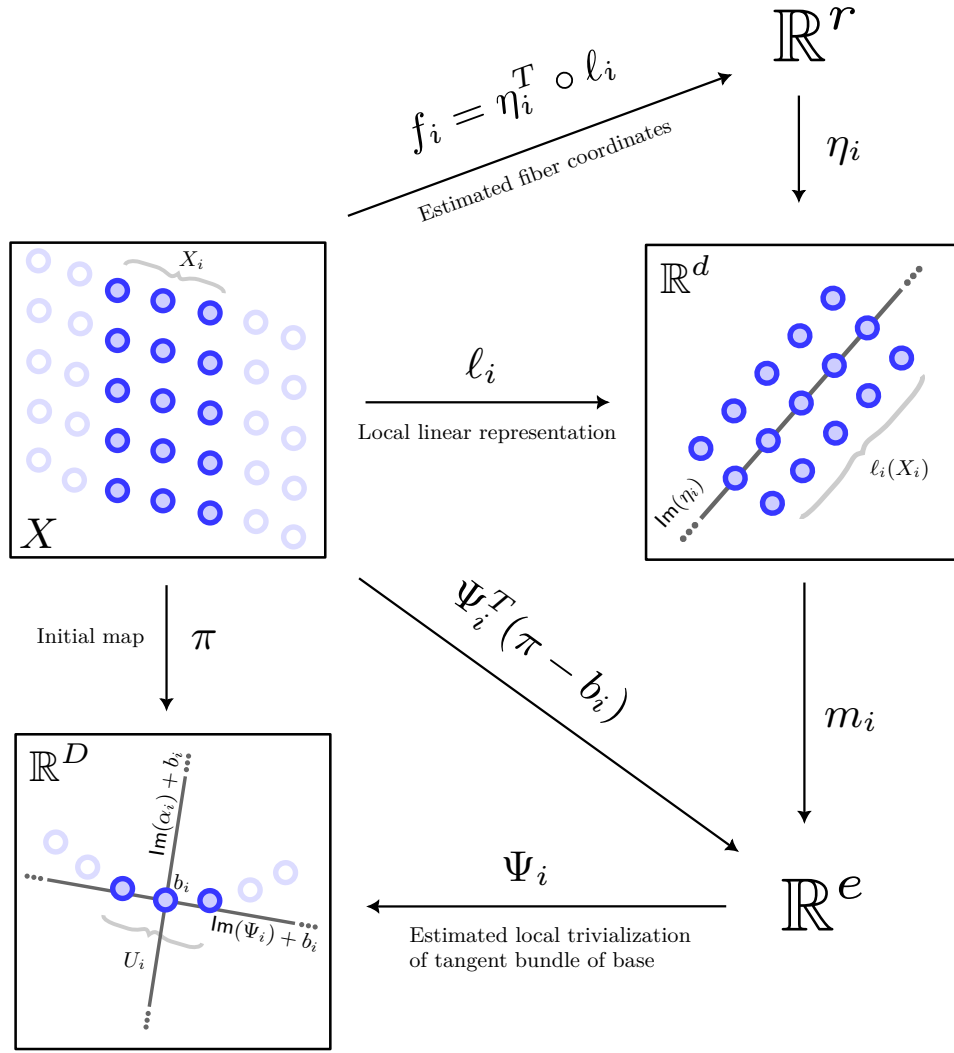
3.1 Main routine

Inputs. A dataset represented by a finite set X together with a distance matrix $\partial : X \times X \rightarrow \mathbb{R}$; and a function $\pi : X \rightarrow \mathbb{R}^D$. We let $B := \pi(X) \subseteq \mathbb{R}^D$.

Parameters. A number $k \in \mathbb{N}$, the number of sets we use to construct a cover of B ; a number $\mathbf{n_iter} \in \mathbb{N}$ used in the ALIGNFIBERS subroutine; an estimate $e \in \mathbb{N}$ of the intrinsic dimension of \mathcal{B} ; an estimate $d \in \mathbb{N}$ of the intrinsic dimension of \mathcal{X} ; a fiber scale $0 < c < 1$.

Output. A map $X \rightarrow \mathbb{R}^D$.

The pseudocode is in Algorithm 1. With this notation, the map ι of Section 2.1 corresponds to the inclusion $B = \pi(X) \subseteq \mathbb{R}^D$ and the rescaling of $\bar{\iota}$ of Equation (3) corresponds to the output of the algorithm.



■ **Figure 2** Schematic representation of the main constructions in the FIBERED pipeline.

■ **Algorithm 1** FIBERED($X, \pi, k, e, d, c, n_iter$).

-
- 1: $\mathcal{U}, \rho \leftarrow \text{COVERANDPARTITIONUNITY}(k, B)$
 - 2: $\mathcal{N} \leftarrow \text{NERVE}(B, \mathcal{U})$
 - 3: **for** $1 \leq i \leq k$ **do**
 - 4: $l_i \leftarrow \text{LOCALLINEARREPRESENTATION}(X, \mathcal{U}, d, i)$
 - 5: $\Psi_i, \alpha_i \leftarrow \text{ESTTANGANDNORMBUN}(B, \mathcal{U}, e, i)$
 - 6: $\bar{f}_i \leftarrow \text{ESTNORMFIBERCOORDINATES}(B, \Psi_i, l_i)$
 - 7: **end for**
 - 8: $\tau \leftarrow \text{ESTREACH}(B, \mathcal{U}, \Psi)$
 - 9: **for** $(ij) \in \mathcal{N}$ **do**
 - 10: $\Omega_{ij}, \Theta_{ij} \leftarrow \text{ESTCOCYCLES}(\bar{f}_i, \bar{f}_j, \alpha_i, \alpha_j)$
 - 11: **end for**
 - 12: $\Phi \leftarrow \text{ALIGNFIBERS}(\mathcal{N}, \Omega, \Theta, n_iter)$
 - 13: **return** ASSEMBLE($\rho, \tau, c, \alpha, \Phi, \bar{f}, \pi$)
-

3.2 Subroutines

Compute cover and partition of unity (COVERANDPARTITIONUNITY). We compute a cover $\mathcal{U} = \{U_i \subseteq B\}_{1 \leq i \leq k}$ of B as follows. We first run on B an approximate algorithm for the k -center problem. We use a simple, greedy approach, but more sophisticated options are available (see, e.g., [17] for a survey). This results in k points $\{b_1, \dots, b_k\} \subseteq B$ and in a radius $c > 0$ such that any point of B is at distance at most c from some b_i . We then let $U_i = \{b \in B : \|b - b_i\| < 3c\}$. The factor of 3 is arbitrary; we choose it to ensure that elements of the cover have sufficiently large intersections.

We compute a partition of unity $\rho = \{\rho_i : U_i \rightarrow \mathbb{R}\}$ subordinate to \mathcal{U} by first defining $p_i(x) = \exp(-1/(1 - (\|x - b_i\|/(3c))^2))$ for $x \in U_i$ and $p_i(x) = 0$ for $x \notin U_i$, and then normalizing as follows $\rho_i(x) = p_i(x) / \sum_j p_j(x)$.

Compute nerve of cover (NERVE). We let \mathcal{N} be the undirected graph with vertices $1 \leq i \leq k$ and an edge (ij) with weight $s_{ij} = |U_i \cap U_j|$ when $U_i \cap U_j \neq \emptyset$.

Compute local linear representation (LOCALLINEARREPRESENTATION). Given $1 \leq i \leq k$, we let $X_i := \pi^{-1}(U_i)$ and apply a linear dimensionality reduction algorithm to each X_i , resulting in a function $\ell'_i : X_i \rightarrow \mathbb{R}^d$. In our implementation, we use classical multidimensional scaling (see, e.g., [9]). We then mean-center ℓ'_i to get a function $\ell_i : X_i \rightarrow \mathbb{R}^d$.

Estimate local trivialization of tangent and normal bundle (ESTTANGANDNORMBUN). Given $1 \leq i \leq k$, we compute an orthonormal frame $\Psi_i \in \mathcal{V}(e, D)$ by applying PCA with target dimension e to $U_i \subseteq \mathbb{R}^D$. We then compute an orthonormal frame $\alpha_i \in \mathcal{V}(D - e, D)$ such that $\alpha_i \perp \Psi_i$.

Estimate normalized fiber coordinates (ESTNORMFIBERCOORDINATES). Given $1 \leq i \leq k$, we define $t : X_i \rightarrow \mathbb{R}^e$ by $t(x) = \Psi_i^T(\pi(x) - b_i)$. We find a linear transformation $m_i : \mathbb{R}^d \rightarrow \mathbb{R}^e$, which has minimal Frobenius norm and minimizes

$$\sum_{x \in X_i} \|t(x) - m_i(\ell_i(x))\|^2, \quad (4)$$

and compute an orthonormal frame $\eta_i \in \mathcal{V}(r, d)$ with image in the kernel of m_i . We let $f_i := \eta_i^T \circ \ell_i : X_i \rightarrow \mathbb{R}^r$, and obtain a normalized fiber coordinate $\bar{f}_i : X_i \rightarrow \mathbb{R}^r$ with image contained in the unit ball by normalizing f_i . We justify these choices in Section 3.5.

Estimate reach (ESTREACH). If $\{b_1, \dots, b_k\} \subseteq B$ are the centers of the k balls used to construct the cover \mathcal{U} in COVERANDPARTITIONUNITY, we compute an estimate of the reach of B by

$$\tau = \inf_{i \neq j} \frac{\|b_j - b_i\|^2}{2\sqrt{\|b_j - b_i\|^2 - \|\Psi_i^T(b_j - b_i)\|^2}}.$$

This formula is equivalent to [1, Equation 6.1], where it is proven that, under suitable assumption, it yields a consistent estimator of the reach.

Estimate cocycles for ν and π (ESTCOCYCLES). Based on Equation (1), given $(ij) \in \mathcal{N}$, we compute an orthogonal matrix $\Omega_{ij} \in O(r)$ which minimizes

$$\sum_{x \in X_i \cap X_j} \|\Omega_{ij} f_j(x) - f_i(x)\|^2.$$

We also compute an orthogonal matrix $\Theta_{ij} \in O(D - e)$ which minimizes $\|\Theta_{ij} - \alpha_i^T \alpha_j\|_F$, where $\|\cdot\|_F$ denotes the Frobenius norm. Both minimizations are instances of the orthogonal Procrustes problem, which can be solved using SVD (see, e.g., [22, Section 7.4]).

Align fibers (ALIGNFIBERS). Based on Equation (2), we compute orthonormal frames $\{\Phi_i \in \mathcal{V}(r, D - e)\}$ minimizing the following expression; we describe the minimization procedure in Section 3.3:

$$\sum_{(ij) \in \mathcal{N}} s_{ij} \|\Phi_i \Omega_{ij} - \Theta_{ij} \Phi_j\|_F. \quad (5)$$

Compute final representation (ASSEMBLE). Based on Eq. 3, we represent $x \in X$ by

$$\sum_{1 \leq i \leq k} \rho_i(x) (c\tau \cdot \alpha_i \Phi_i \bar{f}_i(x) + \pi(x)).$$

3.3 Minimizing Equation (5)

The minimization problem in ALIGNFIBERS is non-convex, so a possible solution is to do gradient descent in a product Stiefel manifold. This is the approach we take, except that we avoid explicitly computing a gradient, and take a sampling based approach, as done in, e.g., LargeVis [51]. Before describing the approach, we note that, in the case $D = r + e$, the Stiefel manifold $\mathcal{V}(r, r)$ is equal to the orthogonal group $O(r)$, which is disconnected. Thus, in this case, any local optimization approach to minimizing Equation (5), such a gradient descent, is bound to fail. In Section 3.4 we describe a procedure based on the notion of synchronization (see, e.g., [48]) that reduces the problem from having to align using matrices in $O(r)$ to using matrices in $SO(r)$, which is connected.

Iterative procedure. We start by initializing $\{\Phi_i \in \mathcal{V}(r, D - e)\}$ at random and setting $a = 1$. For $1 \leq n \leq \text{n_iter}$, we proceed as follows. We sample an edge $(ij) \in \mathcal{N}$ with probability proportional to its weight s_{ij} , let M be an orthonormal frame minimizing $\|M \Omega_{ij} - \Theta_{ij} \Phi_j\|_F$, and replace Φ_i with a closest orthonormal frame to the convex combination $(1 - a)\Phi_i + aM$. Finally, we replace a with $1 - n/\text{n_iter}$.

3.4 Preprocessing in the case $D = r + e$

In this case, the matrices Φ , Ω , and Θ are in $O(r)$. The preprocessing consists of replacing the matrices $\{\Theta_{ij}\}$ by matrices that induce an equivalent problem to the one of minimizing Equation (5), but for which the matrices $\{\Phi_i\}$ we look for can be taken to be in the special orthogonal group $SO(r)$, which is connected.

Note that, if we want $\Phi_i \Omega_{ij}$ and $\Theta_{ij} \Phi_j$ to belong to the same connected component of $O(r)$, then we must have $\det(\Omega_{ij}) \det(\Theta_{ij}) = \det(\Phi_i) \det(\Phi_j) \in O(1) = \{-1, +1\}$. This

suggests that we can let $\omega_{ij} = \det(\Omega_{ij}) \det(\Theta_{ij}) \in O(1)$ and consider first the problem of finding $\{\lambda_i \in O(1)\}$ such that $\lambda_i \lambda_j = \omega_{ij}$, which leads to minimizing the objective function

$$\sum_{(ij) \in \mathcal{N}} s_{ij} |\omega_{ij} - \lambda_i \lambda_j|^2.$$

This is a well known synchronization problem, for which an approximate solution can be found effectively and efficiently with spectral methods [49, 4]. Here, we use [4, Algorithm 2.3], with $d = 1$, which yields an approximate solution $\{\lambda_i \in O(1)\}$.

Given $\lambda \in O(1) = \{-1, +1\}$ let $M(\lambda) \in O(r)$ be the diagonal matrix with all diagonal entries equal to 1, except for the first one, which is equal to λ . With this in mind, we can replace Θ_{ij} by $M(\lambda_i)\Theta_{ij}M(\lambda_j)$. Having done this, we can now restrict the matrices $\{\Phi_i\}$ to belong to the connected component of $O(r)$ of orthogonal matrices with $+1$ as determinant. More specifically, we now can carry out the optimization procedure described above, but restricting the matrices $\{\Phi_i\}$ to be in $SO(r) \subseteq O(r) = V(r, r)$.

3.5 Justification of estimate of fiber coordinates

Let $x_i := s_0(b_i) \in \mathcal{X}$. We interpret the local model $\ell_i : X_i \rightarrow \mathbb{R}^d$ as a projection $\ell_i : X_i \rightarrow T_x \mathcal{X} \cong \mathbb{R}^d$ of X_i onto the tangent space at the origin of the fiber $\pi^{-1}(b_i)$. In the idealized case ([45, Appendix B.1]), the fiber coordinate $f_i : \mathcal{X}_i \rightarrow \mathbb{R}^r$ is given by any map fitting into a fiberwise isometric diffeomorphism $(\pi|_{\mathcal{X}_i}, f_i) : \mathcal{X}_i \rightarrow U_i \times \mathbb{R}^r$. When dealing with finite samples, we use the following composite as a proxy for f_i :

$$X_i \xrightarrow{\ell_i} T_{x_i} \mathcal{X} \xrightarrow{(df_i)_{x_i}} T_{f_i(x_i)} \mathbb{R}^r \cong \mathbb{R}^r.$$

Note that, by assumption, $(df_i)_{x_i}$ is the second component of an isometric isomorphism of Euclidean vector spaces $d(\pi, f_i)_{x_i} : T_{x_i} \mathcal{X} \rightarrow T_{b_i} \mathcal{B} \oplus \mathbb{R}^r$, in which the two direct summands are orthogonal. It is thus sufficient to estimate the first component $d\pi_{x_i} : T_{x_i} \mathcal{X} \rightarrow T_{b_i} \mathcal{B}$ and to then compose ℓ_i with the orthogonal projection onto the orthogonal complement of $d\pi_{x_i}$. We do have an estimate for the composite

$$X_i \xrightarrow{\ell_i} T_{x_i} \mathcal{X} \xrightarrow{d\pi_{x_i}} T_{b_i} \mathcal{B} \cong \mathbb{R}^e,$$

namely $t = \Psi_i^T \circ \pi|_{X_i} : X_i \rightarrow \mathbb{R}^e$, but, since the embedding $\mathcal{B} \subseteq \mathbb{R}^D$ is not required to preserve the Riemannian structure of \mathcal{B} inherited from that of \mathcal{X} , the map t is an approximation of $d\pi_{x_i} \circ \ell_i$ up to a linear map $m_i : \mathbb{R}^d \rightarrow \mathbb{R}^e$. This justifies finding m_i by minimizing Equation (4), and getting the approximate fiber coordinate f_i by composing ℓ_i with the orthogonal projection onto the kernel of m_i .

3.6 Choosing input and parameters

We discuss some guiding principles to choose parameters for our pipeline. We focus mostly on parameter selection for the examples of Section 4.

Parameters. An estimate of the dimensions e of B and d of X can be obtained by analyzing the explained variance of PCA applied to each of the sets U_i and X_i with a range of target dimensions, but more sophisticated algorithms are available; see, e.g., [25]. The parameter k is chosen to be large enough so that the cover \mathcal{U} captures the topology of the base space B , and such that each open ball of the cover is sufficiently small so that it can be approximated

reasonable well by a linear space. Admittedly, this is in general a difficult choice and producing good covers of data is an interesting problem in its own right. In our case, when the base space is the circle, we use $k = 16$; see also [45, Appendix C.5] for a parameter sensitivity analysis. The algorithm is robust to the choice of parameter `n_iter`, which we choose to be 1000 in all of our examples.

Choosing base map and D . We construct the initial map $\pi : X \rightarrow \mathbb{R}^D$ in two ways.

The first way is to use the persistent cohomology of the initial data X to construct circular coordinates $X \rightarrow S^1$ and then embed the circle S^1 as the unit circle in the plane spanned by the first two coordinates of \mathbb{R}^D , $D \geq 2$, which gives us the initial map $\mathcal{X} \rightarrow \mathbb{R}^D$. In order to choose the embedding dimension D , we compute the Stiefel–Whitney obstructions, as in Section 2.3. Since the base space is the circle, which is 1-dimensional, only the first Stiefel–Whitney class provides an obstruction. The Stiefel–Whitney class of the normal bundle of the embedding $S^1 \subseteq \mathbb{R}^D$ is trivial. Thus, if the first Stiefel–Whitney class of the estimated cocycle Ω is trivial, we set $D = r + 1$, and if it is non-trivial, we set $D = r + 2$.

The second way is to use the cut-unfold technique, explained in [45, Appendix C.1], with the circular coordinates and map $\mathcal{X} \rightarrow \mathbb{R}^D$ by embedding the interval $[0, 1)$ as the unit interval of the line spanned by the first coordinate of \mathbb{R}^D . In this case, since the interval is topologically trivial (contractible), the Stiefel–Whitney classes give no obstructions, and thus we set $D = r + 1$.

4 Examples

We apply FIBERED to three examples. We reproduce a dynamical system simulation from [12] and reconstruct an attractor – a torus. We reconstruct the conformation space – a Möbius band – and energy landscape of the pentane molecule from a simulation using RDKit [39]; this is inspired by an analysis in [29]. Finally, we reconstruct the conformation space of the cyclooctane molecule – a Klein bottle glued to a 2-sphere – using the data of [27].

We compare FIBERED to various well known dimensionality reduction algorithms (see [45, Appendix C.4] for more results). Given that we consider topologically non-trivial data, we follow [40, 33] and evaluate the output of algorithms using persistent homology and persistence diagrams (PDs) to quantify the preservation of large scale topology (see [45, Appendix A.1] for background and references). When we do not clarify the field of coefficients used to compute a PD, the PD is independent of this choice. See Table 1 for a summary.

For the initial map $\pi : X \rightarrow B$ we use the implementation of circular coordinates in [54]. The parameters for FIBERED are chosen as in Section 3.6 and the computed Stiefel–Whitney obstructions are in Figure 3. For persistent homology computations, we use `ripser` [5] on geodesic distance, estimated as shortest path distance in a 15-nearest neighbor graph. For other dimensionality reduction algorithms, we use their `scikit-learn` implementation [34].

An implementation and Jupyter notebooks to reproduce the examples is in [44].

Torus from attractor of double-gyre dynamical system. Dynamical systems can be analyzed by studying the topology of their attractors [2, 36]. Given a real-valued time series coming from measurements of a given particle on which a dynamical system acts, one can obtain a pointcloud by constructing a delay embedding of the time series, which, under certain conditions, is concentrated around a diffeomorphic copy of the attractor the particle is converging to [36]. Using the delay embedding method with target dimension 4, it was shown in [12, Section 4.1] that a certain attractor of the double-gyre dynamical system [47]

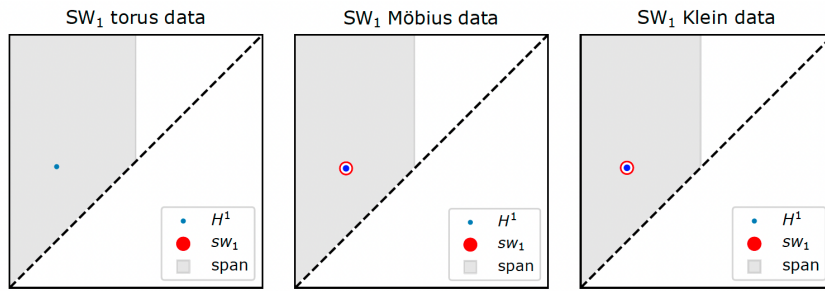


Figure 3 We proceed as in [45, Appendix C.3]. The persistence diagram of $\{\mathcal{N}_r\}_{r \in [0,1]}$ for each of the three examples, with the span of Ω shaded in grey, and the classes summing to $w_1(\Omega)$ circled in red. Recall that in the three examples, the nerve \mathcal{N} is a circle and thus the persistence diagram consists of just one prominent 1-dimensional cohomology class. In the case of the torus, the first Stiefel–Whitney class is zero and thus there is no obstruction to choosing $D = 1 + 2$ (1 being the dimension of the circle and 2 the rank of the vector bundle). In the case of the Möbius band, the first Stiefel–Whitney class coincides with the only point in the persistence diagram and is thus non-trivial, which gives an obstruction to selecting $D = 1 + 1$, which reflects the fact that the Möbius band cannot be embedded in the plane. Similarly, in the case of the Klein bottle, the Stiefel–Whitney computation gives an obstruction to selecting $D = 1 + 2$, which reflects the fact that the Klein bottle cannot be embedded in \mathbb{R}^3 .

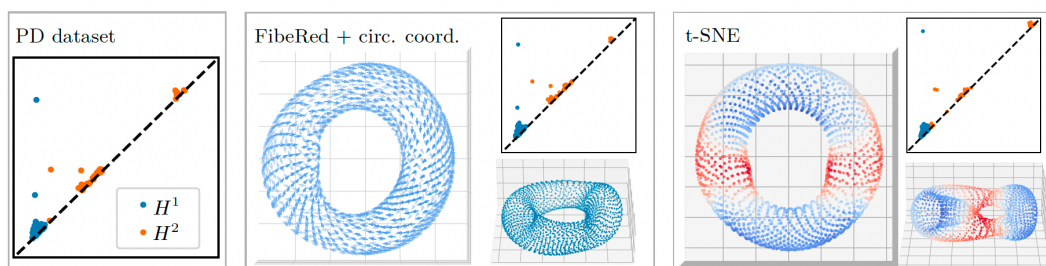
	Optimal	Isomap	t-SNE	LE/DM	LLE	HLLE	LTSA	UMAP	FibeRed
Cyl.	2	3	3	3	3	3	3	3	2
Torus	3	4	4	4	4	4	4	4	3
Möb.	3	4	4	3*	N/A	N/A	N/A	3*	3
Klein	4	5	5	7	7	5	5	4	4

Table 1 The minimal target dimension that can be chosen for each of the algorithms considered in this section, so that there exist parameters that return a topologically faithful embedding of the data. “Optimal” refers to the theoretical minimal embedding dimension. “Cylinder” refers to the dataset of Figure 1. “Torus”, “Möbius band”, and “Klein bottle” refer to the three datasets considered in this section. Since the Möbius band data is not Euclidean, some algorithms cannot be run on these data; we denote this with “N/A”. Asterisks indicate that the data had to be preprocessed with MDS and target dimension 20 in order to get a topologically faithful embedding with the corresponding algorithm and dimension.

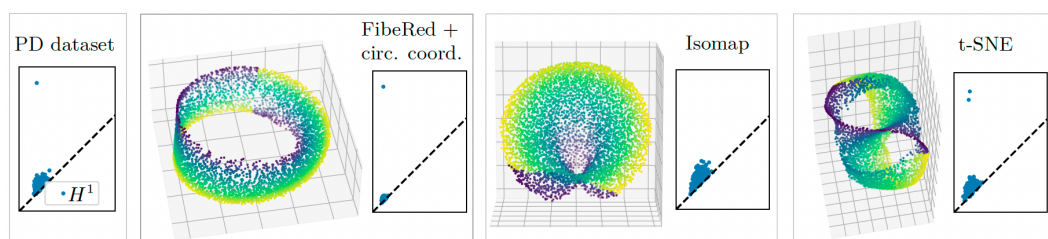
is orientable and has the homology of a torus. Here, we reproduce the simulation of [12] using the code from [16] and apply dimensionality reduction to this 4D pointcloud, with the goal of embedding the attractor and its dynamics in \mathbb{R}^3 .

In Figure 4, we show the results of FIBERED and t-SNE. In order to highlight self-intersections in low-dimensional representations, we use the following function: given a dataset X and a representation of it $f : X \rightarrow Y$ let $\kappa : X \rightarrow \mathbb{R}$ be defined by $\kappa(x) = \min_{y \in X} d_Y(f(x), f(y)) / d_X(x, y)$. The output of t-SNE in Figure 4 is representative of the output with other parameter choices and other dimensionality reduction algorithms we have tried on this data (LE, DM, LLE, HLLE, Isomap, UMAP): if the target dimension is 3, there are always self-intersections or tears. The difficulty faced by metric-based algorithms in this example is that the input torus in 4D has an approximately flat metric and thus it does not admit a smooth isometric embedding in \mathbb{R}^3 .

Möbius band from conformation space of pentane. Any fixed molecule admits different realizations, or *conformations*, in three-dimensional space. In, e.g., molecular dynamics [19],



■ **Figure 4** The PD of the original pointcloud (two prominent 1-dimensional classes, and one prominent 2-dimensional class); the output of FIBERED with the reconstructed dynamics and side view, and the PD of the output (which matches the PD of the original pointcloud well); the output of t-SNE on the same data and side view, colored by κ (red is smaller), there seem to be two self-intersections, and the PD of the output of t-SNE, which has one prominent 1-dimensional hole and two 2-dimensional voids, confirming that the red regions have been pinched.

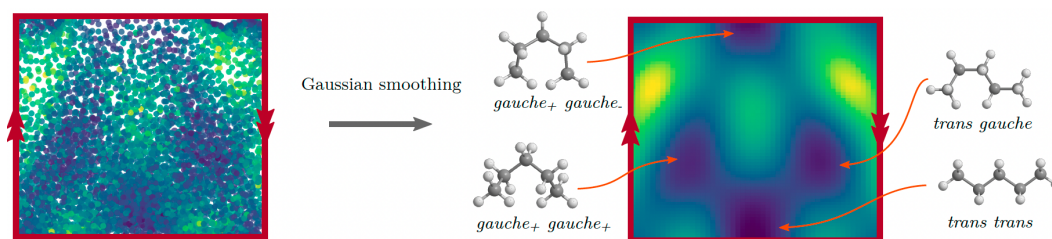


■ **Figure 5** The PD of the original pointcloud, which has one prominent 1-dimensional class; the output of FIBERED and its PD; the output of Isomap and its PD (regardless of the parameter for Isomap, the algorithm is unable to capture the circularity of the data, and thus its PD has no prominent features); the output of t-SNE and its PD (regardless of the parameters for t-SNE, the algorithm is unable to capture the circularity and non-orientability of the data without tears, which cause the output to have two holes). Outputs are colored by the (aligned) fiber coordinates estimated by FIBERED.

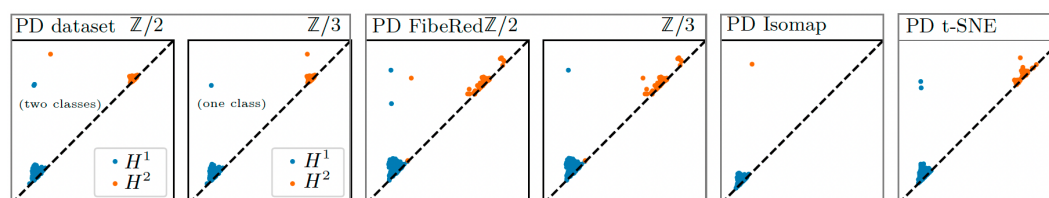
one is interested in understanding all possible conformations of a molecule. The collection of conformations up to rotations and translations is known as the *conformation space* of the molecule. Each conformation has an associated energy and the conformation space together with the energy function is known as the *energy landscape* of the molecule.

We reconstruct the conformation space and energy landscape of the pentane molecule from a simulation (see [45, Appendix C.2] for details). The pentane molecule has two rotational degrees of freedom (modelled as a torus $S^1 \times S^1$) but also has a symmetry which interchanges the two angles of rotation. For this reason, the (unlabeled) conformation space consists of a quotient of the torus, which can be seen to be a Möbius band. In Figure 5, we embed the conformation space of pentane in \mathbb{R}^3 and compare the output of FIBERED to that of Isomap and t-SNE. LE and DM are able to recover a Möbius band in \mathbb{R}^3 ; since UMAP uses LE as initialization, it is also able to recover the Möbius band in \mathbb{R}^3 . In Figure 6, we use the cut-unfold technique to find a fundamental domain of the conformation space and estimate the energy landscape.

The difficulty faced by some of the metric-based dimensionality reduction algorithms in this example is that, with respect to the intrinsic metric, the ratio between the height of the Möbius band and its circumference is approximately $2/3$ and thus there is no isometric embedding in \mathbb{R}^3 [20, Theorem 15.1].



■ **Figure 6** *Left*: the 2D representation of the conformation space of pentane using FIBERED with the cut-unfold technique, colored by energy. Arrows indicate how the data must be glued in order to recover its global topology; this information can be extracted from the cocycle Ω of ESTCOCYCLES. *Right*: a 2D representation of the energy landscape of pentane, where the energy is estimated using the representation on the left and Gaussian smoothing. We see that there are four local minima of the energy function. By going back to the molecule simulation, we confirm that these four minima correspond to the four well known conformations of pentane [3].

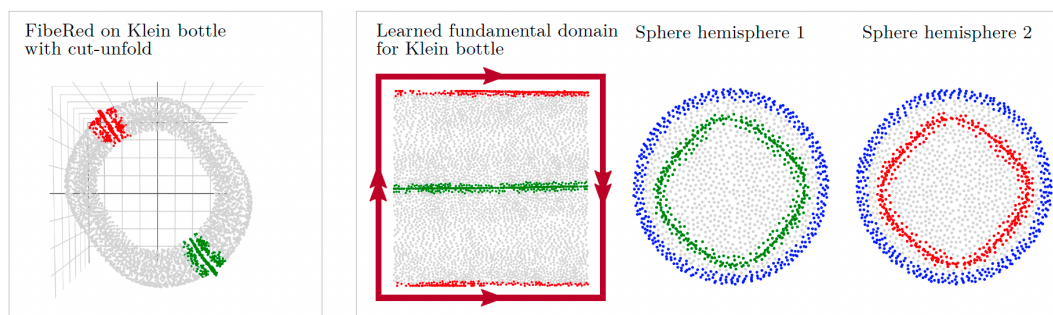


■ **Figure 7** The PD of the original data with $\mathbb{Z}/2$ (two prominent 1-dimensional and one 2-dimensional classes) and $\mathbb{Z}/3$ coefficients (one prominent 1-dimensional class), which suggests the data is a Klein bottle; the PD of the representation obtained using FIBERED, which matches the original topology well; the PD of a representation using Isomap; the PD of a representation using t-SNE. For Isomap and t-SNE, the PD is the same regardless of the field of coefficients.

Klein bottle from conformation space of cyclooctane. In this example, we reconstruct the conformation space and energy landscape of the cyclooctane molecule using the dataset of [27]. In [27], it is shown that the conformation space of cyclooctane consists of a 2-sphere glued to a Klein bottle along two disjoint circles and a parametrization of the dataset is given using Isomap and knowledge about how the data was generated.

By estimating the local dimension of the data, we first separate the Klein bottle part of the dataset from the 2-sphere. In Figure 7, we embed the Klein bottle part of the data in 4D. We were not able to recover the right topology in \mathbb{R}^4 using any of LE, DM, LLE, HLLE, LTSA, Isomap, or t-SNE. Meanwhile, UMAP is able to recover the right topology in \mathbb{R}^4 . In order to evaluate the 4D embeddings, we use the following distinguishing feature of the Klein bottle K : with $\mathbb{Z}/2$ coefficients we have $\dim(H^1(K; \mathbb{Z}/2)) = 2$ and $\dim(H^2(K; \mathbb{Z}/2)) = 1$, while with $\mathbb{Z}/3$ coefficients we have $\dim(H^1(K; \mathbb{Z}/3)) = 1$ and $\dim(H^2(K; \mathbb{Z}/3)) = 0$. In Figure 8, we produce an efficient 2D parametrization of the conformation space of cyclooctane without using a priori knowledge of how the data was generated.

The difficulty faced by metric-based algorithms in this example is that the Klein bottle in high dimensional space has aspect ratio close to 1 (i.e., an isometric representation by a fundamental domain such as the one Figure 8 (left) has commensurable height and width), and thus it does not admit a simple isometric embedding in \mathbb{R}^4 .



■ **Figure 8** *Left*: The output of FIBERED with the cut-unfold technique on the portion of the conformation space belonging to the Klein bottle. Colored in red and green are the two circles that glue the Klein bottle to the 2-sphere. Using this representation – a cylinder – we compute a new circular coordinate, which we combine with the initial circular coordinate to get a fundamental domain for the Klein bottle. *Right*: A 2D model of the conformation space of cyclooctane. The two circles are two hemispheres of the 2D sphere and were obtained using Isomap. Points not colored in grey indicate the gluings that have to be performed to recover the conformation space.

5 Discussion

We have presented a procedure to learn vector bundles from data and demonstrated that it can be used to decouple the global topology from the local geometry in topologically complex data. We showed with examples that this can be helpful for embedding topologically complex data in low dimension, as well as for charting such data. We have also developed a mathematical foundation for this point of view.

Limitations. The theory and methods presented in this paper assume that the data lives in the total space of a vector bundle. There are two main ways in which real data can deviate from these assumptions: (1) There are singularities in the data manifold and thus the base map is not a vector bundle since fibers may have different dimensions; (2) the data contains outliers and only a core subset of the data satisfies the assumptions. Two other important caveats are that (3) the procedure assumes that a base map is given and that (4) success depends on the first step of the procedure finding a good cover of the data. We comment on these remarks below.

Future work. With respect to (1), the situation in which the fibers of the base map can have different dimensions can be abstracted using the theory of stratified vector bundles [7, 6]. We believe that the main algorithm of Section 3 can be adjusted to account for different local dimensions by allowing the cocycle Ω between patches with different dimension to be a matrix in a Stiefel manifold instead of an orthogonal matrix. With respect to (2), our procedures are robust with respect to limited amount of noise and the problem of devising extensions robust to outliers is left as future work.

With respect to (3), there are several ways to obtain non-linear initial representations. First, other cohomological coordinates besides circular coordinates have been developed [35, 38]. Second, one could use standard non-linear representations, such as the ones learned by Diffusion Maps [23, 13]. Third, one could use any of the lens functions [50, Section 4] Mapper uses. Another interesting avenue for constructing coarse topological representations is to build a graph on the data, simplify it while preserving part of its large scale topology, and use a graph layout algorithm. Regarding (4), finding good covers of noisy data is an

interesting problem in itself; we believe the approach presented in this paper can be made more robust by developing a more nuanced subroutine for computing a cover.

Our approach depends on several constructions, some of which are known to be consistent estimators. Addressing the consistency of the entire pipeline is left for future work.

Finally, FIBERED can be interpreted as principal component analysis relative to an initial representation, as it works by linearly embedding the local coordinates of X that are not already accounted by the initial map, in a way that is globally consistent and orthogonal to the coordinates already accounted by the initial map. This suggests considering versions of other popular dimensionality reduction algorithms relative to an initial representation.

References

- 1 Eddie Aamari, Jisu Kim, Frédéric Chazal, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Estimating the reach of a manifold. *Electronic journal of statistics*, 13(1):1359–1399, 2019.
- 2 Henry D. I. Abarbanel. *Analysis of observed chaotic data*. Institute for Nonlinear Science. Springer-Verlag, New York, 1996. doi:10.1007/978-1-4612-0763-4.
- 3 Roman M Balabin. Enthalpy difference between conformations of normal alkanes: Raman spectroscopy study of n-pentane and n-butane. *The Journal of Physical Chemistry A*, 113(6):1012–1019, 2009.
- 4 Afonso S. Bandeira, Amit Singer, and Daniel A. Spielman. A Cheeger inequality for the graph connection Laplacian. *SIAM J. Matrix Anal. Appl.*, 34(4):1611–1630, 2013. doi:10.1137/120875338.
- 5 Ulrich Bauer. Ripser: efficient computation of vietoris-rips persistence barcodes. *Journal of Applied and Computational Topology*, 2021. doi:10.1007/s41468-021-00071-5.
- 6 Hans-Joachim Baues and Davide L. Ferrario. K -theory of stratified vector bundles. *K-Theory*, 28(3):259–284, 2003. doi:10.1023/A:1026215632002.
- 7 Hans-Joachim Baues and Davide L. Ferrario. Stratified fibre bundles. *Forum Math.*, 16(6):865–902, 2004. doi:10.1515/form.2004.16.6.865.
- 8 Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.
- 9 Ingwer Borg and Patrick J. F. Groenen. *Modern multidimensional scaling*. Springer Series in Statistics. Springer, New York, second edition, 2005. Theory and applications.
- 10 Matthew Brand. Charting a manifold. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL: <https://proceedings.neurips.cc/paper/2002/file/8929c70f8d710e412d38da624b21c3c8-Paper.pdf>.
- 11 Elie Joseph Cartan. Sur la possibilité de plonger un espace riemannien donné dans un espace euclidien. *Annales de la Société Polonaise de Mathématique*, 1928.
- 12 Gisela D. Charó, Guillermo Artana, and Denisse Sciamarella. Topology of dynamical reconstructions from Lagrangian data. *Phys. D*, 405:132371, 12, 2020. doi:10.1016/j.physd.2020.132371.
- 13 Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Appl. Comput. Harmon. Anal.*, 21(1):5–30, 2006. doi:10.1016/j.acha.2006.04.006.
- 14 Vin de Silva, Dmitry Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete Comput. Geom.*, 45(4):737–759, 2011. doi:10.1007/s00454-011-9344-x.
- 15 David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, 2003. doi:10.1073/pnas.1031596100.
- 16 Ximena Fernández. Topology of fluid flows. https://github.com/ximenafernandez/topology_fluids, 2022.


- 17 Jesus Garcia-Diaz, Rolando Menchaca-Mendez, Ricardo Menchaca-Mendez, Saúl Pomares Hernández, Julio César Pérez-Sansalvador, and Nouredine Lakouari. Approximation algorithms for the vertex k -center problem: Survey and experimental evaluation. *IEEE Access*, 7:109228–109245, 2019. doi:10.1109/ACCESS.2019.2933875.
- 18 Robert Ghrist. Barcodes: the persistent topology of data. *Bull. Amer. Math. Soc. (N.S.)*, 45(1):61–75, 2008. doi:10.1090/S0273-0979-07-01191-3.
- 19 James M Haile. *Molecular dynamics simulation: elementary methods*. John Wiley & Sons, Inc., 1992.
- 20 B. Halpern and C. Weaver. Inverting a cylinder through isometric immersions and isometric embeddings. *Trans. Amer. Math. Soc.*, 230:41–70, 1977. doi:10.2307/1997711.
- 21 Allen Hatcher. *Algebraic topology*. Cambridge University Press, Cambridge, 2002.
- 22 Roger A. Horn and Charles R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, second edition, 2013.
- 23 S. Lafon and A.B. Lee. Diffusion maps and coarse-graining: a unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006. doi:10.1109/TPAMI.2006.184.
- 24 John Aldo Lee and Michel Verleysen. Nonlinear dimensionality reduction of data manifolds with essential loops. *Neurocomput.*, 67:29–53, August 2005. doi:10.1016/j.neucom.2004.11.042.
- 25 Anna V Little, Jason Lee, Yoon-Mo Jung, and Mauro Maggioni. Estimation of intrinsic dimensionality of samples from noisy low-dimensional manifolds in high dimensions with multiscale svd. In *2009 IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 85–88. IEEE, 2009.
- 26 Zixiang Luo, Chenyu Xu, Zhen Zhang, and Wenfei Jin. A topology-preserving dimensionality reduction method for single-cell rna-seq data using graph autoencoder. *Scientific reports*, 11(1):1–8, 2021.
- 27 Shawn Martin, Aidan Thompson, Evangelos A Coutsias, and Jean-Paul Watson. Topology of cyclo-octane energy landscape. *The journal of chemical physics*, 132(23):234115, 2010.
- 28 Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. doi:10.48550/ARXIV.1802.03426.
- 29 Ingrid Membrillo-Solis, Mariam Pirashvili, Lee Steinberg, Jacek Brodzki, and Jeremy G. Frey. Topology and geometry of molecular conformational spaces and energy landscapes, 2019. doi:10.48550/ARXIV.1907.07770.
- 30 John W. Milnor and James D. Stasheff. *Characteristic classes*. Annals of Mathematics Studies, No. 76. Princeton University Press, Princeton, N. J.; University of Tokyo Press, Tokyo, 1974.
- 31 Michael Moor, Max Horn, Bastian Rieck, and Karsten Borgwardt. Topological autoencoders. In *International conference on machine learning*, pages 7045–7054. PMLR, 2020.
- 32 Steve Y. Oudot. *Persistence theory: from quiver representations to data analysis*, volume 209 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2015. doi:10.1090/surv/209.
- 33 Rahul Paul and Stephan K Chalup. A study on validating non-linear dimensionality reduction using persistent homology. *Pattern Recognition Letters*, 100:160–166, 2017.
- 34 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- 35 Jose A. Perea. Multiscale projective coordinates via persistent cohomology of sparse filtrations. *Discrete Comput. Geom.*, 59(1):175–225, 2018. doi:10.1007/s00454-017-9927-2.
- 36 Jose A. Perea. Topological times series analysis. *Notices Amer. Math. Soc.*, 66(5):686–694, 2019.
- 37 Jose A. Perea. Sparse circular coordinates via principal \mathbb{Z} -bundles. In *Topological data analysis—the Abel Symposium 2018*, volume 15 of *Abel Symp.*, pages 435–458. Springer, Cham, 2020.

- 38 Luis Polanco and Jose A. Perea. Coordinatizing data with lens spaces and persistent cohomology. In Zachary Friggstad and Jean-Lou De Carufel, editors, *Proceedings of the 31st Canadian Conference on Computational Geometry, CCCG 2019, August 8-10, 2019, University of Alberta, Edmonton, Alberta, Canada*, pages 49–58, 2019.
- 39 RDKit developers. RDKit: Open-source cheminformatics. <http://www.rdkit.org>, 2022.
- 40 Bastian Rieck and Heike Leitte. Persistent homology for the evaluation of dimensionality reduction schemes. In *Computer Graphics Forum*, volume 34, pages 431–440. Wiley Online Library, 2015.
- 41 Sam Roweis, Lawrence Saul, and Geoffrey E Hinton. Global coordination of local linear models. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL: <https://proceedings.neurips.cc/paper/2001/file/850af92f8d9903e7a4e0559a98ecc857-Paper.pdf>.
- 42 Sam T Roweis and Lawrence K Saul. Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326, 2000.
- 43 Luis Scoccola, Hitesh Gakhar, Johnathan Bush, Nikolas Schonsheck, Tatum Rask, Ling Zhou, and Jose A. Perea. Toroidal coordinates: Decorrelating circular coordinates with lattice reduction, 2022. doi:10.48550/ARXIV.2212.07201.
- 44 Luis Scoccola and Jose A. Perea. FibeRed implementation. <https://github.com/LuisScoccola/fibered.git>, 2022.
- 45 Luis Scoccola and Jose A. Perea. Fiberwise dimensionality reduction of topologically complex data with vector bundles, 2022. doi:10.48550/ARXIV.2206.06513.
- 46 Luis Scoccola and Jose A. Perea. Approximate and discrete Euclidean vector bundles. *Forum of Mathematics, Sigma*, 11, March 2023. doi:10.1017/fms.2023.16.
- 47 Shawn C. Shadden, Francois Lekien, and Jerrold E. Marsden. Definition and properties of Lagrangian coherent structures from finite-time Lyapunov exponents in two-dimensional aperiodic flows. *Phys. D*, 212(3-4):271–304, 2005. doi:10.1016/j.physd.2005.10.007.
- 48 A. Singer. Angular synchronization by eigenvectors and semidefinite programming. *Appl. Comput. Harmon. Anal.*, 30(1):20–36, 2011. doi:10.1016/j.acha.2010.02.001.
- 49 Amit Singer and Hau-tieng Wu. Orientability and diffusion maps. *Appl. Comput. Harmon. Anal.*, 31(1):44–58, 2011. doi:10.1016/j.acha.2010.10.001.
- 50 Gurjeet Singh, Facundo Mémoli, Gunnar E Carlsson, et al. Topological methods for the analysis of high dimensional data sets and 3d object recognition. *PBG Eurographics*, 2, 2007.
- 51 Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web, WWW '16*, pages 287–297, Republic and Canton of Geneva, CHE, 2016. International World Wide Web Conferences Steering Committee. doi:10.1145/2872427.2883041.
- 52 Yee Teh and Sam Roweis. Automatic alignment of local representations. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. URL: <https://proceedings.neurips.cc/paper/2002/file/3a1dd98341fafc1dfe9bcf36360e6b84-Paper.pdf>.
- 53 Joshua B Tenenbaum, Vin de Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
- 54 Chris Tralie, Tom Mease, and Jose Perea. Dreimac. <https://github.com/ctralie/DREiMac>, 2017.
- 55 Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.
- 56 Alexander Wagner, Elchanan Solomon, and Paul Bendich. Improving metric dimensionality reduction with distributed topology, 2021. doi:10.48550/ARXIV.2106.07613.
- 57 Hassler Whitney. The self-intersections of a smooth n -manifold in $2n$ -space. *Ann. of Math. (2)*, 45:220–246, 1944. doi:10.2307/1969265.

56:18 Fiberwise Dimensionality Reduction

- 58 Lin Yan, Yaodong Zhao, Paul Rosen, Carlos Scheidegger, and Bei Wang. Homology-preserving dimensionality reduction via manifold landmarking and tearing. *arXiv preprint arXiv:1806.08460*, 2018.
- 59 Zhenyue Zhang and Hongyuan Zha. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing*, pages 313–338, 2004.

Toroidal Coordinates: Decorrelating Circular Coordinates with Lattice Reduction

Luis Scoccola ✉ 🏠 

Department of Mathematics, Northeastern University, Boston, MA, USA

Hitesh Gakhar ✉ 🏠 

Department of Mathematics, The University of Oklahoma, Norman, OK, USA

Johnathan Bush ✉ 🏠 


Department of Mathematics, University of Florida, Gainesville, FL, USA

Nikolas Schonsheck ✉ 🏠 

Department of Mathematical Sciences, University of Delaware, Newark, DE, USA

Tatum Rask ✉ 🏠

Department of Mathematics, Colorado State University, Fort Collins, CO, USA

Ling Zhou ✉ 🏠 

Department of Mathematics, The Ohio State University, Columbus, OH, USA

Jose A. Perea ✉ 🏠 

Department of Mathematics and Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA

Abstract

The circular coordinates algorithm of de Silva, Morozov, and Vejdemo-Johansson takes as input a dataset together with a cohomology class representing a 1-dimensional hole in the data; the output is a map from the data into the circle that captures this hole, and that is of minimum energy in a suitable sense. However, when applied to several cohomology classes, the output circle-valued maps can be “geometrically correlated” even if the chosen cohomology classes are linearly independent. It is shown in the original work that less correlated maps can be obtained with suitable integer linear combinations of the cohomology classes, with the linear combinations being chosen by inspection. In this paper, we identify a formal notion of geometric correlation between circle-valued maps which, in the Riemannian manifold case, corresponds to the Dirichlet form, a bilinear form derived from the Dirichlet energy. We describe a systematic procedure for constructing low energy torus-valued maps on data, starting from a set of linearly independent cohomology classes. We showcase our procedure with computational examples. Our main algorithm is based on the Lenstra–Lenstra–Lovász algorithm from computational number theory.

2012 ACM Subject Classification Mathematics of computing → Algebraic topology

Keywords and phrases dimensionality reduction, lattice reduction, Dirichlet energy, harmonic, cocycle

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.57

Related Version *Full Version*: <https://arxiv.org/abs/2212.07201> [25]

Supplementary Material *Software (Proof-of-concept implementation)*: <https://github.com/LuisScoccola/DREiMac> [24]

Funding This material is based upon work initiated during the 2022 Mathematics Research Community *Data Science at the Crossroads of Analysis, Geometry, and Topology* supported by the National Science Foundation under Grant Number DMS 1916439.

Luis Scoccola: supported by the NSF through grants CCF-2006661 and CAREER award DMS-1943758.

Johnathan Bush: supported by the NSF-Simons Southeast Center for Mathematics and Biology



© Luis Scoccola, Hitesh Gakhar, Johnathan Bush, Nikolas Schonsheck, Tatum Rask, Ling Zhou, and Jose A. Perea; licensed under Creative Commons License CC-BY 4.0

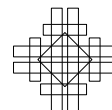
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 57; pp. 57:1–57:20

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



through NSF grant DMS-1764406 and Simons Foundation grant 594594.

Nikolas Schonsheck: supported by the Air Force Office of Scientific Research through award number FA9550-21-1-0266.

Jose A. Perea: supported by the NSF through grants CCF-2006661 and CAREER award DMS-1943758.

1 Introduction

Motivation and problem statement. Given a point cloud $X \subseteq \mathbb{R}^n$ concentrated around a k -dimensional linear subspace, linear dimensionality reduction algorithms such as Principal Component Analysis are effective at finding a low-dimensional representation $X \rightarrow \mathbb{R}^k$ of the data that preserves the linear structure. The problem of finding low-dimensional representations of non-linear data is more involved; one reason being that it is often hard to make principled assumptions about which particular non-linear shape the data may have. Topological Data Analysis provides tools allowing for the extraction of qualitative and quantitative topological information from discrete data. These tools include persistent cohomology, which can be used, in particular, to identify circular features.

Given a dataset X and a class α in the first integral persistent cohomology group of X , the circular coordinates algorithm of [4, 3] constructs a circle-valued representation $\text{cc}_\alpha : X \rightarrow \mathbb{S}^1$, which preserves the cohomology class α in a precise sense [21, Theorem 3.2]. The circular coordinates algorithm is thus a principled non-linear dimensionality reduction algorithm, and has found various applications [16, 28], particularly in neuroscience [11, 7, 23].

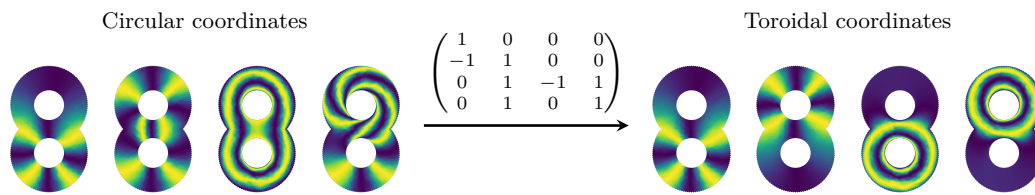
As observed in [3, Section 3.9], and reproduced in Figure 2, when several cohomology classes $\alpha_1, \dots, \alpha_k$ are used to produce a single torus-valued representation $(\text{cc}_{\alpha_1}, \dots, \text{cc}_{\alpha_k}) : X \rightarrow \mathbb{S}^1 \times \dots \times \mathbb{S}^1 = \mathbb{T}^k$, this representation is often not the most natural. Indeed, even when the cohomology classes α_i are linearly independent (l.i.), the maps cc_{α_i} can be “geometrically correlated.” Certain integer linear combinations of the cohomology classes, however, can yield decorrelated representations. The problems of defining an appropriate notion of geometric correlation between circle-valued maps, and of using this notion to systematically decorrelate sets of circle-valued maps are left open in [3]. In this paper, we address these two problems.

Contributions. Given a Riemannian manifold \mathcal{M} , we propose to measure the geometric correlation between smooth maps $f, g : \mathcal{M} \rightarrow \mathbb{S}^1$ using the Dirichlet form $D(f, g) \in \mathbb{R}$. We show that, given smooth maps $f, g : \mathcal{M} \rightarrow \mathbb{S}^1$ obtained by integrating cocycles θ and η defined on the nerve $N(\mathcal{U})$ of an open cover \mathcal{U} of \mathcal{M} , there exists an inner product $\langle -, - \rangle_D$ at the level of cocycles inducing an isometry $\langle \theta, \eta \rangle_D = D(f, g)$ (Theorem 15). This motivates our Toroidal Coordinates Algorithm (Algorithm 2), which works at the level of cocycles on a simplicial complex and produces low energy torus-valued representations of data. We prove that the energy minimization subroutine of the Toroidal Coordinates Algorithm is correct (Theorem 3) and give a geometric interpretation (Proposition 12). We introduce the Sparse Toroidal Coordinates Algorithm (Algorithm 8) – a more scalable version of our main algorithm – implemented in [24], and showcase it on four datasets (Section 6).

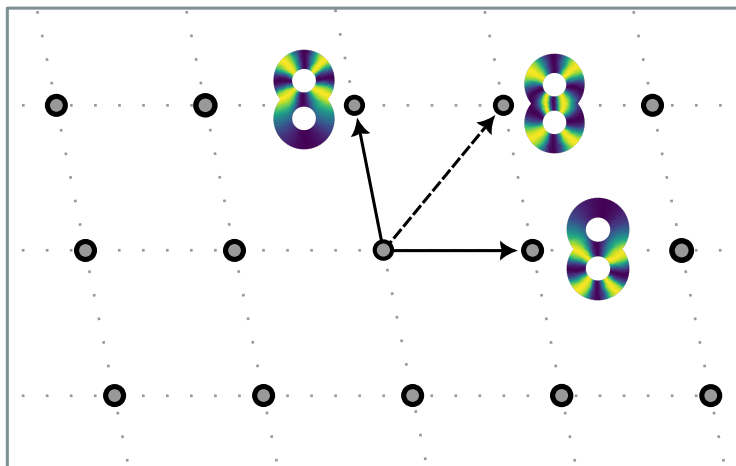
Structure of the paper. Section 2 contains background and can be referred to as needed. The next two sections, 3 and 4, can be read in any order: Section 3 contains a computational description of the Toroidal Coordinates Algorithm, while Section 4 describes an analogous procedure for Riemannian manifolds and serves as motivation. Section 5 describes the Sparse Toroidal Coordinates Algorithm, then demonstrated in the examples of Section 6.



■ **Figure 1** An illustration of how we use colors to display circular coordinates on data. We first color the circle S^1 with a smooth transition between yellow and violet, repeated four times; then, given a function into the circle, we color its domain by pulling back the coloring. Depicted are the colorings induced on a genus two surface by the map that “goes around a longitude” (Left) and by the map that “goes around a meridian” (Right).



■ **Figure 2** We represent circular coordinates as explained in Figure 1. *Left:* Four circle-valued maps obtained by running the (Sparse) Circular Coordinates Algorithm on four generators of the first cohomology of a genus two surface. The generators were obtained using persistent cohomology. Although the cohomology classes are linearly independent, they do not give a particularly efficient representation of the 1-dimensional holes in the data: for instance, the first two maps both vary as one goes around the bottom outer hole. *Right:* Four circle-valued maps obtained by running the (Sparse) Toroidal Coordinates Algorithm, with input the same four cohomology classes used on the left. *Middle:* The change of basis matrix applied to the cohomology classes in order to geometrically decorrelate them. See Section 6.1 for details about this example.



■ **Figure 3** The lattice generated by the two left-most circle-valued maps in Figure 2, using our notion of discrete geometric correlation (Definition 4). These two circle-valued maps are represented as the horizontal vector and the dashed vector. Note that, although these two vectors form a basis of the lattice, there exists a basis of smaller total squared length: the one formed by the two solid vectors. The two solid vectors correspond to two circle-valued returned by the (Sparse) Toroidal Coordinates Algorithm, as shown in Figure 2.

Discussion. In the examples in Section 6, running the Sparse Toroidal Coordinates Algorithm on a set of cohomology classes gives results that are qualitatively and quantitatively better than the results obtained by running the Sparse Circular Coordinates Algorithm separately on each class. This suggests that the Dirichlet form is indeed a useful notion of geometric correlation that can be leveraged for producing geometrically efficient and topologically faithful low-dimensional representations of data. We believe our methods can be extended to representations valued in non-trivial spaces other than tori, such as other Lie groups.

Various interesting problems remain open: Is our lattice reduction problem (Problem 6) provably a hard computational problem? Why is it that the de Silva–Morozov–Vejdemo–Johansson inner product and the inner product estimated in Construction 16 give such similar results (Remark 17)? Are our heuristics for estimating the Dirichlet form from finite samples (Construction 16 and [25, Construction 20]) consistent? Here, consistency refers to convergence in probability to the Dirichlet form as the number of samples goes to infinity.

2 Background

For details about the basics of algebraic topology and Riemannian geometry, we refer the reader to [18] and [10], respectively.

Cohomology. Let K be a finite abstract simplicial complex and let A be either of the rings \mathbb{Z} or \mathbb{R} . Let K_0 denote the set of vertices of K and let $K_1 = \{(i, j) \in K_0 \times K_0 : \{i, j\} \text{ is a 1-simplex of } K\}$. For a function $\theta : K_1 \rightarrow A$, we denote the evaluation of θ on a pair (i, j) by θ^{ij} . The group of 0-cochains $C^0(K; A)$ is the Abelian group of functions $K_0 \rightarrow A$, and the group of 1-cocycles is the Abelian group

$$Z^1(K; A) = \left\{ \theta : K_1 \rightarrow A \mid \begin{array}{l} \theta^{ij} = -\theta^{ji} \text{ for all } (i, j) \in K_1, \\ \theta^{ij} + \theta^{jk} = \theta^{ik} \text{ for every 2-simplex } \{i, j, k\} \text{ of } K \end{array} \right\},$$

The *first cohomology group* of K with coefficients in A is $H^1(K; A) := Z^1(K; A)/\text{Im}(\delta)$, where δ denotes the group morphism $C^0(K; A) \rightarrow Z^1(K; A)$ defined by $\delta(\tau)^{ij} = \tau(j) - \tau(i)$. Given $\theta \in Z^1(K; A)$ we denote its image in $H^1(K; A)$ as $[\theta] \in H^1(K; A)$.

For any topological space B , we let ι denote the homomorphism $\iota : H^1(B; \mathbb{Z}) \rightarrow H^1(B; \mathbb{R})$ induced by the inclusion of coefficients $\mathbb{Z} \hookrightarrow \mathbb{R}$.

The Frobenius inner product. Let W and Z be real, finite dimensional inner product spaces. For a linear map $A : Z \rightarrow W$, let $A^* : Z \rightarrow W$ denotes the adjoint of A with respect to the inner products on W and Z . The *Frobenius inner product* between two linear maps $A, B : W \rightarrow Z$ is defined as $\langle A, B \rangle_F := \text{Tr}(A^*B)$. In particular, the space of linear maps $W \rightarrow Z$ can be endowed with the *Frobenius norm*, given by $\|A\|_F := \sqrt{\text{Tr}(A^*A)}$.

Circle and tori. We define the circle as the quotient of topological Abelian groups $\mathbb{S}^1 = \mathbb{R}/\mathbb{Z}$, with the induced quotient map $\mathbb{R} \xrightarrow{q} \mathbb{S}^1$ given by mapping r to $r \bmod \mathbb{Z}$. We endow \mathbb{S}^1 with the unique Riemannian metric that makes q a local Riemannian isometry. Given $k \in \mathbb{N}$, let $\mathbb{T}^k = (\mathbb{S}^1)^k$ denote the k -dimensional torus with the product Riemannian metric.

Circle-valued maps. Let B be a topological space. Given $f, g : B \rightarrow \mathbb{S}^1$, define $f + g : B \rightarrow \mathbb{S}^1$ by $(f + g)(p) = f(p) + g(p)$ for all $p \in B$. This endows the set of maps $B \rightarrow \mathbb{S}^1$ with the structure of an Abelian group. We say $f : B \rightarrow \mathbb{S}^1$ and $g : B \rightarrow \mathbb{S}^1$ are *rotationally equivalent* if $f - g$ is constant on each connected component of B . Analogously, for a simplicial

complex K , we say that maps on vertices $f : K_0 \rightarrow \mathbb{S}^1$ and $g : K_0 \rightarrow \mathbb{S}^1$ are *rotationally equivalent* if $f - g : K_0 \rightarrow \mathbb{S}^1$ is constant on each connected component of K .

Differential of circle-valued maps. There is a canonical isomorphism $T\mathbb{S}^1 \cong \mathbb{S}^1 \times \mathbb{R}$ of Riemannian vector bundles over \mathbb{S}^1 . Here, $\mathbb{S}^1 \times \mathbb{R} \rightarrow \mathbb{S}^1$ is the trivial Riemannian vector bundle over \mathbb{S}^1 and the isomorphism is given by the linear isometries $d(\cdot - q)_q : T_q\mathbb{S}^1 \rightarrow T_0\mathbb{S}^1 \cong \mathbb{R}$, where $\cdot - q : \mathbb{S}^1 \rightarrow \mathbb{S}^1$ denotes subtracting q , and the isomorphism $T_0\mathbb{S}^1 \cong \mathbb{R}$ is chosen once and for all. Using the isomorphism $T\mathbb{S}^1 \cong \mathbb{S}^1 \times \mathbb{R}$, we can unambiguously treat the differential of a map $f : \mathcal{M} \rightarrow \mathbb{S}^1$ at a point $p \in \mathcal{M}$ as a linear function $df_p : T_p\mathcal{M} \rightarrow \mathbb{R}$. In particular, any smooth map $f : \mathcal{M} \rightarrow \mathbb{S}^1$ induces a 1-form $df \in \Omega^1(\mathcal{M})$ on \mathcal{M} .

Dirichlet energy and Dirichlet form. Given a closed Riemannian manifold \mathcal{M} , we let μ denote its Riemannian measure. The *Dirichlet energy* of a smooth map $f : \mathcal{M} \rightarrow \mathcal{N}$ between Riemannian manifolds is

$$E[f] := \frac{1}{2} \int_{p \in \mathcal{M}} \|df_p\|_F^2 d\mu(p),$$

where $df_p : T_p\mathcal{M} \rightarrow T_{f(p)}\mathcal{N}$ is the differential of f , a map between inner product spaces.

Recall that the inner product on the space of 1-forms $\Omega^1(\mathcal{M})$ is given, for $\theta, \eta \in \Omega^1(\mathcal{M})$, by $\langle \theta, \eta \rangle_{\Omega^1} := \int_{p \in \mathcal{M}} \langle \theta_p, \eta_p \rangle_F d\mu(p)$. One can thus extend the Dirichlet energy of circle-valued maps to a bilinear form, as follows. Given $f, g : \mathcal{M} \rightarrow \mathbb{S}^1$, define their *Dirichlet form* as

$$D(f, g) := \frac{1}{2} \langle df, dg \rangle_{\Omega^1} = \frac{1}{2} \int_{p \in \mathcal{M}} \langle df_p, dg_p \rangle_F d\mu(p).$$

We remark that, as defined, the Dirichlet form makes sense only for circle-valued maps. We conclude by noticing that the Dirichlet form and the Dirichlet energy determine each other. On one hand, we have $E[f] = D(f, f)$. On the other hand, we have $D(f, g) = \frac{1}{4}(E(f + g) - E(f - g))$, by the polarization identity for the inner product space $\Omega^1(\mathcal{M})$.

3 The Toroidal Coordinates Algorithm

3.1 From circular coordinates to toroidal coordinates

We recall the circular coordinates algorithm of [4, 3] and use its main minimization subroutine to motivate the Toroidal Coordinates Algorithm. The most relevant portion of the full pipeline¹ is given as Algorithm 1, which we refer to as the *Circular Coordinates Algorithm*.

As can be easily checked, the minimization subroutine (Algorithm 3) of the Circular Coordinates Algorithm returns a solution to the following problem:

► **Problem 1.** Given $0 \neq \alpha \in H^1(K; \mathbb{Z})$ and an inner product $\langle \cdot, \cdot \rangle$ on $Z^1(K; \mathbb{R})$, find $\theta \in Z^1(K; \mathbb{R})$ of minimum norm such that $[\theta] = \iota(\alpha) \in H^1(K; \mathbb{R})$.

We propose the following extension of Problem 1 to the case in which more than one cohomology class is selected.

► **Problem 2.** Given linearly independent $\alpha_1, \dots, \alpha_k \in H^1(K; \mathbb{Z})$ and inner product $\langle \cdot, \cdot \rangle$ on $Z^1(K; \mathbb{R})$, find $\theta_1, \dots, \theta_k \in Z^1(K; \mathbb{R})$ minimizing $\sum_{j=1}^k \|\theta_j\|^2$, with the property that the sets $\{[\theta_j]\}_{1 \leq j \leq k}$ and $\{\iota(\alpha_j)\}_{1 \leq j \leq k}$ generate the same Abelian subgroup of $H^1(K; \mathbb{R})$.

¹ We refer the reader to [3, Sections 2.2–2.4] for details about the rest of the pipeline.

Simple examples, such as the one depicted in Figure 3, show that Problem 2 does not reduce to solving Problem 1 for each individual cohomology class. Indeed, as explained in Section 3.3, we believe that Problem 2 is significantly harder to solve exactly than Problem 1. Nevertheless, we also show that one can use the Lenstra–Lenstra–Lovász lattice basis reduction algorithm to find an approximate solution to Problem 2. This approximation is the content of the following result, which is proven in [25, Appendix A.1].

► **Theorem 3.** *The output of Algorithm 4 consists of cocycles $\theta_1, \dots, \theta_k$ such that $\sum_{j=1}^k \|\theta_j\|^2$ is at most 2^{k-1} times the optimal solution of Problem 2.*

Algorithm 4 constitutes the main minimization subroutine of the Toroidal Coordinates Algorithm, which is given as Algorithm 2.

3.2 On the choice of inner product

Algorithms 1 and 2 depend on a user-given choice of inner product on $Z^1(K; \mathbb{R})$. In [4, 3], the inner product used is given by

$$\langle \theta, \eta \rangle_{\text{dSMV}} := \sum_{\{i,j\} \in K_1} \theta^{ij} \eta^{ij}. \tag{1}$$

The motivation for this choice is given in [3, Proposition 2], which implies that the map $K_0 \rightarrow \mathbb{S}^1$ returned by Algorithm 1 has the property that it can be extended to a continuous function $|K| \rightarrow \mathbb{S}^1$ which maps each edge $\{i, j\}$ of K to a curve of length $|\theta^{ij}|$. Thus, with this choice of inner product, the circle-valued representation returned by Algorithm 1 is one that stretches the edges of the simplicial complex as little as possible.

There are other natural choices of inner product. In particular, we show in Theorem 15 that there exists an inner product between cocycles that recovers the Dirichlet form between circle-valued maps obtained by integrating these cocycles. Since, as explained in the contributions section, we propose to measure the geometric correlation between maps $f, g : \mathcal{M} \rightarrow \mathbb{S}^1$ on a Riemannian manifold using their Dirichlet form, this motivates the following definition.

► **Definition 4.** *Let K be a simplicial complex and let $\langle -, - \rangle$ be an inner product on $Z^1(K; \mathbb{R})$. Given cocycles $\theta, \eta \in Z^1(K; \mathbb{R})$ with $[\theta], [\eta] \in \text{Im}(\iota : H^1(K; \mathbb{Z}) \rightarrow H^1(K; \mathbb{R}))$, define the discrete geometric correlation between $\text{integrate}_\theta, \text{integrate}_\eta : K_0 \rightarrow \mathbb{S}^1$ as $\langle \theta, \eta \rangle$. Here integrate is as defined in Algorithm 5.*

In Section 4, we give a geometric interpretation of the Toroidal Coordinates Algorithm and provide more details as to why the above notion of discrete geometric correlation is a discrete analogue of the Dirichlet form (Remark 13).

We conclude this section with a remark explaining why an exact or approximate solution to Problem 2 promotes low discrete geometric correlation.

► **Remark 5.** Let $\theta_1, \dots, \theta_k \in Z^1(K; \mathbb{R})$. For any linear map $A : W \rightarrow Z$ between finite dimensional inner product spaces, we have $\|A^*A\|_F \leq \|A\|_F^2$. Thus, if $A : \mathbb{R}^k \rightarrow Z^1(K; \mathbb{R})$ is given by mapping the j th standard basis vector to θ_j , we get

$$\sum_{1 \leq i, j \leq k} \langle \theta_i, \theta_j \rangle^2 = \|A^*A\|_F \leq \|A\|_F^2 = \sum_{j=1}^k \|\theta_j\|^2.$$

This implies that a set of cocycles solving Problem 2 exactly or approximately (right-hand side) induces, by integration (Algorithm 5), a set of circle-valued maps with low pairwise squared discrete geometric correlation (left-hand side).

■ **Algorithm 1** The Circular Coordinates Algorithm.

Input: a non-trivial cohomology class $\alpha \in H^1(K; \mathbb{Z})$ and an inner product $\langle -, - \rangle$ on $Z^1(K; \mathbb{R})$

Output: a function $cc_\alpha : K_0 \rightarrow \mathbb{S}^1$

- 1: Let $\theta := \text{harmonicRepresentative}(\alpha, \langle -, - \rangle)$
- 2: Let $cc_\alpha := \text{integrate}_\theta$

■ **Algorithm 2** The Toroidal Coordinates Algorithm.

Input: l.i. cohomology classes $\alpha_1, \dots, \alpha_k \in H^1(K; \mathbb{Z})$ and inner product $\langle -, - \rangle$ on $Z^1(K; \mathbb{R})$

Output: a function $tc_\alpha : K_0 \rightarrow \mathbb{T}^k$

- 1: Let $\theta_1, \dots, \theta_k := \text{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle -, - \rangle)$
- 2: Let $tc_\alpha := (\text{integrate}_{\theta_1}, \dots, \text{integrate}_{\theta_k})$

■ **Algorithm 3** Harmonic representative.

Input: a non-trivial cohomology class $\alpha \in H^1(K; \mathbb{Z})$ and an inner product $\langle -, - \rangle$ on $Z^1(K; \mathbb{R})$

Output: a cocycle $\text{harmonicRepresentative}(\alpha, \langle -, - \rangle) \in Z^1(K; \mathbb{R})$

- 1: Let $\eta \in Z^1(K; \mathbb{Z})$ be such that $[\eta] = \alpha \in H^1(K; \mathbb{Z})$
- 2: Use least squares, w.r.t. $\langle -, - \rangle$, to solve $\tau = \text{argmin}\{ \|\iota(\eta) - \delta(\tau)\| \mid \tau : K_0 \rightarrow \mathbb{R} \}$
- 3: Let $\text{harmonicRepresentative}(\alpha, \langle -, - \rangle) := \iota(\eta) - \delta(\tau)$

■ **Algorithm 4** Low energy representatives.

Input: l.i. cohomology classes $\alpha_1, \dots, \alpha_k \in H^1(K; \mathbb{Z})$ and inner product $\langle -, - \rangle$ on $Z^1(K; \mathbb{R})$

Output: list of k cocycles $\text{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle -, - \rangle) \subseteq Z^1(K; \mathbb{R})$

- 1: Let $\eta_j := \text{harmonicRepresentative}(\alpha_j, \langle -, - \rangle)$ for $1 \leq j \leq k$
- 2: Compute the Cholesky decomposition $G = CC^*$ of $G \in \mathbb{R}^{k \times k}$ with $G_{ij} = \langle \eta_i, \eta_j \rangle$
- 3: Let $b_1, \dots, b_k := \text{LLL}(C_1, \dots, C_k)$, with C_j the j th row of C and LLL as in Section 3.3
- 4: Let $M \in \mathbb{Z}^{k \times k}$ be the change of basis matrix such that $MC = (b_1, \dots, b_k)^T$
- 5: Let $\text{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle -, - \rangle) := M (\eta_1, \dots, \eta_k)^T$

■ **Algorithm 5** Cocycle integration.

Input: a cocycle $\theta \in Z^1(K; \mathbb{R})$ such that $[\theta] \in \text{Im}(\iota : H^1(K; \mathbb{Z}) \rightarrow H^1(K; \mathbb{R}))$

Output: a function $\text{integrate}_\theta : K_0 \rightarrow \mathbb{S}^1$

- 1: Assume K is connected, otherwise do the following in each connected component
- 2: Choose $x \in K_0$ arbitrarily
- 3: **for** $y \in K_0$ **do**
- 4: Choose a path $x = y_0, y_1, \dots, y_{\ell-1}, y_\ell = y$ from x to y , arbitrarily
- 5: Let $\text{integrate}_\theta(y) := (\theta^{y_0 y_1} + \theta^{y_1 y_2} + \dots + \theta^{y_{\ell-2} y_{\ell-1}} + \theta^{y_{\ell-1} y_\ell}) \pmod{\mathbb{Z}}$

3.3 Minimizing the objective function with lattice reduction

We start by describing the specific lattice reduction problem we are interested in. Fix $k \in \mathbb{N}$ and a k -dimensional real vector space R with an inner product. A full-dimensional *lattice* L in R is a discrete subgroup $L \subseteq R$ which generates R as a real vector space. An ordered *basis* of a lattice $L \subseteq R$ consists of an ordered list $B = \{b_1, \dots, b_k\} \subseteq L$ of linearly independent vectors that generate L as an Abelian group. We are interested in the following problem.

► **Problem 6.** Let $L \subseteq R$ be a lattice. Find a basis B of L minimizing $\|B\|_F^2 = \sum_{i=1}^k \|b_i\|^2$.

We suspect that Problem 6 is in general hard to solve exactly or approximately up to a small multiplicative constant. Formally establishing that this problem is hard is beyond the scope of this work since hardness results for these kinds of problems – like [1] for the shortest vector problem – are usually quite involved; we refer the reader to [12, 22] for surveys. We note that minimizations like the one in Problem 6 have already been considered in the computational number theory literature, see, e.g., [2, Equation 38].

We content ourselves with the following result, which shows the Lenstra–Lenstra–Lovász lattice basis reduction algorithm (LLL-algorithm), a polynomial-time algorithm introduced in [15], provides an approximate solution to Problem 6. For our purposes, the LLL-algorithm takes as input linearly independent vectors $\{b_1, \dots, b_k\}$ in \mathbb{R}^k and returns a *reduced* basis, which we denote by $\text{LLL}(b_1, \dots, b_k)$. We shall not recall the definition of reduced basis here, since all we need to know about them is the following.

► **Lemma 7.** Let $L \subseteq \mathbb{R}^n$ and let V be a solution to Problem 6 for $L \subseteq \mathbb{R}^n$. If B is an *reduced basis*, then $\|B\|_F^2 \leq 2^{k-1} \|V\|_F^2$.

We prove Lemma 7 in [25, Appendix A.1], where we use it to prove Theorem 3. We conclude this section with a practical remark about the LLL-algorithm.

► **Remark 8.** Although the LLL-algorithm can be run with any input $\{b_1, \dots, b_n\} \subseteq L \subseteq \mathbb{R}^n$, it is guaranteed to terminate only if one uses infinite precision arithmetic. In [15], this is dealt with by assuming that the given lattice has rational coordinates, i.e., $L \subseteq \mathbb{Q}^n \subseteq \mathbb{R}^n$; see [15, Remark 1.38]. This is a reasonable assumption in our case, since we expect to be given the input cocycles and inner product with some finite precision.

In our implementation of the LLL-algorithm, we use floating-point arithmetic, for simplicity, and this did not present any problems to us. We note that floating-point algorithms with polynomial guarantees do exist in the case $L \subseteq \mathbb{Z}^n \subseteq \mathbb{R}^n$, see, e.g., [19].

4 Geometric Interpretation of the Toroidal Coordinates Algorithm

Let \mathcal{M} be a closed Riemannian manifold. We propose the following problem as a suitable objective for finding an efficient representation of \mathcal{M} which captures any chosen set of 1-dimensional holes of \mathcal{M} .

► **Problem 9.** Given linearly independent cohomology classes $\alpha_1, \dots, \alpha_k \in H^1(\mathcal{M}; \mathbb{Z})$, find a smooth map $f : \mathcal{M} \rightarrow \mathbb{T}^k$ of minimum Dirichlet energy, with the property that the induced morphism $f^* : H^1(\mathbb{T}^k; \mathbb{Z}) \rightarrow H^1(\mathcal{M}; \mathbb{Z})$ restricts to an isomorphism between $H^1(\mathbb{T}^k; \mathbb{Z}) \cong \mathbb{Z}^k$ and the subgroup of $H^1(\mathcal{M}; \mathbb{Z})$ generated by $\alpha_1, \dots, \alpha_k$.

In this section, we show that the above problem can be solved by an analogue of our Toroidal Coordinates Algorithm, thus providing a geometric interpretation of our algorithm. In Remark 13, at the end of this section, we explain how this interpretation motivates the notion of discrete geometric correlation of Definition 4.

First, we give the analogue of cocycle integration (Algorithm 5) for 1-forms.

► **Construction 10.** Given a closed 1-form $\theta \in \Omega^1(\mathcal{M})$ such that $[\theta] \in \text{Im}(\text{H}^1(\mathcal{M}; \mathbb{Z}) \rightarrow \text{H}^1(\mathcal{M}; \mathbb{R}))$, consider the following procedure, which returns a function $f : \mathcal{M} \rightarrow \mathbb{S}^1$.

1. Assume \mathcal{M} is connected, otherwise do the following in each connected component.
2. Choose $x \in \mathcal{M}$ arbitrarily.
3. For each $y \in \mathcal{M}$, let $p : [0, 1] \rightarrow \mathcal{M}$ be any smooth path from x to y .
4. For each $y \in \mathcal{M}$, define $f(y) = \left(\int_0^1 \theta_{p(t)}(p'(t)) dt \right) \bmod \mathbb{Z}$.

It is worth remarking that, although Construction 10 depends on arbitrary choices, all choices yield rotationally equivalent outputs.

The following procedure is the analogue of the Toroidal Coordinates Algorithm.

► **Construction 11.** Given linearly independent cohomology classes $\alpha_1, \dots, \alpha_k \in \text{H}^1(\mathcal{M}; \mathbb{Z})$, consider the following procedure, which returns a function $f : \mathcal{M} \rightarrow \mathbb{T}^k$.

1. Find closed $\theta_1, \dots, \theta_k \in \Omega^1(\mathcal{M})$ minimizing $\sum_{j=1}^k \|\theta_j\|^2$, with the property that the sets $\{[\theta_j]\}_{1 \leq j \leq k}$ and $\{\iota(\alpha_j)\}_{1 \leq j \leq k}$ generate the same Abelian subgroup of $\text{H}^1(\mathcal{M}; \mathbb{R})$.
2. Return $(f_1, \dots, f_k) : \mathcal{M} \rightarrow \mathbb{T}^k$, where f_j is obtained by integrating θ_j (Construction 10).

► **Proposition 12.** *Construction 11 returns a solution to Problem 9.*

A proof of Proposition 12 is in [25, Appendix A.2]. We conclude with a remark relating the Dirichlet form to our notion of discrete geometric correlation.

► **Remark 13.** Recall from the contributions section that we propose to measure geometric correlation between maps $f, g : \mathcal{M} \rightarrow \mathbb{S}^1$ using the Dirichlet form $D(f, g)$. On one hand, if f and g are obtained using Construction 10 with input 1-forms θ and η , respectively, then $D(f, g) = \frac{1}{2} \langle \theta, \eta \rangle_{\Omega^1}$, by [25, Lemma 19]. On the other hand, given a simplicial complex K with inner product $\langle -, - \rangle$ on $\text{Z}^1(K; \mathbb{R})$, and maps $f', g' : K_0 \rightarrow \mathbb{S}^1$ obtained using Algorithm 5 with inputs cocycles θ' and η' , respectively, we defined the discrete geometric correlation between f' and g' as $\langle \theta', \eta' \rangle$. In this sense, our notion of discrete geometric correlation is a discrete analogue of the Dirichlet form. Theorem 15 makes this analogy precise: when using Algorithm 6, there exists an inner product that exactly recovers the Dirichlet form.

5 The Sparse Toroidal Coordinates Algorithm

Although effective, the Circular Coordinates Algorithm has two practical drawbacks. First, the simplicial complex K is usually taken to be a Vietoris–Rips complex, and thus the cohomology computations scale with the number of data points. Second, the circle-valued representation returned by the algorithm is defined only on the input data and no representation is provided for out-of-sample data points. The sparse circular coordinates algorithm of [21] addresses these shortcomings. We now describe a version of the sparse circular coordinates algorithm² and recall the steps not included here when describing Pipeline 18 in the examples.

² We refer the reader to [21] for the full pipeline.

■ **Algorithm 6** Sparse cocycle integration.

Input: a finite open cover $\mathcal{U} = \{U_x\}_{x \in I}$ of a topological space B , a partition of unity $\Phi = \{\varphi_x\}_{x \in I}$ subordinate to \mathcal{U} , a simplicial complex $K \supseteq N(\mathcal{U})$, and a cocycle $\theta \in Z^1(K; \mathbb{R})$ such that $[\theta] \in \text{Im}(\iota : H^1(K; \mathbb{Z}) \rightarrow H^1(K; \mathbb{R}))$

Output: a function $\text{sparseIntegrate}_\theta^\Phi : B \rightarrow \mathbb{S}^1$

- 1: Assume K is connected, otherwise do the following in each connected component
- 2: Choose $x \in K_0$ arbitrarily
- 3: **for** $y \in K_0$ **do**
- 4: Choose a path $x = y_0, y_1, \dots, y_{\ell-1}, y_\ell = y$ from x to y , arbitrarily
- 5: Let $\tau_y := \theta^{y_0 y_1} + \theta^{y_1 y_2} + \dots + \theta^{y_{\ell-2} y_{\ell-1}} + \theta^{y_{\ell-1} y_\ell}$
- 6: Let $\text{sparseIntegrate}_\theta^\Phi(b) := (\tau_y + \sum_{z \in I} \varphi_z(b) \theta^{yz}) \pmod{\mathbb{Z}}$, where $b \in U_y$

■ **Algorithm 7** The Sparse Circular Coordinates Algorithm.

Input: a finite open cover $\mathcal{U} = \{U_x\}_{x \in I}$ of a topological space B , a partition of unity $\Phi = \{\varphi_x\}_{x \in I}$ subordinate to \mathcal{U} , a simplicial complex $K \supseteq N(\mathcal{U})$, a non-trivial cohomology class $\alpha \in H^1(K; \mathbb{Z})$, and an inner product $\langle -, - \rangle$ on $Z^1(K; \mathbb{R})$

Output: a function $\text{scc}_\alpha : B \rightarrow \mathbb{S}^1$

- 1: Let $\theta := \text{harmonicRepresentative}(\alpha, \langle -, - \rangle)$
- 2: Let $\text{cc}_\alpha := \text{sparseIntegrate}_\theta^\Phi$

■ **Algorithm 8** The Sparse Toroidal Coordinates Algorithm.

Input: a finite open cover $\mathcal{U} = \{U_x\}_{x \in I}$ of a topological space B , a partition of unity $\Phi = \{\varphi_x\}_{x \in I}$ subordinate to \mathcal{U} , a simplicial complex $K \supseteq N(\mathcal{U})$, i.e. cohomology classes $\alpha_1, \dots, \alpha_k \in H^1(K; \mathbb{Z})$, and inner product $\langle -, - \rangle$ on $Z^1(K; \mathbb{R})$

Output: a function $\text{stc}_\alpha : B \rightarrow \mathbb{T}^k$

- 1: Let $\theta_1, \dots, \theta_k := \text{lowEnergyRepresentatives}(\alpha_1, \dots, \alpha_k, \langle -, - \rangle)$
- 2: Let $\text{stc}_\alpha := (\text{sparseIntegrate}_{\theta_1}^\Phi, \dots, \text{sparseIntegrate}_{\theta_k}^\Phi)$

As in previous cases, we remark that, although the sparse cocycle integration subroutine (Algorithm 6) depends on arbitrary choices, all choices yield rotationally equivalent outputs.

We now show that, when B is a closed Riemannian manifold, there is a choice of inner product on cocycles that coincides with the Dirichlet form between the corresponding circle-valued maps, making the analogy in Remark 13 formal.

► **Definition 14.** Let $\mathcal{U} = \{U_x\}_{x \in I}$ be a finite open cover of a closed Riemannian manifold \mathcal{M} and let $\Phi = \{\varphi_x\}_{x \in I}$ be a smooth partition of unity subordinate to \mathcal{U} . Define the inner product $\langle -, - \rangle_D$ on $Z^1(N(\mathcal{U}); \mathbb{R})$ by

$$\langle \theta, \eta \rangle_D := \frac{1}{2} \sum_{w, y, z \in I} D_{wyz} \theta^{wy} \eta^{wz}, \quad \text{where } D_{wyz} := \int_{b \in \mathcal{M}} \langle d(\varphi_y)_b, d(\varphi_z)_b \rangle_F \varphi_w(b) \, d\mu(b).$$

Note that the quantities D_{wyz} do not depend on the cocycles.

► **Theorem 15.** Let $\mathcal{U} = \{U_x\}_{x \in I}$ be a finite open cover of a closed Riemannian manifold \mathcal{M} , let $K \supseteq N(\mathcal{U})$, and let $\Phi = \{\varphi_x\}_{x \in I}$ be a smooth partition of unity subordinate to \mathcal{U} . Assume $\theta, \eta \in Z^1(K; \mathbb{R})$ are such that $[\theta], [\eta] \in H^1(K; \mathbb{R})$ are in the image of $\iota : H^1(K; \mathbb{Z}) \rightarrow H^1(K; \mathbb{R})$. Let $f = \text{sparseIntegrate}_\theta^\Phi : \mathcal{M} \rightarrow \mathbb{S}^1$ and $g = \text{sparseIntegrate}_\eta^\Phi$. Then, f and g are smooth and $D(f, g) = \langle \theta, \eta \rangle_D$.

We prove Theorem 15 in [25, Appendix A.3]. We conclude by giving a heuristic for computing an estimate $\langle -, - \rangle_{\widehat{D}}$ of $\langle -, - \rangle_D$. Addressing the consistency of this heuristic is left for future work.

► **Construction 16.** Let $X \subseteq \mathcal{M} \subseteq \mathbb{R}^n$ be a finite sample of a smoothly embedded closed manifold. Assume given a subsample $I \subseteq X$ as well as $\varepsilon > 0$ such that $\mathcal{M} \subseteq \bigcup_{x \in I} B(x, \varepsilon)$. For $w, y, z \in I$, we seek to estimate D_{wyz} , where the open cover is taken to be $\mathcal{U} = \{B(x, \varepsilon) \cap \mathcal{M}\}_{x \in I}$ and $\Phi = \{\varphi_x\}_{x \in I}$ is a smooth partition of unity subordinate to \mathcal{U} .

1. Form a neighborhood graph G on $X_w := X \cap B(w, \varepsilon)$. For instance, this can be done by selecting $k \in \mathbb{N}$ and using an undirected k -nearest neighbor graph.
2. Compute weights $h(a, b) \geq 0$ for the edges $(a, b) \in G$. For instance, this can be done by selecting a radius $\delta > 0$ and letting $h(a, b) = \exp(-\|a - b\|^2 / \delta^2)$.
3. For $a \in X_w$, let $N(a) = \{b \in G \mid (a, b) \in G\}$, and define

$$\widehat{D}_{wyz} = \sum_{a \in G} \left(\frac{1}{N(a)} \sum_{b \in N(a)} h(a, b) (\varphi_y(b) - \varphi_y(a)) (\varphi_z(b) - \varphi_z(a)) \right) \varphi_w(a).$$

► **Remark 17.** We have implemented the estimated inner product $\langle -, - \rangle_{\widehat{D}}$ in [24]. In all examples we have considered, running the algorithms in this paper with inner product $\langle -, - \rangle_{\widehat{D}}$ on one hand, and with the de Silva, Morozov, and Vejdemo-Johansson inner product $\langle -, - \rangle_{\text{dSMV}}$ (Equation (1)) on the other, gives results that are essentially indistinguishable. For this reason, and for concreteness, in Section 6 we use $\langle -, - \rangle_{\text{dSMV}}$. We leave the question of when and why the two inner products give such similar results for future work.

6 Examples

We compare the output of the Sparse Circular Coordinates Algorithm [21] run independently on several cohomology classes with that of the Sparse Toroidal Coordinates Algorithm. We use the DREiMac [27] implementation of the Sparse Circular Coordinates Algorithm and our extension implementing the Sparse Toroidal Coordinates Algorithm. The code together with Jupyter notebooks replicating the examples here can be found at [24].

The examples include a synthetic genus two surface (Sec. 6.1), a dataset from [14] of two figurines rotating at different speeds (Sec. 6.2), a solution set of the Kuramoto–Sivashinsky equation obtained with Mathematica [29] (Sec. 6.3), and a synthetic dataset modeling neurons tuned to head movement of bats (Sec. 6.4). We use the following pipeline.

- **Pipeline 18.** Assume we are given a point cloud $X \subseteq \mathbb{R}^n$.
1. Compute a subsample $I \subseteq X$ using maxmin sampling (see [5, 8, 21]).
 2. Fix a large prime p ; we take $p = 41$.
 3. Compute Vietoris–Rips persistent cohomology of I in degree 1 with coefficients in $\mathbb{Z}/p\mathbb{Z}$.
 4. Looking at the persistence diagram, identify a filtration step $\varepsilon > 0$ at which cohomology classes $\beta_1, \dots, \beta_k \in H^1(\text{VR}_\varepsilon(I), \mathbb{Z}/p\mathbb{Z})$ of interest to the user are alive. Do this in such a way that $X \subseteq \bigcup_{x \in I} B(x, \varepsilon/2)$.
 5. Note that $\mathcal{U} = \{B(x, \varepsilon/2)\}_{x \in I}$ covers X and define $K := \text{VR}_\varepsilon(I) \supseteq N(\mathcal{U})$.

6. Lift $\beta_1, \dots, \beta_k \in H^1(K, \mathbb{Z}/p\mathbb{Z})$ to classes $\alpha_1, \dots, \alpha_k \in H^1(K, \mathbb{Z})$ (see [3, Section 2.4]).
7. Choose a partition of unity subordinate to \mathcal{U} (see [21, Section 4]). We use the inner product $\langle -, - \rangle_{\text{dSMV}}$ on cocycles (as explained in Remark 17).
8. On one hand, run the Sparse Circular Coordinates Algorithm (Algorithm 7) on each class α_j separately, and get k circle-valued maps $X \rightarrow \mathbb{S}^1$.
9. On the other hand, run the Sparse Toroidal Coordinates Algorithm (Algorithm 8) on all classes $\alpha_1, \dots, \alpha_k$ simultaneously, to again get k circle-valued maps $X \rightarrow \mathbb{S}^1$.

In order to show that the Sparse Toroidal Coordinates Algorithm returns coordinates with lower correlation and energy, we quantify the performance of the two algorithms using the estimated Dirichlet correlation matrix (see [25, Appendix B]) of the circle-valued functions obtained from them. When the functions are obtained from the Sparse Circular Coordinates Algorithm (resp. Sparse Toroidal Coordinates Algorithm), we denote the correlation matrix by D_{SCC} (resp. D_{STC}). Note that diagonal correlation matrices reflect complete independence of coordinates. Hence, we interpret correlations matrices that are close to being diagonal as indicating low correlation and high independence of recovered coordinates.

The correlation computations depend on two parameters (a k for a k -nearest neighbor graph and a choice of edge weights). We use $k = 15$ and weights related to the scale of the data, but note that the results are robust with the respect to these choices.

We also display the change of basis matrix M (as in Algorithm 4) that relates the torus-valued maps output by the two algorithms.

6.1 Genus two surface

We apply Pipeline 18 on a densely sampled surface of genus two (Figure 1), as in [3, Section 3.9]. As expected, persistent cohomology returns four high persistence features. The resulting circular coordinates obtained by applying the Sparse Toroidal Coordinates Algorithm are shown in Figure 2 (Right). For comparison, we show the circular coordinates obtained by applying the Sparse Circular Coordinates Algorithm to each cohomology class separately Figure 2 (Left). The Dirichlet correlation matrices are as follows:

$$D_{SCC} = \begin{pmatrix} 3 & 2.4 & 0 & -2.4 \\ 2.4 & 4.8 & 0 & -4.8 \\ 0 & 0 & 22.6 & 11.2 \\ -2.4 & -4.8 & 11.2 & 19.5 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 3 & -0.6 & 0 & 0 \\ -0.6 & 3 & 0 & 0 \\ 0 & 0 & 14.9 & 3.5 \\ 0 & 0 & 3.5 & 14.8 \end{pmatrix}.$$

6.2 Lederman–Talmon dataset

We run Pipeline 18 on a dataset collected and studied by Lederman and Talmon in [14]. In this example, two figurines *Yoda* (the green figure on the left) and *Dog* (the bulldog figure on the right) are situated on rotating platforms; see Figure 4.

Since each image is characterized by a rotation (φ_1, φ_2) of the two figurines, we interpret the time series of images as an observation of a dynamical system on a two-torus. Because the frequencies of rotation of both figurines have a large least-common-multiple, we expect the set of toroidal angles (φ_1, φ_2) to comprise a dense sample of the torus, and verify this by treating the temporal sequence of images as a vector-valued time series and compute its sliding window persistence with window length $d = 4$ and time delay $\tau = 1$ (see [25, Appendix C.1]).

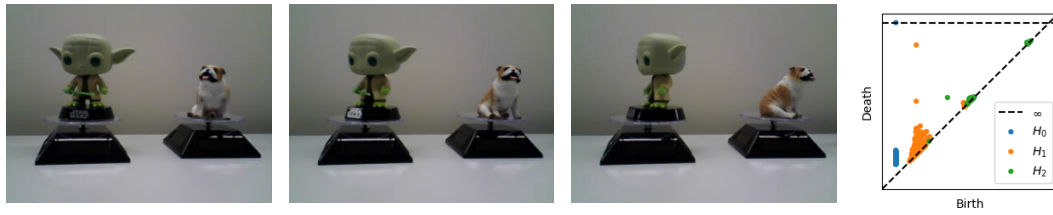


Figure 4 *Left:* A sample of different images in the dataset. *Right:* The sliding window persistence diagram of the data, showing two prominent 1-dimensional cohomology classes. *Yoda's* platform rotates clockwise completing about 310 cycles during the experiment, while in the same time *Dog's* platform completes about 450 cycles rotating counterclockwise. The data we consider are a collection of images of these rotating platforms captured from a fixed viewpoint.

In Figures 5 and 6, we display the result of applying the Sparse Circular Coordinates Algorithm and the Sparse Toroidal Coordinates Algorithm to the sliding window point cloud of the dataset, respectively. Here, we show a sample of images as parameterized by the toroidal coordinates obtained from both algorithms. The Dirichlet correlation matrices and change of basis matrix are as follows:

$$D_{SCC} = \begin{pmatrix} 3.07 & -3.08 \\ -3.08 & 10.48 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 3.07 & 0 \\ 0 & 7.39 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}.$$

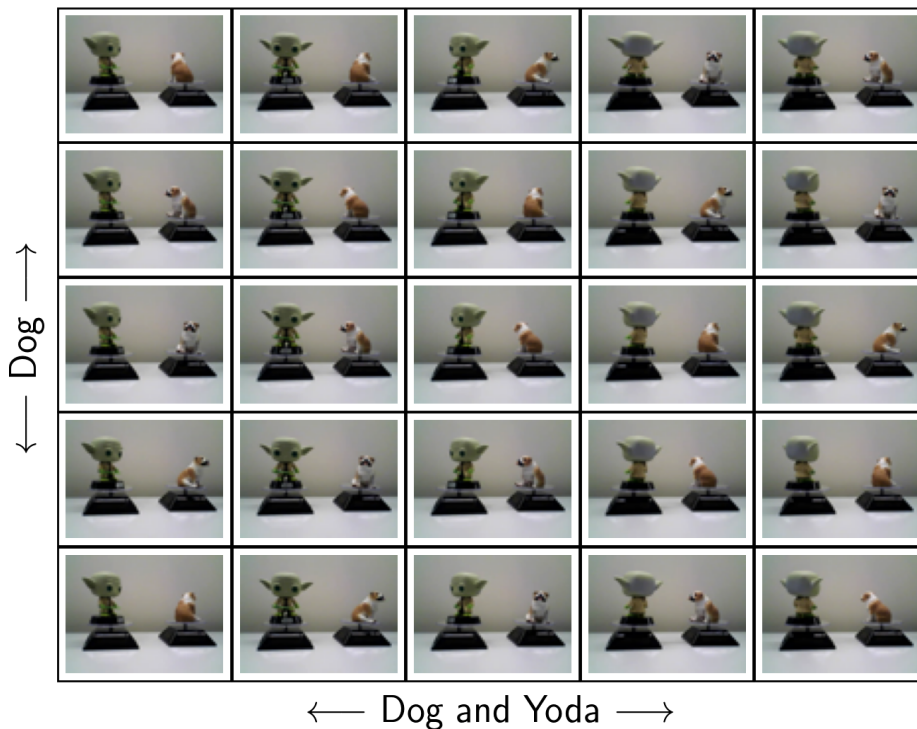
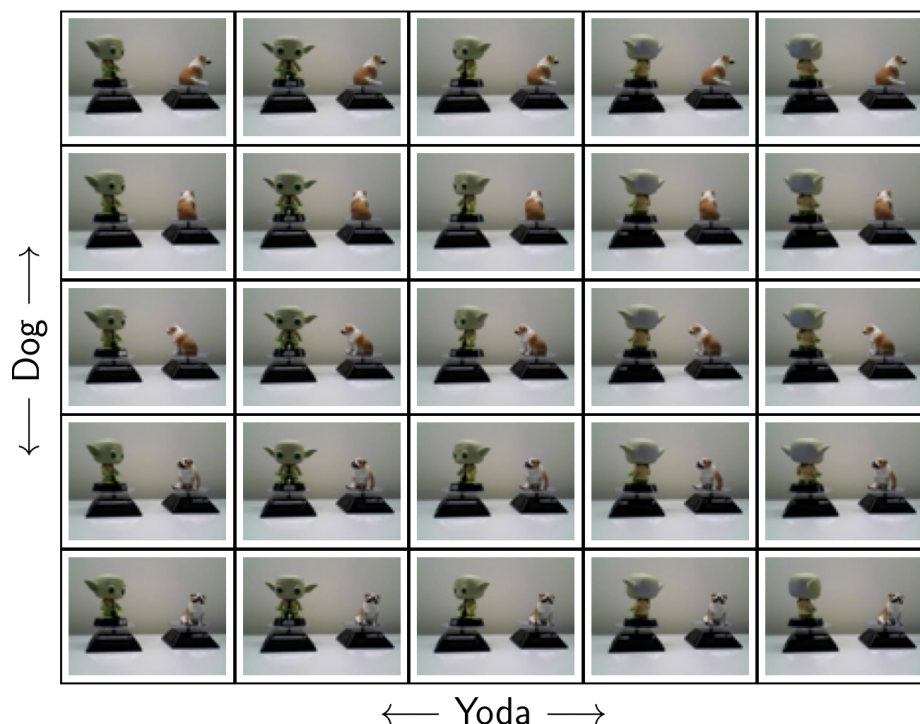


Figure 5 The vertical coordinate parameterizes *Dog's* rotation, while the horizontal coordinate parameterizes the rotation of both figurines.



■ **Figure 6** Vertical coordinate parameterizes *Dog*'s rotation; horizontal parameterizes *Yoda*'s.

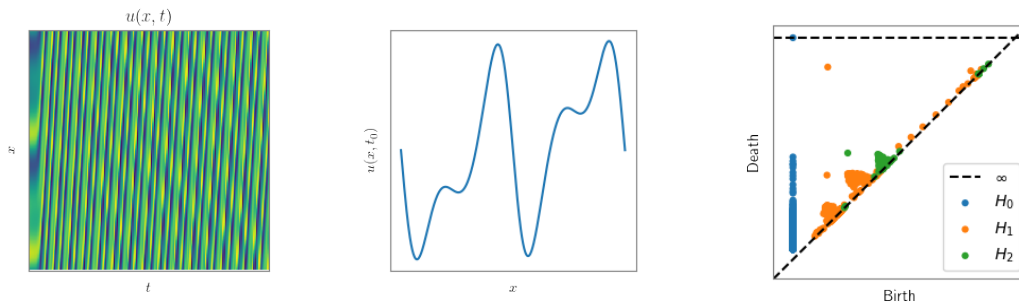
6.3 Kuramoto–Sivashinsky Dynamical Systems

An example of a one-dimensional Kuramoto–Sivashinsky (KS) equation is the following fourth order partial differential equation:

$$\frac{\partial u(x,t)}{\partial t} + 4 \frac{\partial^4 u(x,t)}{\partial x^4} + 53.3 \frac{\partial^2 u(x,t)}{\partial x^2} + 53.3 u(x,t) \frac{\partial u(x,t)}{\partial x} = 0$$

with periodic boundary conditions $u(x,0) = \sin(x)$ and $u(0,t) = u(2\pi,t)$. The general family of KS equations [9] have gained popularity from their simple appearance and their ability to produce chaotic spatiotemporal dynamics. They have been shown to model pattern formations in several physical contexts; for instance [13, 17, 26].

The underlying dynamical system is toroidal. Indeed, it is controlled by two frequencies: one comes from oscillation in time, the other is dictated by the speed of the traveling wave along the periodic domain. However, the dynamic is not periodic and the trajectory of any initial state eventually densely fills out the torus. We represent the solution $u(x,t)$ to this equation as a heatmap in Figure 7 (Left). The horizontal axis refers to time t , the vertical axis refers to the spatial variable x . At each time, $u(x,t)$ is periodic and a slice of it can be seen in Figure 7 (Middle). We treat $u(x,t)$ as a vector valued time series $f(t) := u(-,t)$ and compute the sliding window persistence ([25, Appendix C.1]) of f with parameters $d = 5$ and $\tau = 4$; see Figure 7 (Right) for the resulting persistence diagram.

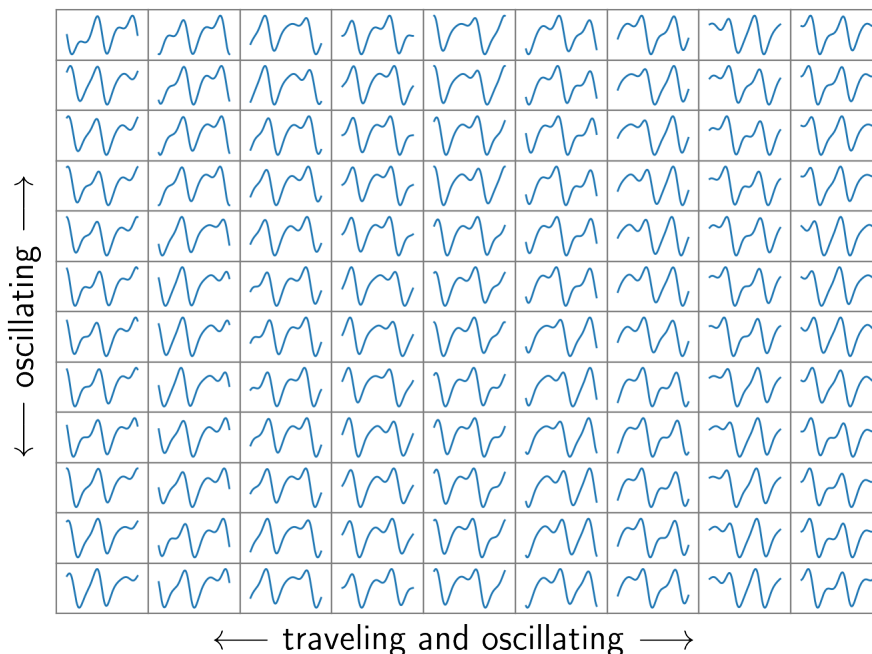


■ **Figure 7** (Left) The solution to the KS equation as a heatmap; (Middle) A slice of $u(x, t)$ at a fixed t_0 ; (Right) The sliding window persistence digram.

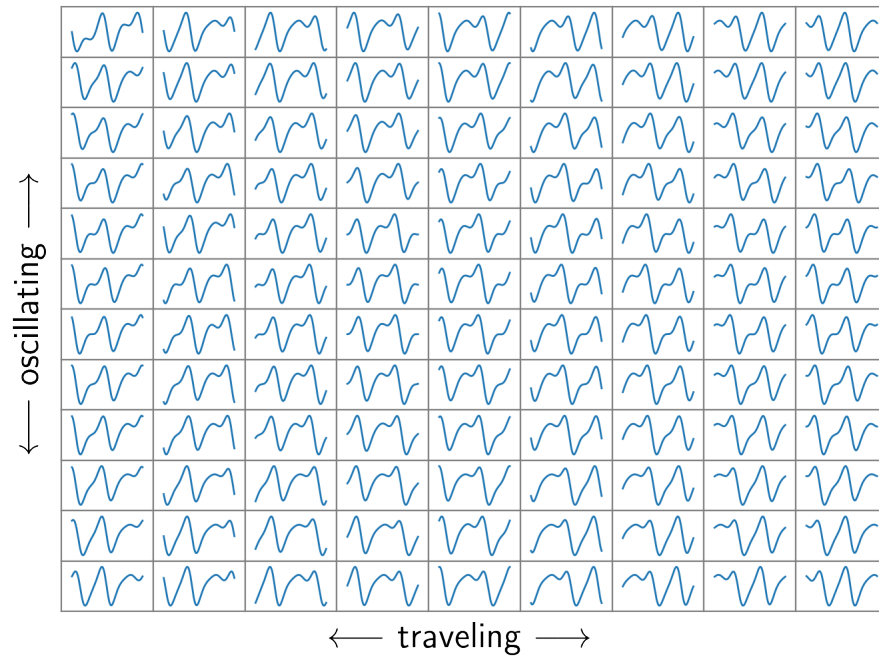
In Figures 8 and 9, we display the result of applying the Sparse Circular Coordinates Algorithm and the Sparse Toroidal Coordinates Algorithm to the sliding window embedding of the dataset, respectively. As in Example 6.2, we show sample data points (in this case waves) as parameterized by the toroidal coordinates obtained from both algorithms.

To verify that the vertical and horizontal components of Figure 9 are indeed parameterizing oscillation and traveling, respectively, we partition the dataset in 50 bins, according to the vertical coordinate, and in each bin we compute all pairwise rotationally invariant L^2 distances between the waves. We show the histogram of distances in Figure 10. The Dirichlet correlation matrices and change of basis are as follows:

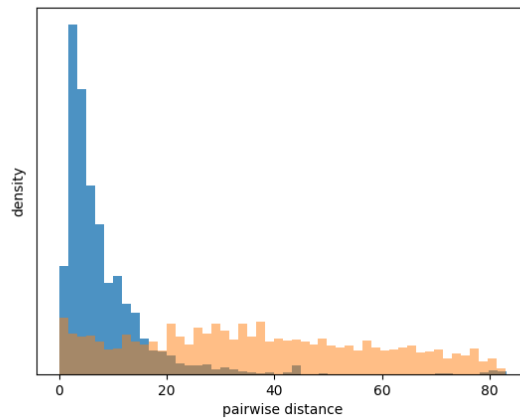
$$D_{SCC} = \begin{pmatrix} 1.1 & 6.1 \\ 6.1 & 112.3 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 1.1 & 1 \\ 1 & 76.6 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 \\ -5 & 1 \end{pmatrix}.$$



■ **Figure 8** Oscillatory behavior vertically and combination of traveling and oscillatory horizontally.



■ **Figure 9** Traveling waves parameterized horizontally and oscillations parameterized vertically.



■ **Figure 10** Density of pairwise distances using the parameterization of the Toroidal Coordinates Algorithm is in blue, while density of pairwise distances using the parameterization given by the Circular Coordinates Algorithm is in red. This suggests that, indeed, the horizontal coordinate of Figure 9 parameterizes rotation only while the one of Figure 8 does not.

6.4 Synthetic Neuroscience Example

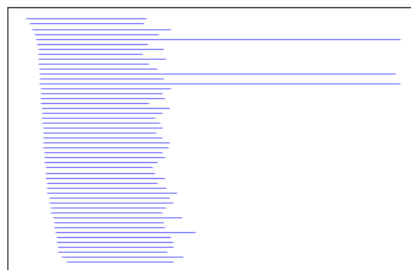
We show that our methods are a viable way of constructing informative circle-valued representations of neuroscientific data. Place cells in the mammalian hippocampus have spatially localized receptive fields that encode position by firing rapidly at specific locations as one navigates an environment [20]. It is shown in [6] that head direction of bats is encoded in a similar way: certain neurons are tuned to pitch, others to azimuth, and others to roll, each using a circular coordinate system to do so.

We consider a synthetic dataset inspired by these types of neuronal responses to stimuli. Suppose three populations of neurons P_1 , P_2 , and P_3 are tuned to elevation, azimuth, and roll, respectively. This means, for instance, that if neuron $n \in P_1$ is tuned to a head elevation of 45 degrees, then n fires most rapidly when the head is at an elevation of 45 degrees, fires less rapidly if the head is at an elevation of, say 35 or 55 degrees, and maintains low activity near an elevation of 0 or 90 degrees. Now, suppose we record the firing rates of neurons in populations P_1 , P_2 , and P_3 for a duration of T time steps while an animal moves its head freely. Letting N denote the total number of recorded neurons, we can record this data as an $N \times T$ matrix M , where $M_{i,j}$ corresponds to the firing rate of neuron i at time step j .

We interpret M as a collection of T points in \mathbb{R}^n and consider the problem of recovering three circular coordinates $M \rightarrow \mathbb{S}^1$ (one each for elevation, azimuth, and roll) that map a point of M , thought of as a time step, to the correct head orientation at that time. We construct a synthetic dataset simulating the situation above (see [25, Appendix C.2]) and run it through Pipeline 18. Figure 11 shows the resulting persistence barcode, which contains three prominent 1-dimensional cohomology classes.

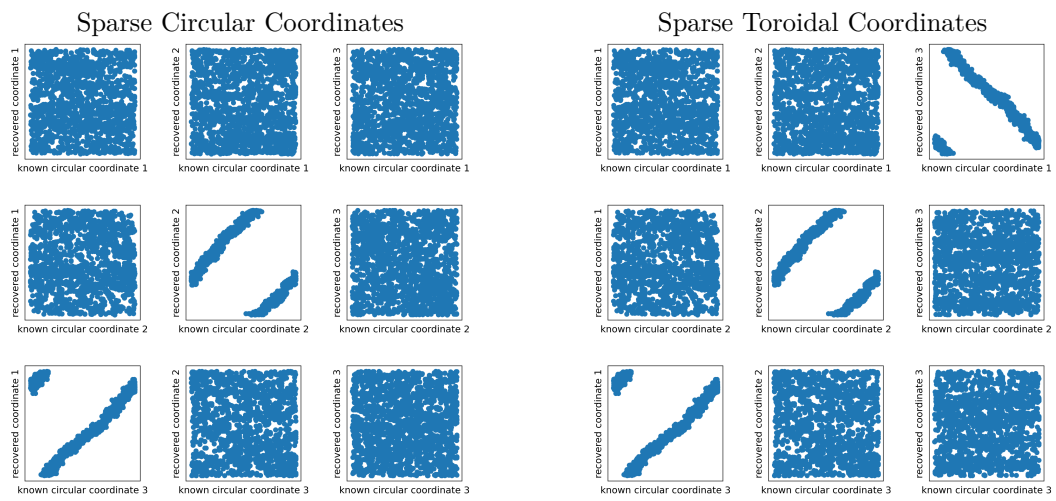
We then use the Sparse Circular Coordinates Algorithm to recover a circle-valued map from each of the three most prominent 1-dimensional cohomology classes. To determine whether we successfully recover all three orientations, we use a scatter plot as in [3, Section 3.2], and display the three recovered maps $M \rightarrow \mathbb{S}^1$ against the ground truth. The Sparse Circular Coordinates Algorithm run independently on each cohomology class fails to recover the three head orientations (Figure 12, left) in many of our runs, while the Toroidal Coordinates Algorithm always recovers the three coordinates (Figure 12, right). The Dirichlet correlation matrices and the change of basis are as follows:

$$D_{SCC} = \begin{pmatrix} 51.6 & -0.9 & 2.7 \\ -0.9 & 50.2 & -99.7 \\ 2.7 & -99.7 & 249.8 \end{pmatrix}, \quad D_{STC} = \begin{pmatrix} 51.6 & -0.9 & 0.8 \\ -0.9 & 50.2 & 0.5 \\ 0.8 & 0.5 & 51.1 \end{pmatrix}, \quad M = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}.$$



■ **Figure 11** Persistence barcode of exhibiting three prominent 1-dimensional cohomology classes.

57:18 Toroidal Coordinates: Decorrelating Circular Coordinates with Lattice Reduction



■ **Figure 12** Recovered versus known circular coordinates using the Sparse Circular Coordinates Algorithm and the Toroidal Coordinates Algorithm.

References

- 1 Miklós Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, STOC '98, pages 10–19, New York, NY, USA, 1998. Association for Computing Machinery. doi:10.1145/276698.276705.
- 2 P. Dayal and M.K. Varanasi. An algebraic family of complex lattices for fading channels with application to space-time codes. *IEEE Transactions on Information Theory*, 51(12):4184–4202, 2005. doi:10.1109/TIT.2005.858923.
- 3 Vin de Silva, Dmitriy Morozov, and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. *Discrete Comput. Geom.*, 45(4):737–759, 2011. doi:10.1007/s00454-011-9344-x.
- 4 Vin De Silva and Mikael Vejdemo-Johansson. Persistent cohomology and circular coordinates. In *Proceedings of the twenty-fifth annual symposium on Computational geometry*, pages 227–236, 2009.
- 5 M.E Dyer and A.M Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985. doi:10.1016/0167-6377(85)90002-1.
- 6 Arseny Finkelstein, Dori Derdikman, Alon Rubin, Jakob N. Foerster, Liora Las, and Nachum Ulanovsky. Three-dimensional head-direction coding in the bat brain. *Nature*, 517(7533):159–164, 2015. doi:10.1038/nature14031.
- 7 Richard J Gardner, Erik Hermansen, Marius Pachitariu, Yoram Burak, Nils A Baas, Benjamin A Dunn, May-Britt Moser, and Edvard I Moser. Toroidal topology of population activity in grid cells. *Nature*, 602(7895):123–128, 2022.
- 8 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.
- 9 James M Hyman and Basil Nicolaenko. The Kuramoto-Sivashinsky equation: a bridge between PDE's and dynamical systems. *Physica D: Nonlinear Phenomena*, 18(1-3):113–126, 1986.
- 10 Jürgen Jost. *Riemannian geometry and geometric analysis*. Universitext. Springer, Cham, seventh edition, 2017. doi:10.1007/978-3-319-61860-9.
- 11 Louis Kang, Boyan Xu, and Dmitriy Morozov. Evaluating state space discovery by persistent cohomology in the spatial representation system. *Frontiers in computational neuroscience*, 15:28, 2021.
- 12 Subhash Khot. Inapproximability results for computational problems on lattices. In *The LLL Algorithm*, pages 453–473. Springer, 2009.
- 13 Yoshiki Kuramoto and Toshio Tsuzuki. Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Progress of theoretical physics*, 55(2):356–369, 1976.
- 14 Roy R. Lederman and Ronen Talmon. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3):509–536, 2018. doi:10.1016/j.acha.2015.09.002.
- 15 A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261(4):515–534, 1982. doi:10.1007/BF01457454.
- 16 Hengrui Luo, Jisu Kim, Alice Patania, and Mikael Vejdemo-Johansson. Topological learning for motion data via mixed coordinates. In *2021 IEEE International Conference on Big Data (Big Data)*, pages 3853–3859, 2021. doi:10.1109/BigData52589.2021.9671525.
- 17 Daniel M Michelson and Gregory I Sivashinsky. Nonlinear analysis of hydrodynamic instability in laminar flames—II. numerical experiments. *Acta astronautica*, 4(11-12):1207–1221, 1977.
- 18 James R. Munkres. *Elements of algebraic topology*. Addison-Wesley Publishing Company, Menlo Park, CA, 1984.
- 19 Phong Q Nguyễn and Damien Stehlé. Floating-point LLL revisited. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 215–233. Springer, 2005.

- 20 John O’Keefe. Place units in the hippocampus of the freely moving rat. *Experimental Neurology*, 51(1):78–109, 1976. doi:10.1016/0014-4886(76)90055-8.
- 21 Jose A Perea. Sparse circular coordinates via principal \mathbb{Z} -bundles. In *Topological Data Analysis*, pages 435–458. Springer, 2020.
- 22 Oded Regev. On the complexity of lattice problems with polynomial approximation factors. In *The LLL algorithm*, pages 475–496. Springer, 2009.
- 23 Erik Rybakken, Nils Baas, and Benjamin Dunn. Decoding of neural data using cohomological feature extraction. *Neural computation*, 31(1):68–93, 2019.
- 24 L. Scoccola, H. Gakhar, J. Bush, N. Schonsheck, T. Rask, L. Zhou, and J. A. Perea. Sparse Toroidal Coordinates. <https://github.com/LuisScoccola/DREiMac>, 2022.
- 25 Luis Scoccola, Hitesh Gakhar, Johnathan Bush, Nikolas Schonsheck, Tatum Rask, Ling Zhou, and Jose A Perea. Toroidal coordinates: Decorrelating circular coordinates with lattice reduction. *arXiv preprint arXiv:2212.07201*, 2022.
- 26 Gregory I Sivashinsky and DM Michelson. On irregular wavy flow of a liquid film down a vertical plane. *Progress of theoretical physics*, 63(6):2112–2114, 1980.
- 27 Christopher Tralie, Tom Mease, and Jose Perea. DREiMac: Dimension Reduction with Eilenberg–MacLane Coordinates. <https://github.com/ctralie/DREiMac>, 2021.
- 28 Bei Wang, Brian Summa, Valerio Pascucci, and Mikael Vejdemo-Johansson. Branching and circular features in high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1902–1911, 2011.
- 29 Wolfram Research Inc. Mathematica, Version 13.1. Champaign, IL, 2022. URL: <https://www.wolfram.com/mathematica>.

Topological Universality of the Art Gallery Problem

Jack Stade 

Department of Mathematics, University of Cambridge, UK

Jamie Tucker-Foltz   

School of Engineering and Applied Sciences, Harvard University, Boston, MA, USA

Abstract

We prove that any compact semi-algebraic set is homeomorphic to the solution space of some art gallery problem. Previous works have established similar universality theorems, but holding only up to homotopy equivalence, rather than homeomorphism, and prior to this work, the existence of art galleries even for simple spaces such as the Möbius strip or the three-holed torus were unknown. Our construction relies on an elegant and versatile gadget to copy guard positions with minimal overhead. It is simpler than previous constructions, consisting of a single rectangular room with convex slits cut out from the edges. We show that both the orientable and non-orientable surfaces of genus n admit galleries with only $O(n)$ vertices.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases Art gallery, Homeomorphism, Exists-R, ETR, Semi-algebraic sets, Universality

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.58

Funding *Jamie Tucker-Foltz*: This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE1745303. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Acknowledgements We are very grateful to Simon Weber for his helpful, detailed feedback on an earlier version of this paper. We are also grateful to Tillmann Miltzow for numerous helpful suggestions.

1 Introduction

An instance of the *art gallery problem* consists of a polygon P (which we refer to as the *art gallery*), and the objective is to find a finite set of points $G \subseteq P$ (the *guards*) of minimal cardinality such that every point in P is *visible* to some guard, meaning that the line segment between the point and the guard is contained within P . This problem was first introduced by Viktor Klee in 1973 and has a long history. In 1986, Lee and Lin [8] showed that the decision problem of determining whether there exists a configuration with at most k guards is NP-hard, but the problem is not known to be in NP since there is no obvious succinct way to represent the guards' coordinates in a solution (they might have to be irrational, even if all polygon vertex coordinates are rational [10]). In 2018, Abrahamsen, Adamaszek, and Miltzow [1] showed that the problem is $\exists\mathbb{R}$ -complete, mostly settling the question of complexity (modulo the longstanding conjectures that $\text{NP} \subsetneq \exists\mathbb{R} \subsetneq \text{PSPACE}$). Approximation algorithms and lower bounds have also been studied; see Bonnet and Miltzow [4] for a recent overview.

Aside from complexity aspects, a parallel line of inquiry concerns the topology of the space of solutions. Supposing that gallery P requires exactly k guards at minimum, we let

$$V(P) := \{G \subseteq P \mid |G| = k \text{ and every } p \in P \text{ is visible to some } g \in G\}.$$

The set $V(P)$ consists of unordered sets of points of cardinality k , which can be turned into a metric space using the Hausdorff distance

$$d_H(G_0, G_1) := \max_{i \in \{0,1\}} \max_{g \in G_i} \min_{g' \in G_{1-i}} d(g, g'),$$

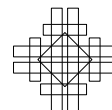


© Jack Stade and Jamie Tucker-Foltz;
licensed under Creative Commons License CC-BY 4.0
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 58; pp. 58:1–58:13



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



where d denotes the Euclidean distance on \mathbb{R}^2 . Thus, for any art gallery P , $V(P)$ is a topological space, so a natural question is, what kinds of topological spaces can occur?

More than a mere mathematical curiosity, this question is relevant to complexity theory due to the connections between $\exists\mathbb{R}$ -hardness and topological universality. The most famous example is Mnëv's theorem [11, 12] that any semi-algebraic set is stably-equivalent to the space of point configurations of some order type, inspiring an eventual proof that order type realizability is $\exists\mathbb{R}$ -complete [13]. A *semi-algebraic* set is a finite union $\bigcup_{i=1}^m S_i$, where each S_i is defined to be the set of points $(x_1, x_2, \dots, x_n) \in \mathbb{R}^n$ satisfying a finite number of constraints of the form $P(x_1, x_2, \dots, x_n) \geq 0$ or $P(x_1, x_2, \dots, x_n) > 0$, where P is a polynomial. The canonical complete problem for $\exists\mathbb{R}$ is called ETR (Existential Theory of the Reals), which asks whether a given semi-algebraic set is nonempty. As a consequence of their reduction from ETR to the art gallery problem, Abrahamsen et al. [1] show, for any compact semi-algebraic set S , how to construct an art gallery P such that $V(P)$ surjects continuously onto S . However, they do not show that the mapping is injective, so this fails to establish universality.

In a recent paper, Bertschinger, El Maalouly, Miltzow, Schnider, and Weber [2] show that any compact semi-algebraic set S is homotopy-equivalent to $V(P)$ for some polygon P . They leave as an open question whether P can be constructed so that $V(P)$ is not just homotopy-equivalent, but homeomorphic to S . Only the following list of spaces are shown to be captured up to homeomorphism:

- k -clover (obtained by joining k circles at a single point)
- k -chain (obtained by connecting k circles in a path with $k - 1$ disjoint line segments)
- $4k$ -necklace (obtained by connecting $4k$ circles in a cycle with $4k$ disjoint line segments)
- k -sphere
- The torus
- The 2-holed torus

The constructions for these spaces are all based on simple galleries with solution spaces homeomorphic to a circle or an interval, which are combined to give Cartesian products and then given simple additional constraints. However, using these methods it is difficult to obtain more general spaces because the geometry significantly limits the types of constraints that can be used. Thus, prior to this work, homeomorphism universality was unknown even for closed surfaces. Galleries for the real projective plane, Klein bottle, and Möbius strip were explicitly left as open questions.

In this work, we settle the question of homeomorphism universality in the affirmative:

► **Theorem 1.** *For every compact semi-algebraic set $S \neq \emptyset$, there exists a polygon P such that S is homeomorphic to $V(P)$.*

In addition to yielding a strictly stronger result than that of Bertschinger et al. [2], our construction is structurally simpler: the art gallery always consists of a single rectangular room with two types of convex slits repeatedly cut out of the edges. As an example, we explicitly draw an art gallery for the Möbius strip (to the best resolution that can reasonably fit on a page) in Figure 6.

The key ingredient in our approach is a novel form of *copying gadget* which enforces a constraint of the form $x_i = x_j$, thereby requiring multiple guards to represent the same underlying variable. Gadgets with similar functions can also be found in Abrahamsen et al. [1]. Our gadget takes a geometrically different form, and is an improvement for the following reasons:

- It is more versatile. In particular the polygon P does not need to intersect the convex hull of the segments being copied. This makes it possible for the gallery to only have a single “room.”
- It works in more general contexts. The previous gadget uses constraints created by the need for guards to see every point on the interior of the polygon P . Our gadget works in the variant of the problem where guards only need to see the boundary of the polygon. Stade [14] uses this gadget in a forthcoming paper proving that this variant is also $\exists\mathbb{R}$ -hard.

Our general construction, which we present in Section 2, does not yield an obvious bound on the size of P (number of vertices) as a function of the complexity of the space S . In Section 3, we refine our construction to show that both the orientable and non-orientable surfaces of genus n can be captured by art galleries with only $O(n)$ vertices.

2 General construction

As in Bertschinger et al. [2], the starting point for our reduction is a theorem by Hironaka [6] that any compact semi-algebraic set can be triangulated as a cubical complex. We begin by reviewing this result. Next, we introduce our variable/copying gadgets and establish its key properties. Finally, we show how to combine the gadgets to construct the art gallery. We illustrate our construction using the Möbius strip as a running example.

2.1 Hironaka’s theorem

An *abstract cubical complex* is a subset K of the set

$$I^n = \{ \{ \mathbf{x} \in [0, 1]^n \mid x_{i_1} = c_1, \dots, x_{i_k} = c_k \} \mid 1 \leq i_1 < \dots < i_k \leq n, (c_1, \dots, c_k) \in \{0, 1\}^k \}$$

of faces of an n dimensional hypercube $[0, 1]^n$ such that if $a \in K$ and $b \subseteq a$, then $b \in K$. We write $|K| := \bigcup K \subseteq [0, 1]^n$ for the union of faces, called the *geometric realization* of K .

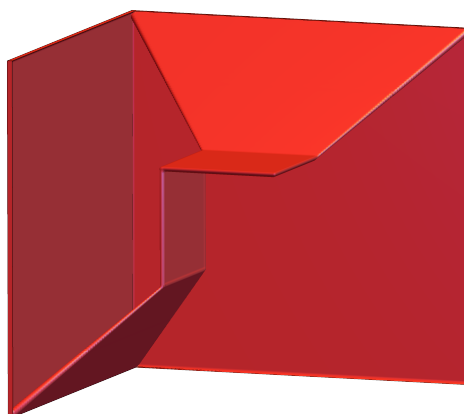
► **Theorem 2** (Bertschinger et al. [2], Lemma 3). *Any compact semi-algebraic set is homeomorphic to the geometric realization of an abstract cubical complex.*

Proof sketch. This fact is well-known so we do not give a complete proof here. Using Hironaka’s theorem [6], it is possible to show that any compact semi-algebraic set is homeomorphic to the geometric realization of an abstract simplicial complex (see Hoffman [7]). Additionally, it can be shown that any abstract simplicial complex has an abstract cubical complex with a homeomorphic geometric realization (see Blass and Włodzimierz [3, Theorem 1.1]). ◀

For example, the circle S^1 can be represented as the abstract cubical complex

$$\{ \{ \mathbf{x} \in [0, 1]^2 \mid x_1 = 0 \}, \{ \mathbf{x} \in [0, 1]^2 \mid x_1 = 1 \}, \{ \mathbf{x} \in [0, 1]^2 \mid x_2 = 0 \}, \{ \mathbf{x} \in [0, 1]^2 \mid x_2 = 1 \} \},$$

in which case its geometric realization is the boundary of the unit square in \mathbb{R}^2 . The Möbius strip can be realized as a subset of the 4-dimensional hypercube as in Figure 1.

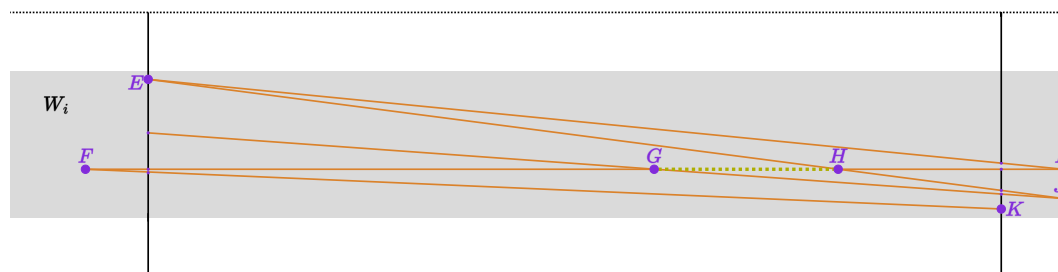


■ **Figure 1** A 4-dimensional geometric realization of a cubical complex homeomorphic to a Möbius strip.

2.2 Variable gadgets

The art gallery we construct will be an axis-aligned rectangle with several slits removed from around the border. We refer to certain slits and combinations of slits as *gadgets*. Our construction involves three kinds of gadgets: *variable gadgets*, *copying gadgets*, and *clause gadgets*.

We begin by establishing the key properties of the variable gadgets, which also appear in Bertschinger et al. [2]. A variable gadget consists of one slit in the left wall and two in the right wall, as depicted in Figure 2.



■ **Figure 2** A variable gadget enforcing that a guard must be placed on the dashed gold line segment GH . The boundary of this section of the art gallery consists of the entire outer profile (continuing above and below the dashed lines at the top and bottom), involving line segments of all colors. No other gadgets may intersect the shaded rectangular region W_i .

► **Lemma 3.** *If slits are drawn as in Figure 2, with no other slits in the region W_i , then at least one guard must be placed within W_i . If there is only one guard in that region, it must be placed on line segment GH . Furthermore, the slits can be drawn so that the height of W_i is arbitrarily small.*

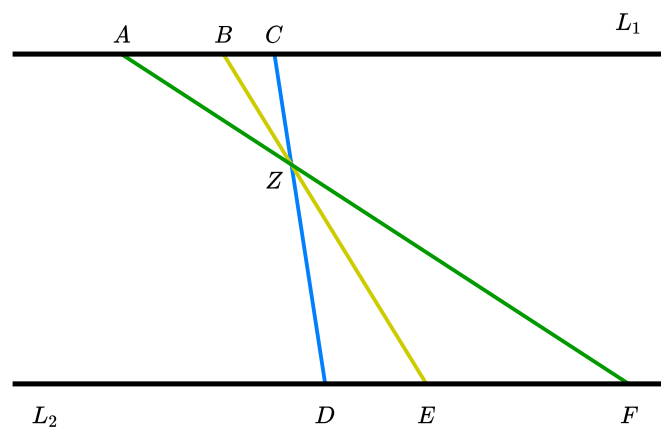
Proof. To see point F , we must have at least one guard placed within the triangle FIK (which also requires there to be some guard in W_i). To see point I , we must have at least one guard placed within the triangle EIF , so if there only one guard in W_i , it must be on the line segment FI . Since it must also see the point J , this further restricts the guard to the segment GH . The entire figure can be scaled in the y direction to give W_i arbitrarily small height without changing the dimensions of the segment GH . ◀

We refer to the line segments GH from Lemma 3 as *guard segments*, which we will number $1, 2, \dots, n$, and we refer to W_i as the *guard region* of guard segment i . Since guard regions can be made arbitrarily small, we may arrange these guard segments any way we wish within the rectangular gallery so long as no two guard segments share the same y -coordinate.

2.3 Copying gadgets

We next introduce our copying gadgets, which are based on the following geometric observation.

► **Lemma 4.** *Let L_1 and L_2 be two distinct parallel lines in the plane, and let Z be a point strictly between L_1 and L_2 . Let $f : L_1 \rightarrow L_2$ be the map defined via inversion through Z , i.e., for $p \in L_1$, $f(p)$ is the intersection of the line pZ with L_2 . Then f is linear (meaning it preserves ratios of distances between points on L_1 and the corresponding points on L_2).*



■ **Figure 3** Illustration of the geometry underlying the copying gadget.

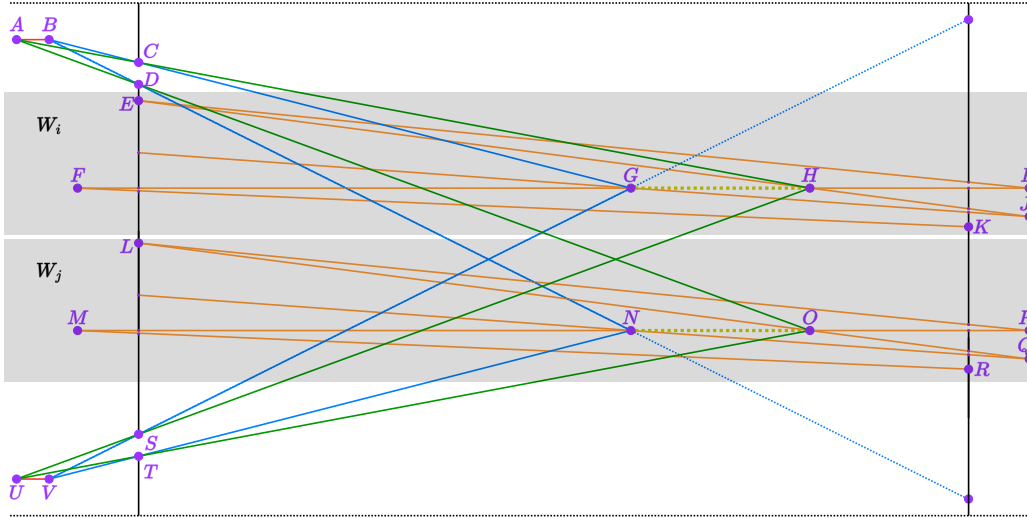
Proof. Assume L_1 and L_2 are drawn aligned with the x -axis as in Figure 3. Let A, B , and C be three arbitrary distinct points on L_1 , and let F, E , and D be their respective images under f . It is easy to see that (AZB, FZE) and (AZC, FZD) are pairs of similar triangles. This means that the ratios of lengths of AB to FE and AC to FD are equal, that is $\frac{B_x - A_x}{F_x - E_x} = \frac{C_x - A_x}{F_x - D_x}$. Rearranging, we have $\frac{B_x - A_x}{C_x - A_x} = \frac{F_x - E_x}{F_x - D_x}$. ◀

We now discuss the key properties of the copying gadgets, which are depicted in Figure 4.

► **Lemma 5.** *Suppose guard segments GH and NO are horizontally aligned and parallel as in Figure 4, with disjoint slits $CBAD$ and $SUVT$ lying outside of the guard regions such that AB, UV, GH and NO are all parallel. Furthermore, suppose $CBAD$ is chosen so the triples (A, C, H) , (B, C, G) , (A, D, O) and (B, D, N) are each colinear, and $SUVT$ is chosen similarly, as in the figure. Then any valid configuration of guards with one guard in each guard region must have two guards sharing an x -coordinate, one on GH and one on NO . Furthermore, if the left wall is moved sufficiently far to the left, $CBAD$ can be placed anywhere above W_i not in a guard region, $SUVT$ can be placed anywhere below W_j not in a guard region, and both of these slits can be made arbitrarily small.*

Proof. Let α be the location of the guard on GH , and let β be the location of guard on NO ; our objective is to show that, in any valid configuration, $\alpha_x = \beta_x$. We define the following four additional points:

- $\bar{\alpha} :=$ the intersection of line segment AB with line αC
- $\underline{\alpha} :=$ the intersection of line segment UV with line αS
- $\bar{\beta} :=$ the intersection of line segment AB with line βD
- $\underline{\beta} :=$ the intersection of line segment UV with line βT



■ **Figure 4** A copying gadget (and two variable gadgets). As in Figure 2, the boundary consists of the entire outer profile. The copying gadgets force the pair of guards on segments GH and NO to have the same x -coordinate. Additional variable and copying gadgets may be placed between W_i and W_j , which would be easier to draw if W_i and W_j were made narrower; they are drawn very large in this figure for the purpose of illustration.

Due to the obstructions by points C and D , the guard at α can only see AB to the left of $\bar{\alpha}$, and the guard at β can only see AB to the right of $\bar{\beta}$. If $\bar{\alpha}_x < \bar{\beta}_x$, the line segment $\bar{\alpha}\bar{\beta}$ will therefore not be seen by any guard, so we must have that $\bar{\alpha}_x \geq \bar{\beta}_x$. Similarly, the guard at α can only see UV to the right of $\underline{\alpha}$, and the guard at β can only see UV to the left of $\underline{\beta}$, so we must have that $\underline{\alpha}_x \leq \underline{\beta}_x$. Thus, in any valid configuration of the two guards,

$$\begin{aligned}
 \frac{\alpha_x - G_x}{H_x - G_x} &= \frac{B_x - \bar{\alpha}_x}{B_x - A_x} && \text{(by Lemma 4 with } Z := C) \\
 &\leq \frac{B_x - \bar{\beta}_x}{B_x - A_x} \\
 &= \frac{\beta_x - N_x}{O_x - N_x} && \text{(by Lemma 4 with } Z := D) \\
 &= \frac{V_x - \underline{\beta}_x}{V_x - U_x} && \text{(by Lemma 4 with } Z := T) \\
 &\leq \frac{V_x - \underline{\alpha}_x}{V_x - U_x} \\
 &= \frac{\alpha_x - G_x}{H_x - G_x} && \text{(by Lemma 4 with } Z := S).
 \end{aligned}$$

Multiplying through by $(H_x - G_x)$ and using the fact that $N_x = G_x$ and $O_x = H_x$, we have

$$\alpha_x - G_x \leq \beta_x - G_x \leq \alpha_x - G_x.$$

Thus, $\alpha_x = \beta_x$ in any valid configuration (and it is obvious that such a configuration is indeed valid).

For the final claim, observe that we can fully define the position of the top slit $CBAD$ as follows. Place AB anywhere outside the art gallery, not in any guard region. Then move the left wall (including all slits) sufficiently to the left so that the y -coordinates of points C and D are sufficiently close to A and B so that they do not lie in any guard region either. Existing slits will have to get stretched in this process, which is fine because this only makes them smaller. Thus, it is possible to make the top slit $CBAD$ arbitrarily small and place it anywhere, so long as it is above W_i and not in any other guard region. We can apply a symmetric procedure for $SUVT$. ◀

2.4 Constructing the art gallery

We are now ready to prove homeomorphism universality. Throughout the proof, as an example, we implement the various steps the construction for the Möbius strip.

Proof of Theorem 1. By Theorem 2, we may assume without loss of generality that S is the geometric realization of a nonempty cubical complex. This means that S can be described as a subset of $[0, 1]^n$ whose coordinates x_1, x_2, \dots, x_n satisfy some disjunctive normal form (DNF) formula

$$\phi_S \equiv \bigvee_{j=1}^{m'} C'_j,$$

where each C'_i is a conjunction of constraints that certain x_i variables take values 0 and 1. The formula will always be in DNF: each C'_i corresponds to a face of the cubical complex, so ϕ_S describes the set of points lying in at least one such face. For example, the formula for the Möbius strip M depicted in Figure 1 has six clauses, one for each of the six 2-dimensional faces. To write an explicit formula ϕ_M , we choose a coordinate system where $x_4 = 0$ corresponds to the points on the outer shell and enumerate the faces as clauses by traversing the strip starting from the left-most depicted face and proceeding next toward the back face:

$$\begin{aligned} \phi_M \equiv & (x_2 = 0 \wedge x_4 = 0) \vee (x_1 = 0 \wedge x_4 = 0) \vee (x_1 = 0 \wedge x_3 = 1) \\ & \vee (x_3 = 1 \wedge x_4 = 1) \vee (x_2 = 0 \wedge x_4 = 1) \vee (x_2 = 0 \wedge x_3 = 0) \end{aligned}$$

We next rewrite ϕ_S in conjunctive normal form (CNF),

$$\phi_S \equiv \bigwedge_{j=1}^m C_i,$$

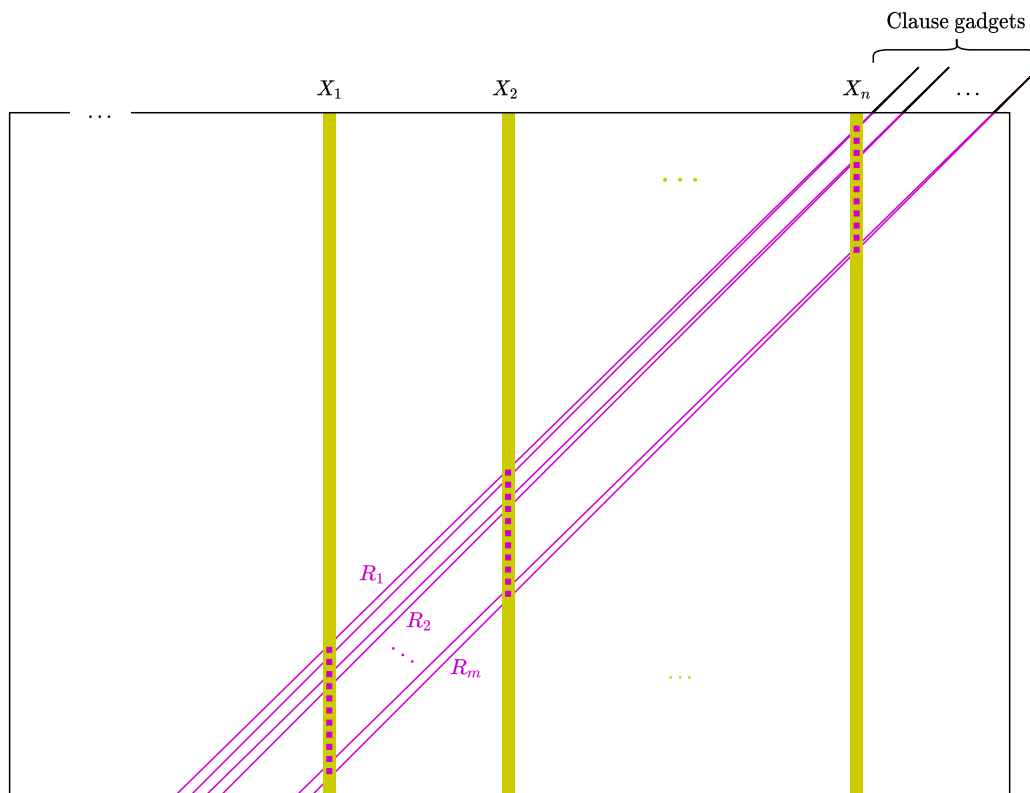
where each C_i is a disjunction of constraints, i.e.,

$$C_i \equiv (x_{i_j,1} = c_{j,1}) \vee (x_{i_j,2} = c_{j,2}) \vee \dots \vee (x_{i_j,\ell} = c_{j,\ell}),$$

with each $i_{j,k} \in \{1, 2, \dots, n\}$ and $c_{j,k} \in \{0, 1\}$. The transformation from DNF to CNF is standard, and can be accomplished by enumerating all tuples of constraints that take one constraint from each clause. For example, since ϕ_M has 6 clauses in DNF, each of size 2, the translation to CNF produces $2^6 = 64$ clauses, each of size 6. However, after eliminating

redundancies, we can simplify ϕ_M to

$$\begin{aligned}
 \phi_M \equiv & (x_1 = 0 \vee x_2 = 0 \vee x_3 = 1) \\
 & \wedge (x_2 = 0 \vee x_3 = 1 \vee x_4 = 0) \\
 & \wedge (x_1 = 0 \vee x_2 = 0 \vee x_4 = 1) \\
 & \wedge (x_3 = 0 \vee x_3 = 1 \vee x_4 = 0 \vee x_4 = 1) \\
 & \wedge (x_1 = 0 \vee x_3 = 0 \vee x_4 = 0 \vee x_4 = 1).
 \end{aligned} \tag{1}$$



■ **Figure 5** Illustration of the first step of the general construction (making clause gadgets), starting from a CNF formula with n variables and m clauses. The full art gallery is shown in more detail for the Möbius strip in Figure 6.

Starting from a rectangular art gallery, we make a narrow diagonal slit for each clause j in the top-right corner such that the regions R_j of the gallery that can see to the end of each slit extend downward to the left and do not overlap, as shown in Figure 5. For each $i \in \{1, 2, 3, \dots, n\}$ we define X_i to be a tall, skinny, axis-aligned rectangular region such that the convex hulls of the sets $X_i \cap (R_1 \cup R_2 \cup \dots \cup R_m)$ do not overlap in y -coordinates (in terms of Figure 5, the pink dashed lines must not overlap in y -coordinates). It is always possible to guarantee this non-overlapping property by making the gallery sufficiently tall and/or making the clauses sufficiently close together.

In every clause j , for every constraint $x_{i_j,k} = c_{j,k}$, we add a variable gadget in the left and right sides of the rectangle enforcing the constraint that there is a guard on a guard segment spanning the width of $X_{i_j,k}$ whose endpoint lies within R_j . If $c_{j,k} = 0$, this will

be the left endpoint, and if $c_{j,k} = 1$, this will be the right endpoint. We then add copying gadgets to enforce that all guard segments placed in X_i for the same i must have the same x -coordinate. By Lemmas 3 and 5, this is possible by shrinking the variable and copying gadgets and moving the left wall sufficiently far away.

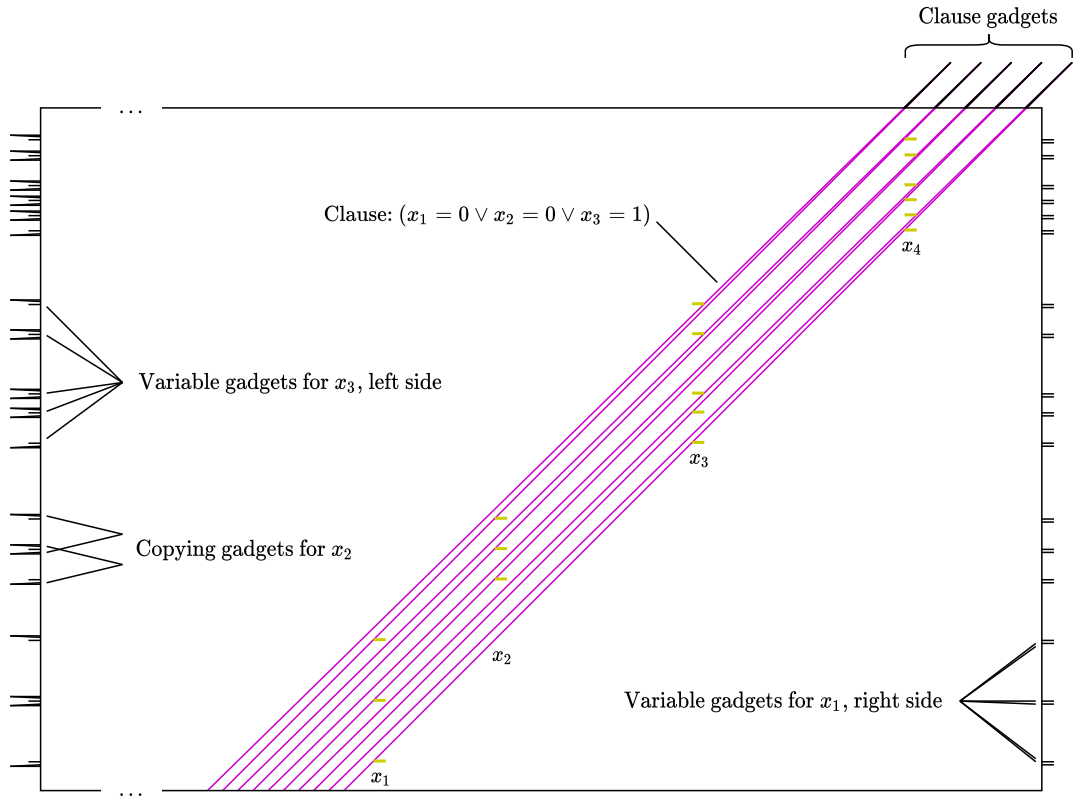


Figure 6 An art gallery whose solution space is homeomorphic to a Möbius strip, using the formula ϕ_M from (1). The horizontal gold dashes above each variable are the line segments on which guards must walk, as in Figure 4. The pink diagonal lines depict the regions R_j that can see to the end of each clause gadget slit; each of these regions must contain at least one guard. The left wall must be placed sufficiently far to the left to ensure that none of the variable/copy gadgets interfere with each other. As drawn, this particular art gallery has 183 vertices.

Letting P_0 be the polygon P without the clause gadget slits, we have that $V(P_0)$ consists of all solutions with one guard on each guard segment, with guards within the same X_i placed at the same x -coordinate. There is thus a natural homeomorphism $h : V(P_0) \rightarrow [0, 1]^n$, and from the way the clause gadgets were constructed, it clearly follows that $V(P)$ consists of all solutions in which ϕ_S is satisfied under h . Thus, $V(P)$ is homeomorphic to S . ◀

3 Efficient construction for closed surfaces

We have argued that our universality construction is qualitatively simpler than its predecessors. In this section, we show how our technique can be used to produce quantitatively simple galleries, in terms of the number of vertices of the polygon. This is not apparent a priori: even if a space can be triangulated as a cubical complex with relatively few faces, the conversion

58:10 Topological Universality of the Art Gallery Problem

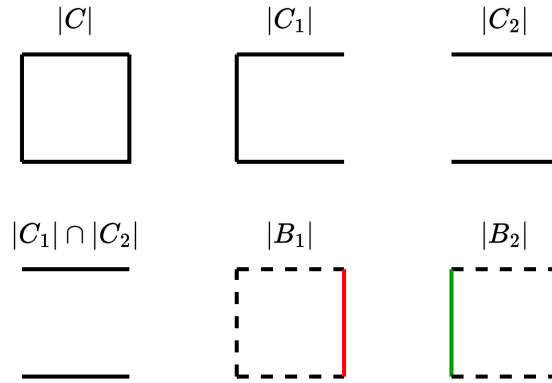
from DNF to CNF can exponentially blow up the number of clauses, and thus the size of the art gallery. Here we prove that an important class of topological spaces, namely the closed surfaces, can occur as solutions to art galleries with linearly many vertices.

► **Theorem 6.** *There are polygons P_g, Q_g with $O(g)$ vertices such that $V(P_g)$ is homeomorphic to the closed orientable surface of genus g and $V(Q_g)$ is homeomorphic to the closed non-orientable surface of genus g .*

Proof. We know by Theorem 1 that such polygons exist for the finite number of cases where $g \in \{0, 1\}$, so it is sufficient to construct these polygons only for $g \geq 2$.

It is well known that, for $g \geq 2$, the orientable surface of genus g can be obtained as the connected sum of g copies of a torus, $T^2 \# T^2 \# \dots \# T^2$, while the non-orientable surface of genus g can be obtained as the connected sum of g copies of the real projective plane $\mathbb{RP}^2 \# \mathbb{RP}^2 \# \dots \# \mathbb{RP}^2$ (see, e.g., the textbook by Massey [9]). The connected sum $R \# R$ is ordinarily defined as removing an open disk from two copies of R and gluing their boundaries together. For 2-dimensional surfaces, this is equivalent to gluing the disk boundaries to opposite ends of a cylindrical tube, which is the formulation we use in this construction. Thus, let R be either T^2 or \mathbb{RP}^2 . By Theorem 2, we know there is some j such that R is homeomorphic to $|C|$, the geometric realization of a cubical complex C with j variables. Let x_1, x_2, \dots, x_j be these variables and write $\mathbf{x} = (x_1, x_2, \dots, x_j)$.

Clearly C has at least two 2-dimensional faces. Let C_1 be a cubical complex obtained by removing a face f_1 of dimension 2 from C , and C_2 obtained by removing a different face f_2 . For $i \in \{1, 2\}$, define B_i to be the cubical complex in \mathbf{x} consisting of f_i and its boundary, so that $|C_i| \cap |B_i|$ is the boundary of the removed disc (see Figure 7).



■ **Figure 7** Visualisation of $|C_1|$, $|C_2|$, $|B_1|$ and $|B_2|$. In reality, C should be a complex for T^2 or \mathbb{RP}^2 , rather than S^1 as shown. The figure is just meant to convey the construction at a schematic level.

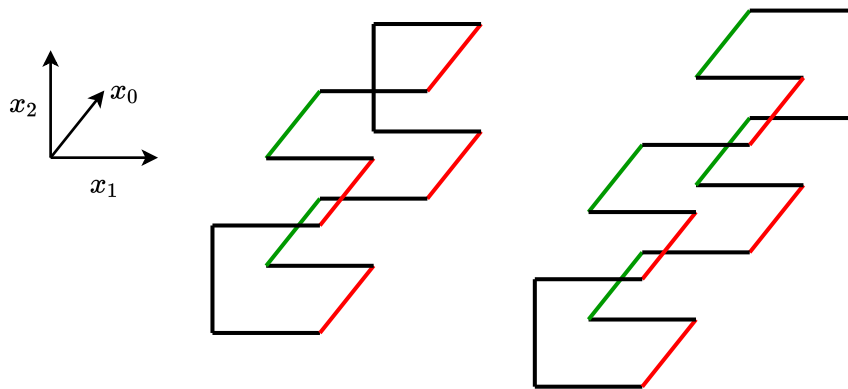
Define a new variable x_0 and fix constants $k_0 = 0 < k_1 < \dots < k_{g-2} < 1 = k_{g-1}$. Then the following formulas express the property that a point \mathbf{x} is contained in $R \# R \# \dots \# R$ (see Figure 8 for a visualization). For g even we have

$$\begin{aligned}
 & (\mathbf{x} \in |C_2| \vee x_0 = 0) \wedge (\mathbf{x} \in |C_1| \vee x_0 = 1) \wedge \\
 & (\mathbf{x} \in |B_1| \vee 0 \leq x_0 \leq k_1 \vee k_2 \leq x_0 \leq k_3 \vee \dots \vee k_{g-3} \leq x_0 \leq k_{g-2} \vee x_0 = 1) \wedge \\
 & (\mathbf{x} \in |B_2| \vee k_1 \leq x_0 \leq k_2 \vee k_3 \leq x_0 \leq k_4 \vee \dots \vee k_{g-2} \leq x_0 \leq 1),
 \end{aligned}$$

and for g odd we have

$$\begin{aligned}
 & (\mathbf{x} \in |C_2| \vee x_0 = 0 \vee x_0 = 1) \wedge \mathbf{x} \in |C_1| \wedge \\
 & (\mathbf{x} \in |B_1| \vee 0 \leq x_0 \leq k_1 \vee k_2 \leq x_0 \leq k_3 \vee \dots \vee k_{g-2} \leq x_0 \leq 1) \wedge \\
 & (\mathbf{x} \in |B_2| \vee k_1 \leq x_0 \leq k_2 \vee k_3 \leq x_0 \leq k_4 \vee \dots \vee k_{g-3} \leq x_0 \leq k_{g-2} \vee x_0 = 1).
 \end{aligned}$$

Note that such a space is typically not the geometric realization of a cubical complex in $x_0, x_1, x_2, \dots, x_j$ because of the constraints on x_0 ; nevertheless, we will construct an art gallery for it.



■ **Figure 8** A schematic visualization for our constructions of $R\#R\#\dots\#R$ for even g (left) and odd g (right). For all values of x_0 , we require $\mathbf{x} \in |C_1|$ and $\mathbf{x} \in |C_2|$, except for $x_0 = 0$ or $x_0 = 1$ where one of these constraints is dropped to cap the hole at the end. For $k_i < x_0 < k_{i+1}$, \mathbf{x} must be in either $|B_1| \cap |C_1|$ or $|B_2| \cap |C_2|$ depending on the parity of i . This creates tubes connecting the copies of R .

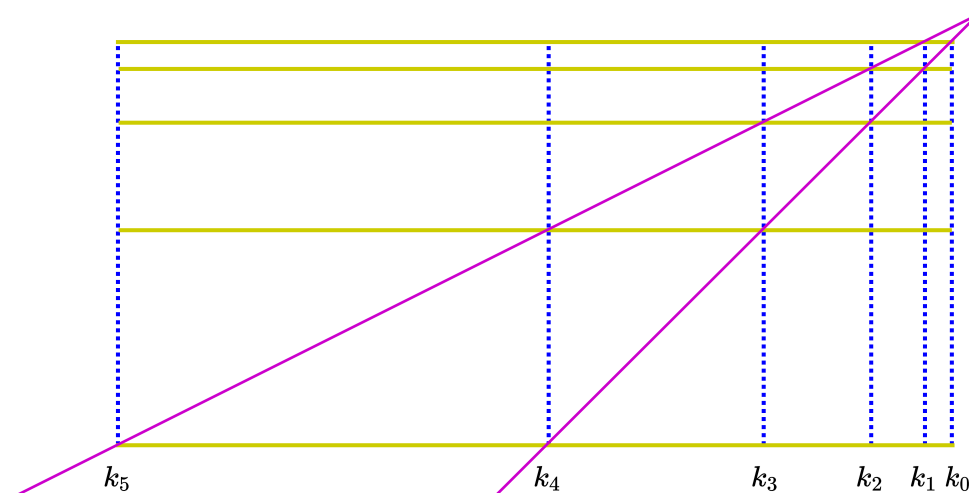
We can write these expressions in CNF with terms of form $x_i = 0$, $x_i = 1$, or $k_i \leq x_0 \leq k_{i+1}$. We write the constraint $\mathbf{x} \in |B_1|$ in CNF as

$$\phi_{B_1} := \bigwedge_{\ell=1}^p \bigvee_{m=1}^{q_\ell} t_{\ell,m},$$

where each $t_{\ell,m}$ is an atomic constraint of the form e.g. $x_i = c$. We may thus rewrite

$$\begin{aligned}
 & \mathbf{x} \in |B_1| \vee 0 \leq x_0 \leq k_1 \vee k_2 \leq x_0 \leq k_3 \vee \dots \vee k_{g-2} \leq x_0 \leq 1 \\
 & \equiv \bigwedge_{\ell=1}^p \left(\left(\bigvee_{m=1}^{q_\ell} t_{\ell,m} \right) \vee 0 \leq x_0 \leq k_1 \vee k_2 \leq x_0 \leq k_3 \vee \dots \vee k_{g-2} \leq x_0 \leq 1 \right).
 \end{aligned}$$

To add the constraints on x_0 to each clause involving x_0 , we must make a small modification to the construction from Section 2. Instead of having the clause gadgets along the top wall, we extend the top wall upward and put these gadgets along the right wall. It is easy to see that this does not affect our ability to create the constraints on \mathbf{x} . We then add the constraints on x_0 to each clause as in Figure 9; if all clause gadgets are vertically translated copies of each other, then all guard segments for x_0 across the different clauses can be placed at the same x -coordinates.



■ **Figure 9** Guard segments for x_0 . We place the first guard segment so that the lower line in the wedge intersects it at 0, and the upper line intersects it at some point k_1 . We can then place another segment which intersects the *lower* line at k_1 , and the intersection of this segment with the upper line gives the value of k_2 . We can repeat until we have $g - 1$ segments. Each clause depending on x_0 will have a copy of this setup but with some of the segments removed depending on which terms $k_i \leq x_0 \leq k_{i+1}$ it contains.

Using a similar expansion for the other terms, we obtain a CNF expression that cuts out the space we want. Since p and each q_ℓ are constants, the total number of clauses does not depend on g , and each clause has at most $O(g)$ terms. Thus, when we add the constraints involving x_0 to the constant-sized formula ϕ_{B_1} , the number of vertices increases linearly in g .

We may similarly add x_0 constraints to the formula for B_2 with linear blowup. The result is a gallery with $O(g)$ vertices whose solution space is homeomorphic to $R\#R\#\dots\#R$. ◀

We have shown that such P_g, Q_g exist having $O(g)$ vertices. In case it is of interest, we leave it to the reader to verify that the positions of the vertices can additionally be chosen to be rational numbers that require only $O(g)$ bits to describe.¹

4 Conclusion

In this work we have settled the open question of Bertschinger et al. [2] by showing that solution spaces to the art gallery problem can capture the topology of any semi-algebraic set up to homeomorphism. In doing so, we have introduced a new form of copying gadget that enables simpler arguments about the structure of valid solutions to art gallery problem instances.

Beyond the art gallery problem, our main result raises intriguing possibilities for the broader theory of $\exists\mathbb{R}$ -hardness. To the best of our knowledge, this is the first paper showing that the topological structure of semi-algebraic sets can be carried into a different problem

¹ In general, if we fix the length of the guard segments for x_0 then the wedge parameters that would give $k_{g-1} = 1$ are not rational numbers. Instead, we fix a sufficiently thin slit and choose the length of the guard segments for x_0 appropriately.

domain even up to the fine-grain notion of homeomorphism.² Perhaps this holds for other $\exists\mathbb{R}$ -hard problems as well.

References

- 1 Mikkel Abrahamsen, Anna Adamaszek, and Tillmann Miltzow. The art gallery problem is $\exists\mathbb{R}$ -complete. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 65–73, 2018. doi:10.1145/3188745.3188868.
- 2 Daniel Bertschinger, Nicolas El Maalouly, Tillmann Miltzow, Patrick Schnider, and Simon Weber. Topological art in simple galleries. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 87–116. SIAM, 2022.
- 3 Josef Blass and Włodzimierz Holsztynski. Cubical polyhedra and homotopy, III. *Atti della Accademia Nazionale dei Lincei. Classe di Scienze Fisiche, Matematiche e Naturali. Rendiconti*, 53(3-4):275–279, September 1972. URL: <http://eudml.org/doc/295890>.
- 4 Édouard Bonnet and Tillmann Miltzow. An approximation algorithm for the art gallery problem. In Boris Aronov and Matthew J. Katz, editors, *33rd International Symposium on Computational Geometry, SoCG 2017, July 4-7, 2017, Brisbane, Australia*, volume 77 of *LIPICs*, pages 20:1–20:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPICs.SoCG.2017.20.
- 5 Michael G. Dobbins, Andreas Holmsen, and Tillmann Miltzow. A universality theorem for nested polytopes. arXiv preprint arXiv:1908.02213, 2019.
- 6 Heisuke Hironaka. Triangulations of algebraic sets. In *Algebraic geometry (Proceedings of Symposia in Pure Mathematics, Volume 29, Humboldt State University, Arcata, California, 1974)*, volume 29, pages 165–185, 1975.
- 7 Kyle Roger Hoffman. *Triangulation of locally semi-algebraic spaces*. PhD thesis, University of Michigan, 2009.
- 8 D. T. Lee and Arthur K. Lin. Computational complexity of art gallery problems. *IEEE Transactions on Information Theory*, 32(2):276–282, 1986. doi:10.1109/TIT.1986.1057165.
- 9 William S. Massey. *A basic course in algebraic topology*, volume 127 of *Graduate Texts in Mathematics*, chapter 1. Springer, 1991.
- 10 Lucas Meijer and Tillmann Miltzow. Sometimes two irrational guards are needed. arXiv preprint arXiv:2212.01211, 2022.
- 11 Nikolai E Mnëv. The universality theorems on the classification problem of configuration varieties and convex polytopes varieties. In *Topology and geometry—Rohlin seminar*, pages 527–543. Springer, 1988.
- 12 Jürgen Richter-Gebert. Mnëv’s universality theorem revisited. *Séminaire Lotharingien de Combinatoire [electronic only]*, 34:15 p.–15 p., 1995. URL: <http://eudml.org/doc/119012>.
- 13 Peter W. Shor. Stretchability of pseudolines is NP-hard. In *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, pages 531–554, 1990. doi:10.1090/dimacs/004/41.
- 14 Jack Stade. The point-boundary art gallery problem is $\exists\mathbb{R}$ -hard. arXiv preprint arXiv:2210.12817, 2022. doi:10.48550/ARXIV.2210.12817.

² We are aware of one similar work by Dobbins, Holmsen, and Miltzow [5] for the Nested Polytope Problem, which is also $\exists\mathbb{R}$ -complete. They show universality up to rational homeomorphism for bounded real varieties, which is a smaller class than semi-algebraic sets.

On Higher Dimensional Point Sets in General Position

Andrew Suk 

Department of Mathematics, University of California San Diego, La Jolla, CA, USA

Ji Zeng 

Department of Mathematics, University of California San Diego, La Jolla, CA, USA

Abstract

A finite point set in \mathbb{R}^d is in general position if no $d + 1$ points lie on a common hyperplane. Let $\alpha_d(N)$ be the largest integer such that any set of N points in \mathbb{R}^d with no $d + 2$ members on a common hyperplane, contains a subset of size $\alpha_d(N)$ in general position. Using the method of hypergraph containers, Balogh and Solymosi showed that $\alpha_2(N) < N^{5/6+o(1)}$. In this paper, we also use the container method to obtain new upper bounds for $\alpha_d(N)$ when $d \geq 3$. More precisely, we show that if d is odd, then $\alpha_d(N) < N^{\frac{1}{2} + \frac{1}{2d} + o(1)}$, and if d is even, we have $\alpha_d(N) < N^{\frac{1}{2} + \frac{1}{d-1} + o(1)}$.

We also study the classical problem of determining the maximum number $a(d, k, n)$ of points selected from the grid $[n]^d$ such that no $k + 2$ members lie on a k -flat. For fixed d and k , we show that

$$a(d, k, n) \leq O\left(n^{\frac{d}{2\lfloor(k+2)/4\rfloor} \left(1 - \frac{1}{2\lfloor(k+2)/4\rfloor d + 1}\right)}\right),$$

which improves the previously best known bound of $O\left(n^{\frac{d}{\lfloor(k+2)/2\rfloor}}\right)$ due to Lefmann when $k + 2$ is congruent to 0 or 1 mod 4.

2012 ACM Subject Classification Mathematics of computing \rightarrow Combinatorics

Keywords and phrases independent sets, hypergraph container method, generalised Sidon sets

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.59

Related Version Full Version: <https://arxiv.org/abs/2211.15968>

Funding Andrew Suk: Supported by NSF CAREER award DMS-1800746 and NSF award DMS-1952786.

Ji Zeng: Supported by NSF grant DMS-1800746.

1 Introduction

A finite point set in \mathbb{R}^d is said to be in *general position* if no $d + 1$ members lie on a common hyperplane. Let $\alpha_d(N)$ be the largest integer such that any set of N points in \mathbb{R}^d with no $d + 2$ members on a hyperplane, contains $\alpha_d(N)$ points in general position.

In 1986, Erdős [8] proposed the problem of determining $\alpha_2(N)$ and observed that a simple greedy algorithm shows $\alpha_2(N) \geq \Omega(\sqrt{N})$. A few years later, Füredi [10] showed that

$$\Omega(\sqrt{N \log N}) < \alpha_2(N) < o(N),$$

where the lower bound uses a result of Phelps and Rödl [20] on partial Steiner systems, and the upper bound relies on the density Hales-Jewett theorem [11, 12]. In 2018, a breakthrough was made by Balogh and Solymosi [3], who showed that $\alpha_2(N) < N^{5/6+o(1)}$. Their proof was based on the method of hypergraph containers, a powerful technique introduced independently by Balogh, Morris, and Samotij [1] and by Saxton and Thomason [24], that reveals an underlying structure of the independent sets in a hypergraph. We refer interested readers to [2] for a survey of results based on this method.



© Andrew Suk and Ji Zeng;

licensed under Creative Commons License CC-BY 4.0

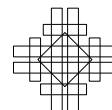
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 59; pp. 59:1–59:13

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



In higher dimensions, the best lower bound for $\alpha_d(N)$ is due to Cardinal, Tóth, and Wood [5], who showed that $\alpha_d(N) \geq \Omega((N \log N)^{1/d})$, for every fixed $d \geq 2$. For upper bounds, Milićević [18] used the density Hales-Jewett theorem to show that $\alpha_d(N) = o(N)$ for every fixed $d \geq 2$. However, these upper bounds in [18], just like that in [10], are still almost linear in N . Our main result is the following.

► **Theorem 1.** *Let $d \geq 3$ be a fixed integer. If d is odd, then $\alpha_d(N) < N^{\frac{1}{2} + \frac{1}{2d} + o(1)}$. If d is even, then $\alpha_d(N) < N^{\frac{1}{2} + \frac{1}{d-1} + o(1)}$.*

Our proof of Theorem 1 is also based on the hypergraph container method. A key ingredient in the proof is a new supersaturation lemma for $(k+2)$ -tuples of the grid $[n]^d$ that lie on a k -flat, which we shall discuss in the next section. Here, by a k -flat we mean a k -dimensional affine subspace of \mathbb{R}^d .

We also study the classical problem of determining the maximum number of points selected from the grid $[n]^d$ such that no $k+2$ members lie on a k -flat. The key ingredient of Theorem 1 mentioned above can be seen as a supersaturation version of this Turán-type problem. When $k=1$, this is the famous *no-three-in-line problem* raised by Dudeney [7] in 1917: Is it true that one can select $2n$ points in $[n]^2$ such that no three are collinear? Clearly, $2n$ is an upper bound as any vertical line must contain at most 2 points. For small values of n , many authors have published solutions to this problem obtaining the bound of $2n$ (e.g. see [9]), but for large n , the best known general construction is due to Hall et al. [13] with slightly fewer than $3n/2$ points.

More generally, we let $a(d, k, r, n)$ denote the maximum number of points from $[n]^d$ such that no r points lie on a k -flat. Since $[n]^d$ can be covered by n^{d-k} many k -flats, we have the trivial upper bound $a(d, k, r, n) \leq (r-1)n^{d-k}$. For certain values d, k , and r fixed and n tends to infinity, this bound is known to be asymptotically best possible: Many authors [22, 4, 17] noticed that $a(d, d-1, d+1, n) = \Theta(n)$ by looking at the modular moment curve over a finite field \mathbb{Z}_p ; In [21], Pór and Wood proved that $a(3, 1, 3, n) = \Theta(n^2)$; Very recently, Sudakov and Tomon [25] showed that $a(d, k, r, n) = \Theta(n^{d-k})$ when $r > d^k$.

We shall focus on the case when $r = k+2$ and write $a(d, k, n) := a(d, k, k+2, n)$. Surprisingly, Lefmann [17] (see also [16]) showed that $a(d, k, n)$ behaves much differently than $\Theta(n^{d-k})$. In particular, he showed that

$$a(d, k, n) \leq O\left(n^{\frac{d}{\lfloor (k+2)/2 \rfloor}}\right).$$

Our next result improves this upper bound when $k+2$ is congruent to 0 or 1 mod 4.

► **Theorem 2.** *For fixed d and k , as $n \rightarrow \infty$, we have*

$$a(d, k, n) \leq O\left(n^{\frac{d}{2\lfloor (k+2)/4 \rfloor} \left(1 - \frac{1}{2\lfloor (k+2)/4 \rfloor d + 1}\right)}\right).$$

For example, we have $a(4, 2, n) \leq O(n^{\frac{16}{9}})$ while Lefmann's bound in [17] gives us $a(4, 2, n) \leq O(n^2)$, which coincides with the trivial upper bound. In particular, Theorem 2 tells us that, if 4 divides $k+2$, then $a(d, k, n)$ only behaves like $\Theta(n^{d-k})$ when $d = k+1$. This is quite interesting compared to the fact that $a(3, 1, n) = \Theta(n^2)$ proved in [21]. Lastly, let us note that the current best lower bound for $a(d, k, n)$ is also due to Lefmann [17], who showed that $a(d, k, n) \geq \Omega\left(n^{\frac{d}{k+1} - k - \frac{k}{k+1}}\right)$.

For integer $n > 0$, we let $[n] = \{1, \dots, n\}$, and $\mathbb{Z}_n = \{0, 1, \dots, n-1\}$. We systemically omit floors and ceilings whenever they are not crucial for the sake of clarity in our presentation. All logarithms are in base two.

2 $(k + 2)$ -tuples of $[n]^d$ on a k -flat

In this section, we establish two lemmas that will be used in the proof of Theorem 1.

Given a set T of $k + 2$ points in \mathbb{R}^d that lie on a k -flat, we say that T is *degenerate* if there is a subset $S \subset T$ of size j , where $3 \leq j \leq k + 1$, such that S lies on a $(j - 2)$ -flat. Otherwise, we say that T is *non-degenerate*. We establish a supersaturation lemma for non-degenerate $(k + 2)$ -tuples of $[n]^d$.

► **Lemma 3.** *For real number $\gamma > 0$ and fixed positive integers d, k , such that k is even and $d - 2\gamma > (k - 1)(k + 2)$, any subset $V \subset [n]^d$ of size $n^{d-\gamma}$ spans at least $\Omega(n^{(k+1)d-(k+2)\gamma})$ non-degenerate $(k + 2)$ -tuples that lie on a k -flat.*

Proof. Let $V \subset [n]^d$ such that $|V| = n^{d-\gamma}$. Set $r = \frac{k}{2} + 1$ and $E_r = \binom{V}{r}$ to be the collection of r -tuples of V . Notice that the sum of a r -tuple from V belongs to $[rn]^d$. For each $v \in [rn]^d$, we define

$$E_r(v) = \{\{v_1, \dots, v_r\} \in E_r : v_1 + \dots + v_r = v\}.$$

Then for $T_1, T_2 \in E_r(v)$, where $T_1 = \{v_1, \dots, v_r\}$ and $T_2 = \{u_1, \dots, u_r\}$, we have

$$v_1 + \dots + v_r = v = u_1 + \dots + u_r,$$

which implies that $T_1 \cup T_2$ lies on a common k -flat. Let

$$E_{2r} = \bigcup_{v \in [rn]^d} \bigcup_{T_1, T_2 \in E_r(v)} \{T_1, T_2\}.$$

Hence, for each $\{T_1, T_2\} \in E_{2r}$, $T_1 \cup T_2$ lies on a k -flat. Moreover, by Jensen's inequality, we have

$$|E_{2r}| = \sum_{v \in [rn]^d} \binom{|E_r(v)|}{2} \geq (rn)^d \binom{\sum_v |E_r(v)|}{(rn)^d} = (rn)^d \binom{|E_r|/(rn)^d}{2} \geq \frac{|E_r|^2}{4(rn)^d}.$$

Since k and d are fixed and $r = \frac{k}{2} + 1$ and $|V| = n^{d-\gamma}$,

$$|E_r|^2 = \binom{|V|}{r}^2 = \binom{|V|}{(k/2) + 1}^2 \geq \Omega(n^{(k+2)(d-\gamma)}).$$

Combining the two inequalities above gives

$$|E_{2r}| \geq \Omega(n^{(k+1)d-(k+2)\gamma}).$$

We say that $\{T_1, T_2\} \in E_{2r}$ is *good* if $T_1 \cap T_2 = \emptyset$, and the $(k + 2)$ -tuple $(T_1 \cup T_2)$ is non-degenerate. Otherwise, we say that $\{T_1, T_2\}$ is *bad*. In what follows, we will show that at least half of the pairs (i.e. elements) in E_{2r} are good. To this end, we will need the following claim.

► **Claim 4.** If $\{T_1, T_2\} \in E_{2r}$ is bad, then $T_1 \cup T_2$ lies on a $(k - 1)$ -flat.

Proof. Write $T_1 = \{v_1, \dots, v_r\}$ and $T_2 = \{u_1, \dots, u_r\}$. Let us consider the following cases.

Case 1. Suppose $T_1 \cap T_2 \neq \emptyset$. Then, without loss of generality, there is an integer $j < r$ such that

$$v_1 + \dots + v_j = u_1 + \dots + u_j,$$

59:4 On Higher Dimensional Point Sets in General Position

where $v_1, \dots, v_j, u_1, \dots, u_j$ are all distinct elements, and $v_t = u_t$ for $t > j$. Thus $|T_1 \cup T_2| = 2j + (r - j)$. The $2j$ elements above lie on a $(2j - 2)$ -flat. Adding the remaining $r - j$ points implies that $T_1 \cup T_2$ lies on a $(j - 2 + r)$ -flat. Since $r = \frac{k}{2} + 1$ and $j \leq \frac{k}{2}$, $T_1 \cup T_2$ lies on a $(k - 1)$ -flat.

Case 2. Suppose $T_1 \cap T_2 = \emptyset$. Then $T_1 \cup T_2$ must be degenerate, which means there is a subset $S \subset T_1 \cup T_2$ of j elements such that S lies on a $(j - 2)$ -flat, for some $3 \leq j \leq k + 1$. Without loss of generality, we can assume that $v_1 \notin S$. Hence, $(T_1 \cup T_2) \setminus \{v_1\}$ lies on a $(k - 1)$ -flat. On the other hand, we have

$$v_1 = u_1 + \dots + u_r - v_2 - \dots - v_r.$$

Hence, v_1 is in the affine hull of $(T_1 \cup T_2) \setminus \{v_1\}$ which implies that $T_1 \cup T_2$ lies on a $(k - 1)$ -flat. ◀

We are now ready to prove the following claim.

▷ **Claim 5.** At least half of the pairs in E_{2r} are good.

Proof. For the sake of contradiction, suppose at least half of the pairs in E_{2r} are bad. Let H be the collection of all the j -flats spanned by subsets of V for all $j \leq k - 1$. Notice that if $S \subset V$ spans a j -flat h , then h is also spanned by only $j + 1$ elements from S . So we have

$$|H| \leq \sum_{j=0}^{k-1} |V|^{j+1} \leq kn^{k(d-\gamma)}.$$

For each bad pair $\{T_1, T_2\} \in E_{2r}$, $T_1 \cup T_2$ lies on a j -flat from H by Claim 4. By the pigeonhole principle, there is a j -flat h with $j \leq k - 1$ such that at least

$$\frac{|E_{2r}|/2}{|H|} \geq \frac{\Omega(n^{(k+1)d-(k+2)\gamma})}{2kn^{k(d-\gamma)}} = \Omega(n^{d-2\gamma})$$

bad pairs from E_{2r} have the property that their union lies in h . On the other hand, since h contains at most n^{k-1} points from $[n]^d$, h can correspond to at most $O(n^{(k-1)(k+2)})$ bad pairs from E_{2r} . Since we assumed $d - 2\gamma > (k - 1)(k + 2)$, we have a contradiction for n sufficiently large. ◀

Each good pair $\{T_1, T_2\} \in E_{2r}$ gives rise to a non-degenerate $(k + 2)$ -tuple $T_1 \cup T_2$ that lies on a k -flat. On the other hand, any such $(k + 2)$ -tuple in V will correspond to at most $\binom{k+2}{r}$ good pairs in E_{2r} . Hence, by Claim 5, there are at least

$$\frac{|E_{2r}|}{2} \bigg/ \binom{k+2}{r} = \Omega(n^{(k+1)d-(k+2)\gamma})$$

non-degenerate $(k + 2)$ -tuples that lie on a k -flat, concluding the proof. ◀

In the other direction, we will use the following upper bound.

► **Lemma 6.** For real number $\gamma > 0$ and fixed positive integers d, k, ℓ , such that $\ell < k + 2$, suppose $U, V \subset [n]^d$ satisfy $|U| = \ell$ and $|V| = n^{d-\gamma}$, then V contains at most $n^{(k+1-\ell)(d-\gamma)+k}$ non-degenerate $(k + 2)$ -tuples that lie on a k -flat and contain U .

Proof. If U spans a j -flat for some $j < \ell - 1$, then by definition no non-degenerate $(k + 2)$ -tuple contains U . Hence we can assume U spans a $(\ell - 1)$ -flat. Observe that a non-degenerate $(k + 2)$ -tuple T , which lies on a k -flat and contains U , must contain a $(k + 1)$ -tuple $T' \subset T$ such that T' spans a k -flat and $U \subset T'$. Then there are at most $n^{(k+1-\ell)(d-\gamma)}$ ways to add $k + 1 - \ell$ points to U from V to obtain such T' . After T' is determined, there are at most n^k ways to add a final point from the affine hull of T' to obtain T . So we conclude the proof by multiplication. ◀

3 The container method: Proof of Theorem 1

In this section, we use the hypergraph container method to prove Theorem 1. We follow the method outlined in [3]. Let $\mathcal{H} = (V(\mathcal{H}), E(\mathcal{H}))$ denote a $(k + 2)$ -uniform hypergraph. For any $U \subset V(\mathcal{H})$, its degree $\delta(U)$ is the number of edges containing U . For each $\ell \in [k + 2]$, we use $\Delta_\ell(\mathcal{H})$ to denote the maximum $\delta(U)$ among all U of size ℓ . For parameter $\tau > 0$, we define the following quantity

$$\Delta(\mathcal{H}, \tau) = \frac{2^{\binom{k+2}{2}-1} |V(\mathcal{H})|}{(k + 2) |E(\mathcal{H})|} \sum_{\ell=2}^{k+2} \frac{\Delta_\ell(\mathcal{H})}{\tau^{\ell-1} 2^{\binom{\ell-1}{2}}}.$$

Then we have the following hypergraph container lemma from [3], which is a restatement of Corollary 3.6 in [24].

► **Lemma 7.** *Let \mathcal{H} be a $(k + 2)$ -uniform hypergraph and $0 < \epsilon, \tau < 1/2$. Suppose that $\tau < 1/(200 \cdot (k + 2) \cdot (k + 2)!)$ and $\Delta(\mathcal{H}, \tau) \leq \epsilon/(12 \cdot (k + 2)!)$. Then there exists a collection \mathcal{C} of subsets (containers) of $V(\mathcal{H})$ such that*

1. *Every independent set in \mathcal{H} is a subset of some $C \in \mathcal{C}$;*
2. *$\log |\mathcal{C}| \leq 1000 \cdot (k + 2) \cdot ((k + 2)!)^3 \cdot |V(\mathcal{H})| \cdot \tau \cdot \log(1/\epsilon) \cdot \log(1/\tau)$;*
3. *For every $C \in \mathcal{C}$, the induced subgraph $\mathcal{H}[C]$ has at most $\epsilon |E(\mathcal{H})|$ many edges.*

The main result in this section is the following theorem.

► **Theorem 8.** *Let k, r be fixed integers such that $r \geq k \geq 2$ and k is even. Then for any $0 < \alpha < 1$, there are constants $c = c(\alpha, k, r)$ and $d = d(\alpha, k, r)$ such that the following holds. For infinitely many values of N , there is a set V of N points in \mathbb{R}^d such that no $r + 3$ members of V lie on an r -flat, and every subset of V of size $cN^{\frac{r+2}{2(k+1)} + \alpha}$ contains $k + 2$ members on a k -flat.*

Before we prove Theorem 8, let us show that it implies Theorem 1. In dimensions $d_0 \geq 3$ where d_0 is odd, we apply Theorem 8 with $k = r = d_0 - 1$ to obtain a point set V in \mathbb{R}^d with the property that no $d_0 + 2$ members lie on a $(d_0 - 1)$ -flat, and every subset of size $cN^{\frac{1}{2} + \frac{1}{2d_0} + \alpha}$ contains $d_0 + 1$ members on a $(d_0 - 1)$ -flat. By projecting V to a generic d_0 -dimensional subspace of \mathbb{R}^d , we obtain N points in \mathbb{R}^{d_0} with no $d_0 + 2$ members on a common hyperplane, and no $cN^{\frac{1}{2} + \frac{1}{2d_0} + \alpha}$ members in general position.

In dimensions $d_0 \geq 4$ where d_0 is even, we apply Theorem 8 with $k = d_0 - 2$ and $r = d_0 - 1$ to obtain a point set V in \mathbb{R}^d with the property that no $d_0 + 2$ members on a $(d_0 - 1)$ -flat, and every subset of size $cN^{\frac{1}{2} + \frac{1}{d_0 - 1} + \alpha}$ contains d_0 members on a $(d_0 - 2)$ -flat. By adding another point from this subset, we obtain $d_0 + 1$ members on a $(d_0 - 1)$ -flat. Hence, by projecting to V a generic d_0 -dimensional subspace of \mathbb{R}^d , we obtain N points in \mathbb{R}^{d_0} with no $d_0 + 2$ members on a common hyperplane, and no $cN^{\frac{1}{2} + \frac{1}{d_0 - 1} + \alpha}$ members in general position. This completes the proof of Theorem 1.

Proof of Theorem 8. We set $d = d(\alpha, k, r)$ to be a sufficiently large integer depending on α , k , and r . Let \mathcal{H} be the hypergraph with $V(\mathcal{H}) = [n]^d$ and $E(\mathcal{H})$ consists of non-degenerate $(k+2)$ -tuples T such that T lies on a k -flat. Let $\mathcal{C}^0 = [n]^d$, $\mathcal{C}^0 = \{C^0\}$, and $\mathcal{H}^0 = \mathcal{H}$. In what follows, we will apply the hypergraph container lemma to \mathcal{H}^0 to obtain a family of containers \mathcal{C}^1 . For each $C_j^1 \in \mathcal{C}^1$, we consider the induced hypergraph $\mathcal{H}_j^1 = \mathcal{H}[C_j^1]$, and we apply the hypergraph container lemma to it. The collection of containers obtained from all \mathcal{H}_j^1 will form another collection of containers \mathcal{C}^2 . We iterate this process until each container in \mathcal{C}^i is sufficiently small, and moreover, we will only produce a small number of containers. As a final step, we apply the probabilistic method to show the existence of the desired point set. We now flesh out the details of this process.

We start by setting $\mathcal{C}^0 = [n]^d$, $\mathcal{C}^0 = \{C^0\}$, and set $\mathcal{H}^0 = \mathcal{H}[C^0] = \mathcal{H}$. Having obtained a collection of containers \mathcal{C}^i , for each container $C_j^i \in \mathcal{C}^i$ with $|C_j^i| \geq n^{\frac{k}{k+1}d+k}$, we set $\mathcal{H}_j^i = \mathcal{H}[C_j^i]$. Let $\gamma = \gamma(i, j)$ be defined by $|V(\mathcal{H}_j^i)| = n^{d-\gamma}$. So, $\gamma \leq \frac{d}{k+1} - k$. We set $\tau = \tau(i, j) = n^{-\frac{k}{k+1}d+\gamma+\alpha}$ and $\epsilon = \epsilon(i, j) = c_1 n^{-\alpha}$, where $c_1 = c_1(d, k)$ is a sufficiently large constant depending on d and k . Then we can verify the following condition.

▷ **Claim 9.** $\Delta(\mathcal{H}_j^i, \tau) \leq \epsilon / (12 \cdot (k+2)!)$.

Proof. Since $|V(\mathcal{H}_j^i)| = n^{d-\gamma}$, $\gamma \leq \frac{d}{k+1} - k$, and d is sufficiently large, Lemma 3 implies that $|E(\mathcal{H}_j^i)| \geq c_2 n^{(k+1)d-(k+2)\gamma}$ for some constant $c_2 = c_2(d, k)$. Hence, we have

$$\frac{|V(\mathcal{H}_j^i)|}{|E(\mathcal{H}_j^i)|} \leq \frac{n^{d-\gamma}}{c_2 n^{(k+1)d-(k+2)\gamma}} = \frac{1}{c_2 n^{kd-(k+1)\gamma}}.$$

On the other hand, by Lemma 6, we have

$$\Delta_\ell(\mathcal{H}_j^i) \leq n^{(d-\gamma)(k+1-\ell)+k} \quad \text{for } \ell < k+2,$$

and obviously $\Delta_{k+2}(\mathcal{H}_j^i) \leq 1$.

Applying these inequalities together with the definition of Δ , we obtain

$$\begin{aligned} \Delta(\mathcal{H}_j^i, \tau) &= \frac{2^{\binom{k+2}{2}-1} |V(\mathcal{H}_j^i)|}{(k+2) |E(\mathcal{H}_j^i)|} \sum_{\ell=2}^{k+2} \frac{\Delta_\ell(\mathcal{H}_j^i)}{\tau^{\ell-1} 2^{\binom{\ell-1}{2}}} \\ &\leq \frac{c_3}{n^{kd-(k+1)\gamma}} \left(\sum_{\ell=2}^{k+1} \frac{n^{(k+1-\ell)(d-\gamma)+k}}{\tau^{\ell-1}} + \frac{1}{\tau^{k+1}} \right) \\ &= \sum_{\ell=2}^{k+1} \frac{c_3}{\tau^{\ell-1} n^{(\ell-1)d-k-\ell\gamma}} + \frac{c_3}{\tau^{k+1} n^{kd-(k+1)\gamma}}, \end{aligned}$$

for some constant $c_3 = c_3(d, k)$. Let us remark that the summation above is where we determined our τ and γ . In order to make the last term small, we choose $\tau = n^{-\frac{k}{k+1}d+\gamma+\alpha}$. Having determined τ , in order for the first term in the summation to be small, we choose $\gamma \leq \frac{d}{k+1} - k$.

By setting $\epsilon = c_1 n^{-\alpha}$ with $c_1 = c_1(d, k)$ sufficiently large, we have

$$\begin{aligned} \Delta(\mathcal{H}_j^i, \tau) &\leq c_3 \left(\sum_{\ell=2}^{k+1} n^{-\frac{\ell-1}{k+1}d+\gamma+k-(\ell-1)\alpha} + n^{-(k+1)\alpha} \right) \\ &\leq c_3 k n^{-\alpha} + c_3 n^{-(k+1)\alpha} \\ &< \frac{\epsilon}{12(k+2)!}. \end{aligned}$$

This verifies the claimed condition. ◀

Given the condition above, we can apply Lemma 7 to \mathcal{H}_j^i with chosen parameters τ and ϵ . Hence we obtain a family of containers \mathcal{C}_j^{i+1} such that

$$\begin{aligned} |\mathcal{C}_j^{i+1}| &\leq 2^{10^3(k+2)((k+2)!)^3|V(\mathcal{H}_j^i)|\tau \log(1/\epsilon) \log(1/\tau)} \\ &\leq 2^{c_4 n^{\frac{d}{k+1} + \alpha} \log^2 n}, \end{aligned}$$

for some constant $c_4 = c_4(d, k)$. In the other case where $|\mathcal{C}_j^i| < n^{\frac{k}{k+1}d+k}$, we just define $\mathcal{C}_j^{i+1} = \{\mathcal{C}_j^i\}$. Then, for each container $C \in \mathcal{C}_j^{i+1}$, we have either $|C| < n^{\frac{k}{k+1}d+k}$ or $|E(\mathcal{H}[C])| \leq \epsilon|E(\mathcal{H}_j^i)| \leq \epsilon^i|E(\mathcal{H})|$. After applying this procedure for each container in \mathcal{C}^i , we obtain a new family of containers $\mathcal{C}^{i+1} = \bigcup \mathcal{C}_j^i$ such that

$$|\mathcal{C}^{i+1}| \leq |\mathcal{C}^i| 2^{c_4 n^{\frac{d}{k+1} + \alpha} \log^2 n} \leq 2^{(i+1)c_4 n^{\frac{d}{k+1} + \alpha} \log^2 n}.$$

Notice that the number of edges in \mathcal{H}_j^i shrinks by a factor of $c_1 n^{-\alpha}$ whenever i increases by one, while on the other hand, Lemma 3 tells us that every large subset $C \subset [n]^d$ induces many edges in \mathcal{H} . Hence, after at most $t \leq c_5/\alpha$ iterations, for some constant $c_5 = c_5(d, k)$, we obtain a collection of containers $\mathcal{C} = \mathcal{C}^t$ such that: each container $C \in \mathcal{C}$ satisfies $|C| < n^{\frac{k}{k+1}d+k}$; every independent set of \mathcal{H} is a subset of some $C \in \mathcal{C}$; and

$$|\mathcal{C}| \leq 2^{(c_5/\alpha)c_4 n^{\frac{d}{k+1} + \alpha} \log^2 n}.$$

Before we construct the desired point set, we make the following crude estimate.

▷ **Claim 10.** The grid $[n]^d$ contains at most $O(n^{(r+1)d+2r})$ many $(r+3)$ -tuples that lie on a r -flat.

Proof. Let T be an arbitrary $(r+3)$ -tuple that spans a j -flat. There are at most $n^{(j+1)d}$ ways to choose a subset $T' \subset T$ of size $j+1$ that spans the affine hull of T . After this T' is determined, there are at most $n^{(r+2-j)j}$ ways to add the remaining $r+2-j$ points from the j -flat spanned by T' . Then the total number of $(r+3)$ -tuples that lie on a r -flat is at most

$$\sum_{j=1}^r n^{(j+1)d+(r+2-j)j} \leq \sum_{j=1}^r n^{(j+1)d+(r+2-j)r} \leq r n^{(r+1)d+2r},$$

since we can assume $d > r$. ◀

Now, we randomly select a subset of $[n]^d$ by keeping each point independently with probability p . Let S be the set of selected elements. Then for each $(r+3)$ -tuple T in S that lies on an r -flat, we delete one point from T . We denote the resulting set of points by S' . By the claim above, the number of $(r+3)$ -tuples in $[n]^d$ that lie on a r -flat is at most $c_6 n^{(r+1)d+2r}$ for some constant $c_6 = c_6(r)$. Therefore,

$$\mathbb{E}[|S'|] \geq pn^d - c_6 p^{r+3} n^{(r+1)d+2r}.$$

By setting $p = (2c_6)^{-\frac{1}{r+2}} n^{-\frac{r}{r+2}(d+2)}$, we have

$$\mathbb{E}[|S'|] \geq \frac{pn^d}{2} = \Omega\left(n^{\frac{2(d-r)}{r+2}}\right).$$

Finally, we set $m = (c_7/\alpha)n^{\frac{d}{k+1} + 2\alpha}$ for some sufficiently large constant $c_7 = c_7(d, k, r)$. Let X denote the number of independent sets of size m in S' . Using the family of containers

\mathcal{C} , we have

$$\begin{aligned}
 \mathbb{E}[X] &\leq |\mathcal{C}| \cdot \binom{n^{\frac{k}{k+1}d+k}}{m} p^m \\
 &\leq \left(2^{(c_5/\alpha)c_4 n^{\frac{d}{k+1}+\alpha} \log^2 n} \right) \left(\frac{en^{\frac{k}{k+1}d+k} p}{m} \right)^m \\
 &\leq \left(2^{(c_5/\alpha)c_4 n^{\frac{d}{k+1}+\alpha} \log^2 n} \right) \left(c_8 \alpha \frac{n^{\frac{k}{k+1}d+k} \cdot n^{-\frac{r}{r+2}(d+2)}}{n^{\frac{d}{k+1}+2\alpha}} \right)^m \\
 &\leq \left(2^{(c_5/\alpha)c_4 n^{\frac{d}{k+1}+\alpha} \log^2 n} \right) \left(c_8 \alpha n^{\frac{2(k-r-1)d}{(k+1)(r+2)} + k - \frac{2r}{r+2} - 2\alpha} \right)^{(c_7/\alpha) n^{\frac{d}{k+1}+2\alpha}},
 \end{aligned}$$

for some constant $c_8 = c_8(d, k, r)$. Since $r \geq k$, $0 < \alpha < 1$, and d is large, for n sufficiently large, we have

$$c_8 \alpha n^{\frac{2(k-r-1)d}{(k+1)(r+2)} + k - \frac{2r}{r+2} - 2\alpha} < 1/2.$$

Hence, we have $\mathbb{E}[X] \leq o(1)$ as n tends to infinity. Notice that $|S'|$ is exponentially concentrated around its mean by Chernoff's inequality. Therefore, some realization of S' satisfies: $|S'| = N = \Omega(n^{2(d-r)/(r+2)})$; S' contains no $(r+3)$ -tuples on a r -flat; and $\mathcal{H}[S']$ does not contain an independent set of size

$$m = (c_7/\alpha) n^{\frac{d}{k+1}+2\alpha} \leq cN^{\frac{r+2}{2(k+1)} + \frac{(r+2)r}{2(k+1)(d-r)} + \frac{r+2}{d} 2\alpha} \leq cN^{\frac{r+2}{2(k+1)} + \alpha},$$

for some constant $c = c(\alpha, d, k, r)$. Here we assume d is sufficiently large so that

$$\frac{(r+2)r}{2(k+1)(d-r)} + \frac{r+2}{d} 2\alpha \leq \alpha.$$

This completes the proof. ◀

4 Avoiding non-trivial solutions: Proof of Theorem 2

In this section, we will give a proof of Theorem 2. Let $V \subset [n]^d$ such that there are no $k+2$ points that lie on a k -flat. In [17], Lefmann showed that $|V| \leq O\left(n^{\frac{d}{\lceil (k+2)/2 \rceil}}\right)$. To see this, assume that k is even and consider all elements of the form $v_1 + \dots + v_{\frac{k}{2}+1}$, where $v_i \neq v_j$ and $v_i \in V$. All of these elements are distinct, since otherwise we would have $k+2$ points on a k -flat. In other words, the equation

$$\left(\mathbf{x}_1 + \dots + \mathbf{x}_{\frac{k}{2}+1} \right) - \left(\mathbf{x}_{\frac{k}{2}+2} + \dots + \mathbf{x}_{k+2} \right) = \mathbf{0},$$

does not have a solution with $\{\mathbf{x}_1, \dots, \mathbf{x}_{\frac{k}{2}+1}\}$ and $\{\mathbf{x}_{\frac{k}{2}+2}, \dots, \mathbf{x}_{k+2}\}$ being two different $(\frac{k}{2}+1)$ -tuples of V . Therefore, we have $\binom{|V|}{\frac{k}{2}+1} \leq (kn)^d$, and this implies Lefmann's bound.

More generally, let us consider the equation

$$c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \dots + c_r \mathbf{x}_r = \mathbf{0}, \tag{1}$$

with constant coefficients $c_i \in \mathbb{Z}$ and $\sum_i c_i = 0$. Here, the variables \mathbf{x}_i takes value in \mathbb{Z}^j . A solution $(\mathbf{x}_1, \dots, \mathbf{x}_r)$ to equation (1) is called *trivial* if there is a partition $\mathcal{P} : [r] = \mathcal{I}_1 \cup \dots \cup \mathcal{I}_t$, such that $\mathbf{x}_j = \mathbf{x}_\ell$ if and only if $j, \ell \in \mathcal{I}_i$, and $\sum_{j \in \mathcal{I}_i} c_j = 0$ for all $i \in [t]$. In other words,

being trivial means that, after combining like terms, the coefficient of each \mathbf{x}_i becomes zero. Otherwise, we say that the solution $(\mathbf{x}_1, \dots, \mathbf{x}_r)$ is *non-trivial*. A natural extremal problem is to determine the maximum size of a set $A \subset [n]^d$ with only trivial solutions to (1). When $d = 1$, this is a classical problem in additive number theory, and we refer the interested reader to [23, 19, 15, 6].

By combining the arguments of Cilleruelo and Timmons [6] and Jia [14], we establish the following theorem.

► **Theorem 11.** *Let d, r be fixed positive integers. Suppose $V \subset [n]^d$ has only trivial solutions to each equation of the form*

$$c_1((\mathbf{x}_1 + \dots + \mathbf{x}_r) - (\mathbf{x}_{r+1} + \dots + \mathbf{x}_{2r})) = c_2((\mathbf{x}_{2r+1} + \dots + \mathbf{x}_{3r}) - (\mathbf{x}_{3r+1} + \dots + \mathbf{x}_{4r})), \tag{2}$$

for integers c_1, c_2 such that $1 \leq c_1, c_2 \leq n^{\frac{d}{2r(d+1)}}$. Then we have

$$|V| \leq O\left(n^{\frac{d}{2r}\left(1 - \frac{1}{2r(d+1)}\right)}\right).$$

Notice that Theorem 2 follows from Theorem 11. Indeed, when $k + 2$ is divisible by 4, we set $r = (k + 2)/4$. If $V \subset [n]^d$ contains $k + 2$ points $\{v_1, \dots, v_{k+2}\}$ that is a non-trivial solution to (2) with $\mathbf{x}_i = v_i$, then $\{v_1, \dots, v_{k+2}\}$ must lie on a k -flat. Hence, when $k + 2$ is divisible by 4, we have

$$a(d, k, n) \leq O\left(n^{\frac{d}{(k+2)/2}\left(1 - \frac{1}{(k+2)d/2+1}\right)}\right).$$

Since we have $a(d, k, n) < a(d, k - 1, n)$, this implies that for all $k \geq 2$, we have

$$a(d, k, n) \leq O\left(n^{\frac{d}{2\lfloor (k+2)/4 \rfloor}\left(1 - \frac{1}{2\lfloor (k+2)/4 \rfloor d+1}\right)}\right).$$

In the proof of Theorem 11, we need the following well-known lemma (see e.g. [6]Lemma 2.1 and [23]Theorem 4.1). For $U, T \subset \mathbb{Z}^d$ and $x \in \mathbb{Z}^d$, we define

$$\Phi_{U-T}(x) = \{(u, t) : u - t = x, u \in U, t \in T\}.$$

► **Lemma 12.** *For finite sets $U, T \subset \mathbb{Z}^d$, we have*

$$\frac{(|U||T|)^2}{|U+T|} \leq \sum_{x \in \mathbb{Z}^d} |\Phi_{U-U}(x)| \cdot |\Phi_{T-T}(x)|.$$

Proof of Theorem 11. Let d, r , and V be as given in the hypothesis. Let $m \geq 1$ be an integer that will be determined later. We define

$$S_r = \{v_1 + \dots + v_r : v_i \in V, v_i \neq v_j\},$$

and a function

$$\sigma : \binom{V}{r} \rightarrow S_r, \{v_1, \dots, v_r\} \mapsto v_1 + \dots + v_r.$$

Notice that σ is a bijection. Indeed, suppose on the contrary that

$$v_1 + \dots + v_r = v'_1 + \dots + v'_r$$

59:10 On Higher Dimensional Point Sets in General Position

for two different r -tuples in V . Then by setting $(\mathbf{x}_1, \dots, \mathbf{x}_r) = (v_1, \dots, v_r)$, $(\mathbf{x}_{r+1}, \dots, \mathbf{x}_{2r}) = (v'_1, \dots, v'_r)$, $(\mathbf{x}_{2r+1}, \dots, \mathbf{x}_{3r}) = (\mathbf{x}_{3r+1}, \dots, \mathbf{x}_{4r})$ arbitrarily, and $c_1 = c_2 = 1$, we obtain a non-trivial solution to (2), which is a contradiction. In particular, we have $|S_r| = \binom{|V|}{r}$.

For $j \in [m]$ and $w \in \mathbb{Z}_j^d$, we let

$$U_{j,w} = \{u \in \mathbb{Z}^d : ju + w \in S_r\}.$$

Notice that for fixed $j \in [m]$, we have

$$\sum_{w \in \mathbb{Z}_j^d} |U_{j,w}| = \sum_{w \in \mathbb{Z}_j^d} |\{v \in S_r : v \equiv w \pmod{j}\}| = |S_r|.$$

Applying Jensen's inequality to above, we have

$$\sum_{w \in \mathbb{Z}_j^d} |U_{j,w}|^2 \geq |S_r|^2 / j^d. \quad (3)$$

For $i \geq 0$, we define

$$\Phi_{U_{j,w}-U_{j,w}}^i(x) = \{(u_1, u_2) \in \Phi_{U_{j,w}-U_{j,w}}(x) : |\sigma^{-1}(ju_1 + w) \cap \sigma^{-1}(ju_2 + w)| = i\}.$$

It's obvious that these sets form a partition of $\Phi_{U_{j,w}-U_{j,w}}(x)$. We also make the following claims.

▷ **Claim 13.** For a fixed $x \in \mathbb{Z}^d$, we have

$$\sum_{j \in [m]} \sum_{w \in \mathbb{Z}_j^d} |\Phi_{U_{j,w}-U_{j,w}}^0(x)| \leq 1,$$

Proof. For the sake of contradiction, suppose the summation above is at least two, then we have $(u_1, u_2) \in \Phi_{U_{j,w}-U_{j,w}}^0(x)$ and $(u_3, u_4) \in \Phi_{U_{j',w'}-U_{j',w'}}^0(x)$ such that either $(u_1, u_2) \neq (u_3, u_4)$ or $(j, w) \neq (j', w')$.

Let $s_1, s_2, s_3, s_4 \in S_r$ such that $s_1 = ju_1 + w$, $s_2 = ju_2 + w$, $s_3 = j'u_3 + w'$, $s_4 = j'u_4 + w'$ and write $\sigma^{-1}(s_i) = \{v_{i,1}, \dots, v_{i,r}\}$. Notice that $u_1 - u_2 = x = u_3 - u_4$. Putting these equations together gives us

$$j'((v_{1,1} + \dots + v_{1,r}) - (v_{2,1} + \dots + v_{2,r})) = j((v_{3,1} + \dots + v_{3,r}) - (v_{4,1} + \dots + v_{4,r})). \quad (4)$$

It suffices to show that (4) can be seen as a non-trivial solution to (2). The proof now falls into the following cases.

Case 1. Suppose $j \neq j'$. Without loss of generality we can assume $j' > j$. Notice that $(u_1, u_2) \in \Phi_{U_{j,w}-U_{j,w}}^0(x)$ implies

$$\{v_{1,1}, \dots, v_{1,r}\} \cap \{v_{2,1}, \dots, v_{2,r}\} = \emptyset.$$

Then after combining like terms in (4), the coefficient of v_1^1 is at least $j' - j$, which means this is indeed a non-trivial solution to (2).

Case 2. Suppose $j = j'$, then we must have $s_1 \neq s_3$. Indeed, if $s_1 = s_3$, we must have $w = w'$ (as s_1 modulo j equals s_3 modulo j') and $s_2 = s_4$ (as $j'(s_1 - s_2) = j(s_3 - s_4)$). This is a contradiction to either $(u_1, u_2) \neq (u_3, u_4)$ or $(j, w) \neq (j', w')$.

Given $s_1 \neq s_3$, we can assume, without loss of generality, $v_{1,1} \notin \{v_{3,1}, \dots, v_{3,r}\}$. Again, we have $\{v_{1,1}, \dots, v_{1,r}\} \cap \{v_{2,1}, \dots, v_{2,r}\} = \emptyset$. Hence, after combining like terms in (4), the coefficient of v_1^1 is positive and we have a non-trivial solution to (2). ◀

▷ Claim 14. For a finite set $T \subset \mathbb{Z}^d$, and fixed integers $i, j \geq 1$, we have

$$\sum_{w \in \mathbb{Z}_j^d} \sum_{x \in \mathbb{Z}^d} |\Phi_{U_{j,w}-U_{j,w}}^i(x)| \cdot |\Phi_{T-T}(x)| \leq |V|^{2r-i}|T|.$$

Proof. The summation on the left-hand side counts all (ordered) quadruples (u_1, u_2, t_1, t_2) such that $(u_1, u_2) \in \Phi_{U_{j,w}-U_{j,w}}^i(t_1 - t_2)$. For each such a quadruple, let $s_1, s_2 \in S_r$ such that

$$s_1 = ju_1 + w \quad \text{and} \quad s_2 = ju_2 + w.$$

There are at most $|V|^{2r-i}$ ways to choose a pair (s_1, s_2) satisfying $|\sigma^{-1}(s_1) \cap \sigma^{-1}(s_2)| = i$. Such a pair (s_1, s_2) determines (u_1, u_2) uniquely. Moreover, (s_1, s_2) also determines the quantity

$$t_1 - t_2 = u_1 - u_2 = \frac{s_1 - w}{j} - \frac{s_2 - w}{j} = \frac{1}{j}(s_1 - s_2).$$

After such a pair (s_1, s_2) is chosen, there are at most $|T|$ ways to choose t_1 and this will also determine t_2 . So we conclude the claim by multiplication. ◀

Now, we set $T = \mathbb{Z}_\ell^d$ for some integer ℓ to be determined later. Notice that $U_{j,w} + T \subset \{0, 1, \dots, \lfloor rn/j \rfloor + \ell - 1\}^d$, which implies

$$|U_{j,w} + T| \leq (rn/j + \ell)^d. \tag{5}$$

By Lemma 12, we have

$$\frac{|U_{j,w}|^2|T|^2}{|U_{j,w} + T|} \leq \sum_{x \in \mathbb{Z}^d} |\Phi_{U_{j,w}-U_{j,w}}(x)| \cdot |\Phi_{T-T}(x)|.$$

Summing over all $j \in [m]$ and $w \in \mathbb{Z}_j^d$, and using Claims 13 and 14, we can compute

$$\begin{aligned} \sum_{j \in [m]} \sum_{w \in \mathbb{Z}_j^d} \frac{|U_{j,w}|^2|T|^2}{|U_{j,w} + T|} &\leq \sum_{j \in [m]} \sum_{w \in \mathbb{Z}_j^d} \sum_{x \in \mathbb{Z}^d} |\Phi_{U_{j,w}-U_{j,w}}(x)| \cdot |\Phi_{T-T}(x)| \\ &= \sum_{x \in \mathbb{Z}^d} \sum_{j \in [m]} \sum_{w \in \mathbb{Z}_j^d} \left(|\Phi_{U_{j,w}-U_{j,w}}^0(x)| + \sum_{i=1}^r |\Phi_{U_{j,w}-U_{j,w}}^i(x)| \right) |\Phi_{T-T}(x)| \\ &\leq \sum_{x \in \mathbb{Z}^d} |\Phi_{T-T}(x)| \sum_{j \in [m]} \sum_{w \in \mathbb{Z}_j^d} |\Phi_{U_{j,w}-U_{j,w}}^0(x)| + \sum_{j \in [m]} \sum_{i=1}^r |V|^{2r-i} \ell^d \\ &\leq \sum_{x \in \mathbb{Z}^d} |\Phi_{T-T}(x)| + \sum_{j \in [m]} \sum_{i=1}^{r-1} |V|^{2r-i} \ell^d \\ &\leq \ell^{2d} + rm|V|^{2r-1} \ell^d, \end{aligned}$$

On the other hand, using (3) and (5), we can compute

$$\begin{aligned}
 \sum_{j \in [m]} \sum_{w \in \mathbb{Z}_j^d} \frac{|U_{j,w}|^2 |T|^2}{|U_{j,w} + T|} &\geq \sum_{j \in [m]} \sum_{w \in \mathbb{Z}_j^d} \frac{|U_{j,w}|^2 \ell^{2d}}{(rn/j + \ell)^d} \\
 &\geq \sum_{j \in [m]} \frac{|S_r|^2 \ell^{2d}}{j^d (rn/j + \ell)^d} \\
 &= \sum_{j \in [m]} \frac{|S_r|^2 \ell^{2d}}{(rn + j\ell)^d} \\
 &\geq \frac{m|S_r|^2 \ell^{2d}}{(rn + m\ell)^d},
 \end{aligned}$$

Combining the two inequalities above gives us

$$\begin{aligned}
 \frac{m|S_r|^2 \ell^{2d}}{(rn + m\ell)^d} &\leq \ell^{2d} + rm|V|^{2r-1} \ell^d \\
 \implies |S_r|^2 &\leq \frac{(rn + m\ell)^d}{m} + r|V|^{2r-1} \frac{(rn + m\ell)^d}{\ell^d}.
 \end{aligned}$$

By setting $m = n^{\frac{d}{2r d+1}}$ and $\ell = n^{1 - \frac{d}{2r d+1}}$, we get

$$\binom{|V|}{r}^2 = |S_r|^2 \leq cn^{d - \frac{d}{2r d+1}} + c|V|^{2r-1} n^{\frac{d^2}{2r d+1}},$$

for some constant c depending only on d and r . We can solve from this inequality that

$$|V| = O\left(n^{\frac{d}{2r} \left(1 - \frac{1}{2r d+1}\right)}\right),$$

completing the proof. \blacktriangleleft

5 Concluding remarks

1. One can consider a generalization of the quantity $\alpha_d(N)$. We let $\alpha_{d,s}(N)$ be the largest integer such that any set of N points in \mathbb{R}^d with no $d+s$ members on a hyperplane, contains $\alpha_{d,s}(N)$ points in general position. Hence, $\alpha_d(N) = \alpha_{d,2}(N)$. Following the arguments in our proof of Theorem 1 with a slight modification, we show the following.

► **Theorem 15.** *Let $d, s \geq 3$ be fixed integers. If d is odd and $\frac{2d+s-2}{2d+2s-2} < \frac{d-1}{d}$, then $\alpha_{d,s}(N) \leq N^{\frac{1}{2} + o(1)}$. If d is even and $\frac{2d+s-2}{2d+2s-2} < \frac{d-2}{d-1}$, then $\alpha_{d,s}(N) \leq N^{\frac{1}{2} + o(1)}$.*

For example, when we fix $d = 3$ and $s \geq 5$, we have $\alpha_{d,s}(N) \leq N^{\frac{1}{2} + o(1)}$. In the other direction, it is easy to show that $\alpha_{d,s}(N) \geq \Omega(N^{1/d})$ for any fixed $d, s \geq 2$ (see [8]).

► **Problem 16.** *Are there fixed integers $d, s \geq 3$ such that $\alpha_{d,s}(N) \leq o(N^{\frac{1}{2}})$?*

2. We call a subset $V \subset [n]^d$ an m -fold B_g -set if V only contains trivial solutions to the equations

$$c_1 \mathbf{x}_1 + c_2 \mathbf{x}_2 + \cdots + c_g \mathbf{x}_g = c_1 \mathbf{x}'_1 + c_2 \mathbf{x}'_2 + \cdots + c_g \mathbf{x}'_g,$$

with constant coefficients $c_i \in [m]$. We call 1-fold B_g -sets simply B_g -sets. By counting distinct sums, we have an upper bound $|V| \leq O(n^{\frac{d}{g}})$ for any B_g -set $V \subset [n]^d$.

Our Theorem 11 can be interpreted as the following phenomenon: by letting m grow as some proper polynomial in n , we have an upper bound for m -fold B_g -sets, where g is even, which gives a polynomial-saving improvement from the trivial $O(n^{\frac{d}{g}})$ bound. We believe this phenomenon should also hold without the parity condition on g .

References

- 1 József Balogh, Robert Morris, and Wojciech Samotij. Independent sets in hypergraphs. *Journal of the American Mathematical Society*, 28(3):669–709, 2015.
- 2 József Balogh, Robert Morris, and Wojciech Samotij. The method of hypergraph containers. In *Proceedings of the International Congress of Mathematicians: Rio de Janeiro 2018*, pages 3059–3092. World Scientific, 2018.
- 3 József Balogh and József Solymosi. On the number of points in general position in the plane. *Discrete Analysis*, 16:20pp, 2018.
- 4 Peter Braß and Christian Knauer. On counting point-hyperplane incidences. *Computational Geometry*, 25(1-2):13–20, 2003.
- 5 Jean Cardinal, Csaba D Tóth, and David R Wood. General position subsets and independent hyperplanes in d -space. *Journal of geometry*, 108:33–43, 2017.
- 6 Javier Cilleruelo and Craig Timmons. k -fold Sidon sets. *Electronic Journal of Combinatorics*, 21(4):P4–12, 2014.
- 7 Henry E Dudeney. *Amusements in Mathematics*. Nelson, London, 1917.
- 8 Paul Erdős. On some metric and combinatorial geometric problems. *Discrete Mathematics*, 60:147–153, 1986.
- 9 Achim Flammenkamp. Progress in the no-three-in-line problem, ii. *Journal of Combinatorial Theory, Series A*, 81(1):108–113, 1998.
- 10 Zoltán Füredi. Maximal independent subsets in Steiner systems and in planar sets. *SIAM Journal on Discrete Mathematics*, 4(2):196–199, 1991.
- 11 H Furstenberg and Y Katznelson. A density version of the Hales–Jewett theorem for $k = 3$. *Discrete Mathematics*, 75(1-3):227–241, 1989.
- 12 Hillel Furstenberg and Yitzhak Katznelson. A density version of the Hales–Jewett theorem. *Journal d’Analyse Mathématique*, 57(1):64–119, 1991.
- 13 Richard R Hall, Terence H Jackson, Anthony Sudbery, and Ken Wild. Some advances in the no-three-in-line problem. *Journal of Combinatorial Theory, Series A*, 18(3):336–341, 1975.
- 14 Xing De Jia. On finite Sidon sequences. *Journal of number theory*, 44(1):84–92, 1993.
- 15 Felix Lazebnik and Jacques Verstraëte. On hypergraphs of girth five. *Electronic Journal of Combinatorics*, 10(1):R25, 2003.
- 16 Hanno Lefmann. No ℓ grid-points in spaces of small dimension. In *Algorithmic Aspects in Information and Management: 4th International Conference, AAIM 2008, Shanghai, China, June 23-25, 2008. Proceedings 4*, pages 259–270. Springer, 2008.
- 17 Hanno Lefmann. Extensions of the no-three-in-line problem. *preprint*, 2012. URL: www.tu-chemnitz.de/informatik/THIS/downloads/publications/lefmann_no_three_submitted.pdf.
- 18 Luka Milićević. Sets in almost general position. *Combinatorics, Probability and Computing*, 26(5):720–745, 2017.
- 19 Kevin O’Bryant. A complete annotated bibliography of work related to sidon sequences. *Electronic Journal of Combinatorics*, pages 39–p, 2004.
- 20 Kevin T Phelps and Vojtech Rödl. Steiner triple systems with minimum independence number. *Ars combinatoria*, 21:167–172, 1986.
- 21 Attila Pór and David R Wood. No-three-in-line-in-3D. *Algorithmica*, 47(4):481–488, 2007.
- 22 Klaus F Roth. On a problem of Heilbronn. *Journal of the London Mathematical Society*, 1(3):198–204, 1951.
- 23 Imre Z Ruzsa. Solving a linear equation in a set of integers I. *Acta Arithmetica*, 65(3):259–282, 1993.
- 24 David Saxton and Andrew Thomason. Hypergraph containers. *Inventiones mathematicae*, 201(3):925–992, 2015.
- 25 Benny Sudakov and István Tomon. Evasive sets, covering by subspaces, and point-hyperplane incidences. *arXiv preprint arXiv:2207.13077*, 2022.

Slice, Simplify and Stitch: Topology-Preserving Simplification Scheme for Massive Voxel Data

Hubert Wagner  

University of Florida, Gainesville, FL, USA

Abstract

We focus on efficient computations of topological descriptors for voxel data. This type of data includes 2D greyscale images, 3D medical scans, but also higher-dimensional scalar fields arising from physical simulations. In recent years we have seen an increase in applications of topological methods for such data. However, computational issues remain an obstacle.

We therefore propose a streaming scheme which simplifies large 3-dimensional voxel data – while provably retaining its persistent homology. We combine this scheme with an efficient boundary matrix reduction implementation, obtaining an end-to-end tool for persistent homology of large data. Computational experiments show its state-of-the-art performance. In particular, we are now able to robustly handle complex datasets with several billions voxels on a regular laptop.

A software implementation called Cubicle is available as open-source: <https://bitbucket.org/hubwag/cubicle>.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Combinatorial algorithms

Keywords and phrases Computational topology, topological data analysis, topological image analysis, persistent homology, persistence diagram, discrete Morse theory, algorithm engineering, implementation, voxel data, volume data, image data

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.60

Supplementary Material *Software:* <https://bitbucket.org/hubwag/cubicle>
archived at `swh:1:dir:5f25601ea576ea1a004c37d94e2e2f5b94b9c00d`

Funding Supported by the 2022 Google Research Scholar Award in Algorithms and Optimization.

Acknowledgements I would like to thank Herbert Edelsbrunner, Teresa Heiss, Kevin Knudson, Marian Mrozek, Georg Osang and Vanessa Robins for their helpful comments.

1 Introduction

Persistent homology is one of the most popular tools offered by the field of Topological Data Analysis (TDA). It provides a rich geometric-topological descriptor of data called the persistence diagram. Point cloud data is a natural choice and a significant portion of theory, and algorithms and software focuses on this setting.

However, persistent homology is also becoming increasingly useful for other types of data – especially when used in conjunction with modern machine learning tools [23, 24, 37]. In this paper, we turn our interest to scalar voxel data in dimension 2 and 3. This type of data includes 2D gray-scale images, 3D medical scans, but also scalar fields coming from physical simulations. Intermediate results of convolutional neural networks (i.e. feature maps) are another interesting case. In all these settings, voxel data encodes potentially useful – and often intricate – geometry and topology.

One key challenge is that such datasets are often large – counted in billions of voxels or more. Currently, for such data we can compute topological descriptors such as connected components [29], merge trees [32], contour trees [17] and the Euler characteristic curve [21, 38]. While these are useful tools, there is a compromise: the first three discard higher-order topological information, and the last one forfeits information about the persistence of topological features.



© Hubert Wagner;

licensed under Creative Commons License CC-BY 4.0

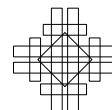
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 60; pp. 60:1–60:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



Our goal is therefore to provide a method for exact computation of persistent homology of large 3D voxel data. We propose an efficient streaming computation scheme, prove its correctness and benchmark its implementation¹. In particular, we experimentally show that for large data it is the fastest method available. Our method also uses significantly less memory, which is crucial because memory usage is the main bottleneck of existing methods. With these improvements we were able to apply our method to several practical datasets with up to 2048^3 voxels on a regular laptop. Existing software was limited to 256^3 or 512^3 .

On a technical side, the efficiency of our approach is achieved by (1) *streaming* the input slice by slice, (2) efficient parallel implementation and (3) realizing selected parts of the pipeline as *external-memory* algorithms. This allows us to save memory, while ensuring that topological features spanning multiple slices are correctly captured.

Focus of the paper. We focus on: (1) conveying the main ideas behind our computation scheme; (2) contrasting it with existing approaches; (3) experimentally comparing our implementation with existing ones using practical datasets. We also discuss the most important algorithmic and implementational decisions.

Method preview. Input is a 3-dimensional array of scalar values. It is often called a **volume** and is composed of 3-dimensional **voxels**. Output is the persistence diagram of the input volume, based on an simplified intermediate representation.

Our scheme cuts the input volume into *slices*. A small number of slices is streamed from disk and processed in parallel. Each slice is *simplified* independently using discrete Morse theory, and boundary information of this smaller representation is output to disk. Carefully handling the *border* between adjacent slices allows us to *stitch* this local information back together. In practice, we do this by constructing a global boundary matrix using the partial information coming from each slice. Despite containing extra stitching information, the resulting boundary matrix is much smaller than the boundary matrix of the input. Finally, we retrieve the persistence diagram by running a specialized version of Gaussian elimination on the resulting matrix.

2 Standard background

In this section we cover the usual theoretical background relevant for topology of voxel data. This includes cubical complexes, discrete Morse theory and persistent homology. Whenever possible we choose simple definitions that help map concepts to computations. In the next section we cover some additional definitions specific to our approach.

Cubical cells and complexes. Following [26] we define a degenerate interval $[k, k] \subset \mathbb{R}$, and a regular interval $[k, k + 1] \subset \mathbb{R}$ for a natural number k . Taking a product of D intervals, p of which are regular, gives us a p -dimensional **cubical cell** in embedding dimension D . We call them cubical p -cells, or simply cells; $\dim(\sigma)$ gives the dimension of cell σ . For $p = 0, 1, 2, 3$, we talk about vertices, edges, squares and cubes. Cell σ is a **face (coface)** of another cell τ whenever $\sigma \subset \tau$ ($\tau \subset \sigma$); it is a **proper** face or coface if additionally their dimensions differ by one. A D -dimensional cell is called the top-dimensional cell which we identify with a **voxel**. A **cubical complex**, \mathbf{K} , of dimension D is a finite collection of cells of dimension at

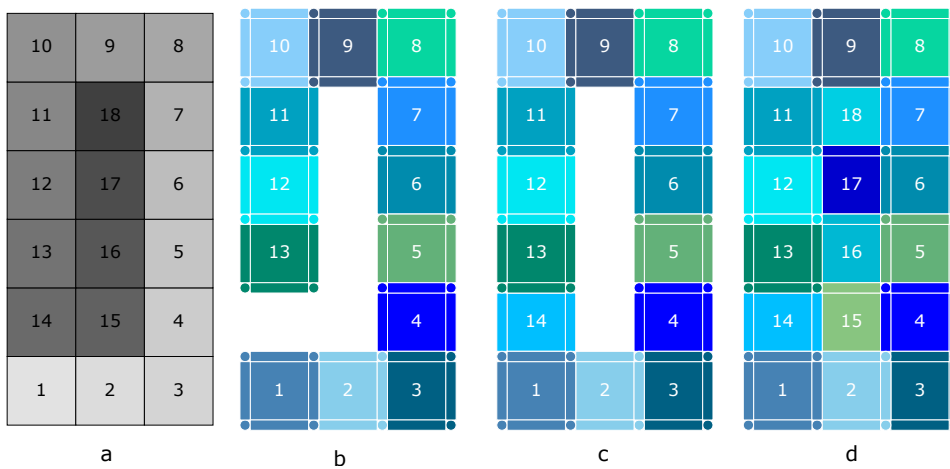
¹ Software is available as open source: <https://bitbucket.org/hubwag/cubicle/>

most D that is closed under taking faces. Namely, $\sigma \subset \tau$ implies $\sigma \in \mathbf{K}$. Given a cubical complex with n cells indexed from 1 to n , its **boundary matrix** is an $n \times n$ binary matrix M ; $M[i, j] = 1$ if and only if the j -th cell is a face of the i -th cell in the complex. The columns of M encode the **boundary** of each cell.

Filtered cubical complexes. We assume a common convention in which the values assigned to input voxels are interpreted as the values of the top-dimensional cells of a cubical complex. These values are then extended to the lower dimensional faces: each cell inherits the minimum value of its top-dimensional cofaces: $val(\sigma) = \min\{val(\tau) : \sigma \subset \tau \text{ and } dim(\tau) = D\}$. For us a **filtered cubical complex** is simply a cubical complex with cell values assigned as above. With this we talk about the **sublevel complex** at a value threshold t . Denoted $\mathbf{K}_{\leq t}$, it contains all cells in \mathbf{K} with value not exceeding t . Clearly $\mathbf{K}_{\leq s} \subseteq \mathbf{K}_{\leq t}$ whenever $s \leq t$, which lets us define a **filtration** of cubical complexes: $\emptyset = \mathbf{K}_0 \subset \mathbf{K}_1 \cdots \subset \mathbf{K}_n = \mathbf{K}$.

Cubical homology and persistent homology. Chains of cubical cells, as well as chain, cycle, boundary, homology and persistent homology groups can be defined [33] in complete analogy to the standard simplicial case [13]. We will work with homology and persistent homology with \mathbb{Z}_2 coefficients, which is standard in applications.

Instead of defining these concepts formally, we offer a brief intuitive overview, which may be more useful. Intuitively, for 3-dimensional cubical complexes, homology groups capture the connected components, 1-dimensional closed loops, and voids made of cells. We call them **homological features** of dimension 0, 1, 2 respectively, as shown in Figure 1.



■ **Figure 1** (a) Input volume. (b)–(d) three subcomplexes of the corresponding filtered cubical complex subdivided into blocks. At value 14 a 1-dimensional cycle is formed and it is filled in at value 18, forming a persistence pair (14,18) with persistence 4.

Persistent homology works with filtered complexes and assumes a dynamic view of data: we add voxels one by one ordered by non-decreasing values. Each voxel introduces all its faces – unless they were already present. Now, theory tells us that each p -cell can either create a p -dimensional homological feature, or destroy a $(p - 1)$ -dimensional one [13]. We keep track of the values of birth and death of each feature. Each such pair is called a **persistence pair**, and **persistence** itself is the lifetime of each feature. A multiset of persistence pairs forms a **persistence diagram** also called a persistence barcode. It turns out to be a powerful topological descriptor, largely due to its stability properties [14].

A **filtered boundary matrix** is the boundary matrix of a filtered cubical complex such that the rows and columns are sorted by the value of the corresponding cells. Persistent homology can be efficiently computed using Gaussian elimination of this matrix [13, 4]. The main shortcoming of this approach is the large size of this boundary matrix arising from the original input, even if stored in sparse format.

Discrete Morse theory. Since cubical complexes can be viewed as CW-complexes, Forman’s discrete Morse theory [15] – **DMT** for short – can be used in this setting. We emphasize that restricting the setting to cubical complexes is crucial for computational efficiency reasons. In short, we use DMT to simplify the input while retaining crucial topological information.

DMT relies on a **discrete Morse matching** which pairs cells in a certain way; it is often called a discrete Morse vector field or discrete Morse gradient. We represent it by a directed graph, which we call the **matching graph**. Its nodes correspond to cells in the cubical complex and initially each directed edge corresponds to proper face relation between two cells. Pairing two cells corresponds to flipping an edge in this graph. A matching is a **valid** discrete Morse matching if: (1) each node in the corresponding matching graph is incident to at most one flipped edge, and (2) the corresponding matching graph is acyclic. A cell is called **critical** if it is not paired with any other cell. We define an **alternating path** between a p -cell σ and $(p - 1)$ -cell τ as any path in the modified graph that starts at σ ends at τ and alternates between cells in dimension p and $p - 1$. They are also called V -paths, discrete Morse flow lines and gradient paths.

A **discrete Morse complex** arising from a matching is generated by its critical cells; the boundary relation between elements σ and τ is 1 if and only if there is an **odd** number of alternating paths between them, and 0 otherwise. As before, we encode this information in a boundary matrix. We remark that the parity criterion comes from using homology with \mathbb{Z}_2 coefficients – in more general setups this is slightly more complicated [31]. The key consequence of DMT is that the discrete Morse complex of a given complex yields homology groups which are isomorphic to the homology groups of the original complex.

In the context of filtrations, we restrict ourselves to pairing cells with the same filtration value. This process yields **filtered discrete Morse complexes** in which the filtration values are inherited from the critical cubical cells. As before, we encode it as a filtered boundary matrix. We also view it as a filtration of discrete Morse complexes: $\mathbb{M}(\mathbf{K}_0) \subset \mathbb{M}(\mathbf{K}_1) \subset \dots \subset \mathbb{M}(\mathbf{K}_n) = \mathbb{M}(\mathbf{K})$.

3 Setup: blocks, slices and borders

In this section we define a number of non-standard definitions which are crucial for our approach. First, we define the **extended value** of a voxel as the pair (input value, index of the voxel in the input volume). These pairs are compared lexicographically. This is a technicality which breaks ties between voxels of equal value. The **block** of a voxel is the subset of its faces which share its extended value. It is useful to imagine that voxels appear one by one starting with the lowest value. Each voxel introduces all its faces which were not yet present in the complex – namely the block. We remark that an individual block is not a cubical complex – just a subset of cubes. However, the blocks of all voxels in a cubical complex disjointly decompose this complex. See Figure 1 for an illustration.

We cut the input volume – and its cubical complex – in the following way. We imagine a horizontal hyperplane at integer height h intersecting the cubical complex. The cells contained in this plane form the **border** which is also the intersection of the two resulting **slices**. This border forms a cubical complex with the dimension of the highest dimensional cell equal to $D - 1$. If we cut k times, we generally get $k + 1$ slices separated by k borders.

We define the **interior** of each slice as the slice itself minus its borders. By a **stratum** we mean either the interior or a border of a slice. The complex disjointly decomposes into strata. Later we will compute discrete Morse matchings separately for each stratum, noting that internal strata are not cubical complexes, but all border strata are.

We remark that in actual computations we will cut the input volume into overlapping slices. More precisely, with each slice we load an extra layer of voxels belonging to each adjacent slice. Each extra layer has height one and is called an **overlap**. It will allow us to assign values to cells belonging to the border strata consistently across adjacent slices.

4 Related work

In this section we overview of existing literature in the topic. We focus on work related to persistent homology and discrete Morse complexes in the context of voxel data. In particular, we emphasize existing techniques which we use in our current approach.

To the best of our knowledge, applying algebraic topology in the context of voxel data were pioneered in the form of shape functions by Verri, Uras, Frosini and Ferri [35]. Using higher-degree homological tools go back to the work of Kaczyński, Mischaikow and Mrozek in the context of dynamical systems [30], with extensions to other application domains [26].

The early 2010s saw an increased interest in computing topological descriptors for voxel data. Bendich, Edelsbrunner and Kerber proposed an efficient approximation scheme for persistent homology of 3D voxel data [6]. In contrast, our approach aims at exact computations. An efficient approach for exact computation of persistence of voxel data of arbitrary dimension is due to Wagner, Chen and Vucini [36]. It relies on efficient generation of the boundary matrix of the entire complex. We reuse some of the techniques used in this paper, in particular the data-structure for compact storage of information for each cell. An efficient implementation of this approach is provided in Gudhi [12] and DIPHA [3]. One downside of this scheme is the large size of the boundary matrix – which prompted development of simplification methods, like the one below.

Work by Robins, Wood, Sheppard [33] marks a breakthrough in computing topological descriptors for voxel data. We refer to this work as \mathcal{RWS} and describe it in more detail in Section 5. In short, it preprocesses the input using discrete Morse theory typically resulting in a much smaller boundary matrix encoding the same topology. In [19], certain algorithmic aspects were improved and a memory-efficient storage of the matching was proposed. This allowed for handling large data – but significantly complicated the implementation. Since our new approach needs to store only a single slice of data at a time, we use a simpler but equally efficient implementation without affecting memory usage.

Work by Mischaikow and Nanda [31] generalizes the theory and algorithms presented in \mathcal{RWS} beyond cubical complexes. A related software package, Perseus, has been a popular tool capable of computing persistence for a variety of input types. One lesson learned from this work was the trade-off between flexibility and efficiency. For voxel data the preprocessing step turned out to be more costly than computing persistence directly [4]. With that in mind, our approach specializes in voxel data. This paper also reports the first approach for streaming preprocessing – each levelset was stored separately, simplified and merged together later. In contrast, we use a spatial decomposition of the volume.

Discrete Morse theory was also used to perform more aggressive simplification. This proved to be useful in data visualization. One particularly impressive approach is due to Gyulassy, Bremer, Hamann and Pascucci [20]. Already in 2008 it handled a volume with a billion voxels on commodity hardware in a day. However, the simplified complex computed

in this approach cannot be used to compute persistent homology of the original data [9]. Still, our approach is inspired by the data subdivision scheme used in this approach. Despite certain similarities, the details of our subdivision scheme, subsequent computations and final goal are all different. Our method can be viewed in terms of discrete stratified Morse theory of Knudson and Wang [28].

Concurrently to the development of DMT-based methods, matrix reduction algorithms were significantly improved [7, 2, 4, 22, 3]. In Section 7 we describe how we adapted certain techniques from the PHAT library [4] at the preprocessing level.

Bauer’s Ripser [1] was originally devised for the Vietoris–Rips construction arising from point cloud data. Recently Ripser was adapted to voxel data with CubicalRipser [27].

Finally, we remark that the proposed scheme was first conceived in 2015 and preliminary results were informally presented at two international meetings at 2017. Software development began in 2015, and the first stable version of the software was published in 2018.

5 Persistence-aware simplification scheme

In this section, we overview an existing approach by by Robins, Wood and Sheppard [33], (\mathcal{RWS}), for simplifying cubical complexes stored in memory.

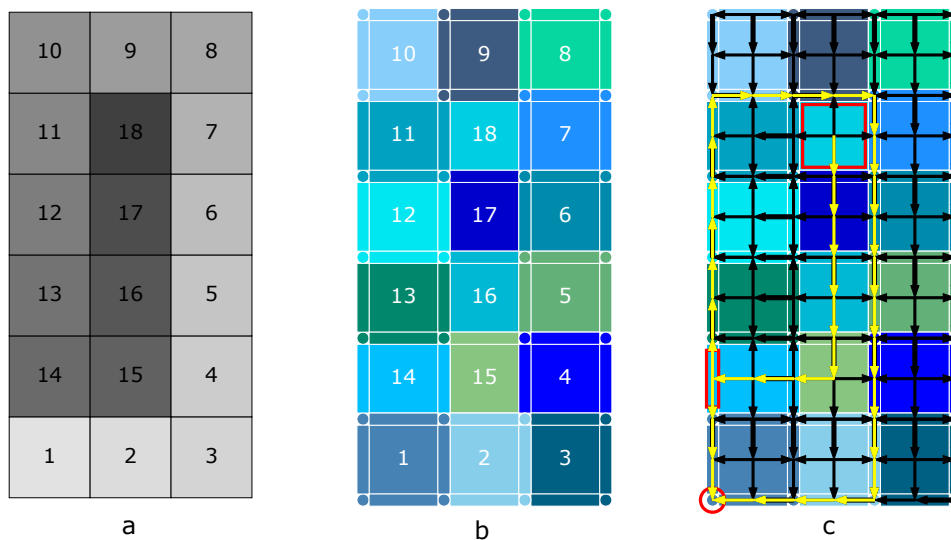
The goal is to compute a smaller representation of data that encodes the same persistent homology as the input volume. More precisely output persistence diagrams are allowed to differ only by **ephemeral persistence pairs** – each corresponding to a feature that is born and dies at the same filtration value. Such pairs are discarded in practice anyway. The crucial observation by Robins and collaborators is that ephemeral features abound in cubical filtrations, allowing them to propose an efficient simplification scheme.

This method computes a valid discrete Morse matchings *separately* for each block. It then computes the boundary matrix of a filtered discrete Morse complex which encodes the same persistent homology. This step is done by computing the parity of the numbers of alternating paths between pairs of critical cells. See Figure 2 for illustration. We discuss correctness of this approach for filtered cubical complexes.

Validity of the matching. We show that if the matching graph within each block is acyclic, then the global one is acyclic as well. We observe that the *extended* value is generally non-increasing along cells in an alternating path. However leaving a block necessarily decreases the extended value. Since forming a cycle would require leaving a block and returning, cycles cannot form since the extended values along paths cannot increase.

Topological correctness of simplification. Applying DMT on each block yields a complex that encodes the same persistent homology – up to ephemeral pairs. Indeed, this block-wise construction yields the filtration: $\mathbb{M}(\mathbf{K}_0) \subset \mathbb{M}(\mathbf{K}_1) \subset \dots \mathbb{M}(\mathbf{K}_n) = \mathbb{M}(\mathbf{K})$. Now, Forman’s theory tells us that $H_*(\mathbb{M}(\mathbf{K}_i)) = H_*(\mathbf{K}_i)$. Applying the Persistence Equivalence Theorem [13] yields the desired result.

Hardness of matchings. Generally, finding discrete Morse matchings minimizing the number of critical cells is a computationally hard problem [25]. However, \mathcal{RWS} showed a simple $\Theta(b \log b)$ time algorithm for optimal matching for a block of a 3-dimensional cubical complex with b cells.



■ **Figure 2** (a) Input volume; (b) Filtered cubical complex (c) Depiction of the \mathcal{RWS} method. Black arrows show the matching graph. The yellow paths mark the alternating paths whose parity determine the boundary relations between the red critical cells. There is one nonzero relation: between the critical 2-cell with value 18 and the critical 1-cell with value 14, which itself has empty boundary. In this simple case the resulting boundary matrix is already reduced, so the persistence pairs are readily available. Typically the matrix has to be reduced first.

6 Streaming simplification scheme

In this section we describe our algorithmic scheme, which is an efficient streaming version of the \mathcal{RWS} scheme. It yields a reduced representation encoding the same persistent homology as the input volume. More precisely, we output – to disk – information about the boundary matrix of the corresponding filtered discrete Morse complex. This boundary matrix is then reconstructed on disk and the persistent homology is computed using an existing matrix reduction algorithm. We focus on a high-level overview, noting that the actual implementation is intricate and contains 2500 lines of terse C++ code (not counting external libraries).

Algorithm 1 outlines the streaming simplification scheme. In overview, we independently simplify each stratum using the method outlined in the previous section and put global information back together. We now outline the algorithm and prove its correctness.

Correctness. We need a few new concepts. A **border-crossing path** is an alternating path which contains cells belonging to two or more internal strata. A **global matching** is the union of discrete Morse matchings computed separately on each stratum of a filtered cubical complex. A **global Discrete Morse complex** is the filtered discrete Morse complex arising from the global matching.

We propose a lemma and its three corollaries. Together they show that the information extracted from all slices is sufficient to reconstruct the boundary matrix of a global discrete Morse complex that captures the correct persistent homology of the entire original dataset.

► **Lemma 1** (Border Blocking Lemma). *Suppose an acyclic discrete Morse matching is computed for each block of each stratum of a filtered cubical complex. Consider a directed path $p = (p_1, p_2, \dots, p_i, \dots, p_n)$ in the matching graph G corresponding to the global matching. If any p_i belongs to a border stratum B , then the suffix $(p_i, p_{i+1}, \dots, p_n)$ is contained in B .*

■ **Algorithm 1** Streaming simplification.

Require: V : volume on disk; (w_1, w_2, \dots, w_D) : size of V ; h : maximum height of slice

Ensure: Boundary matrix on disk with the same persistent homology as V

```

1: for index  $i$  in  $1 \dots \lceil w_1/h \rceil$  do
2:    $n = h \prod_{d=2}^D w_d$ 
3:    $D =$  read next  $n$  voxels of  $V$  from disk (less for last slice)
4:    $C =$  filtered cubical complex of  $D$  representing the slice
5:    $O =$  load 1-voxel tall overlap for each border of  $C$ 
6:   update the values of border cells in  $C$  using extra information in  $O$ 
7:    $S =$  decompose  $C$  into an internal stratum and up to 2 border strata
8:   for stratum  $s$  of  $S$  do
9:     for block  $b$  in  $s$  do
10:      compute acyclic Morse matching within  $b$ 
11:      mark critical cells in  $S$ 
12:   for critical cell  $\sigma$  in  $C$  do
13:     if  $\sigma$  does not belong to the bottom border stratum of  $C$  then
14:       write value and dimension information:  $(\text{ind}(\sigma), \text{val}(\sigma), \text{dim}(\sigma))$  to disk
15:        $p = \text{dim}(\sigma)$ 
16:        $T =$  critical  $(p - 1)$ -cells reachable from  $\sigma$  by odd number of paths
17:       for critical  $(p - 1)$ -cell  $\tau$  in  $T$  do
18:         write boundary information:  $(\text{ind}(\sigma), \text{ind}(\tau))$  to disk
19:   unload all data from memory
20: sort the indices by corresponding value and dimension (on disk)
21: map these sorted indices to a contiguous range of indices (on disk)
22: save the filtered boundary matrix in appropriate sparse format (on disk)

```

Proof. We proceed by induction, with $p_i \in B$ by assumption. To show that $p_k \in B$ implies $p_{k+1} \in B$ for $i \leq k < n$, we consider two cases. Namely, there are two potential outgoing edges from p_k in G : to a proper face of p_k , which belongs to the border stratum B because each border stratum is a cubical complex; or to a *matched* (paired) proper coface of p_k , which belongs to the border stratum B because the corresponding matching is restricted to B by construction. In any case, p_{k+1} lies in B and by induction so does each p_j for $i \leq j \leq n$. ◀

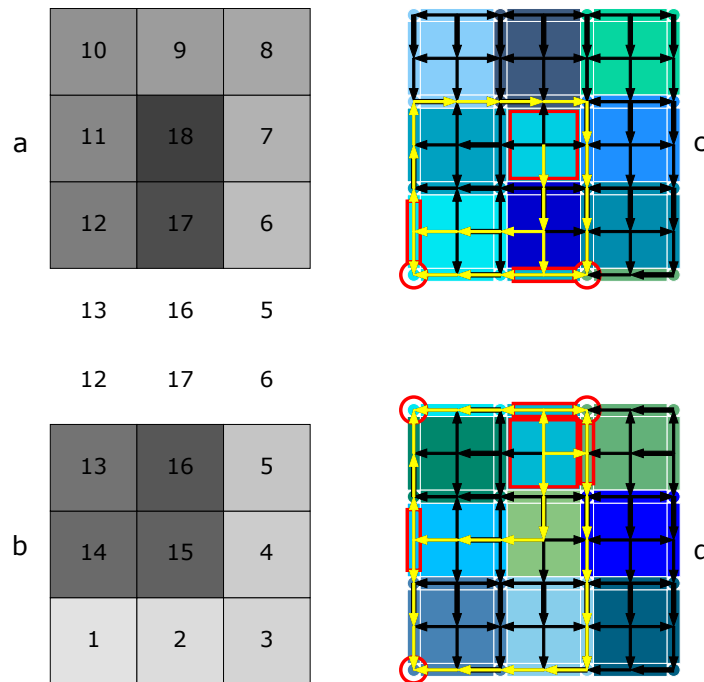
We recall that the internal strata are generally not cubical complexes. This means that paths can lead from an internal cell to a border cell. Still, the following corollary reassures us that no cycles can be formed.

► **Corollary 1** (Global Acyclicity). *The global discrete Morse matching is acyclic.*

Proof. First, there are no cycles within each stratum by construction. To form a directed cycle spanning multiple strata, a path would have to go back and forth between internal and border strata. This is however impossible, since paths entering a border stratum remain inside this stratum, as shown in Lemma 1. ◀

With the above, it is easy to see that we preserve information about persistent homology.

► **Corollary 2** (Global Correctness). *The global discrete Morse complex encodes the same persistent homology.*



■ **Figure 3** (a,b) Input split into two overlapping parts; (c,d) Simplification applied on each slice. The slices agree on the values and matching on the shared border. The path that crossed the border in Figure 2 is now split into *three* paths. One can construct the filtered boundary matrix with 8 columns and 13 nonzero entries to verify that, in particular, the 1-dimensional feature created in one slice and destroyed in another is correctly captured; also that the extra critical cells result only in features of zero persistence.

Proof. By Corollary 1, the global matching is a valid – if suboptimal – matching on the entire input filtered cubical complex. This reduces the proof to the case of the \mathcal{RWS} approach we covered in the previous section. Therefore, the simplified complex encodes the same persistent homology (up to ephemeral pairs) as the original cubical filtration. ◀

Our scheme introduces extra critical cells, even if all the voxel values are unique. These extra cells allow us to stitch together discrete Morse complexes coming from different slices. More specifically, these extra cells split each border-crossing path into multiple non-crossing paths as shown in Figure 3. Instead of performing costly pruning at this stage [20, 10], we welcome these cells into the final boundary matrix, reduce it, and simply discard the resulting ephemeral pairs from the resulting diagram.

Finally, we show that the information available in each slice is sufficient.

► **Corollary 3** (Slice Locality). *The boundary matrix of the global discrete Morse complex can be computed from information contained within each slice.*

Proof. Lemma 1 implies that there are no border-crossing paths. This means that all the boundary information contained in the resulting boundary matrix can be computed locally within in each slice. Additionally, the dimension and value of each critical cell is available within each slice. ◀

One subtlety: in a practical implementation the values and matching assigned to each border stratum must be consistent between the adjacent slices. To ensure this, we load the overlap, namely the extra layers of voxels, in line 5 of Algorithm 1.

Computational complexity. Computations are dominated by tracking the parity of alternating paths [33], which is done in line 16. Overall, a single slice is handled in $\Theta(v^3)$ worst case time [33], where v is the number of voxels in a slice. In our case, a dataset with v voxels divided into s slices, the worst-case complexity is $\Theta(s(\frac{v}{s})^3) = \Theta(\frac{v^3}{s^2})$. However, this worst-case behaviour is theoretical, and the experiments we report in Section 7 show roughly linear scaling for *all* practical datasets.

7 Technicalities

In this section we mention several technicalities which make our implementation efficient in practice. Balancing speed and memory usage was the key challenge in our streaming setup.

Encoding the information about cells. All auxiliary information about cells are stored in a cube-map format [36], which simply arranges cells as an array of size $2w_1 + 1, 2w_2 + 1, \dots, 2w_D + 1$ for input of size w_1, w_2, \dots, w_D . We use an efficient implementation of multidimensional arrays provided by the blitz++ library.

Global indexing of cells. Each slice needs to assign a globally unique index to each of its critical cells. Since additionally the indices of border cells must be consistent across slices, we simply use the global index of cells in the entire complex. To compute this, each slice must know its offset and the dimensions of the volume (except w_1). These indices are compactified in line 13 of Algorithm 1.

Computing and storing the matching. Focusing on $D \leq 3$ allows us to adapt the ProcessLowerStar procedure of \mathcal{RWS} . Since each block can have at most 27 cells, these computations are unlikely to be the performance bottleneck.

However, storing the matching could dominate memory usage, which was alleviated in [19] by using a succinct bit-level encoding of the matching graph. Our sliced setting allows for more relaxed memory management, so such techniques are not necessary. Instead we used a simpler encoding, exploiting the fact that each edge can only point at at most 6 directions.

Computing the parity of alternating paths. This is hidden in line 16 and is crucial for performance of the simplification part. In $D \geq 3$, the alternating paths can both split and merge, which complicates the algorithms compared to lower dimensions. One consequence is that the number of alternating paths between 3-cells and 2-cells can grow exponentially in input size – which actually occurs in practical datasets [19]. This makes simple enumeration prohibitive. However, the matching graph is an acyclic directed graph, allowing us to use a basic dynamic programming algorithm for counting the parity of alternating paths.

One important addition compared to existing implementations is the usage of the bit-tree data-structure developed for the PHAT library. We re-purpose it for compact storage and manipulation of indices of cells maintained during path-counting. This removes the main memory bottleneck present in [19], which used a red-black tree. It required significantly more storage in the worst case and is also generally slower.

Reconstructing boundary matrix and persistence pairs. During simplification we output partial boundary and filtration information on disk in lines 14 and 18. In lines 20–22, we reconstruct the full filtered boundary matrix using a simple external-memory (on disk) algorithm. We do this to avoid storing both the partial information and the full matrix at

the same time. In line 20 of Algorithm 1, we sort the indices of cells by their values, since we need a filtered boundary matrix. In line 20, we map global indices of cells into contiguous indices, as required by most matrix reduction packages. Already in line 13 we made sure that the information for each border cell is output from exactly one of the two slices. In line 22 we rearrange the data in the format required by matrix reduction software; usually each column is represented by the dimension of the cell, the size of the column followed by its nonzero indices. Additionally, after the matrix reduction, we transform the reduced matrix into the final persistence diagram in a similar way. All these simple computations are implemented using the STXXL [11] external algorithms library.

Lock-free parallelization scheme. Each slice is handled by one thread. More precisely, we maintain a thread-pool of a fixed size k , which allows k slices to be handled in parallel. Each thread loads its slice, processes it, and outputs partial results to one of k output buffers on disk and unloads the slice. This allows for simple, lock-free parallelism, circumventing the usual synchronization problems. We use the CTL thread-pool library.

Experiments We present experiments which compare the end-to-end performance of the proposed approach with existing solutions.

Datasets. We use datasets provided at the free Open Scientific Visualization Datasets (OSVD) repository. This repository contains several 3D voxel datasets, mostly coming from real-world applications. The sizes range from 41^3 to 4096^3 , with one 900GB file of size $10240 \times 7680 \times 1536$ voxels. For comparison we used the 15 smallest datasets in the repository, with up to 150 millions voxels. We also test our method on images up to 8 billion voxels.

Benchmarked software. A recent paper [18] includes a comprehensive benchmark with a multitude of software packages. We restrict our benchmark to three fastest end-to-end alternative approaches: CubicalRipser, DIPHA and DiscreteMorseSandwich. **CubicalRipser** is a modern method [27] based on implicit boundary matrix representation. It is a single-threaded implementation. We used a newer version than in the original benchmark [18], and got significantly better results. **DIPHA** [3] implements [36] and explicitly stores and reduces the full boundary matrix. **DiscreteMorseSandwich** [18] is part of TTK [34] and implements the \mathcal{RWS} scheme. It works both in cubical and triangulated setting.

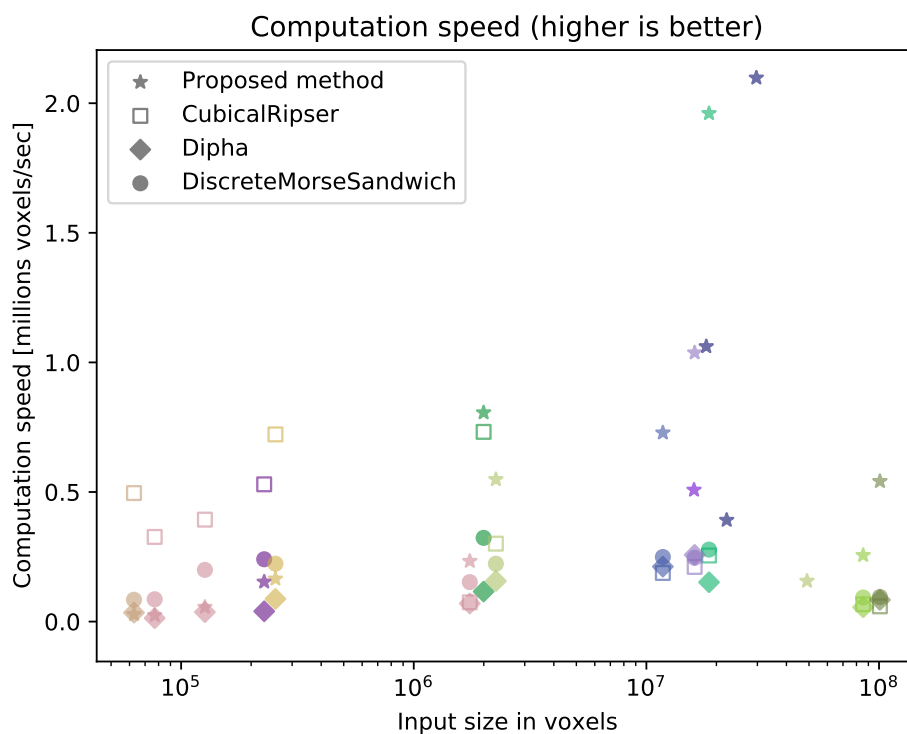
Result consistency. Our approach uses the data interpretation described earlier in the paper, sometimes called the T interpretation [16]. We verified that the produced persistence diagrams are consistent with the results of CubicalRipser with appropriate options. DIPHA uses the dual V interpretation, which however encodes equivalent information [16]. As already noted in [27], DiscreteMorseSandwich returns different diagrams presumably due to an alternative data interpretation.

Hardware. We use a commodity laptop with an i7-1165G7@2.80GHz CPU with 5MB L2 cache, 8 logical cores, 32GB of RAM and Toshiba XG6 M.2 NVMe SSD. All software is compiled and run on Ubuntu 20.04.5 using g++ 10.3.0. All parallel implementation are run on 8 threads.

Performance comparison on smaller data. We now discuss results of the experiments. We start with a comparison with other software on smaller datasets. Then we turn to much larger data to understand performance characteristics and limits of our approach.

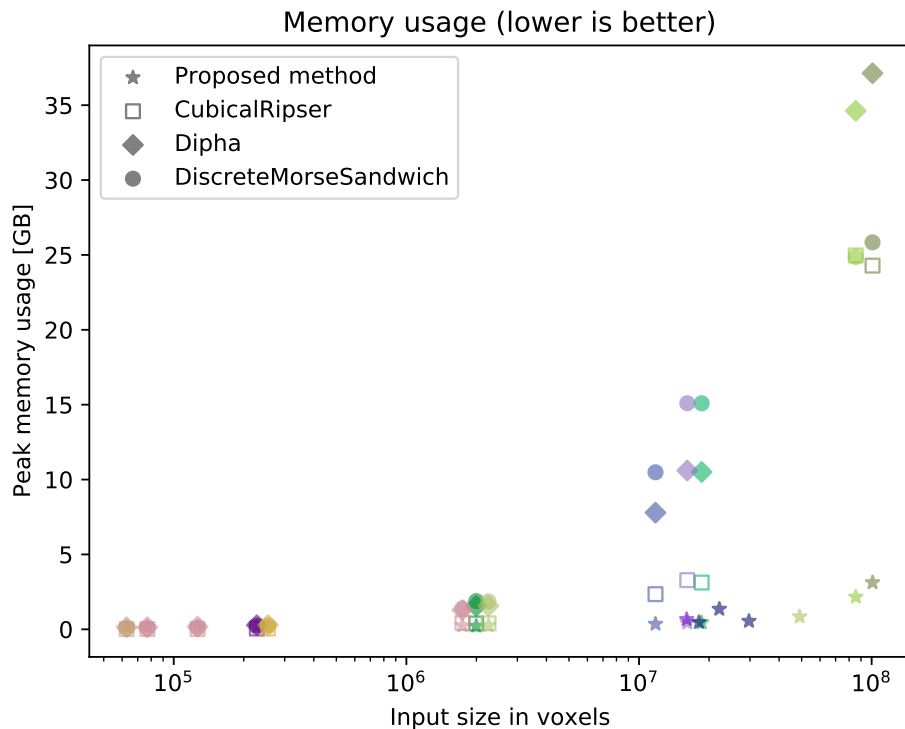
Figures 4 and 5 depict the speed and memory efficiency of the proposed method and existing implementations. Our goal was to provide an efficient method for massive voxel data running on commodity hardware. For small datasets the overhead related to streaming and external memory operations dominates the computations – which is a trade-off we made by focusing on massive data. In this case, the new version of CubicalRipser is the dominant solution. For *all* datasets larger than $256^3 \approx 10^6$ voxels, our approach is both faster and more memory efficient.

Speed. For large enough data, our approach achieves speed between 0.25 and 2 million voxels per second and is usually an order of magnitude faster than the fastest alternative. The computations scale linearly with size, and also depend on the complexity of the data, which we investigate in a moment.



■ **Figure 4** Speed comparison in millions of processed voxels per second. Methods are marked with different symbols. Random colors help distinguish between datasets. For datasets exceeding 1 million voxels, the proposed method offers the highest execution speed. In some cases competing software failed to finish in reasonable time, which explains why some points are missing.

Memory. For large enough data, the proposed approach achieves much smaller memory footprint. For the tested datasets the peak memory usage was kept below 3.5GB, whereas alternative approaches required > 24 GB of memory. Since memory consumption was the obstacle preventing analysis of large volumes, the low memory usage is the main selling point of the new approach.



■ **Figure 5** Peak memory usage comparison. For datasets exceeding 1 million voxels the proposed method offers significantly lower memory usage. Memory usage is negligible for smaller datasets.

Performance on large data. We turn to much larger datasets counted in billions of voxels – beyond the scope of the other approaches, at least on our test hardware. Results are summarized in Table 1.

The technicalities described in Section 7 aimed to balance memory usage and speed. In particular, storing boundary matrix information in memory would significantly increase the memory footprint. On the other hand, a slow on-disk implementation could impact the overall performance. On a more fundamental level, one big unknown was the number of extra critical cells our method will generate. They could easily overwhelm the computations or lead to huge boundary matrices. These experiments show that we found a reasonable balance and that memory usage was significantly decreased slowing things down.

We mention two data-points not collected in the benchmark. First, the final matrix reduction accounts only for a small portion of the execution time. Second, storage of the boundary matrix often dominates the memory usage. Therefore, further research into memory-efficient matrix reduction algorithms would benefit our implementation. The work reported in [5] is a step in this direction. We also mention that for some of the examples the memory usage could be further reduced by setting the slice size parameter to a smaller value.

Richtmyer–Meshkov instability. The last dataset in Table 1 is a snapshot of a 3D simulation of the Richtmyer–Meshkov instability [8]. It describes impulsive mixing of two different density fluids – and often leads to multi-scale behaviour exhibiting topological patterns. This particular file has size $2048 \times 2048 \times 1920$, roughly 8 billion voxels and 64 billion cells. Its full boundary matrix takes $\approx 1.5\text{TB}$, and reducing it would require at least 3TB of RAM. Our method requires 150 times less allowing it to work on a regular laptop.

■ **Table 1** Columns represent **file**: original input filename; **t[s]**: total time in seconds; **Mvox/s**: speed in millions of input voxels per second; **sim**: percentage of time spent in the simplification step; **B[M]**: millions of nonzero elements in the resulting boundary matrix; **c[M]**: millions of critical cells; **p[M]**: millions of persistence pairs; **m[GB]**: peak memory usage in GB.

file	t[s]	sim	Mvox/s	B[M]	c[M]	p[M]	m[GB]
vertebra_512x512x512	107	74%	1.2	20	7.6	2.8	1.3
zeiss_680x680x680	190	87%	1.7	17	7.7	0.4	2.2
prone_512x512x463	231	54%	0.5	73	29.5	12.9	3.1
neocortical_..._1464x1033x76	266	59%	0.4	81	32.1	11.9	3.1
present_492x492x442	290	53%	0.4	100	37.8	15.5	4.2
stent_512x512x174	291	92%	0.2	16	5.6	2.5	0.8
christmas_tree_512x499x512	333	42%	0.4	132	53.1	20.7	5.3
marmoset_..._1024x1024x314	662	40%	0.5	273	111.4	34.5	10.4
kingsnake_1024x1024x795	1345	62%	0.6	337	140.8	51.6	13.8
pawpawsaurus_958x646x1088	1809	30%	0.4	573	232.0	109.5	21.3
chameleon_1024x1024x1080	2152	48%	0.5	619	261.4	122.3	21.8
richtmyer_..._2048x2048x1920	11477	86%	0.7	828	292.5	88.4	22.3

8 Outlook

The main contribution of this work is a streaming preprocessing scheme which reduces the representation of voxel data without affecting its topology. In particular, it provably preserves persistent homology of the data. We combined our scheme with an existing boundary matrix reduction algorithm, yielding an end-to-end solution for persistent homology computations. Our experiments show that for large data our solution is the most efficient option.

Our method achieves speed between 0.2 and $2Mvox/s$ depending on input complexity. It handles complex data with 2048^3 voxels on a laptop.

We offer three interesting future directions the proposed scheme opens up:

- We can now handle multi-scale datasets of several billion voxels. Data coming from physical simulations, astrophysics and nanotechnology often exhibit multi-scale structure – and using persistent homology on such data is an exciting prospect.
- With little extra technical effort, we can stream huge data from a network location. This is important since raw voxel volumes of up to 900GB are readily available but copying and storing them can be a nuisance.
- Further progress in matrix reduction algorithms will benefit our approach. In particular, an external-memory matrix reduction algorithm would complement our scheme well.



References

- 1 Ulrich Bauer. Ripser: efficient computation of vietoris–rips persistence barcodes. *Journal of Applied and Computational Topology*, 5(3):391–423, 2021.
- 2 Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Clear and compress: Computing persistent homology in chunks. In *Topological Methods in Data Analysis and Visualization*, 2014.
- 3 Ulrich Bauer, Michael Kerber, and Jan Reininghaus. Distributed computation of persistent homology. In *2014 proceedings of the sixteenth workshop on algorithm engineering and experiments (ALENEX)*, pages 31–38. SIAM, 2014.
- 4 Ulrich Bauer, Michael Kerber, Jan Reininghaus, and Hubert Wagner. Phat: Persistent homology algorithms toolbox. *Journal of Symbolic Computation*, 78:76–90, 2017. Algorithms and Software for Computational Topology. doi:10.1016/j.jsc.2016.03.008.

- 5 Ulrich Bauer, Talha Bin Masood, Barbara Giunti, Guillaume Houry, Michael Kerber, and Abhishek Rathod. Keeping it sparse: Computing persistent homology revised. *arXiv preprint arXiv:2211.09075*, 2022.
- 6 Paul Bendich, Herbert Edelsbrunner, and Michael Kerber. Computing robustness and persistence for images. *IEEE transactions on visualization and computer graphics*, 16(6):1251–1260, 2010.
- 7 Chao Chen and Michael Kerber. Persistent homology computation with a twist. In *Proceedings 27th European workshop on computational geometry*, volume 11, pages 197–200, 2011.
- 8 Ronald H Cohen, William P Dannevik, Andris M Dimits, Donald E Eliason, Arthur A Mirin, Ye Zhou, David H Porter, and Paul R Woodward. Three-dimensional simulation of a richtmyer–meshkov instability with a two-scale initial perturbation. *Physics of Fluids*, 14(10):3692–3709, 2002.
- 9 Olaf Delgado-Friedrichs, Vanessa Robins, and Adrian Sheppard. Skeletonization and partitioning of digital images using discrete morse theory. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):654–666, 2014.
- 10 Olaf Delgado-Friedrichs, Vanessa Robins, and Adrian Sheppard. Skeletonization and partitioning of digital images using discrete morse theory. *IEEE transactions on pattern analysis and machine intelligence*, 37(3):654–666, 2015.
- 11 Roman Dementiev, Lutz Kettner, and Peter Sanders. Stxxl: standard template library for xxl data sets. *Software: Practice and Experience*, 38(6):589–637, 2008.
- 12 Pawel Dlotko. Cubical complex. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2015. URL: http://gudhi.gforge.inria.fr/doc/latest/group__cubical__complex.html.
- 13 Herbert Edelsbrunner and John Harer. *Computational topology: an introduction*. American Mathematical Soc., 2010.
- 14 Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological persistence and simplification. In *Proceedings 41st annual symposium on foundations of computer science*, pages 454–463. IEEE, 2000.
- 15 Robin Forman. A user’s guide to discrete morse theory. *Séminaire Lotharingien de Combinatoire [electronic only]*, 48:B48c–35, 2002.
- 16 Adélie Garin, Teresa Heiss, Kelly Maggs, Bea Bleile, and Vanessa Robins. Duality in persistent homology of images. *arXiv preprint arXiv:2005.04597*, 2020.
- 17 Charles Gueunet, Pierre Fortin, Julien Jomier, and Julien Tierny. Contour forests: Fast multithreaded augmented contour trees. In *2016 IEEE 6th Symposium on Large Data Analysis and Visualization (LDAV)*, pages 85–92. IEEE, 2016.
- 18 Pierre Guillou, Jules Vidal, and Julien Tierny. Discrete morse sandwich: Fast computation of persistence diagrams for scalar data—an algorithm and a benchmark. *arXiv preprint arXiv:2206.13932*, 2022.
- 19 David Günther, Jan Reininghaus, Hubert Wagner, and Ingrid Hotz. Efficient computation of 3D morse–smale complexes and persistent homology using discrete morse theory. *The Visual Computer*, pages 1–11, 2012.
- 20 Attila Gyulassy, Peer-Timo Bremer, Bernd Hamann, and Valerio Pascucci. A practical approach to morse-smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.
- 21 Teresa Heiss and Hubert Wagner. Streaming algorithm for Euler characteristic curves of multidimensional images. In Michael Felsberg, Anders Heyden, and Norbert Krüger, editors, *Computer Analysis of Images and Patterns - 17th International Conference, CAIP*, volume 10424 of *Lecture Notes in Computer Science*, pages 397–409. Springer, 2017. doi:10.1007/978-3-319-64689-3_32.
- 22 G. Henselman and R. Ghrist. Matroid Filtrations and Computational Persistent Homology. *ArXiv e-prints*, June 2016. arXiv:1606.00199.

- 23 Xiaoling Hu, Fuxin Li, Dimitris Samaras, and Chao Chen. Topology-preserving deep image segmentation. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- 24 Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-aware segmentation using discrete morse theory. *arXiv preprint arXiv:2103.09992*, 2021.
- 25 Michael Joswig and Marc E Pfetsch. Computing optimal morse matchings. *SIAM Journal on Discrete Mathematics*, 20(1):11–25, 2006.
- 26 Tomasz Kaczynski, Konstantin Mischaikow, and Marian Mrozek. *Computational Homology*. Springer-Verlag, New York, 2004.
- 27 Shizuo Kaji, Takeki Sudo, and Kazushi Ahara. Cubical ripser: Software for computing persistent homology of image and volume data. *arXiv preprint arXiv:2005.12692*, 2020.
- 28 Kevin Knudson and Bei Wang. Discrete stratified morse theory: Algorithms and a user’s guide. *arXiv preprint arXiv:1801.03183*, 2018.
- 29 Jonathan Shewchuk Martin Isenburg. Streaming connected component computation for trillion voxel images. In *Workshop on Massive Data Algorithmics*, 2009.
- 30 Konstantin Mischaikow and Marian Mrozek. Chaos in the lorenz equations: a computer-assisted proof. *Bulletin of the American Mathematical Society*, 32(1):66–72, 1995.
- 31 Konstantin Mischaikow and Vidit Nanda. Morse theory for filtrations and efficient computation of persistent homology. *Discrete & Computational Geometry*, 50(2):330–353, 2013.
- 32 Arnur Nigmatov and Dmitriy Morozov. Local-global merge tree computation with local exchanges. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–13, 2019.
- 33 Vanessa Robins, Peter John Wood, and Adrian P Sheppard. Theory and algorithms for constructing discrete morse complexes from grayscale digital images. *IEEE Transactions on pattern analysis and machine intelligence*, 33(8):1646–1658, 2011.
- 34 Julien Tierny, Guillaume Favelier, Joshua A Levine, Charles Gueunet, and Michael Michaux. The topology toolkit. *IEEE transactions on visualization and computer graphics*, 24(1):832–842, 2017.
- 35 Alessandro Verri, Claudio Uras, Patrizio Frosini, and Massimo Ferri. On the use of size functions for shape analysis. *Biological cybernetics*, 70(2):99–107, 1993.
- 36 Hubert Wagner, Chao Chen, and Erald Vuçini. Efficient computation of persistent homology for cubical data. In *Workshop on Topology-based Methods in Data Analysis and Visualization*, 2011.
- 37 Fan Wang, Saarthak Kapse, Steven Liu, Prateek Prasanna, and Chao Chen. Topotxr: A topological biomarker for predicting treatment response in breast cancer. In *International Conference on Information Processing in Medical Imaging*, pages 386–397. Springer, 2021.
- 38 Fan Wang, Hubert Wagner, and Chao Chen. Gpu computation of the euler characteristic curve for imaging data. In *38th International Symposium on Computational Geometry (SoCG 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

Maximum Overlap Area of a Convex Polyhedron and a Convex Polygon Under Translation

Honglin Zhu  

Massachusetts Institute of Technology, Cambridge, MA, USA

Hyuk Jun Kweon  

Massachusetts Institute of Technology, Cambridge, MA, USA

Abstract

Let P be a convex polyhedron and Q be a convex polygon with n vertices in total in three-dimensional space. We present a deterministic algorithm that finds a translation vector $v \in \mathbb{R}^3$ maximizing the overlap area $|P \cap (Q + v)|$ in $O(n \log^2 n)$ time. We then apply our algorithm to solve two related problems. We give an $O(n \log^3 n)$ time algorithm that finds the maximum overlap area of three convex polygons with n vertices in total. We also give an $O(n \log^2 n)$ time algorithm that minimizes the symmetric difference of two convex polygons under scaling and translation.

2012 ACM Subject Classification Theory of computation \rightarrow Computational geometry

Keywords and phrases computational geometry, shape matching, arrangement

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.61

Related Version *Full Version:* <https://arxiv.org/abs/2301.02949>

Acknowledgements This paper is the result of the MIT SPUR 2022, a summer undergraduate research program organized by the MIT math department. The authors would like to thank the faculty advisors David Jerison and Ankur Moitra for their support and the math department for providing this research opportunity. We thank the anonymous referees for providing helpful comments that increased the quality of this paper.

1 Introduction

Shape matching is an important topic in computational geometry, with useful applications in areas such as computer graphics. In a typical problem of shape matching, we are supplied two or more shapes, and we want to determine how much the shapes resemble each other. More precisely, given a similarity measure and a set of allowed transformations, we want to transform the shapes to maximize their similarity measure.

There are many candidates for the similarity measure, such as the Hausdorff distance and the Fréchet distance between the boundaries of the shapes. We can also consider the area/volume of overlap or of symmetric difference. The advantage to these is that they are more robust against noise on the boundary of the shapes [6].

The maximum overlap problem of convex polytopes has been studied by many. In dimension 2, de Berg et al. [6] give an $O(n \log n)$ time algorithm for finding a translation maximizing the area of intersection of two convex polygons (where n denotes the total number of vertices of the polygons). In dimension 3, Ahn et al. [1] give an $O(n^3 \log^4 n)$ expected time algorithm finding the maximum overlap of two convex polyhedra under translation. For the same problem, Ahn et al. [3] present an algorithm that runs in $O(n \log^{3.5} n)$ time with probability $1 - n^{-O(1)}$ and an additive error. For $d > 3$, given two convex polytopes of dimension d with n facets in total, Ahn et al. [3] give an algorithm that finds the maximum overlap under translation in $O(n^{\lfloor d/2 \rfloor + 1} \log^d n)$ time with probability $1 - n^{-O(1)}$ and an additive error.



© Honglin Zhu and Hyuk Jun Kweon;

licensed under Creative Commons License CC-BY 4.0

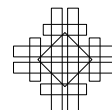
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 61; pp. 61:1–61:16

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



61:2 Maximum Overlap of Polyhedron and Polygon

In the plane, when all rigid motions are allowed, Ahn et al. [4] give an approximate algorithm that finds a rigid motion realizing at least $1 - \epsilon$ times the maximal overlap in $O((1/\epsilon) \log n + (1/\epsilon^2) \log(1/\epsilon))$ time. In dimension 3, Ahn et al. [2] present an approximate algorithm that finds a rigid motion realizing at least $1 - \epsilon$ times the maximal overlap in $O(\epsilon^{-3} n \log^{3.5} n)$ with probability $1 - n^{-O(1)}$.

When considering the maximum overlap as a similarity measure, we obviously can only allow area/volume-preserving transformations. However, we may want to allow scaling as a transformation – two similar triangles are supposed to be very “similar,” though they may have different sizes. In this case, the area of symmetric difference is a better measure of similarity. Yon et al. [14] give an algorithm minimizing the symmetric difference of two convex polygons under translation and scaling in $O(n \log^3 n)$ expected time.

Our results

While many have studied the matching problem for two convex polytopes of the same dimension, to our knowledge no one has examined the problem for polytopes of different dimensions or matching more than two polytopes.

The main result in this paper is a deterministic algorithm for the problem of matching a convex polyhedron and a convex polygon under translation in three-dimensional space.

► **Theorem 1.** *Let P be a convex polyhedron and Q a convex polygon with n vertices in total. We can find a vector $v \in \mathbb{R}^3$ that maximizes the overlap area $|P \cap (Q + v)|$ in $O(n \log^2 n)$ time.*

We also present two applications of our algorithm to other problems in computational geometry. First, we give a deterministic algorithm for maximizing the overlap of three convex polygons under translations.

► **Theorem 2.** *Let P, Q, R be three convex polygons with n vertices in total in the plane. We can find a pair of translations $(v_Q, v_R) \in \mathbb{R}^4$ that maximizes the overlap area $|P \cap (Q + v_Q) \cap (R + v_R)|$ in $O(n \log^3 n)$ time.*

We also give a deterministic $O(n \log^2 n)$ time algorithm for minimizing the symmetric difference of two convex polygons under a homothety (a translation and a scaling), which is an improvement to Yon et al.’s randomized algorithm [14].

► **Theorem 3.** *Let P and Q be convex polygons with n vertices in total. Then we can find a homothety φ that minimizes the area of symmetric difference $|P \setminus \varphi(Q)| + |\varphi(Q) \setminus P|$ in $O(n \log^2 n)$ time.*

The main ingredient in the proof of Theorem 1 is a new technique we introduce which generalizes Megiddo’s prune-and-search [13]. This allows us to efficiently prune among n groups of m parallel lines.

Let $S = \bigcup_{i=1}^n S_i$ be a union of n sets of $O(m)$ parallel lines in the plane, none of which are parallel to the x -axis, and suppose the lines in each S_i are indexed from left to right.

► **Lemma 4.** *In $O(n)$ time, \mathbb{R}^2 can be partitioned into six regions R_1, \dots, R_6 by three lines, and we can find six subsets $S^{R_1}, \dots, S^{R_6} \subset S$ such that for each $i \leq 6$, S^{R_i} contains all lines intersecting the interior of R_i and $|S^{R_i}| \leq \frac{17}{18}|S|$.*

With this lemma, we can employ divide-and-conquer to obtain the following.

► **Theorem 5.** *Suppose there is an unknown point $p^* \in \mathbb{R}^2$ and we are given an oracle that decides in time T the relative position of p^* to any line in the plane. Then we can find the relative position of p^* to every line in S in $O((T+n)\log(mn))$ time.*

The omitted proofs can be found in the full version of this paper [12].

2 Preliminaries

Let $P \subset \mathbb{R}^3$ be a convex polyhedron and $Q \subset \mathbb{R}^2$ be a convex polygon with n vertices in total. Throughout the paper, we assume that Q is in the xy -plane, and that the point in P with minimal z coordinate is on the xy -plane. We want to find a translation vector $v = (x, y, z) \in \mathbb{R}^3$ that maximizes the overlap area $f(v) = |P \cap (Q + v)|$.

It is easy to observe that $f(v)$ is continuous and piecewise quadratic on the interior of its support. As noted in [6, 1, 3], f is smooth on a region R if $P \cap (Q + v)$ is combinatorially equivalent for all $v \in R$, that is, if we have the same set of face-edge incidences between P and Q . Following the convention of [1], we call the polygons that form the boundaries of these regions the *event polygons*, and as in [6], we call the space of translations of Q the *configuration space*. The arrangement of the event polygons partition the configuration space into cells with disjoint interiors. The overlap function $f(v)$ is quadratic on each cell. Thus, to locate a translation maximizing f , we need to characterize the event polygons.

For two sets $A, B \subset \mathbb{R}^d$, we write the *Minkowski sum* of A and B as

$$A + B := \{a + b \mid a \in A, b \in B\}.$$

We will make no distinction between the translation $A + v$ and the Minkowski sum $A + \{v\}$ for a vector v . We also write $A - B$ for the Minkowski sum of A with $-B = \{-b \mid b \in B\}$. We categorize the event polygons into three types and describe them in terms of Minkowski sums:

- (I) When $Q + v$ contains a vertex of P . For each vertex u of P , we have an event polygon $u - Q$. There are $O(n)$ event polygons of this type.
- (II) When a vertex of $Q + v$ is contained in a face of P . For each face F of P and each vertex v of Q , we have an event polygon $F - v$. There are $O(n^2)$ event polygons of this type.
- (III) When an edge of $Q + v$ intersects an edge of P . For each edge e of P and each edge e' of Q , we have an event polygon $e - e'$. There are $O(n^2)$ event polygons of this type.

The reason that convexity is fundamental is due to the following standard fact, as noted and proved in [6, 14].

► **Proposition 6.** *Let P be a d' -dimensional convex polytope and let Q be a d -dimensional convex polytope. Suppose $d' \geq d$. Let $f(v) = \text{Vol}(P \cap (Q + v))$ be the volume of the overlap function. Then, $f(v)^{1/d}$ is concave on its support $\text{supp}(f) = \{v \mid f(v) > 0\}$.*

As in [5], we say a function $f : \mathbb{R} \rightarrow \mathbb{R}$ is *unimodal* if it increases to a maximum value, possibly stays there for some interval, and then decreases. It is *strictly unimodal* if it strictly increases to the maximum and then strictly decreases. Furthermore, we say a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ is (strictly) unimodal if its restriction to any line is (strictly) unimodal.

The following corollary of Proposition 6 allows us to employ a divide-and-conquer strategy in our algorithm.

► **Corollary 7** ([5]). *For any line l parameterized by $l = p + vt$ in \mathbb{R}^d for $v \neq 0$, the function $f_1(t) = f(p + vt)$ is strictly unimodal.*

We also use the following two techniques in our algorithm.

► **Lemma 8** ([11]). *Let M be an $m \times n$ matrix of real numbers, where $m \leq n$. If every row and every column of M is in increasing order, then we say M is a sorted matrix. For any positive integer k smaller or equal to mn , the k -th smallest entry of M can be found in $O(m \log(2n/m))$ time, assuming an entry of M can be accessed in $O(1)$ time.*

For our purposes, we will use this result in the weaker form of $O(m + n)$.

► **Lemma 9** ([8]). *Given n hyperplanes in \mathbb{R}^d and a region $R \subset \mathbb{R}^d$, a $(1/r)$ -cutting is a collection of simplices with disjoint interiors, which together cover R and such that the interior of each simplex intersects at most n/r hyperplanes. A $(1/r)$ -cutting of size $O(r^d)$ can be computed deterministically in $O(nr^{d-1})$ time. In addition, the set of hyperplanes intersecting each simplex of the cutting is reported in the same time.*

3 Generalized two-dimensional prune-and-search

In this section, we prove Theorem 5, our generalization of Megiddo's prune-and-search technique [13]. This technique is of independent interest and can likely be applied to other problems.

In [13], Megiddo proves the following:

► **Theorem 10** ([13]). *Suppose there exists a point $p^* \in \mathbb{R}^2$ not known to us. Suppose further that we have an oracle that can tell us for any line $l \subset \mathbb{R}^2$ whether $p^* \in l$, and if $p^* \notin l$, the side of l that p^* belongs to. Let T be the running time of the oracle. Then given n lines in the plane, we can find the position of p^* relative to each of the n lines in $O(n + T \log n)$ time.*

We are interested in a generalized version of Megiddo's problem. Suppose, instead of n lines, we are given n sets of parallel lines S_1, S_2, \dots, S_n , each of size $O(m)$. In addition, suppose the lines in each S_i are indexed from left to right (assuming none of the lines are parallel to the x -axis). Again, we want to know the position of p^* relative to every line in $S = \bigcup_{i=1}^n S_i$. Megiddo's algorithm solves this problem in $O(mn + T \log(mn))$ time, but we want a faster algorithm for large m by exploiting the structure of S .

Without loss of generality, suppose that there are no lines parallel to the y -axis. For each i between 1 and n , let $S_i = \{l_i^1, l_i^2, \dots\}$ where l_i^j lies strictly to the left of l_i^{j+1} for all valid j . Suppose that $p^* = (x^*, y^*) \in \mathbb{R}^2$. To report our final answer, we need to provide, for each S_i , the two consecutive indices a and $a + 1$ such that p^* lies strictly between l_i^a and l_i^{a+1} or the single index a such that $p^* \in l_i^a$.

In our algorithm, we keep track of a feasible region R containing P^* , which is either the interior of a (possibly unbounded) triangle or an open line segment if we find a line l that p^* lies on. Together with R , we keep track of the $2n$ indices $\text{lower}(i)$ and $\text{upper}(i)$ such that $S^R = \bigcup_{i=1}^n S_i^R = \{l_i^j \mid j \in (\text{lower}(i), \text{upper}(i))\}$ contains the set of lines intersecting R . In the beginning, $R = \mathbb{R}^2$. Each step, we find $O(1)$ lines to run the oracle on to find a new feasible region $R' \subset R$ such that $|S^{R'}| \leq \frac{17}{18}|S^R|$ and recurse on R' . An outline is given in Algorithm 3.1.

We will use the following well-known result:

► **Lemma 11** ([10]). *Suppose we are given n distinct real numbers with positive weights that sum to 1. Then we can find the weighted median of these numbers in $O(n)$ time.*

Given S^R and R , we want to find $R' \subset R$ to recurse on, as well as efficiently update $S^{R'}$. Note that $S^{R'}$ need not be exactly the set of lines intersecting the interior of R' ; we only need it to contain those lines and have size a constant fraction smaller than S^R .

```

Algorithm 3.1 Pseudocode for Theorem 5.
input : A set  $S = \bigcup_{i=1}^n S_i = \{l_i^j\}$  of  $O(mn)$  lines
output : A list of indices that indicate the position of  $p^*$  to each  $S_i$ 
1  $R \leftarrow \mathbb{R}^2$ 
2  $S^R \leftarrow S$ 
3 while  $|S^R| \geq 18$  do
4   Find  $O(1)$  lines to run the oracle on
5   Compute the piece  $R' \subset R$  containing  $p^*$ 
   /* We guarantee that  $R'$  intersects at most 17/18 of the lines that
   intersect  $R$  */
6   Triangulate  $R'$  with  $O(1)$  lines to run the oracle on
7   Update  $S^R \leftarrow S^{R'}$ 
8 end
9 Compute relative position of  $p^*$  to the remaining lines in  $|S^R|$  by brute force
    
```

Proof of Lemma 4. We write $S^R = S = \bigcup_{i=1}^n S_i = \{l_i^j\}$. We first find the weighted median of the slopes of the lines in S , where the slope of the lines of S_i is weighted by $|S_i|/|S|$. This can be done in $O(n)$ time by Lemma 11.

If this slope is equal to the slope of some line in S_i and $|S_i| \geq \frac{1}{9}|S|$, then we can simply divide the plane using the median line of S_i and the x -axis and the interior of each quadrant will avoid at least $1/18$ of the lines of S . The subsets S^{R_i} can be formed by removing either half of the lines of S_i .

Otherwise, at least $4/9$ of the lines have slopes strictly greater than/less than the median slope. Without loss of generality, we assume at least $4/9$ of the lines have positive slope and at least $4/9$ of the lines have negative slope. Now let $S_+ = \bigcup_{i=1}^k S_i$ and $S_- = \bigcup_{i=k+1}^n S_i$ denote the set of lines with positive/negative slope, respectively. We remove lines from the larger of the two sets until they have the same size.

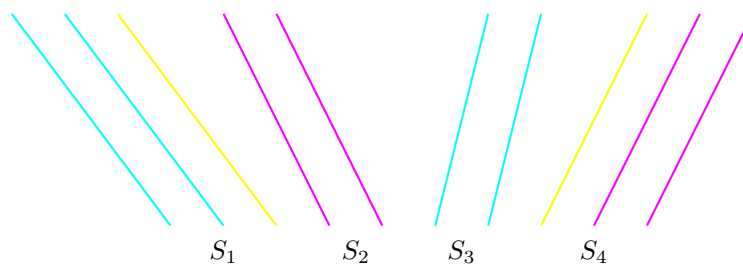
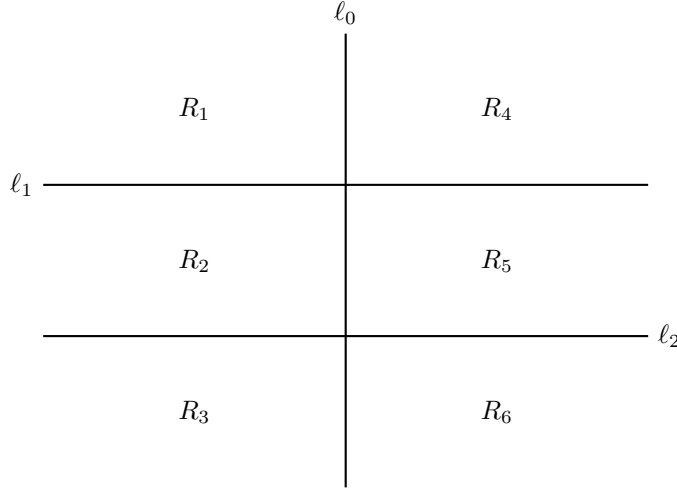


Figure 1 P_1, P_2 are P_3 are represented by colors.

We partition $S_+ \cup S_-$ into $O(n)$ subsets P_i each containing the same number of lines from S_+ and S_- in the following way: going in lexicographical order by the indices of the lines, we put a line from S_1 and a line from S_{k+1} into P_1 until we exhaust one of the sets (say it is S_{k+1}). Then, we move on to put a line from the remaining S_1 and a line from S_{k+2} into P_2 until we exhaust one of them, and so on. Each P_i is then of the form $\{l_{a(i)}^{b(i)}, \dots, l_{a(i)}^{b(i)+|P_i|/2-1}, l_{c(i)}^{d(i)}, \dots, l_{c(i)}^{d(i)+|P_i|/2-1}\}$, and can be represented by the indices $(a(i), b(i))$ and $(c(i), d(i))$ (see Figure 1). We can compute this partition in $O(n)$ time. For each P_i , we compute the intersection $p_i = (x_i, y_i)$ of the median line in P_i with positive slope and the median line with negative slope, and assign p_i a weight $w_i = |P_i|/(2|S_+|)$. Then,

61:6 Maximum Overlap of Polyhedron and Polygon

the weights of the p_i sum to 1. The significance of this is that the interior of each of the four quadrants of the plane defined by $x = x_i$ and $y = y_i$ is avoided by at least $1/4$ of the lines in P_i , which is at least $\frac{2}{9}w_i$ of all the lines in $|S|$.



■ **Figure 2** Dividing the plane into six regions.

We find the median point $q_0 = (x_q, y_q)$ of the p_i 's by weight in x -coordinate in $O(n)$ time by Lemma 11. We have the line $\ell_0 : x = x_{q_0}$. We then find the median point of the p_i 's to the left of ℓ_0 and the median point of those to the right of ℓ_0 by weight in y -coordinates, respectively. Suppose these are $q_1 = (x_{q_1}, y_{q_1})$ and $q_2 = (x_{q_2}, y_{q_2})$. Then let $\ell_1 : y = y_{q_1}$ and $\ell_2 : y = y_{q_2}$. The three lines ℓ_0, ℓ_1 , and ℓ_2 partition the plane into six closed regions R_1, \dots, R_6 as in Figure 2. By our construction, the weights of points in each of $R_1, R_2 \cup R_3, R_4 \cup R_5, R_6$ sum to at least $1/4$. Thus, the interiors of $R_5 \cup R_6, R_4, R_3, R_1 \cup R_2$ each avoids at least $\frac{2}{9} \cdot \frac{1}{4} = \frac{1}{18}$ of all the lines in S . In particular, the interior of each R_i intersects no more than $17/18$ of the lines in S .

We show how to compute $|S^{R_1}|$, and the others follow similarly. If $p_i \in R_6$, then the lines in P_i with positive slope and to the right of p_i avoid $R_1 \cup R_2$. We can remove these lines by updating the indices of the associated set S_j or parallel lines. This updating takes $O(1)$ time for each p_i and $O(n)$ time in total. ◀

Applying Lemma 4 on S^R , we obtain three lines on which we can run the oracle to get a new feasible region R'_i and a subset $S^{R'_i}$. We then triangulate it with $O(1)$ more oracle calls to get R' and set $S^{R'} = S^{R'_i}$, in $O(T + n)$ time total.

Proof of Theorem 5. After $O(\log mn)$ recursive iterations of Lemma 4, we arrive at a feasible region whose interior intersects less than 18 lines in S , and we can finish by brute force. Therefore, our algorithm runs in $O((T + n) \log(mn))$ time. ◀

► **Remark 12.** A simpler and probably more practical algorithm for Lemma 4 is simply choosing a random line from S_+ and S_- to intersect and run the oracle on the horizontal and vertical line through the intersection. This method gives the same run time in expectation.

4 Maximum overlap of convex polyhedron and convex polygon

In this section, we give the algorithm that finds a translation $v \in \mathbb{R}^3$ maximizing the area of overlap function f . Following the convention in [6], we call such a translation a *goal placement*. In the algorithm, we keep track of a closed *target region* R which we know contains a goal placement and decrease its size until for each event polygon F , either $F \cap \text{interior}(R) = \emptyset$ or $F \supset R$. Then, f is quadratic on R and we can find the maximum of f on R using standard calculus. Thus, the goal of our algorithm is to efficiently trim R to eliminate event polygons that intersect it.

In the beginning of the algorithm, the target region is the interior of the Minkowski sum $P - Q$, where the overlap function is positive. By the unimodality of the overlap function, the set of goal placements is convex. Thus, for a plane in the configuration space, either it contains a goal placement, or all goal placements lie on one of the two open half spaces separated by the plane. If we have a way of knowing which case it is for any plane, we can decrease the size of our target region by cutting it with planes and finding the piece to recurse. More precisely, we need a subroutine **PlaneDecision** that decides the relative position of the set of goal placements to a plane S .

Whenever **PlaneDecision** reports that a goal placement is found on a plane, we can let the algorithm terminate. Thus, we can assume it always reports a half-space containing a goal placement.

As in Algorithm 4.1, we break down our algorithm into three stages.

Algorithm 4.1 Pseudocode for Theorem 1.

<p>input : A convex polyhedron $P \in \mathbb{R}^3$ and a convex polygon $Q \in \mathbb{R}^3$ with n vertices in total</p> <p>output : A translation $v \in \mathbb{R}^3$ maximizing the area $P \cap (Q + v)$</p> <ol style="list-style-type: none"> 1 Locate a horizontal slice containing a goal placement that does not contain any vertices of P and replace P by this slice of P 2 Find a “tube” $D + l_y$ whose interior contains a goal placement and intersects $O(n)$ event polygons, where D is a triangle in the xz-plane and l_y is the y-axis 3 Recursively construct a (1/2)-cutting of the target region $D + l_y$ to find a simplex containing a goal placement that does not intersect any event polygon
--

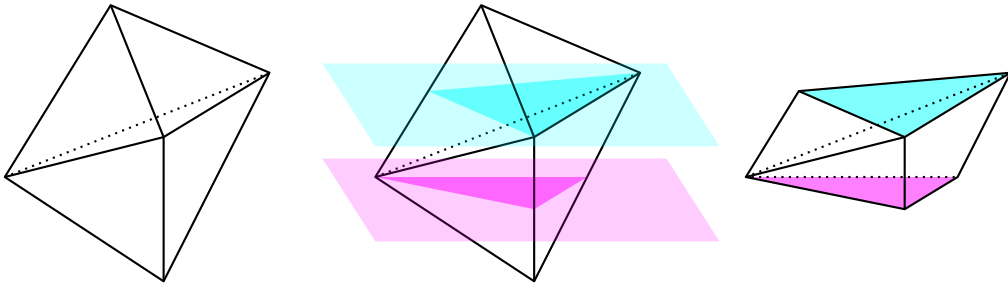
4.1 Stage 1

In the first stage of our algorithm, we make use of [6] to simplify our problem so that P can be taken as a convex polyhedron with all of its vertices on two horizontal planes.

We sort the vertices of P by z -coordinate in increasing order and sort the vertices of Q in counterclockwise order. Next, we trim the target region with horizontal planes (planes parallel to the xy -plane) to get to a slice that does not contain any vertices of P .

► **Lemma 13.** *In $O(n \log^2 n)$ time, we can locate a strip $R = \{(x, y, z) | z \in [z_0, z_1]\}$ whose interior contains a goal placement and P has no vertices with $z \in [z_0, z_1]$.*

Proof. Starting with the median z -coordinate of the vertices of P , we perform a binary search on the levels containing a vertex of P . For a horizontal plane S , [6, Theorem 3.8] allows us to compute the maximum overlap of $P \cap S$ and Q under translation in $O(n \log n)$ -time. The



■ **Figure 3** The slice of P with $z \in [z_0, z_1]$.

two planes S_1 and S_2 with the largest maximum values will be the bounding planes for the slice containing a goal placement by the unimodality of f . Thus, by a binary search, we can locate this slice in $O(n \log^2 n)$ time. ◀

By Chazelle's algorithm [7], the convex polyhedron $P' = \{(x, y, z) \in P \mid z \in [z_0, z_1]\}$ can be computed in $O(n)$ time. From now on, we replace P with P' (see Figure 3). Without loss of generality, assume $z_0 = 0$ and $z_1 = 1$.

The region in the configuration space where $|P \cap (Q + v)| > 0$ is the Minkowski sum $P - Q$. Since P only has two levels $P_0 = \{(x, y, z) \in P \mid z = 0\}$ and $P_1 = \{(x, y, z) \in P \mid z = 1\}$ that contain vertices, the Minkowski sum $P - Q$ is simply the convex hull of $(P_0 - Q) \cup (P_1 - Q)$, which has $O(n)$ vertices. We can compute $P_0 - Q$ and $P_1 - Q$ in $O(n)$ time and compute their convex hull in $O(n \log n)$ time by Chazelle's algorithm [9].

4.2 PlaneDecision

With the simplification of the problem in Stage 1, we now show that the subroutine **PlaneDecision** can be performed in $O(n \log n)$ time. Let S be a fixed plane in the configuration space. We call a translation v that achieves $\max_{v \in S} f(v)$ a *good placement*. First, we can compute the intersection of S with $P - Q$ in $O(n)$ time by Chazelle's algorithm [7]. If the intersection is empty, we just report the side of S containing $P - Q$. From now on assume this is not the case.

The following lemma shows that **PlaneDecision** runs in the same time bound as the algorithm that just finds the maximum of f on a plane.

► **Lemma 14.** *Suppose we can compute $\max_{v \in S} f(v)$ for any plane $S \subset \mathbb{R}^3$ in time T , then we can perform **PlaneDecision** for any plane in time $O(T)$.*

Proof. The idea is to compute $\max_{v \in S'} f(v)$ for certain S' that are perturbed slightly from S to see in which direction relative to S does f increase.

We compute over an extension of the reals $\mathbb{R}[\omega]/(\omega^3)$, where $\omega > 0$ is smaller than any real number. Let $A > 0$ be the maximum of f over a plane S . Let S_+ and S_- be the two planes parallel to S that have distance ω from S . We compute $A_+ = \max_{v \in S_+} f(v)$ and $A_- = \max_{v \in S_-} f(v)$ in $O(T)$ time. Since f is piecewise quadratic, A_+ and A_- as symbolic expression will only involve quadratic terms in ω . Since f is strictly unimodal on $P - Q$, there are three possibilities:

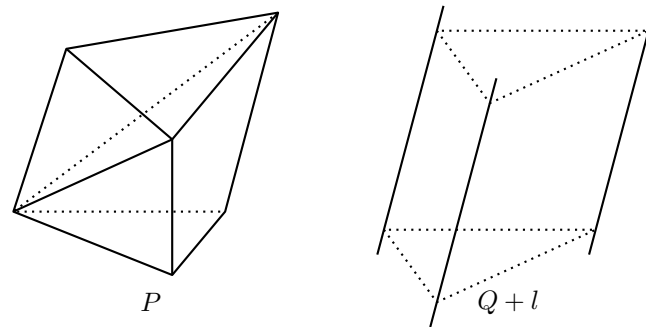
1. If $A_+ > A$, then halfspace on the side of S_+ contains the set of goal placements.
2. If $A_- > A$, then halfspace on the side of S_- contains the set of goal placements.
3. If $A \geq A_+$ and $A \geq A_-$, then A is the global maximum of f .

Thus, in $O(T)$ time, we can finish **PlaneDecision**. ◀

Finding a good placement on S is similar to finding a goal placement on the whole configuration space. S is partitioned into cells by the intersections of event polygons with S . We need to find a region on S containing a good placement that does not intersect any event polygons.

We present a subroutine **LineDecision** that finds, for a line $l \subset S$, the relative position of the set of good placements on S to l .

► **Proposition 15.** *For a line $l \subset S$, we can perform **LineDecision** in $O(n)$ time.*



■ **Figure 4** The convex polyhedron I is formed by intersecting P and $(Q + l)$.

Proof. First, we compute $\max_{v \in l} f(v)$ and a vector achieving the maximum. We parameterize the line l by $p + vt$ where t is the parameter and $p, v \in \mathbb{R}^3$. The horizontal cross-section of $I = P \cap (Q + l)$ at height t has area $f(p + vt)$. Since I is the intersection of two convex polytopes with $O(n)$ vertices (see Figure 4), Chazelle’s algorithm [7] computes I in $O(n)$ time. Then, [5, Theorem 3.2] computes the maximum cross-section in $O(n)$ time.

Now, by the same argument and method as in the proof of Lemma 14, we can finish **LineDecision** in $O(n)$ time. In the case where $\max_{v \in l} f(v) = 0$, we report the side of l containing $S \cap (P - Q)$. ◀

Whenever our subroutine **LineDecision** reports a good placement is found on a line, we can let the algorithm terminate. Thus, we can assume it always reports a half-plane of S containing a good placement.

We now present **PlaneDecision**. If S is horizontal, then we only need to find the maximum overlap of the convex polygons $P \cap S$ and Q using De Berg et al.’s algorithm [6], which takes $O(n \log n)$ time. Thus, we assume S is non-horizontal.

■ **Algorithm 4.2** Pseudocode for **PlaneDecision**.

input : A plane $S \subset \mathbb{R}^3$
output : A translation $v \in S$ maximizing the area $|P \cap (Q + v)|$

- 1 Compute $S \cap (P - Q)$ and set it to be our initial target region
- 2 Locate a strip on S containing a good placement whose interior intersects $O(n)$ event polygons
- 3 Recursively construct a $(1/2)$ -cutting of the strip to find a triangle containing a good placement that does not intersect any event polygon

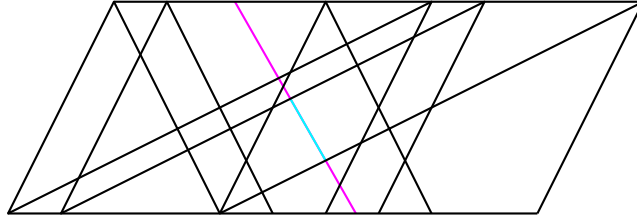
As in Algorithm 4.2, we break down **PlaneDecision** into three steps. We have already explained Step 1, where we compute $S \cap (P - Q)$, so we begin with Step 2.

4.2.1 PlaneDecision: Step 2

We want to find a strip on S strictly between $z = 0$ and $z = 1$ that intersects $O(n)$ event polygons. Since there are no vertices of P with z -coordinate in the interval $(0, 1)$, there are no event polygons of type I in this range, and we will only need to consider event polygons of type II and type III.

We look at the intersection points of S with the edges of the event polygons. These edges come from the set $\{e_i - v_j | e_i \text{ non-horizontal edge of } P, v_j \text{ vertex of } Q\}$. Without loss of generality, assume that S is parallel to the y -axis. We are interested in the z -coordinates of the intersections, so we project everything into the xz -plane. Then, S becomes a line, which we denote by l_S , and each edge $e_i - v_j$ becomes a segment whose endpoints lie on $z = 0$ and $z = 1$. Suppose each edge e_i projects to a segment s_i , and each v_j projects to a point x_j on the x -axis. Then, we get $O(n^2)$ segments $s_i - x_j$ with endpoints on $z = 0$ and $z = 1$, and the line l_S that intersect them in some places.

► **Lemma 16.** *In $O(n \log n)$ time, we can locate a strip $R = \{(x, y, z) \in S | z \in [z_0, z_1]\}$ whose interior contains a good placement and intersects none of the edges of the event polygons.*



■ **Figure 5** Projecting the configuration space onto the xz -plane. The projection of S is the magenta line segment, and the projection of the strip R obtained from Lemma 16 is the cyan line segment.

Our current target region, the strip R we obtained from Lemma 16 (see Figure 5), intersects few event polygons and we can compute them efficiently.

► **Lemma 17.** *The interior of the region R intersects $O(n)$ event polygons, and we can compute them in $O(n \log n)$ time.*

4.2.2 PlaneDecision: Step 3

Now we have a target region R as well as the $O(n)$ intersections it makes with the event polygons.

► **Lemma 18.** *In $O(n \log n)$ time, we can find a region $R' \subset R$ containing a good placement that does not intersect any of the $O(n)$ event polygons.*

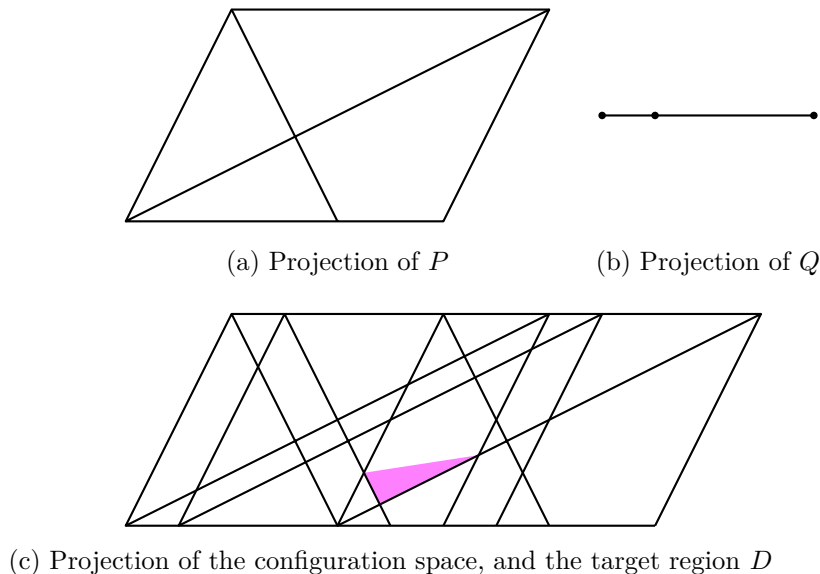
Proof. We recursively construct a $(1/2)$ -cutting of the target region. By Lemma 9, a $(1/2)$ -cutting of constant size can be computed in $O(n)$ time. We perform **LineDecision** on the lines of the cutting to decide on which triangle to recurse. After $O(\log n)$ iterations, we have a target region R' that intersects no event polygons. This procedure runs in $O(n \log n)$ time. ◀

Finally, since the overlap function is quadratic on our final region R' , we can solve for the maximum using standard calculus. After finding $\max_{v \in S} f(v)$ and a vector achieving it $O(n \log n)$ time, by Lemma 14, we can perform **PlaneDecision** on S in the same time bound.

► **Proposition 19.** *For a plane S , we can perform **PlaneDecision** in $O(n \log n)$ time.*

4.3 Stage 2

With the general **PlaneDecision** at our disposal, we now move on to Stage 2, the main component of our algorithm. We project the entire configuration space and the event polygons onto the xz -plane in order to find a target region D whose preimage $D + l_y$ intersects few event polygons, where l_y is the y -axis (see Figure 6).



■ **Figure 6** Projecting onto the xz -plane.

The non-horizontal edges of the event polygons project to segments on the strip $0 < z < 1$ on the xz -plane. We characterize our desired region D in the following lemma.

► **Lemma 20.** *For a region D that does not intersect any of the segments that are the projections of the non-horizontal edges of the event polygons, the preimage $D + l_y$ intersects $O(n)$ event polygons.*

Now it remains to efficiently find such a region D with $D + l_y$ containing a goal placement and compute the $O(n)$ event polygons that intersect its interior.

► **Lemma 21.** *In $O(n \log^2 n)$ time, we can find a triangle D in the xz -plane such that the interior of $D + l_y$ contains a goal placement and intersects $O(n)$ event polygons. We can compute these $O(n)$ event polygons in the same time bound.*

Proof. The computation of D is a direct application of Theorem 5, where $m = O(n)$. Calling the oracle on a line l in the xz -plane is running the **PlaneDecision** algorithm on the plane parallel to the y -axis that projects to l . We compute a triangle for each of the four groups of segments, take their intersection, and triangulate the intersection using $O(1)$ calls to **PlaneDecision**. Thus, we can compute the desired triangle D in $O(n \log^2 n)$ time.

To compute the event polygons intersecting the interior of $D + l_y$ is simple, since we have shown in the proof of Lemma 20 that D intersects at most one projection of an event polygon of each type in each of the four groups for a fixed vertex x_j (for type II) or segment s_i (for type III). Once we have D , we can compute these polygons by binary search on each of the $O(n)$ groups of $O(n)$ non-intersecting segments to find the two between which R lies. Also, the event polygons all have constant complexity so computing all of them takes linear

61:12 Maximum Overlap of Polyhedron and Polygon

time. We can recover the event polygons from their projections and compute the planes that contain them in linear time. Thus, this entire process can be done in $O(n \log n)$ time. ◀

4.4 Stage 3

Now, we have a target region $R = D + l_y$ whose interior contains a goal placement, and we have the $O(n)$ event polygons that intersect it.

► **Lemma 22.** *In $O(n \log^2 n)$ time, we can find a region $R' \subset R$ containing a goal placement that does not intersect any of the $O(n)$ event polygons.*

Proof. We recursively construct a $(1/2)$ -cutting of the target region. By Lemma 9, a $(1/2)$ -cutting of constant size can be computed in $O(n)$ time. We perform **PlaneDecision** on the planes of the cutting to decide on which simplex to recurse. After $O(\log n)$ iterations, we have a target region R' that intersects no event polygons. This procedure runs in $O(n \log^2 n)$ time. ◀

Finally, since the overlap function is quadratic on our final region R' , we can solve for the maximum using standard calculus. This concludes the proof of Theorem 1.

5 Maximum overlap of three convex polygons

Let P, Q, R be three convex polygons with n vertices in total in the plane. We want to find a pair of translations $(v_Q, v_R) \in \mathbb{R}^4$ that maximizes the overlap area $g(v_Q, v_R) = |P \cap (Q + v_Q) \cap (R + v_R)|$.

In this problem, the configuration space is four-dimensional. An easy extension of Proposition 6 and Corollary 7 shows that the function of overlap area is again unimodal. This time, we have four-dimensional *event polyhedra* instead of event polygons that divide the configuration space into four-dimensional cells on which $g(v_Q, v_R)$ is quadratic. We call a hyperplane containing an event polyhedron an *event hyperplane*, and they are defined by two types of events:

- (I) When one vertex of $P, Q + v_Q$ or $R + v_R$ lies on an edge of another polygon. There are $O(n)$ groups of $O(n)$ parallel event hyperplanes of this type.
- (II) When an edge from each of the three polygons intersect at one point. There are $O(n^3)$ event hyperplanes of this type.

To overcome the difficulty of dealing with the $O(n^3)$ event hyperplanes of type II, we first prune the configuration space to a region intersecting no event hyperplanes of type I. We then show that the resulting region only intersects $O(n)$ event hyperplanes of type II.

Similar to Theorem 1, we want an algorithm **HyperplaneDecision** that computes, for a hyperplane $H \subset \mathbb{R}^4$, the maximum $\max_{(v_Q, v_R) \in H} g(v_Q, v_R)$ and the relative location of the goal placement to H . In fact, we will only need to perform **HyperplaneDecision** on some hyperplanes.

► **Proposition 23.** *Suppose H is a hyperplane that satisfies one of the following three conditions:*

- (1) H is orthogonal to a vector $(x_1, y_1, 0, 0)$ for some $x_1, y_1 \in \mathbb{R}$.
- (2) H is orthogonal to a vector $(0, 0, x_2, y_2)$ for some $x_2, y_2 \in \mathbb{R}$.
- (3) H is orthogonal to a vector $(x_1, y_1, -x_1, -y_1)$ for some $x_1, y_1 \in \mathbb{R}$.

*Then, we can perform **HyperplaneDecision** on H in $O(n \log^2 n)$ time.*

Using Proposition 23, we can prune the configuration space to a region that intersects no event hyperplanes of type I and $O(n)$ event hyperplanes of type II.

► **Proposition 24.** *We can compute a 4-polytope T_{PQR} of complexity $O(1)$ in $O(n \log^3 n)$ time such that*

- (1) *the goal placement lies on T_{PQR} ,*
- (2) *no hyperplane of type I intersects the interior of T_{PQR} , and*
- (3) *only $O(n)$ event polyhedrons of type II passes through T_{PQR} .*

The hyperplanes of type II intersecting the interior of T_{PQR} are obtained in the same time bound. Furthermore, the 3-tuples of edges of P , Q and R defining the hyperplanes are also obtained in the same time bound.

In the rest of the section, we fix T_{PQR} as in Proposition 24. Moreover, let

$$f(v_P, v_Q) = \begin{cases} |P \cap (Q + v_Q) \cap (R + v_R)| & \text{if } (v_Q, v_R) \in T_{PQR} \\ 0 & \text{otherwise.} \end{cases}$$

► **Proposition 25.** *Let S be any m -flat in the configuration space. In $O(n)$ time, we can find a point in $S \cap \text{supp } f$, or report that $S \cap \text{supp } f$ is empty.*

Proof. Notice that $\text{supp } f$ is a convex 4-polytope whose face are hyperplanes of type I or type II. Let H be a hyperplane of type II intersecting the interior of T_{PQR} . Then H contains a face of $\text{supp } f$ if and only if a polygon $P \cap Q$ is tangent to R in $H \cap T_{PQR}$. This can be tested in constant time, so we can find all faces of $\text{supp } f$ in $O(n)$ time. Our problem become a feasibility test of a linear programming of size $O(n)$, which can be solved in $O(n)$ time by Megiddo’s algorithm [13]. ◀

Proof of Theorem 2. Take T_{PQR} as in Proposition 24. Let

$$f(v_P, v_Q) = \begin{cases} |P \cap (Q + v_Q) \cap (R + v_R)| & \text{if } (v_Q, v_R) \in T_{PQR} \\ 0 & \text{otherwise.} \end{cases}$$

Then f is unimodal and the maximum of f is the goal placement. Given an m -flat S , we want to compute the maximum of f on S in $O(n \log^{m-1})$ time by induction on $m \in \{1, 2, 3, 4\}$.

If $m = 1$, this can be done in $O(n)$ time by Proposition 15. Assume that $m > 1$. Then $S \cap T_{PQR}$ can be computed in $O(1)$ time. Given an $(m - 1)$ -flat $l \subset S$, we can use Proposition 25 and the perturbation method as in Lemma 14 to report the relative position of the maximum over S . There are $O(n)$ event hyperplane intersecting $S \cap T_{PQR}$. Thus, by Lemma 9, we can recursively construct (1/2)-cuttings to give an $O(n \log^{m-1})$ time algorithm to find the maximum of f on S . ◀

6 Minimum symmetric difference of two convex polygons under homothety

A homothety $\varphi: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is a composition of a scaling and a translation. Let $\lambda > 0$ be the scaling factor and v be the translation vector of φ . Then

$$\varphi(A) = \lambda A + v = \{\lambda p + v \mid p \in A\}.$$

Define the *symmetric difference* of sets $A, B \subset \mathbb{R}^2$ to be

$$\begin{aligned} A \Delta B &:= (A \cup B) \setminus (A \cap B) \\ &= (A \setminus B) \cup (B \setminus A). \end{aligned}$$

61:14 Maximum Overlap of Polyhedron and Polygon

Let P and Q be convex polygons with n vertices in total. We want to find a homothety φ of Q that minimizes the area of symmetric difference

$$h(\varphi) = h(x, y, \lambda) = |P \Delta \varphi(Q)|,$$

where $\varphi(Q) = \lambda Q + (x, y)$.

Yon et al. [14] consider a slightly more general problem, where they minimize the function

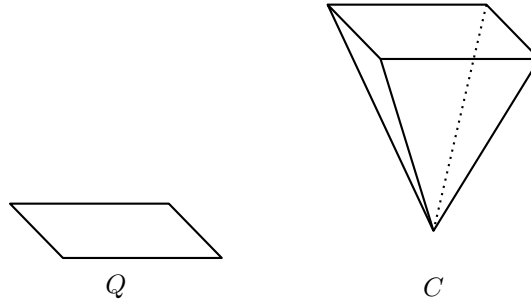
$$h(\varphi) = (2 - 2\kappa)|P \setminus \varphi(Q)| + 2\kappa|\varphi(Q) \setminus P|,$$

where $\kappa \in (0, 1)$ is some constant. When $\kappa = 1/2$, this is the area of symmetric difference function. They give a randomized algorithm that solves this problem in $O(n \log^3 n)$ expected time. We present a faster deterministic algorithm by relating this problem to the polyhedron-polygon matching problem and then applying a modified version of Theorem 1.

As in [14], we rewrite the objective function $h(\varphi)$:

$$\begin{aligned} h(\varphi) &= 2(1 - \kappa)|P| + 2\kappa|\varphi(Q)| - 2|P \cap \varphi(Q)| \\ &= 2(1 - \kappa)|P| + 2\kappa|Q|\lambda^2 - 2|P \cap \varphi(Q)|. \end{aligned}$$

Thus, minimizing $h(\varphi)$ is the same as maximizing $f(\varphi) = |P \cap \varphi(Q)| - c\lambda^2$, where $c = \kappa|Q|$.



■ **Figure 7** Formation of the cone C .

Consider the cone $C = \{(x, y, \lambda) | \lambda \in [0, M], (x, y) \in \lambda Q\}$, where $M = \sqrt{|P|/c}$ (see Figure 7). Then f is negative for $\lambda > M$ so it is never maximized. We also put P into \mathbb{R}^3 by $P = \{(x, y, 0) | (x, y) \in P\}$. Since $f(x, y, \lambda) = |C \cap (P + (-x, -y, \lambda))| - c\lambda^2$, the problem reduces to maximizing the overlap area of the cone C and P under translation subtracted by a quadratic function. To show that we can still use a divide-and-conquer strategy, we identify a region where f is strictly unimodal.

► **Lemma 26** ([14]). *The closure \mathcal{D} of the set $\{\varphi \in \mathbb{R}^3 | f(\varphi) > 0\}$ is convex. Furthermore, $f(x, y, \lambda)$ is strictly unimodal on \mathcal{D} .*

Proof. This follows from [14, Lemma 2.2] and [14, Lemma 2.7]. ◀

Although it is difficult to directly compute \mathcal{D} , note that $-P \subset \mathcal{D}$. With this observation, we show that we can still find the relative position of the set of goal placements to certain planes S in $O(n \log n)$ time with some modifications to **LineDecision** and **PlaneDecision**.

► **Lemma 27.** *For any $l \subset \mathbb{R}^3$, we can compute $\max_{\varphi \in l} f(\varphi)$ or report it is a negative number in $O(n)$ time.*

► **Proposition 28.** *Let $S \subset \mathbb{R}^3$ be a plane. If S is horizontal or if S intersects the polygon $-P \subset \mathcal{D}$, then we can perform **PlaneDecision** on S in $O(n \log n)$ time.*

► **Theorem 29.** *Let P and Q be convex polygons with n vertices in total. Suppose $\kappa \in (0, 1)$ is a constant. We can find a homothety φ that minimizes*

$$h(\varphi) = 2(1 - \kappa)|P \setminus \varphi(Q)| + 2\kappa|\varphi(Q) \setminus P|$$

in $O(n \log^2 n)$ time.

Proof. We want to maximize $f(x, y, \lambda) = |C \cap (P + (-x, -y, \lambda))| - c\lambda^2$ over \mathbb{R}^3 , where $c = \kappa|Q|$. In order to apply our algorithm for Theorem 1, we need to show that we only run **PlaneDecision** on horizontal planes and planes that intersect $-P$.

In the first stage (as outlined in Algorithm 4.1), we only run **PlaneDecision** on horizontal planes.

In the second stage, we apply Theorem 5 to the $O(n)$ groups of $O(n)$ lines that are the projections of the lines containing edges of event polygons on the xz -plane. Observe that these lines all intersect the projection of $-P$ on the xz -plane. In each recursive step of our algorithm, we query a horizontal (parallel to the x -axis) line and a line that goes “between” two lines in the $O(n^2)$ lines. The planes they represent both satisfy the condition for Proposition 28. Then we run **PlaneDecision** $O(1)$ more times to triangulate our feasible region. Here, we make a small modification: instead of maintaining a triangular feasible region, we maintain a trapezoidal one by making $O(1)$ horizontal cuts to make the region a trapezoid.

In the third stage, we have a “tube” and $O(n)$ event polygons that intersect it. As usual, we recursively construct a $(1/2)$ -cutting by Lemma 9. Chazelle’s algorithm [8] picks $O(1)$ planes intersecting the target region as the cutting, along with $O(1)$ extra planes to triangulate each piece. All the planes containing the event polygons intersect $-P$, so we can run **PlaneDecision** on them. Instead of triangulating our target region, it suffices to reduce it to constant complexity. We do this by cutting it with $O(1)$ horizontal planes such that the remaining region only has vertices on two levels. Then, let e be any non-horizontal edge. With $O(1)$ planes through e , we can cut the target region into prisms and pyramids with triangular bases. These planes all intersect $-P$ since they are between the two faces of the target region containing e , and the planes containing them intersect $-P$.

Therefore, with slight modifications to Theorem 1, we obtain a deterministic $O(n \log^2 n)$ algorithm for minimizing $h(\varphi)$. ◀

Theorem 3 follows as a direct corollary of Theorem 29.

References

- 1 Hee-Kap Ahn, Peter Brass, and Chan-Su Shin. Maximum overlap and minimum convex hull of two convex polyhedra under translations. *Comput. Geom.*, 40(2):171–177, 2008. doi:10.1016/j.comgeo.2007.08.001.
- 2 Hee-Kap Ahn, Siu-Wing Cheng, Hyuk Jun Kweon, and Juyoung Yon. Overlap of convex polytopes under rigid motion. *Comput. Geom.*, 47(1):15–24, 2014. doi:10.1016/j.comgeo.2013.08.001.
- 3 Hee-Kap Ahn, Siu-Wing Cheng, and Iris Reinbacher. Maximum overlap of convex polytopes under translation. *Comput. Geom.*, 46(5):552–565, 2013. doi:10.1016/j.comgeo.2011.11.003.
- 4 Hee-Kap Ahn, Otfried Cheong, Chong-Dae Park, Chan-Su Shin, and Antoine Vigneron. Maximizing the overlap of two planar convex sets under rigid motions. *Comput. Geom.*, 37(1):3–15, 2007. doi:10.1016/j.comgeo.2006.01.005.

- 5 David Avis, Prosenjit Bose, Thomas C. Shermer, Jack Snoeyink, Godfried Toussaint, and Binhai Zhu. On the sectional area of convex polytopes. In *Communication at the 12th Annu. ACM Sympos. Comput. Geom.*, page C. Association for Computing Machinery, New York, NY, 1996.
- 6 Mark de Berg, Olivier Devillers, Marc van Kreveld, Otfried Schwarzkopf, and Monique Teillaud. Computing the maximum overlap of two convex polygons under translations. In *International Symposium on Algorithms and Computation*, pages 126–135. Springer, 1996.
- 7 Bernard Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM J. Comput.*, 21(4):671–696, 1992. doi:10.1137/0221041.
- 8 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(2):145–158, 1993. doi:10.1007/BF02189314.
- 9 Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10(4):377–409, 1993. doi:10.1007/BF02573985.
- 10 Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to algorithms*. MIT Press, Cambridge, MA, third edition, 2009.
- 11 Greg N. Frederickson and Donald B. Johnson. Generalized selection and ranking: sorted matrices. *SIAM J. Comput.*, 13(1):14–30, 1984. doi:10.1137/0213002.
- 12 Hyuk Jun Kweon and Honglin Zhu. Maximum overlap area of a convex polyhedron and a convex polygon under translation, 2023. doi:10.48550/ARXIV.2301.02949.
- 13 Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *J. Assoc. Comput. Mach.*, 31(1):114–127, 1984. doi:10.1145/2422.322418.
- 14 Juyoung Yon, Sang Won Bae, Siu-Wing Cheng, Otfried Cheong, and Bryan T. Wilkinson. Approximating convex shapes with respect to symmetric difference under homotheties. In *32nd International Symposium on Computational Geometry*, volume 51 of *LIPICs. Leibniz Int. Proc. Inform.*, pages Art. No. 63, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016.

Godzilla Onions: A Skit and Applet to Explain Euclidean Half-Plane Fractional Cascading

Richard Berger

Ursinus College Computer Science, Collegeville, PA, USA

Vincent Ha

Ursinus College Computer Science, Collegeville, PA, USA

David Kratz

Ursinus College Computer Science, Collegeville, PA, USA

Michael Lin

Ursinus College Computer Science, Collegeville, PA, USA

Jeremy Moyer

Ursinus College Computer Science, Collegeville, PA, USA

Christopher J. Tralie   

Ursinus College Mathematics And Computer Science, Collegeville, PA, USA

Abstract

We provide a skit and an applet to illustrate fractional cascading in the context of half-plane range search for points in the Euclidean plane, which takes $O(\log N + h)$ output-sensitive time. In the video, a group of news anchors struggles to find the correct data structure to efficiently send out an early warning to the residents of Philadelphia who will be overtaken by a marching line of Godzillas. After exploring several options, the group eventually settles on onions and fractional cascading, only to discover that they themselves are in the line of fire! In the applet, we show step by step details of preprocessing to build the onions with fractional cascading and the subsequent query of the “Godzilla line” against the onion layers. Our video skit and applet can be found at <https://ctralie.github.io/GodzillaOnions/>

2012 ACM Subject Classification Human-centered computing → Visualization toolkits; Theory of computation → Randomness, geometry and discrete structures

Keywords and phrases convex hulls, onions, fractional cascading, visualization, d3

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.62

Category Media Exposition

Supplementary Material *Software (Source Code)*: <https://ctralie.github.io/GodzillaOnions/>

1 Background

Given N points X in the Euclidean plane and a query line ℓ , a naive algorithm to determine the points in X above ℓ would be to check each $x_i \in X$ in turn. However, this approach takes $O(N)$ time.

Alternatively, one can obtain an output-sensitive algorithm if one preprocesses X into an “onions” data structure with a nested sequence of convex hulls from the outside to the inside [2]. For N_L onion layers, refer to the i^{th} layer as L_i , where i indexes the layers in the order in which they are constructed, so L_0 is the outermost layer. If there are h points above the line, each onion layer L_i can be queried with binary search in $O(\log N)$ time to find the line segment with the closest slope to that of ℓ , and this line segment contains the furthest point in that layer from ℓ . If this point is above the line in layer L_i , one can walk to the left and to the right to gather all points above the line in this layer. Overall, this takes



© Richard Berger, Vincent Ha, David Kratz, Michael Lin, Jeremy Moyer, and Christopher J. Tralie;

licensed under Creative Commons License CC-BY 4.0

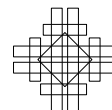
39th International Symposium on Computational Geometry (SoCG 2023).

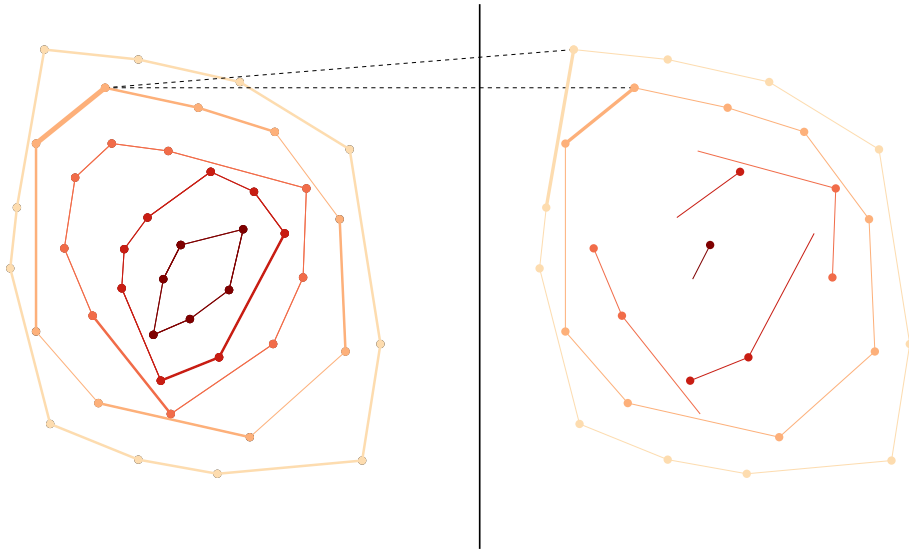
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 62; pp. 62:1–62:3

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** A screenshot from our Javascript d3 interface showing how each point in M_0 has a pointer to a point in L_0 (the outer layer, duplicated on the right) and M_1 (the second from the outer cascaded layer, duplicated on the right) with the closest slope. Preprocessing and storing such pointers allows quick traversal through the structure. Note also: the thickest line segment on the left shows the particular line segment whose pointers the app is highlighting. The middle thick line segments on the left show L_0 and M_1 in the context of the whole onion.

$O(N_i \log N + h)$ time. However, as we note in our video, there may be too many layers; in particular, N_L is $\Theta(N^{2/3})$ for points distributed uniformly and independently at random within any bounded 2D region that contains a disc [3]. The ensuing $O(N^{2/3} \log N + h)$ algorithm is still marginally better than the brute force $O(N)$ approach, but one can do better.

A superior output-sensitive algorithm relies on a more involved onion-based preprocessed data structure that uses fractional cascading [4]. In addition to storing the layers L_i , one constructs parallel layers M_i . The last layer $M_{N_L} = L_{N_L}$. From there, one iteratively constructs M_i as the union of L_i and every other element of M_{i+1} , sorted by slope. Each element in M_i also stores a pointer to the points in L_i and M_{i+1} with the nearest slope. After preprocessing, one starts querying the fractionally cascaded onions by first searching for the point in the outer M layer M_0 with the slope closest to ℓ using binary search. Since M_i only takes every other point in M_{i+1} , the number of points in $\cup_i M_i$ is $O(N)$, so the binary search query on M_0 takes $O(\log N)$ time. From there, one follows the pointer to L_0 to extract all points at that layer that are above ℓ . Then, one follows the pointer to M_1 and continues the process until getting to a layer with no points above ℓ . The preprocessed pointers allow one to walk from layer to points above the line in the subsequent layer in constant time, so the overall process takes only $O(\log N + h)$ time.

2 Applet Details

One of our major contributions is an applet to incrementally construct this rather intricate data structure. To that end, we create an interactive applet using d3 in Javascript [1] to construct and query and onions data structure on top of user selected points. We first show the process of constructing the L_i layers, which we color code. Then, we show how each M_i

is created by merging by slope L_i and every other element of M_{i+1} . Once that is finished, we show a few examples of pointers from M_0 to L_0 and M_1 (Figure 1). Finally, the user queries a “Godzilla line” ℓ , and the applet shows how to incrementally walk through the layers and follow the pointers to obtain all of the points above ℓ .



References

- 1 Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D³ data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.
- 2 Bernard Chazelle. On the convex layers of a planar set. *IEEE Transactions on Information Theory*, 31(4):509–517, 1985.
- 3 Ketan Dalal. Counting the onion. *Random Structures & Algorithms*, 24(2):155–165, 2004.
- 4 Leonidas Guibas and Bernard Chazelle. Fractional cascading: I. a data structuring technique. *Algorithmica*, 1:133–162, 1986.

Interactive 2D Periodic Graphs

Alexandra Camero  

University of Massachusetts - Amherst, MA, USA

Ileana Streinu  

Smith College, Northampton, MA, USA

Abstract

We present an educational web app for interactively drawing and editing 2D periodic graphs. The user defines the unit cell and the finite set of vertex and edge representatives, from which a sufficiently large fragment of the periodic graph is created for the visualization. The periodic graph can also be modified by applying several transformations, including isometries and relaxations of the unit cell. A finite representation of the infinite periodic graph can be saved in an external file as a quotient graph with Z^2 -marked edges. Its geometry is recorded using fractional (crystallographic) coordinates. The facial structure of non-crossing periodic graphs can be revealed by the user interactively selecting face representatives. An accompanying video demonstrates the functionality of the web application.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Periodic graphs, isometric transformations

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.63

Category Media Exposition

Funding Ileana Streinu: NSF CCF-2212309

1 Introduction

Motivated by applications in crystallography and materials science [4], periodic graphs have been studied intensely in recent years in the context of rigidity of bar-and-joint frameworks [1, 2, 3, 5]. They also appear in computational topology as the universal cover of geodesic toroidal drawings [6, 7]. We present an educational web app to create, edit, visualize and generate finite descriptions of 2D geometric periodic graphs (Fig. 1).

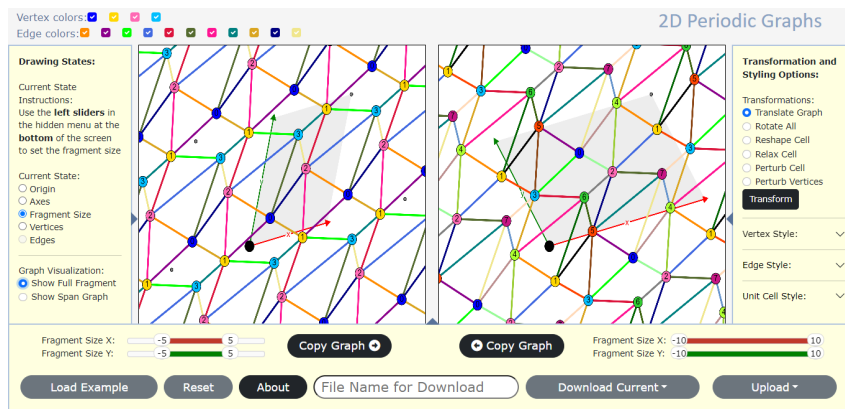


Figure 1 The drawing canvas (left) and transformation canvas (right) display a fragment of the same (infinite) crystal up to isometries and relaxation of periodicity. Translation, rotation and relaxation transformations have been applied on the left graph to obtain the one on the right. Vertices or edges of the same color belong to the same orbit. The shaded parallelogram represents the unit cell, shown together with its origin and generating vectors.

Informally, a *periodic graph* is an infinite graph with a translationally-repeating finite pattern of vertices and edges. In 2D, periodic graphs can be generated from two independent



© Alexandra Camero and Ileana Streinu;

licensed under Creative Commons License CC-BY 4.0

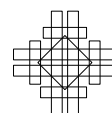
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 63; pp. 63:1–63:4

Leibniz International Proceedings in Informatics



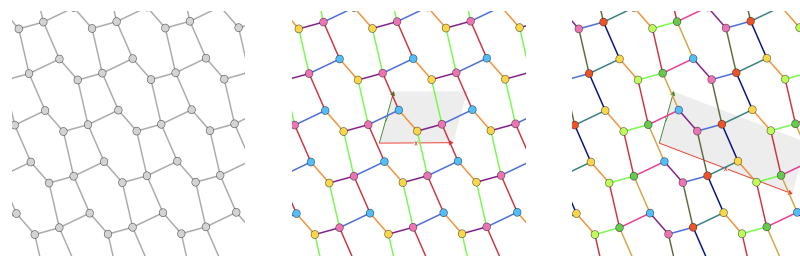
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



translation vectors that induce a *unit cell* and a finite number of vertices and edges, called *representatives*. The infinite periodic graph is partitioned into a finite set of vertex and edge orbits, which receive distinct *colors*. We describe an interactive web app, available at <http://linkage.cs.umass.edu/pergraph/>, which can facilitate the study and visualization of various properties of periodic graphs.

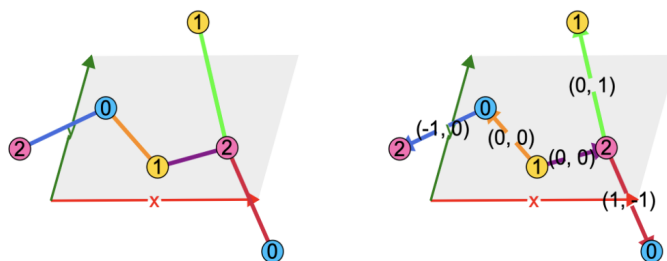
The web application has two canvases, one for drawing and one for applying transformations. There are three hidden menus (left, right, and bottom) that can be opened by hovering over them. Fig. 1 shows all three of them. An accompanying video, accessed through the *Video* option in the bottom menu or directly at <http://linkage.cs.umass.edu/pergraph/about/>, briefly demonstrates the functionality of the app and includes additional information and features not discussed in this abstract.

2 Crystals and their representations



■ **Figure 2** (Left) A 2D crystal. (Middle and right) Two of its possible periodic graphs. The highlighted unit cells represent the periodicity groups, and vertices or edges in the same orbit get identical colors.

Crystals and Periodic Graphs. A *2D crystal* (Fig. 2, left) is an infinite graph which (a) is locally finite (each vertex is incident to finitely many edges) and (b) is subject to the action of *some* periodicity group. Condition (b) actually implies the existence of infinitely many periodicity groups acting on the same crystal: if we fix the group action, we obtain a specific *periodic graph* [1]. Fig. 2 (middle and right) shows two distinct periodic graphs obtained from the same crystal. We specify a periodicity group by its generators (two vectors, the red *x*-axis and the green *y*-axis inducing the gray *unit cell*) and use colors to indicate the group action: similarly colored vertices (resp. edges) belong to the same *orbit*.



■ **Figure 3** An undirected span-graph (left) and the (directed) shift-span-graph (right) of the periodic graph from Fig. 2 (middle). The pair of integers on a directed edge representative indicates the shift of the head-vertex; the tail of the edge representative is always chosen to be in the unit cell.

Quotient and shift-quotient graphs. To each periodic graph we associate a *quotient-graph*, whose vertices V (resp. edges E) correspond to orbits (colors) of vertices (resp. edges) in the original periodic graph. It is, in general, a multi-graph. The information in the quotient graph (V, E) is insufficient for reconstructing the periodic graph [1, 2] and must be supplemented by Z^2 -markings (called *shifts*) on directed versions of the quotient graph edges. This type of *marked directed multi-graph* associated to a periodic graph is called a *shift-quotient-graph*. The *shift* is a pair $(i, j) \in Z^2$ of integers. A *shift-edge* is a pair of an edge $(u, v) \in E$ in the quotient graph and a shift (i, j) . It stands for an edge representative whose tail is the representative of vertex orbit u and whose head is a translation (that goes i -times in the direction of the x -axis and j -times in the direction of the y -axis) of the vertex representative v . The x and y -axes are the generators of the periodicity lattice. The shift-quotient-graph is a complete finite description of a periodic graph and is used in the `.sqf` file format for exporting the graphs produced by our web app.

Span and shift-span graphs. For visualization purposes we use a geometric version of (shift)-quotient-graphs called *(shift)-span-graphs*. A *span-graph* (Fig. 3 (left)) is a subgraph of the geometric periodic graph. It contains all the edge-representatives selected as follows. We first choose vertex representatives to be in the unit cell. An edge representative for an orbit of edges is chosen so that one of its endpoints (the tail) is in the unit cell; the second endpoint (the head) may be either inside or outside of the unit cell. In a span-graph each edge representative appears exactly once (each with its unique edge color). However, several geometric points with the same color (representing vertices from the same vertex orbit) may be present among the edge representative endpoints. The span-graph is thus a *simple* (no multi-edges) colored geometric graph. A *shift-span-graph* (Fig. 3 (right)) orients the edges of the span graph and marks them with shifts such that the *tail* is always inside the unit cell. In short: *the (shift)-quotient graph is obtained by identifying vertices of the same color in the (shift)-span-graph*.

Fractional coordinates. The geometry of the crystal vertices can be expressed using *fractional coordinates* of the vertex representatives relative to the unit cell axes. We use them in the geometric version of the shift-quotient-graphs recorded in the external (file) representation of the graphs produced by our web app.

Crystal fragment. Given the two generators of the unit cell (x - and y -axes), a shift-quotient graph and fractional coordinates for the vertices, we can reconstruct a *fragment* of the infinite periodic graph. We need a position for the *origin* to build the *unit cell* from the two axes. From this information we compute the Cartesian coordinates of the vertex representatives from the fractional ones. Finally, we use two integer-intervals (*windows*) indicating the range of translations of the unit cell in the x and y -directions necessary to build a fragment. The windows can be selected by the user so that the fragment covers the canvas.

Finite descriptions of crystals. Periodicity groups already provide infinitely many ways to describe the same geometric crystal. Furthermore, each periodicity group may be described in infinitely many ways by different choices of generators (unit cells). Two choices of unit cells lead to the same quotient graph as long as they have the same index. If the unit cell is relaxed, then the number of vertices and edges in the quotient graph is scaled by an integer factor. In Fig. 2, the graph on the right is a relaxation of index 2 of the periodic graph in the middle and thus the number of vertices and edges of its quotient-graph are doubled.

Transformations of crystals and periodic graphs. Geometric crystals can be translated or rotated, or be represented by different periodicity groups. Each periodicity group can be represented by different unit cells, resulting in the same number of vertices in the unit cell. All these different representations may result in different fractional coordinates of the vertex representatives. Finally, we may choose different orientations for the edge representatives and obtain different shifts. *Our web app software captures this multitude of different ways of obtaining finite periodic graph representations of the same infinite crystal.*

References

- 1 Ciprian S Borcea and Ileana Streinu. Periodic frameworks and flexibility. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2121):2633–2649, 2010. doi:10.1098/rspa.2009.0676.
- 2 Ciprian S. Borcea and Ileana Streinu. Minimally rigid periodic graphs. *Bulletin of the London Mathematical Society*, 43(6):1093–1103, 2011. doi:10.1112/blms/bdr044.
- 3 Ciprian S. Borcea and Ileana Streinu. Frameworks with crystallographic symmetry. *Philosophical Transactions of the Royal Society of London Series A: Mathematical, Physical and Engineering Sciences*, 372(2008), December 2013. doi:10.1098/rsta.2012.0143.
- 4 Ciprian S. Borcea and Ileana Streinu. Geometric auxetics. *Proceedings of the Royal Society A*, 471(20150033), December 2015. doi:10.1098/rspa.2015.0033.
- 5 Ciprian S. Borcea and Ileana Streinu. Liftings and stresses for planar periodic frameworks. *Discrete and Computational Geometry*, 53(4):747–782, June 2015. doi:10.1007/s00454-015-9689-7.
- 6 Erin Wolf Chambers, Jeff Erickson, Patrick Lin, and Salman Parsa. How to morph graphs on the torus. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2759–2778. SIAM, 2021. doi:10.1137/1.9781611976465.164.
- 7 Jeff Erickson and Patrick Lin. A toroidal Maxwell-Cremona-Delaunay correspondence. *Journal of Computational Geometry*, 12(2), 2022. doi:10.20382/jocg.v12i2a4.

Greedy Permutations and Finite Voronoi Diagrams

Oliver A. Chubet ✉ 


North Carolina State University, Raleigh, NC, USA

Paul Macnichol ✉

North Carolina State University, Raleigh, NC, USA

Parth Parikh ✉

North Carolina State University, Raleigh, NC, USA

Donald R. Sheehy ✉ 

North Carolina State University, Raleigh, NC, USA

Siddharth S. Sheth ✉

North Carolina State University, Raleigh, NC, USA

Abstract

We illustrate the computation of a greedy permutation using finite Voronoi diagrams. We describe the neighbor graph, which is a sparse graph data structure that facilitates efficient point location to insert a new Voronoi cell. This data structure is not dependent on a Euclidean metric space. The greedy permutation is computed in $O(n \log \Delta)$ time for low-dimensional data using this method [4, 6].

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases greedy permutation, Voronoi diagrams

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.64

Category Media Exposition

Supplementary Material *Audiovisual*: <https://youtu.be/zM1pHV6Y1SM>

Funding This research was supported by the NSF under grant CCF-2017980.

1 Introduction

Given a finite metric space X , and a subset P , there is a natural decomposition of X into *Voronoi cells* defined for each $p \in P$ as

$$V(p) := \{x \in X \mid \mathbf{d}(x, p) = \mathbf{d}(x, P)\}$$

This is a discrete analog of the Voronoi diagrams studied in geometry.

The most glaring difference between finite Voronoi diagrams and, say, Euclidean Voronoi diagrams is that there is no natural dual (i.e., no Delaunay triangulation) that connects neighboring cells. To store some local structure, a finite Voronoi diagram is endowed with a *neighbor graph* G . The requirement is that if there are two cells $V(a)$ and $V(b)$ such that inserting a new point $a' \in V(a)$ will cause some points to move from $V(b)$ to the newly formed $V(a')$, then ab is an edge of G . In other words, G stores sufficient information to perform an incremental insertion of a new cell. The movement of points into a new cell is called *point location*. The neighbor graph also stores all the information needed to update itself efficiently after an insertion.

Greedy permutations, also known as farthest point traversals) are a standard way to provide a sequence of good samples of a metric space at different scales. The points are ordered so that each point is the farthest among the remaining points. We describe a simple, deterministic algorithm to compute greedy permutations.



© Oliver A. Chubet, Paul Macnichol, Parth Parikh, Donald R. Sheehy, and Siddharth S. Sheth;

licensed under Creative Commons License CC-BY 4.0

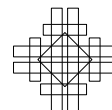
39th International Symposium on Computational Geometry (SoCG 2023).

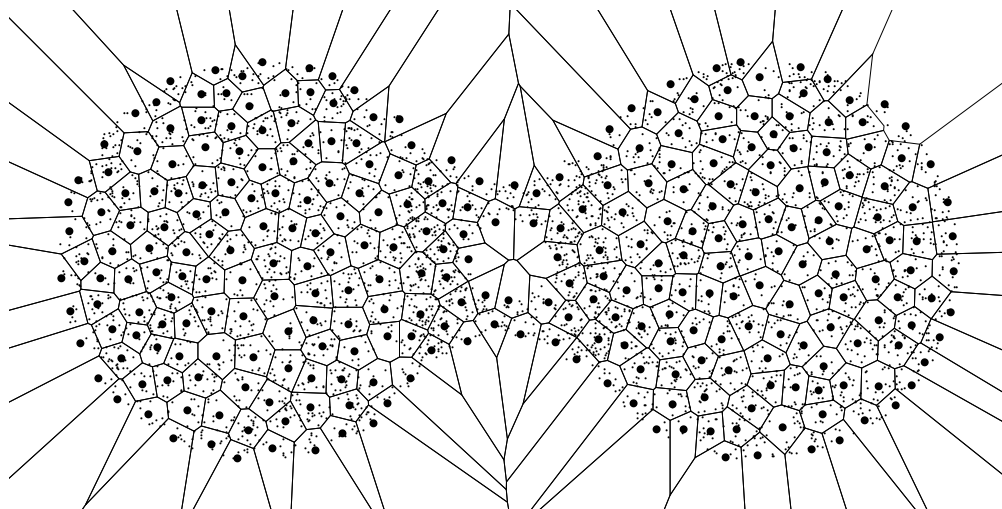
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 64; pp. 64:1–64:5

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** The geometric Voronoi cells cover the points in the cells of the finite Voronoi diagram.

2 Greedy Permutations

The greedy permutation, or the farthest first traversal, is a powerful heuristic used to approximate many hard problems since 1977 from TSP [5] to k -center clustering [3, 2]. Let $P = (p_0, \dots, p_{n-1})$ be a finite sequence of points in a metric space with distance \mathbf{d} . The i^{th} -prefix is the set $P_i = \{p_0, \dots, p_{i-1}\}$ containing the first i points of P . The sequence P is a *greedy permutation* if for all i ,

$$\mathbf{d}(p_i, P_i) = \max_{p \in P} \mathbf{d}(p, P_i).$$

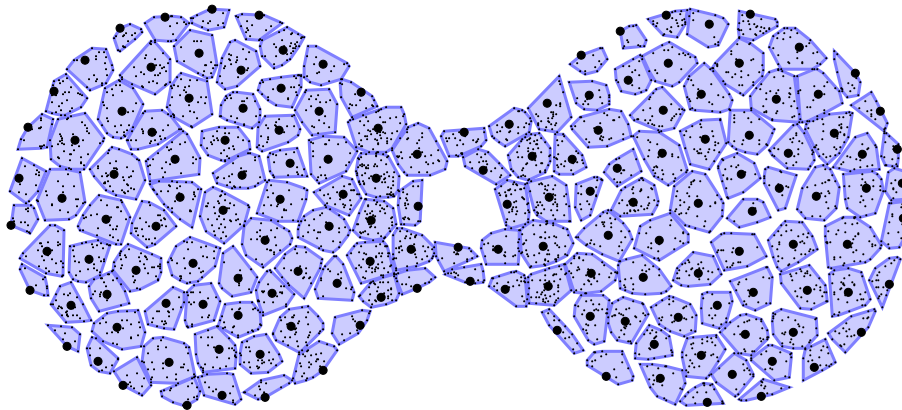
The point $p_0 \in P$ is the *seed* of the greedy permutation and may be chosen arbitrarily. A greedy permutation can be computed in $O(n \log \Delta)$ time for low-dimensional data [4, 6]. An $O(n \log n)$ -time randomized approximation algorithm exists [4], but, to our knowledge, it has never been implemented.

3 Finite Voronoi Diagram

A *finite Voronoi diagram* consists of the cells $V(p)$ and a neighbor graph G . It can be constructed incrementally as follows. It starts with a single cell containing all the points. We insert a point $p' \in V(p)$ by constructing $V(p')$ as a subset of the points $V(q)$ where q is a neighbor of p . We then update the neighbor graph by selecting neighbors of $V(p')$ among the neighbors of neighbors of p . There is a final step in which we prune excess edges that are clearly no longer needed in the neighbor graph. This maintains sparsity of the neighbor graph, resulting in efficient point location.

4 Point Location

An obvious disadvantage of finite Voronoi diagrams as compared to their geometric cousins is that the points of the cell must be explicitly enumerated and moved when there is an insertion. The analysis of Clarkson's algorithm by Har-Peled and Mendel [4] counts the number of times that each point is considered for moving. They show that by a standard



■ **Figure 2** We depict the finite Voronoi cells by drawing the convex hull of the points it contains.

packing argument, only a constant number (i.e., $2^{O(d)}$) of points can touch a given point before the maximum distance of the farthest point from the center of a cell goes down by at least half. It follows immediately that each point is touched at most $2^{O(d)} \log \Delta$ times.

5 Incremental Construction of Finite Voronoi Diagrams

An incremental construction inserts the points one at a time, updating the cells and a neighbor graph. Let (X, \mathbf{d}) be a metric space and let $P \subseteq X$ be a finite subset. The points of X are partitioned into cells $V(p)$, where $p \in P$. The cells of a finite Voronoi diagram satisfy the following invariant.

The Cell Invariant: For all $p \in P$ and all $p' \in V(q)$, $\mathbf{d}(p, p') = \min_{q \in P} \mathbf{d}(p', q)$.

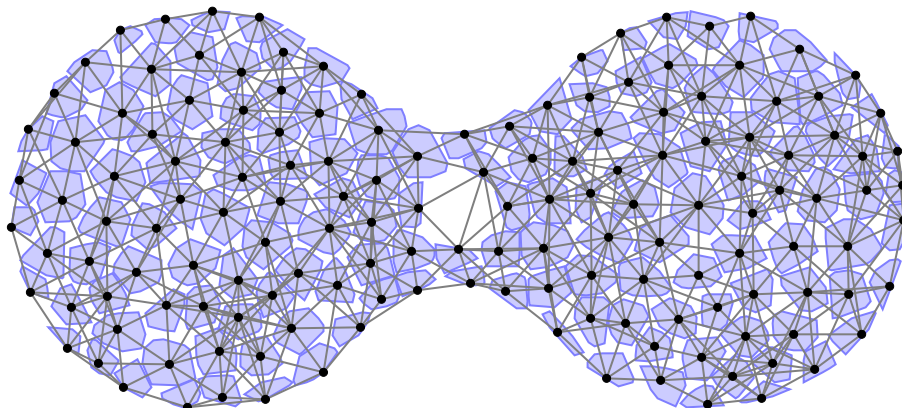
The cell invariant says that if $a' \in V(a)$, then a is a nearest neighbor of a' among the points of P . If we insert a' into the finite Voronoi diagram, then we add a' to P . The point a is called the *parent* of a' . To recover the cell invariant after an insertion, points from other cells are moved into the new cell in a process called *point location*. For a cell $V(q)$, its *out-radius* is $R_q := \max_{a \in V(q)} \mathbf{d}(q, a)$.

In addition to the cells, we also maintain the *neighbor graph* on P that allows point location to be performed locally. That is, there is an edge from a to b if inserting a new point into P from $V(a)$ would require a point from $V(b)$ to be moved (or vice versa). Formally, the neighbor graph maintains the following invariant.

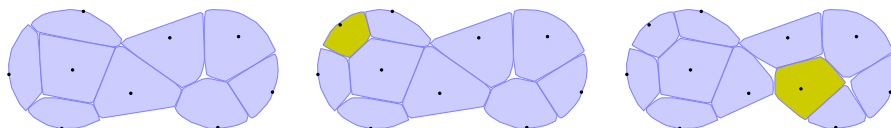
The Neighbor Invariant: For all cells $V(a)$ and $V(b)$, if there exist points $a' \in V(a)$ and $b' \in V(b)$ such that $\mathbf{d}(a', b') < \mathbf{d}(b', b)$, then there is an edge from a to b .

Suppose we are inserting a' with parent a . The neighbor invariant directly implies that every point b' that must move into a new cell centered at a point a' is contained in a cell $V(b)$, where b is a neighbor of a . Thus, point location only requires iterating over the cells of the neighbors of a to find all the points that will move into the newly constructed $V(a')$.

To avoid excess distance computations, the stored neighbor graph contains some edges that are not strictly required by the neighbor invariant. After inserting a' , the neighbors



■ **Figure 3** The neighbor graph during the finite Voronoi diagram construction.



■ **Figure 4** From left to right, we see two iterations of the incremental finite Voronoi diagram construction.

are selected from point a , the neighbors of a , and the neighbors of the neighbors of a . One can simply add edges from a' to each of these points. Any graph that satisfies the neighbor invariant will continue to satisfy the neighbor invariant after this “neighbors of neighbors” update to support insertions.

As Clarkson observed [1], it is possible to use the triangle inequality to identify edges that are too long. Specifically, if the out-radii of two cells $V(a)$ and $V(b)$ are R_a and R_b respectively, then if the neighbor invariant requires an edge from a to b , then $\mathbf{d}(a, b) \leq R_a + R_b + \max\{R_a, R_b\}$. Edges longer than this length can be *pruned* from the neighbor graph.

So, inserting a new point a' with parent a into a finite Voronoi diagram has three steps:




1. Compute $V(a')$ by moving points from cells of neighbors of a .
2. Compute the neighbors of a' by iterating over neighbors of neighbors of a .
3. Prune the edges incident to any point whose cell or neighbors changed.

By the analysis of Har-Peled and Mendel [4], the total point location cost is at most $2^{O(d)}n \log \Delta$. Storing points in a cell as a max-heap keyed by distance from the center gives ready access to the cell’s out-radius. Heaps are updated for all cells which are modified in an iteration. If centers are added in a greedy fashion, it follows from a packing argument that the degree of a vertex in the neighbor graph is $2^{O(d)}$. So, the heap operations per iteration cost $2^{O(d)} \log n$. Updating the neighbor graph also takes $2^{O(d)}$ time per iteration. Therefore, a finite Voronoi diagram can be computed in $2^{O(d)}n \log \Delta$ time.

References

- 1 Kenneth L. Clarkson. Nearest neighbor queries in metric spaces. *Discrete & Computational Geometry*, 22(1):63–93, 1999.
- 2 M.E Dyer and A.M Frieze. A simple heuristic for the p-centre problem. *Operations Research Letters*, 3(6):285–288, 1985. doi:10.1016/0167-6377(85)90002-1.
- 3 Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi:10.1016/0304-3975(85)90224-5.
- 4 Sarel Har-Peled and Manor Mendel. Fast construction of nets in low dimensional metrics, and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.
- 5 Daniel J Rosenkrantz, Richard E Stearns, and Philip M Lewis, II. An analysis of several heuristics for the traveling salesman problem. *SIAM journal on computing*, 6(3):563–581, 1977.
- 6 Donald R. Sheehy. greedypermutations. <https://anonymous.4open.science/r/greedypermutation-C50B>, 2020.

The Sum of Squares in Polycubes

Donald R. Sheehy   

North Carolina State University, Raleigh, NC, USA

Abstract

We give several ways to derive and express classic summation problems in terms of polycubes. We visualize them with 3D printed models. The video is here: http://go.ncsu.edu/sum_of_squares.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Archimedes, polycubes, sum of squares

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.65

Category Media Exposition

Funding This research was supported by the NSF under grant CCF-2017980.

1 Introduction

In 1960, Martin Gardner popularized a generalization of dominoes called polyominoes [3]. These shapes, made from squares glued together along edges, led to many computational problems associated with their enumeration including higher-dimensional variants called *polycubes* [6, 2]. In his book on polyominoes, Solomon Golomb describes many puzzle problems based on the idea of reconfiguring a set polyominoes into different shapes [4]. In this video, we show how polycube reconfiguration problems lead to closed form solutions to classic summation problems.

2 A Closed Form for the Sum of Squares

In his work *On Spirals*[1], Archimedes used tangents of a spiral to solve the hardest computational problems of his day, trisecting an angle and squaring the circle. This treatise also includes a formula for the sum of squares. In modern notation, the identity said

$$3 \sum_{k=1}^n k^2 = n^2(n+1) + \sum_{k=1}^n k \tag{1}$$

Equivalently,

$$\sum_{k=1}^n k^2 = \frac{1}{3}n(n+1)(n+1/2). \tag{2}$$

In Chapter 2 of their book *Concrete Mathematics* [5], Graham, Knuth, and Patashnik give seven different ways to derive this formula for the sum of squares analytically. Below, we show four more ways one could derive this sum from manipulating polycubes.

3 A Proof Without Words

In 1984, in *Mathematics Magazine*, Man-Keung Siu published a proof without words of the sum of squares formula by representing the sum of squares as a pyramid [8]. Each layer is a square. Let $\text{pyr}(n)$ denote the sum of squares.

$$\text{pyr}(n) = \sum_{k=1}^n k^2.$$



© Donald R. Sheehy;

licensed under Creative Commons License CC-BY 4.0

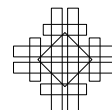
39th International Symposium on Computational Geometry (SoCG 2023).

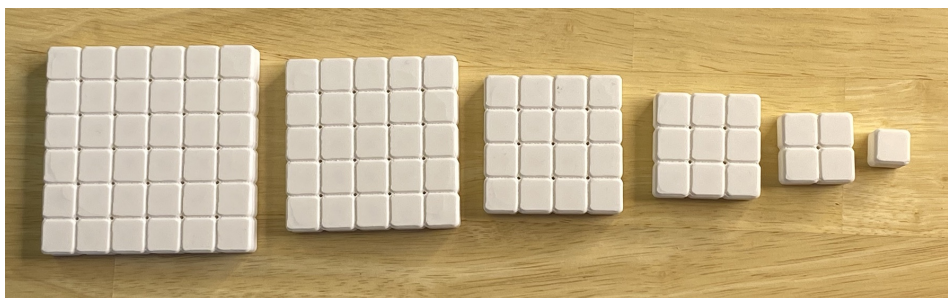
Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 65; pp. 65:1–65:6

Leibniz International Proceedings in Informatics

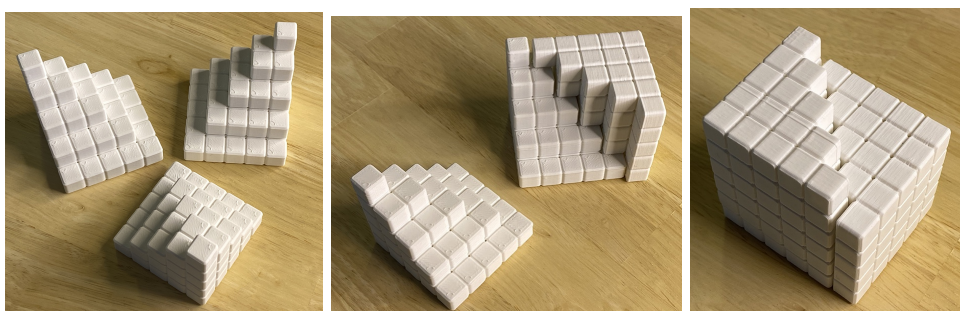


LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany





■ **Figure 1** The sum of squares can be viewed as a collection of polycubes.



■ **Figure 2** The photo shows a 3D printed version of Siu's proof without words.

Three pyramids can be put together to make a cube, and the leftover pieces give the low-order terms.

Specifically, there is one extra $n \times n$ square and a (discrete) triangle. Let $\text{tri}(n)$ denote the size of the triangle, i.e.,

$$\text{tri}(n) = \sum_{k=1}^n k = \binom{n+1}{2}.$$

So, if one believes the picture, the result is

$$3\text{pyr}(n) = \text{cube}(n) + \text{square}(n) + \text{tri}(n) = n^3 + n^2 + \binom{n+1}{2}.$$

More recently, Siu's picture was posted to Math Stack Exchange¹ and by far, the most upvoted comment was one that said they didn't think the picture was convincing on its own, but maybe would be more believable with a physical model. The concern is that it depends on believing that there are no holes in the interior of the cube and that everything indeed, fits perfectly together. This comment was the motivation for 3D printing physical models.

4 Pyramids and Tetrahedra

A different way to derive a sum of squares formula is to observe that a pyramid is the sum of two tetrahedra. A discrete tetrahedron is a sum of triangles:

$$\text{tet}(n) = \sum_{k=1}^n \binom{k+1}{2} = \binom{k+2}{3}.$$

¹ <http://math.stackexchange.com/a/48152/301977>

The closed form can be checked by applying induction and Pascal's identity. It can also be understood as counting the number of ways to choose integers a, b, c, d such that $a + b + c + d = n - 1$. These sums correspond to the barycentric coordinates of the integer points of a tetrahedron embedded in \mathbb{R}^4 with vertices at $n - 1$ times the standard basis vectors. The same intuition explains why $\text{tri}(n) = \binom{n+1}{2}$ and implies that the discrete d -simplex has size

$$s_d(n) := \sum_{k=1}^n s_{d-1}(k) = \binom{n+d-1}{d}.$$

Splitting the pyramid into two tetrahedra as in Figure 4, we get that

$$\text{pyr}(n) = \text{tet}(n) + \text{tet}(n-1).$$

Equivalently,

$$\sum_{k=1}^n k^2 = \binom{n+2}{3} + \binom{n+1}{3},$$

which is another way to express the closed forms in (1) and (2).

Breaking the pyramid into two tetrahedra allows them to be rearranged into another shape with the same volume. This new shape can be stacked with two other pyramids to form a (discrete) triangular prism with base $\text{tri}(n)$ and height $2n + 1$.

Thus, we get another equivalent construction as shown in Figure 5.

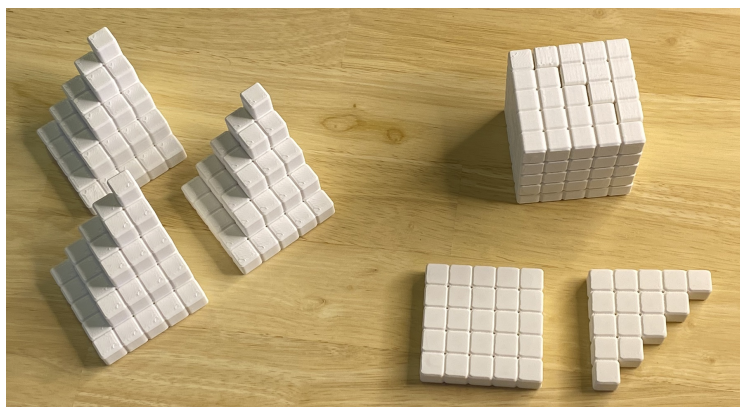
$$3\text{pyr}(n) = \text{tri}(n)(2n+1) = \binom{n+1}{2}(2n+1).$$

5 Four Pyramids

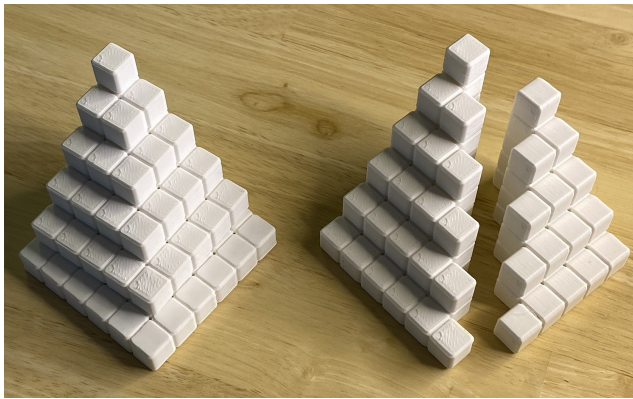
If we put four pyramids together, they make a new pyramid shape that is not as steep. Each layer is a square, but we only get even squares. Separating this pyramid into two tetrahedra results in a sum of odd triangles plus a sum of even triangles. Rearranging the triangles results in a regular tetrahedron that is twice as tall.

That is,

$$4\text{pyr}(n) = \text{tet}(2n),$$



■ **Figure 3** The three pyramids are equal to a cube, a square, and a triangle.



■ **Figure 4** The pyramid can be divided into two tetrahedra.

and thus,

$$\sum_{k=1}^n k^2 = \frac{1}{4} \binom{2n+2}{3}.$$

This is yet another way to express the same closed form.

6 A Sum of Cubes

Writing a sum of squares as a sum of two tetrahedra follows from the fact that each square is a sum of two triangles. To extend this idea to a sum of squares, we might decompose a cube into six (discrete) tetrahedra. In this case, we would use the following construction of a cube.

$$\text{cube}(n) = \text{tet}(n) + 4\text{tet}(n-1) + \text{tet}(n-2).$$

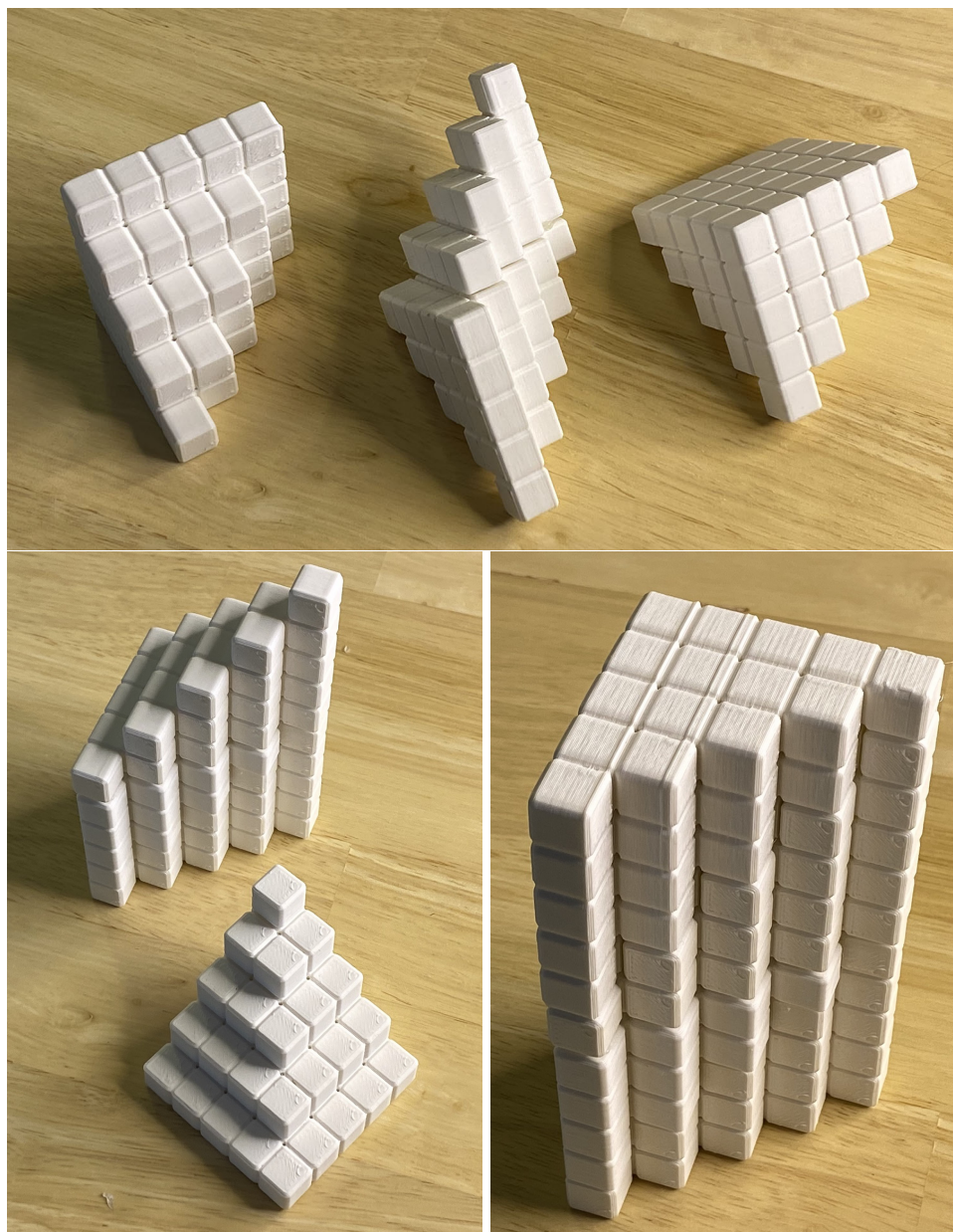
It follows that

$$\begin{aligned} \sum_{k=1}^n k^3 &= \sum_{k=1}^n (\text{tet}(k) + 4\text{tet}(k-1) + \text{tet}(k-2)) \\ &= \sum_{k=1}^n \left(\binom{k+2}{3} + 4\binom{k+1}{3} + \binom{k}{3} \right) \\ &= \binom{n+3}{4} + 4\binom{n+2}{4} + \binom{n+1}{4}. \end{aligned}$$

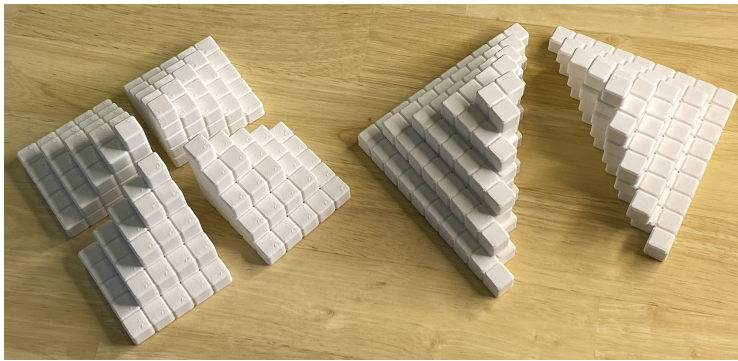
However, this is not the most popular way to express the sum of cubes, because it can be written more simply as

$$\sum_{k=1}^n k^3 = \binom{n+1}{2}^2.$$

In other words, it is the product of two triangles in \mathbb{R}^4 . Perhaps the reader will find a way to visualize this 4-dimensional object as polycubes. An open source library is available to generate the models printed for this project[7]. It is based on the OpenSCAD software, which uses the CGAL library [9] to perform solid geometry constructions.



■ **Figure 5** A triangular prism is constructed from three pyramids.



■ **Figure 6** Four pyramids come together to make a pyramid of even squares.

References

- 1 Archimedes. *On Spirals*, pages 151–188. Cambridge Library Collection - Mathematics. Cambridge University Press, 2009. doi:10.1017/CB09780511695124.014.
- 2 Gill Barequet and Mira Shalah. Automatic Proofs for Formulae Enumerating Proper Polycubes. In Lars Arge and János Pach, editors, *31st International Symposium on Computational Geometry (SoCG 2015)*, volume 34 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19–22, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.SOCG.2015.19.
- 3 Martin Gardner. More about the shapes that can be made with complex dominoes (mathematical games). *Scientific American*, 203(5):186–201, November 1960.
- 4 Solomon W. Golomb. *Polyominoes: Puzzles, Patterns, Problems, and Packings - Revised and Expanded Second Edition*. Princeton University Press, Princeton, 1994. doi:10.1515/9780691215051.
- 5 Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, Reading, 1989.
- 6 W. F. Lunnon. Counting multidimensional polyominoes. *The Computer Journal*, 18(4):366–367, January 1975. doi:10.1093/comjnl/18.4.366.
- 7 Donald R. Sheehy. polycube. <https://github.com/donsheehy/polycube>, 2023.
- 8 Man-Keung Siu. Proof without words: Sum of squares. *Mathematics Magazine*, 57(2):92–92, 1984. doi:10.1080/0025570X.1984.11977083.
- 9 The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023. URL: <https://doc.cgal.org/5.5.2/Manual/packages.html>.

Constructing Concise Convex Covers via Clique Covers

Mikkel Abrahamsen  

University of Copenhagen, Denmark

William Bille Meyling 

University of Copenhagen, Denmark

André Nusser  

University of Copenhagen, Denmark

Abstract

This work describes the winning implementation of the CG:SHOP 2023 Challenge. The topic of the Challenge was the convex cover problem: given a polygon P (with holes), find a minimum-cardinality set of convex polygons whose union equals P . We use a three-step approach: (1) Create a suitable partition of P . (2) Compute a visibility graph of the pieces of the partition. (3) Solve a vertex clique cover problem on the visibility graph, from which we then derive the convex cover. This way we capture the geometric difficulty in the first step and the combinatorial difficulty in the third step.

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Combinatorial algorithms

Keywords and phrases Convex cover, Polygons with holes, Algorithm engineering, Vertex clique cover

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.66

Category CG Challenge

Supplementary Material

Software: <https://github.com/willthbill/ExtensionCC>

archived at `swh:1:rev:ad78739911ab5733f600d4fbc08acae6d69e115`

Text (Thesis): <https://github.com/willthbill/ExtensionCC/blob/main/bachelorthesis.pdf>

Funding *Mikkel Abrahamsen:* Supported by Starting Grant 1054-00032B from the Independent Research Fund Denmark under the Sapere Aude research career programme. Part of BARC, supported by the VILLUM Foundation grant 16582.

William Bille Meyling: Supported by Starting Grant 1054-00032B from the Independent Research Fund Denmark under the Sapere Aude research career programme.

André Nusser: Part of BARC, supported by the VILLUM Foundation grant 16582.

Acknowledgements We want to thank the CG:SHOP 2023 organizers and the other participants (especially Guilherme Dias da Fonseca) for creating such a fun challenge and for helpful comments on our write-up. We also want to thank Martin Aumüller and Rasmus Pagh for access and help with using their server. Finally, we want to thank Darren Strash for quick and last-minute support using the ReduVCC implementation.

1 Introduction

Covering a polygon with the minimum number of convex pieces is a fundamental problem in computational geometry and the problem chosen for the CG:SHOP 2023 Challenge. In this problem we are given a polygon P (potentially with holes) and we have to find a smallest possible set of convex polygons whose union equals P . This problem is NP-hard [3] and was later shown to be even $\exists\mathbb{R}$ -complete [1]. Note that in the Challenge, all coordinates of the solutions had to be rational, and then the decision problem is not even known to be decidable. Thus, it is not expected that there exists any fast algorithm that always finds



© Mikkel Abrahamsen, William Bille Meyling, and André Nusser;
licensed under Creative Commons License CC-BY 4.0

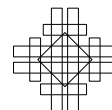
39th International Symposium on Computational Geometry (SoCG 2023).

Editors: Erin W. Chambers and Joachim Gudmundsson; Article No. 66; pp. 66:1–66:9

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



the optimal solution. Likewise, we are aware of no previously described algorithm which is fast in practice. In this 5th CG:SHOP Challenge, there were a total of 22 teams who signed up, out of which 18 submitted solutions. Our team – named *DIKU (AMW)* – obtained the highest total score, despite finding fewer smallest solutions than the runner-up [4], as we achieved significantly smaller solutions on many of the largest Challenge instances. See [5] for a survey about the Challenge.

Our algorithm consists of three steps. In the first step (see Section 2.1), the aim is to capture the geometry of the problem. We do this by partitioning the input polygon P into triangles. Note that the corners of these triangles do not have to be corners of P but can be Steiner points. In the second step (see Section 2.2), we move from the geometric structure to a combinatorial structure. We do this by computing a visibility graph G of the partition, with each triangle corresponding to a vertex and an edge is inserted for two vertices only if the convex hull of the corresponding triangles lies within P (i.e., the convex hull would be a valid piece of the convex cover). Finally, in the third step (see Section 2.3), we solve a combinatorial problem: we find a vertex clique cover (VCC) of G with small cardinality. Recall that a *vertex clique cover* is a set of cliques in G , for which each vertex in G appears in one of the cliques. When possible, we use the convex hull H of the triangles of a clique C as a piece of our convex cover. However, H may intersect holes of P , which makes H an invalid piece. This happens rarely for the Challenge instances, but in that case we split C into smaller cliques. For us, the main insights and highlights of our approach are:

1. Assembling the pieces and the cover at the same time (instead of first deciding on the pieces and then assembling the cover) allows for great flexibility and adaptivity.
2. Reduction to a fundamental graph problem allows for usage of a powerful set of already existing tools.
3. As we can arbitrarily choose a partition in the first step of the algorithm, our approach is very adaptive with respect to input structure and instance size (simpler partitions can be chosen for larger instances).

2 Algorithm

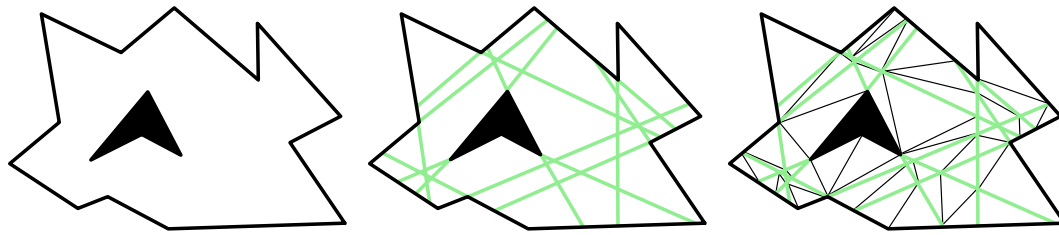
In this section we describe our algorithmic approach to solve the convex cover problem.

2.1 Partition

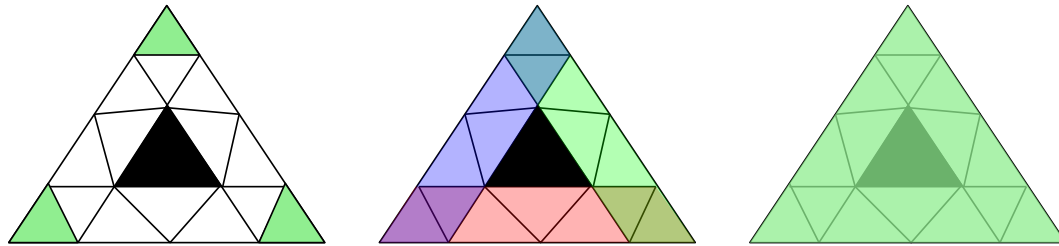
First, we partition the polygon. While our approach in principle works with any kind of partition, for simplicity we only used partitions consisting of triangles. Recall that the goal of the partition is to obtain triangles from which we can later assemble good pieces for a convex cover and that the corners of these triangles are not restricted to lie on the corners of P . In fact, to obtain good solutions one often needs Steiner points.

The simplest partition that we use is a Delaunay triangulation. We prefer a Delaunay triangulation over an arbitrary triangulation because it leads to fat triangles, which we intuitively assume to create better pieces for the convex cover. The main issues of using a Delaunay triangulation of P as partition are that its vertices are restricted to the corners of P and that the pieces can be too coarse to merge into convex pieces. See Figure 6 for an example for which this leads to a suboptimal cover. Thus, the question is: which Steiner points should we introduce to obtain better solutions?

Consider a directed edge e of P and suppose that the interior of P is to the left of e . We define the *extension* of e to be the maximal directed segment s such that $e \subseteq s \subseteq P$; see Figure 1 (middle). Note that a piece Q of a convex cover can contain an interior point of



■ **Figure 1** Left: Polygon P (left), extensions of P (middle), and extension partition of P (right).



■ **Figure 2** For the set of green triangles (left), every pairwise convex hull is contained in P (middle), but the convex hull of all the green triangles is not (right).

e only if Q does not contain a point to the right of s . Thus, intuitively it make sense to include pieces of the cover that are bounded by s . This intuition is captured by the *extension partition*, which is the constrained Delaunay triangulation of the extensions of all edges of P ; see Figure 1 (right).

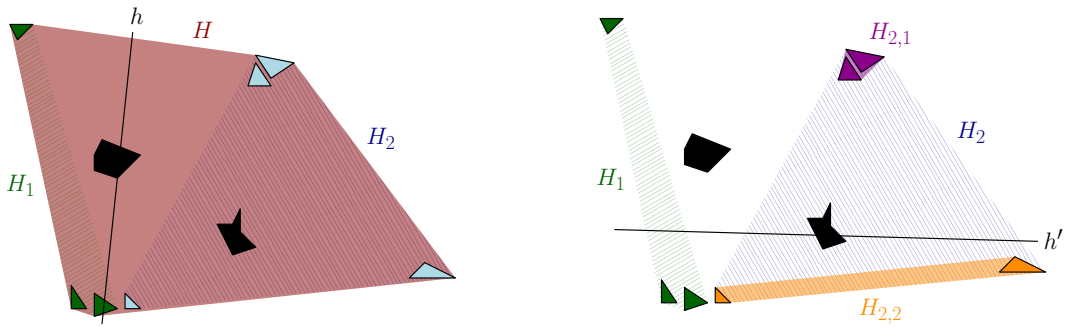
2.2 Visibility Graph

In order to create a convex cover, we first want to understand which triangles we can potentially combine to form pieces for the cover. Given a partition \mathcal{P} of the polygon P and two triangles $p, q \in \mathcal{P}$, we say that q is *fully visible* from p if every point in p sees all of q and *partially visible* if every point in p sees some point in q . We define the visibility graph $G = (\mathcal{P}, E)$, which contains an edge pq if the convex hull of $p \cup q$ is contained in P . We can compute G naively by checking for each pair $p, q \in \mathcal{P}$ whether its convex hull is contained in P . However, the running time $\Omega(|\mathcal{P}|^2)$ renders this impractical. A simple observation comes in handy here: For any triangles $q \in \mathcal{P}$ fully visible from $p \in \mathcal{P}$, there exists a path from p to q in the dual graph¹ of \mathcal{P} using only vertices that correspond to triangles that are partially visible from p . Thus, instead of checking all pairwise visibilities, we can simply perform a BFS on the dual graph, only using partially visible triangles and stop exploring on triangles that are not partially visible. While this significantly reduces the running time in practice, it can still be too expensive. For further speedup, we resort to building a subgraph of G by only exploring fully visible triangles in the BFS. To speed up the visibility graph construction, we engineered fast visibility checks that we do not further describe here.

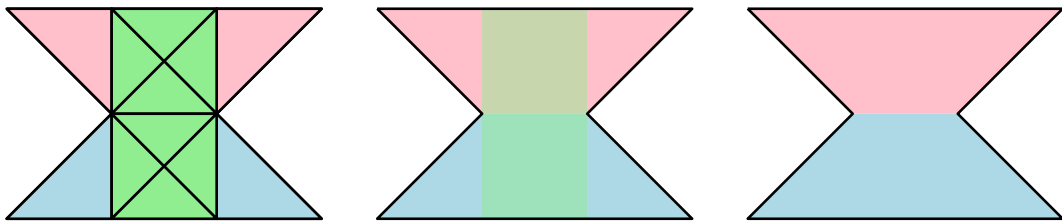
2.3 Compute Cover

We employ the following three steps to compute a convex cover using the visibility graph.

¹ The dual graph of a partition is defined as follows: the vertex set consists of the triangles of the partition and there is an edge between two vertices iff the two corresponding triangles touch.



■ **Figure 3** Fixing an invalid clique: All visible triangles form the initial piece H that we then split into pieces H_1 and H_2 using the half-plane h (left). As H_2 still contains a hole, we again split it into pieces $H_{2,1}$ and $H_{2,2}$ using half-plane h' (right). The result is the valid pieces H_1 , $H_{2,1}$, and $H_{2,2}$.



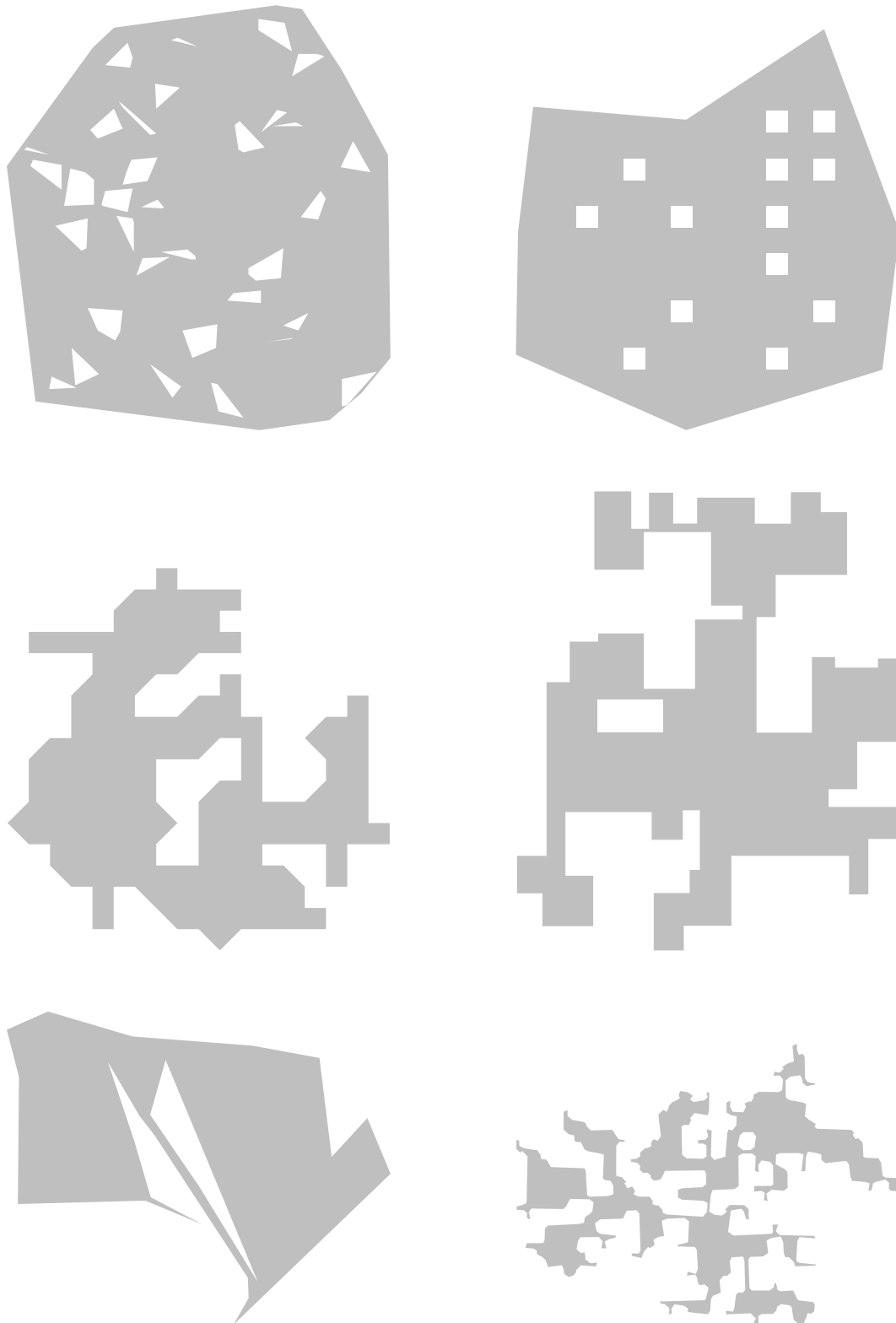
■ **Figure 4** Extension partition with suboptimal clique cover (left), the corresponding convex cover (middle), and a convex cover without the unnecessary green polygon (right).

Compute Vertex Clique Cover: We first compute a vertex clique cover (VCC) on the visibility graph. The problem of finding a minimum VCC is one of Karp’s classical NP-hard problems, and there exists no $n^{1-\epsilon}$ -approximation algorithm for any $\epsilon > 0$ unless $P = NP$. However, there exist implementations that compute small solutions on practical instances. Namely, Chalupa [2] presented a randomized clique-growing approach that was subsequently used as a subroutine by Strash and Thompson [8] in their state-of-the-art solver **ReduVCC** that uses sophisticated reduction rules with a branch-and-reduce approach.

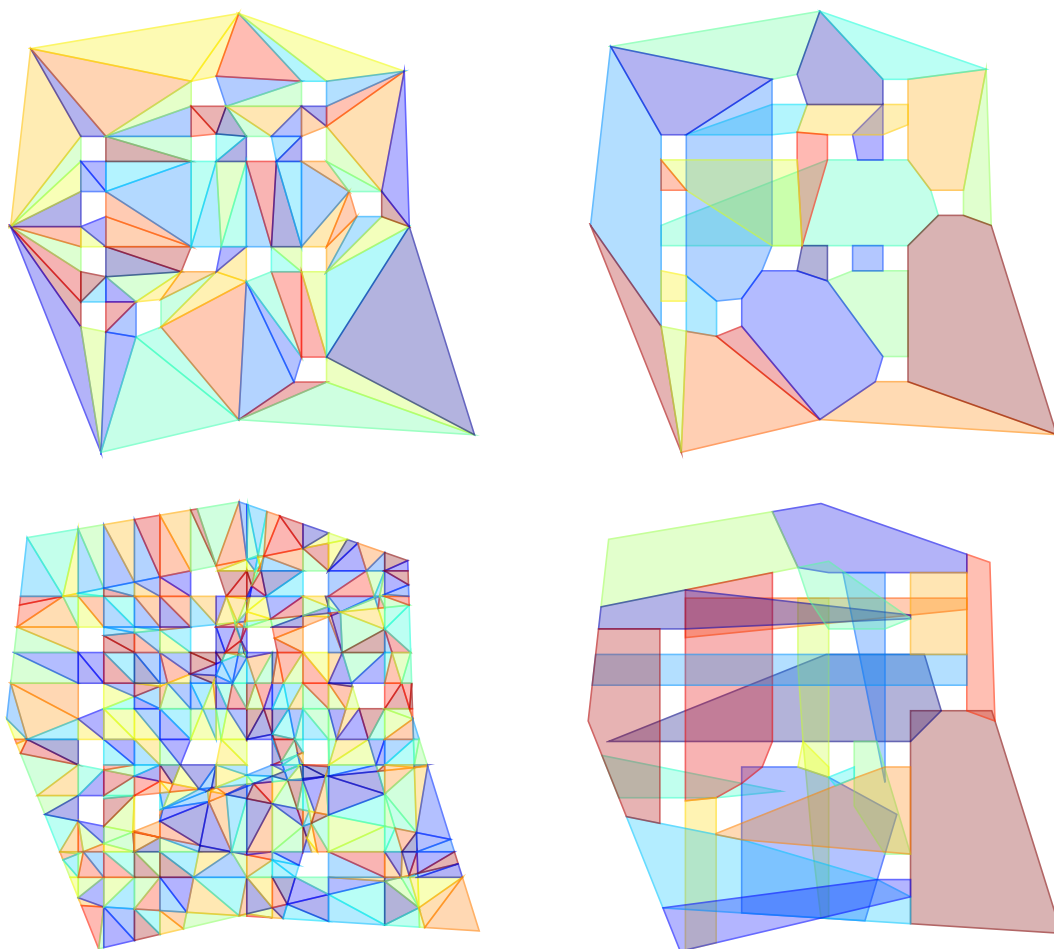
Fix Cover: Recall that a clique C corresponds to a set of triangles that are pairwise fully visible. We would like to use the convex hull H of the triangles as a piece in our convex cover, but H may not be contained in P ; see Figure 2 for a simple example. While this rarely happens on the Challenge instances (see Section 3.3), we nonetheless have to post-process such pieces to obtain a feasible convex cover.

First, note that the only way a piece H can be invalid is by containing a hole of P ; in particular, it is not possible that H intersects the unbounded region of the complement of P . We fix an invalid piece H as follows: Pick any hole h that invalidates H and consider an arbitrary half-plane whose boundary intersects h . Now partition the triangles of C according to whether they intersect the half-plane or not. This creates two new pieces, which both do not intersect the hole h and which partition the remaining holes in H . We apply this procedure recursively to the new pieces (always reducing the number of holes intersected by these pieces by at least one) until all newly created pieces are valid; see Figure 3. We omit the proof of correctness of this procedure due to space constraints.

Make Cover Minimal: At this point, we may end up with a non-minimal cover, i.e., there may exist redundant pieces; see Figure 4. To make the solution minimal, we iterate over the pieces and remove them from the cover if their removal does not invalidate it.



■ **Figure 5** The smallest instances of some of the Challenge instance types. From left to right and top to bottom, these are: `cheese`, `maze`, `octa`, `iso`, `fpg`, `srpg_mc`.



■ **Figure 6** The Delaunay triangulation (top left) and the resulting cover of size 27 (top right). The extension partition (bottom left) and the resulting cover of size 23 (bottom right).

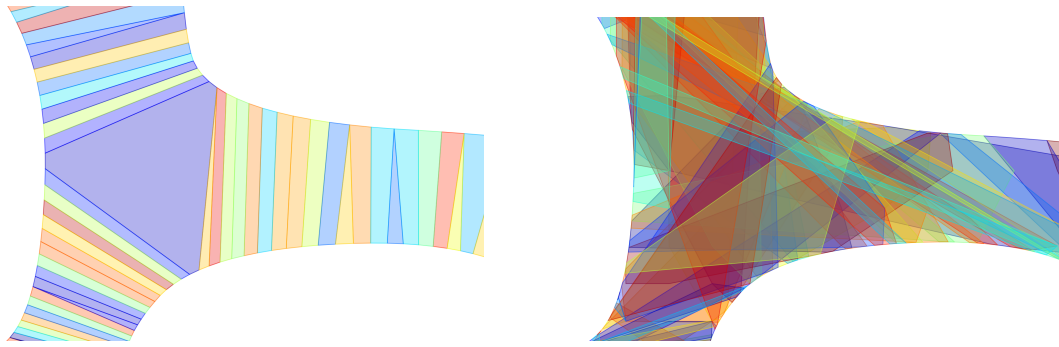
3 Evaluation

3.1 Implementation and Data

The competition code is written in C++ and compiled using GCC 11.3 with `-O3` optimization turned on. We use CGAL [9] for all geometric primitives with a Kernel that uses a number type that saves numbers as fractions and performs exact computations. For the partitioning and to construct the visibility graph, we use the triangulation, visibility, and convex hull packages of CGAL [6, 7, 10]. To compute the vertex cover, we use ReduVCC [8]. We show different types of instances of the problem set in Figure 5.

3.2 Examples

An important part of our approach is the choice of the partition. In particular, while the Delaunay triangulation is fast to compute, the extension partition creates partitions with significantly more pieces and thus slows down our approach. To justify this kind of partition, it must lead to significantly better solutions. Figure 6 shows a cover of the same instance using the Delaunay partition and the extension partition – one can clearly see that the extension partition better adapts to the geometry of the input polygon. Unfortunately, extension



■ **Figure 7** Part of a polygon with multiple long concave chains. While the cover using a Delaunay triangulation almost exclusively creates pieces adjacent to only two concave chains (left), local triangulation allows for creation of pieces that contain parts of all three concave chains (right).

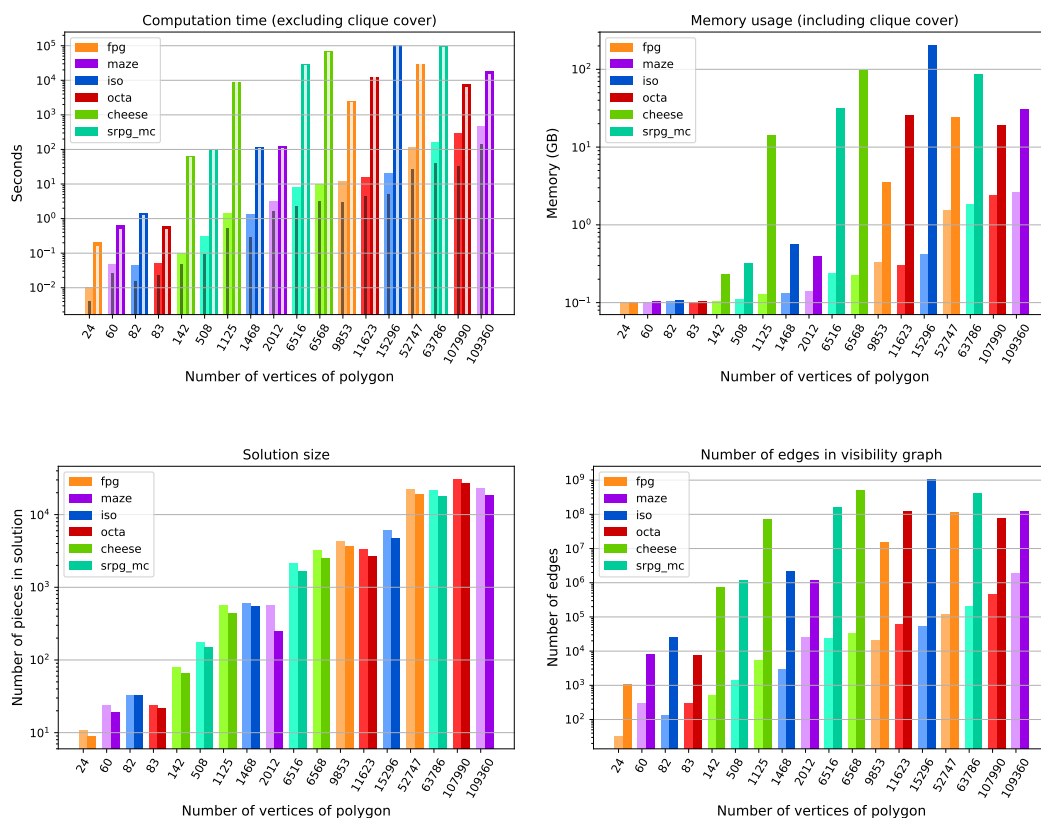
partitions can lead to a blow-up of the partition size that makes the approach practically infeasible for some instances. This blow-up happens when many extensions intersect. We circumvent this issue by computing restricted extension partitions in two different ways.

1. We want to preserve extensions locally. Thus, we only choose a subset of the extensions favoring short extensions. We either only insert extensions below a certain length, or we randomly sample extensions inversely proportional to their length.
2. Some Challenge instances have long concave chains on the boundary of P . Note that the midpoint of each edge of such a chain has to be part of a distinct piece in the convex cover. To allow for creation of pieces that combine segments from multiple concave chains, we locally triangulate long concave chains instead of creating extensions; see Figure 7.

3.3 Experiments

For this section, we selected a subset of the instances for more thorough experiments and subsequently only refer to these. See the sizes and types in the plots of Figure 8. Recall that we compute an intermediate, potentially infeasible solution via a vertex clique cover that is subsequently fixed. We argue above that we expect that only few cliques have to be fixed on practical instances. Indeed, on all except the `cheese` instances, the solution size increased by at most 6 pieces when fixing cliques, while most small instances did not have any invalid clique. However, the largest increase in solution size was for the largest `cheese` instances with an increase of 110 pieces. Our algorithm may create redundant cliques that are removed in a post-processing step, so it is interesting to consider how much this post-processing reduces the size of the solution. This decrease in pieces is very much dependent on the instance: While for `octa` the maximal decrease was 2 pieces, it was 83 pieces for `cheese` instances and all larger `cheese` instances saw significant improvements.

For the competition and our experiments we used a server with two Intel Xeon E5-2690 v4 CPUs with 14 cores (28 threads) each, and a total of 504GB RAM. All reported running times are single-threaded. The bottleneck of our approach is the visibility graph computation discussed in Section 2.2. To better understand the trade-off between running time, memory usage, and solution quality with respect to the choice of partition, we conduct experiments comparing Delaunay triangulation and extension partition; see Figure 8. The extension partition introduces a large overhead in running time and memory consumption compared to the Delaunay triangulation, but it reduces the solution size by a significant fraction. While for the extension partition the visibility graph computation clearly dominates the running time, for the Delaunay triangulation it only makes up 32.8% of the running time on average.





■ **Figure 8** Experiments with the Delaunay triangulation (left bars) and the extension partition (right bars) as underlying partitions for a selected set of instances. We measure the running time (top left; thin bars showing the running time of the visibility graph computation), memory usage (top right), solution size (bottom left), and number of edges in the visibility graph (bottom right).

References

- 1 Mikkel Abrahamsen. Covering polygons is even harder. In *Symposium on Foundations of Computer Science (FOCS)*, pages 375–386, 2021. doi:10.1109/FOCS52979.2021.00045.
- 2 David Chalupa. Construction of near-optimal vertex clique covering for real-world networks. *Comput. Informatics*, 34(6):1397–1417, 2015. URL: <http://www.cai.sk/ojs/index.php/cai/article/view/1276>.
- 3 Joseph C. Culbertson and Robert A. Reckhow. Covering polygons is hard. *J. Algorithms*, 17(1):2–44, 1994. doi:10.1006/jagm.1994.1025.
- 4 Guilherme D. da Fonseca. Shadoks approach to convex covering. In *Symposium on Computational Geometry (SoCG)*, volume 258, 2023. URL: <https://pageperso.lis-lab.fr/guilherme.fonseca/CGSHOP23conf.pdf>.
- 5 Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Stefan Schirra. Minimum coverage by convex polygons: The CG:SHOP Challenge 2023, 2023. arXiv:2303.07007.
- 6 Michael Hemmer, Kan Huang, Francisc Bungiu, and Ning Xu. 2D visibility computation. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023. URL: <https://doc.cgal.org/5.5.2/Manual/packages.html#PkgVisibility2>.
- 7 Susan Hert and Stefan Schirra. 2D convex hulls and extreme points. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023. URL: <https://doc.cgal.org/5.5.2/Manual/packages.html#PkgConvexHull2>.

- 8 Darren Strash and Louise Thompson. Effective data reduction for the vertex clique cover problem. In *Symposium on Algorithm Engineering and Experiments (ALENEX)*, pages 41–53, 2022. doi:10.1137/1.9781611977042.4.
- 9 The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023. URL: <https://doc.cgal.org/5.5.2/Manual/packages.html>.
- 10 Mariette Yvinec. 2D triangulations. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition, 2023. URL: <https://doc.cgal.org/5.5.2/Manual/packages.html#PkgTriangulation2>.

Shadoks Approach to Convex Covering

Guilherme D. da Fonseca  

LIS, Aix-Marseille Université, France

Abstract

We describe the heuristics used by the Shadoks team in the CG:SHOP 2023 Challenge. The Challenge consists of 206 instances, each being a polygon with holes. The goal is to cover each instance polygon with a small number of convex polygons. Our general strategy is the following. We find a big collection of large (often maximal) convex polygons inside the instance polygon and then solve several set cover problems to find a small subset of the collection that covers the whole polygon.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Set cover, covering, polygons, convexity, heuristics, enumeration, simulated annealing, integer programming, computational geometry

Digital Object Identifier 10.4230/LIPIcs.SoCG.2023.67

Category CG Challenge

Related Version *Full Version*: <https://arxiv.org/abs/2303.07696>

Supplementary Material *Software (Source Code)*: github.com/gfonsecabr/shadoks-CGSHOP2023
archived at `swh:1:dir:1ebe4fb460d0533472d052a52e3d4476ce15393e`

Funding Work supported by the French ANR PRC grant ADDS (ANR-19-CE48-0005).

Acknowledgements We would like to thank the Challenge organizers and other competitors for their time, feedback, and making this whole event possible. We would also like to thank the second year undergraduate students Rayis Berkat, Ian Bertin, Ulysse Holzinger, and Julien Jamme for coding a solution visualisation tool that allows you to view all our best solutions in <https://pageperso.lis-lab.fr/guilherme.fonseca/cgshop23view/>.

1 Introduction

CG:SHOP Challenge is an annual geometric optimization challenge. The fifth edition in 2023 considers the problem of covering a polygon with holes P using a small set of convex polygons \mathcal{S} that lie inside P . In total, 206 polygons have been given as instances, ranging from 24 to 109,360 vertices. The instances are of several different types, including orthogonal

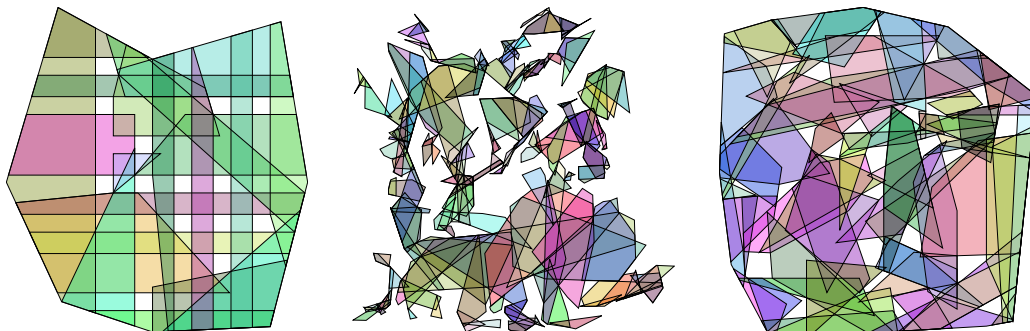
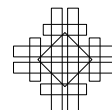


Figure 1 Solutions to `maze_79_50_5_5`, `fpg-poly_400_h2`, and `cheese163` instances. The instances have 64, 462, and 163 vertices and the solutions have 23, 172, and 68 polygons, respectively.



polygons and polygons with many small holes. The team Shadoks won second place with the best solution (among the 22 participating teams) to 128 instances. More details about the Challenge and this year’s problem are available in the organizers’ survey paper [5].

Our general strategy consists of two distinct phases. First, we produce a large *collection* \mathcal{C} of large convex polygons inside P . Second, we find a small subset $\mathcal{S} \subseteq \mathcal{C}$ that covers P , which is returned as the solution. Figure 1 shows three small solutions and we can observe that most convex polygons are maximal and often much larger than necessary. Our approach is different from that of the winning team DIKU (AMW), that uses clique cover [1].

To construct the collection \mathcal{C} in phase 1, we used either a modified version of the Bron-Kerbosch algorithm or a randomized bloating procedure starting from a constrained Delaunay triangulation (Section 2). To solve the set cover problem in phase 2, we used integer programming and simulated annealing. The key element for the efficiency of phase 2 is to iteratively generate constraints as detailed in Section 3. Generally speaking, the initial constraints ensure that all input vertices are covered and supplementary constraints ensure that a point in each uncovered area is covered in the following iteration. In fact, to obtain our best solutions, we repeat phase 2 using the union of the solutions from independent runs of the first two phases as the collection \mathcal{C} . Our results are discussed in Section 4.

2 Collections

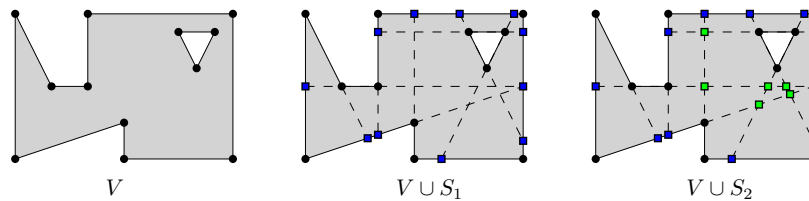
We now describe phase 1 of our strategy: building a collection. Throughout, the instance is a polygon with holes P with vertex set V . Formally speaking, a *collection* \mathcal{C} is defined exactly as a *solution* \mathcal{S} : a finite set of convex polygons whose union is P . However, while we want a solution \mathcal{S} to have as few elements as possible, the most important aspect of a collection \mathcal{C} is that it contains a solution $\mathcal{S} \subseteq \mathcal{C}$ with few elements. Ideally, $|\mathcal{C}|$ is also not too big so the second phase solver is not overloaded, but the size of \mathcal{C} is of secondary importance.

Given a set of points S , a convex polygon $C \subseteq P$ is *S-maximal* if the vertices of C are in S and there exists no point $s \in S$ with $\text{conv}(C \cup \{s\}) \subseteq P$. Next, we show how to build a collection with all *S-maximal* convex polygons.

Bron-Kerbosch. The Bron-Kerbosch algorithm [2] is a classic algorithm to enumerate all maximal cliques in a graph (in our case the visibility graph) with good practical performance [6]. The algorithm recursively keeps the following three sets. **R**: Vertices in the current maximal clique. **S**: Vertices that may be added to the current maximal clique. **X**: Vertices that may not be added to the current maximal clique because otherwise the same clique would be reported multiple times. Initially, $R = X = \emptyset$ and $S = S$.

If the polygon P has no holes and S , then there is a bijection between the maximal cliques in the visibility graph of S on P and the *S-maximal* convex polygons. While this is no longer true in the version with holes, we can adapt the Bron-Kerbosch algorithm to enumerate all *S-maximal* convex polygons as shown in Listing 1. Figure 3 shows how the number of *V-maximal* convex polygons grows for different instances and that we can compute all *V-maximal* convex polygons quickly for instances with around 10 thousand vertices.

Let S_1 be the set of the endpoints of the largest segments inside P that contain each edge of P (Figure 2). It is easy to see that $|S_1| \leq 2|V|$. However, as shown in Figure 4, we are only able to compute all $(V \cup S_1)$ -maximal convex polygons for instances with less than one thousand vertices. It is possible that a modified version of the Bron-Kerbosch algorithm gives better results, either by using a pivot or choosing a particular order for the points, but we have not succeeded in obtaining significant improvements. Another natural set of points



■ **Figure 2** Definitions of V , S_1 , and S_2 .

is the set $S_2 \supseteq S_1$ defined as the intersection points (inside P) of the lines containing the edges of P (Figure 2). The set S_2 may however have size roughly $|V|^2$. Hence, computing all $(V \cup S_2)$ -maximal convex polygons is only feasible for very small instances.

Random Bloating. As a V -maximal convex polygon C is generally not P -maximal, we also grow C with an operation we call bloating. Given a convex polygon C and a set of points S , we construct an S -bloating convex polygon C' by iteratively trying to add a random point from S to C and taking the convex hull, verifying at each step that C' lies inside the instance polygon P . There are two sets of points that may compose S . First, $S_1(C)$ is the set of endpoints of the largest segment in P that contains each edge of C . Second, $S_2(C)$ is the union of $S_1(C)$ and the intersection points of the lines containing the edges of C , if the points are inside P . Notice that $|S_1(C)| = O(|C|)$, but $|S_2(C)| = O(|C|^2)$.

To start the bloating operation, we need a convex polygon C . One approach is to use the V -maximal convex polygons produced by Bron-Kerbosch. A much faster approach for large instances is to use a constrained Delaunay triangulation of the instance polygon. In this case, we start by V -bloating the triangles into a convex polygon C , and then possibly $S_1(C)$ -bloating or $S_2(C)$ -bloating the polygon C . Since the procedure is randomized, we can replicate the triangles multiple times to obtain larger collections of large convex polygons.

3 Set Cover

Given a collection \mathcal{C} of convex polygons that covers P , our covering problem consists of finding a small subset of \mathcal{C} that still covers P . In contrast to the classic set cover problem, in our case P is an infinite set of points. Nevertheless, it is easy to create a finite set of witnesses W , that satisfy that W is covered by a subset \mathcal{S} of \mathcal{C} if and only if P is. To do that, we place a point inside each region (excluding holes) of the arrangement of line segments defining the boundaries of the polygons in \mathcal{C} and P (Figure 5(a)). The size of such set W is however very large in practice and potentially quadratic in the number of segments of \mathcal{C} .

Producing small sets of witnesses has been studied in the context of art gallery problems [3]. However, we do not know if small sets of witnesses exist for our problem. Hence, we use a loose definition of witness as any finite set of points $W \subset P$. In practice, we want W to be such that if W is covered, then most of P is covered. Next, we show how to build such set of witnesses and afterwards we describe how we solved the finite set cover problem.

Witnesses. A set of witnesses W that gave very good results, which we call *vertex witnesses*, consists of one witness inside each cell of the arrangement that contains a vertex of the instance polygon P , as shown in Figure 5(b). This set guarantees that if W is covered, then all points that are arbitrarily close to the *vertices* of P are covered. However, trivially computing W requires building the arrangement of the collection \mathcal{C} , which is too slow and memory consuming for large \mathcal{C} .

■ **Listing 1** Modified Bron-Kerbosch algorithm.

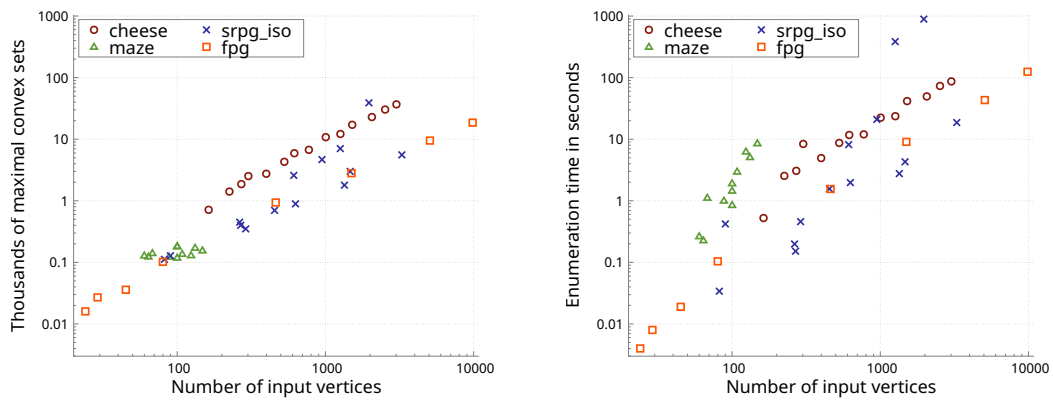
```

BronKerbosch(R, S, X):
  if S and X are both empty: report conv(R) and return
  if  $X \cap \text{conv}(R) \neq \{\}$ : return // Only for improved performance

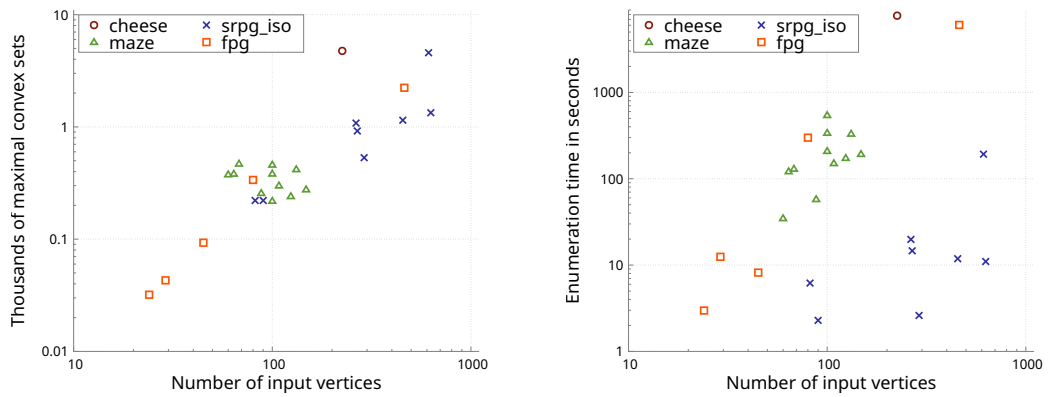
  for each vertex v in S:
    S' ← X' ← {}
    for each vertex u in S:
      if  $v \neq u$  and  $\text{conv}(R \cup \{u,v\}) \subseteq P$ : S' ← S' ∪ {u}

    for each vertex u in X:
      if  $v \neq u$  and  $\text{conv}(R \cup \{u,v\}) \subseteq P$ : X' ← X' ∪ {u}

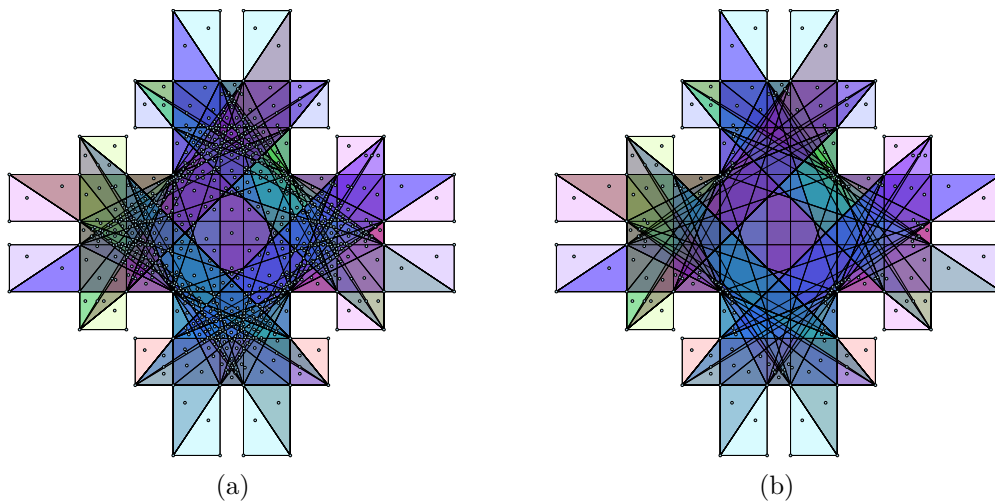
  BronKerbosch(R ∪ {v}, S', X')
  S ← S \ {v}, X ← X ∪ {v}
    
```



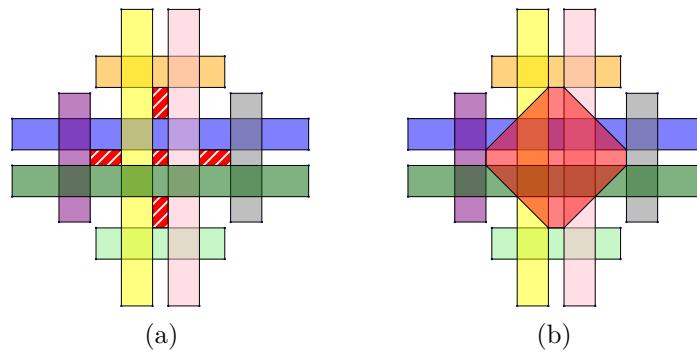
■ **Figure 3** Number of V -maximal convex polygons and the time to enumerate them.



■ **Figure 4** Number of $(V \cup S_1)$ -maximal convex polygons and the time to enumerate them.



■ **Figure 5** All the 82 V -maximal convex polygons for the `socg_fixed60` instance with (a) the 1009 arrangement witnesses and (b) the 200 vertex witnesses.

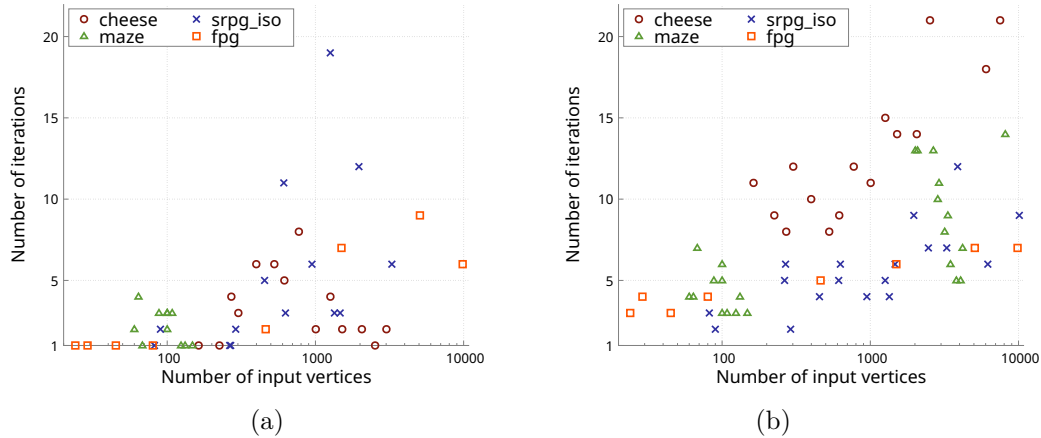


■ **Figure 6** (a) A solution that covers all vertex witnesses of the `socg_fixed60` instance but not the whole polygon. Uncovered regions are marked in striped red. (b) The optimal solution obtained from the previous one by merging the uncovered regions.

A set of witnesses W that also gives excellent results and is much faster to compute is called *quick vertex witnesses*. For each vertex v of P , we consider all edges in \mathcal{C} and also P that are adjacent to v . We order these edges around v starting and ending with the edges of P . For each pair of consecutive edges, we add a point w to W that is between the two consecutive edges and infinitely close to v . Notice that the number of vertex witnesses is linear in the number of edges of \mathcal{C} and it can also be built in near linear time, avoiding the construction of the whole arrangement of \mathcal{C} . If P has not colinear points, then the quick vertex witnesses give the same vertex coverage guarantee as the vertex witnesses. We represent points that are arbitrarily close to v implicitly as a point and a direction.

Given a set \mathcal{S}' of convex polygons that cover W , there are two natural options to produce a valid solution \mathcal{S} . The first option is to make $\mathcal{S} = \mathcal{S}' \cup \mathcal{R}$ for a set \mathcal{R} built as follows. The *uncovered region* $P \setminus \cup_{C \in \mathcal{S}'} C$ consists of a set \mathcal{U} of disjoint polygons, possibly with holes (Figure 6(a)). However, most of the time the polygons in \mathcal{U} are in fact convex. For each polygon $U \in \mathcal{U}$, if U is convex, then we add U to \mathcal{R} . Otherwise, we triangulate U and add the triangles to \mathcal{R} . Furthermore, we can greedily merge convex polygons in \mathcal{R} to reduce their number, as long as the convex hull of the union remains inside P , which works very well for the SoCG logo solution shown in Figure 6(b).

A second option is normally preferable and is based on the constraint generation technique, widely used in integer programming. We build the set \mathcal{R} as before, but for each convex polygon $R \in \mathcal{R}$ we add to W a point inside R . Then, we run the solver again and repeat until a valid solution is found (or one with a very few uncovered regions). It is perhaps surprising how few iterations are normally needed, as shown in Figure 7.



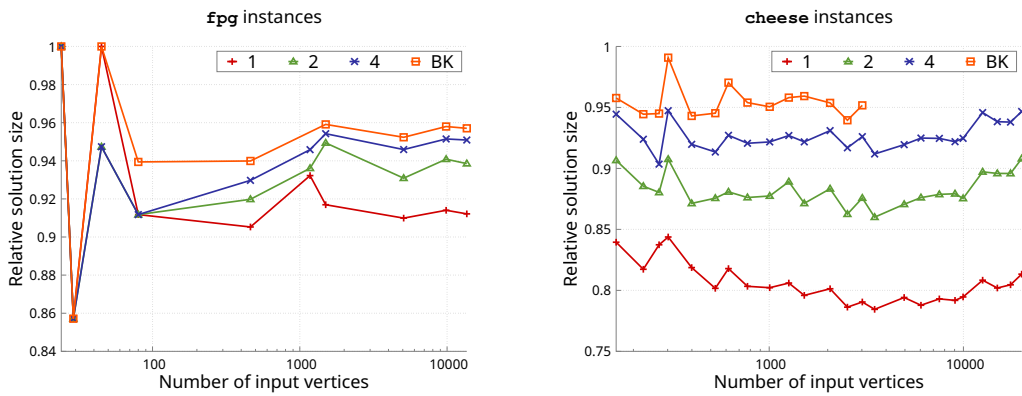
■ **Figure 7** Number of iterations to find a valid solution starting from quick vertex witnesses using IP as the solver, setting \mathcal{C} as (a) all V -maximal convex polygons and (b) 2 times triangulation ($V \cup S_2(\mathcal{C})$)-bloated convex polygons.

Set Cover Solver. A simple and often efficient way to solve a set cover problem (W, \mathcal{C}) is to model the problem as *integer programming* (IP) and then use the CPLEX solver [4]. Each set in \mathcal{C} becomes a binary variable and each witness point $w \in W$ becomes a constraint forcing the sum of the sets that contain w to be at least 1. As discussed in the next section, this approach can optimally solve fairly large problems in seconds and give good approximation guarantees to some extremely large problems. However, for some large problems the solution found is extremely bad (sometimes worse than a greedy algorithm). Another solver we used is based on *simulated annealing*, which is described in the full version.

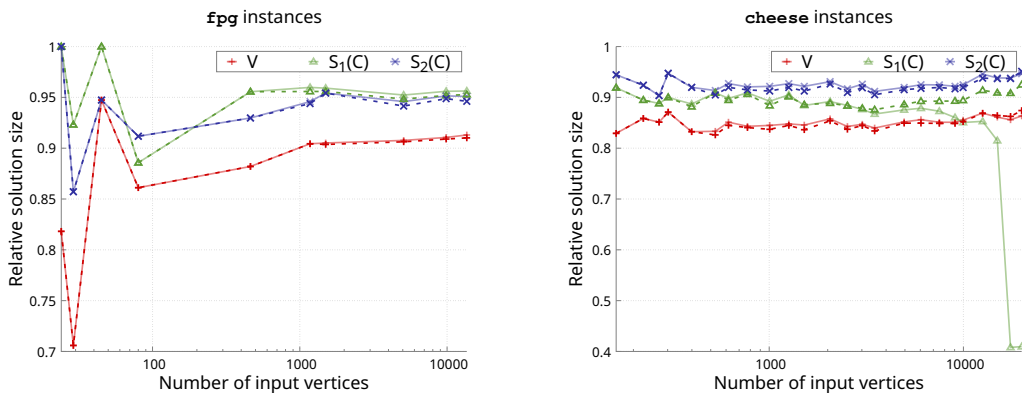
4 Results

We now discuss the quality of the solutions obtained for each technique. Our C++ code uses CGAL [8] and CPLEX [4] and is run on Fedora Linux on a Dell Precision 7560 laptop with an Intel Core i7-11850H and 128GB of RAM. All times refer to a single core execution with scheduling coordinated by GNU Parallel [7]. Our plots for a solution \mathcal{S} use the *relative solution size*, defined as $|\mathcal{S}^*|/|\mathcal{S}|$, where \mathcal{S}^* is the best solution submitted among all teams. This corresponds to the square root of the Challenge *score* of \mathcal{S} .

Figure 8 compares the different techniques to obtain V -maximal convex polygons before bloating them. As the figure shows, using 4 replications of each constrained Delaunay triangle gives solutions that are almost as good as Bron-Kerbosch, but works on all instance sizes. Hence, we use this setting for Figures 9 and 10. Figure 9 shows the relative solution sizes using different bloating approaches and comparing the simulated annealing and the IP solvers. You can see that the simulated annealing solver is only slightly worse than IP for small instances, but better for large **cheese** instances. We limited the running time of IP to 10 minutes per iteration. The total running times of the solvers are compared in Figure 10.

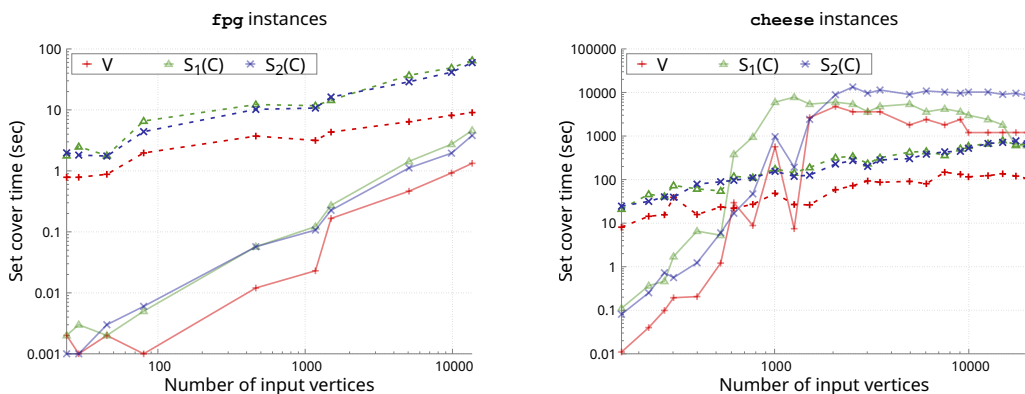


■ **Figure 8** Solution sizes relative to the best Challenge solution. Data is based on a triangulation replicated 1, 2, or 4 times and randomly bloated using V . Alternatively, we use Bron-Kerbosch (BK) to obtain all V -maximal polygons, when the running time is not too long. Afterwards, all collections have bloated again using $V \cup S_2(C)$ and solved using IP.

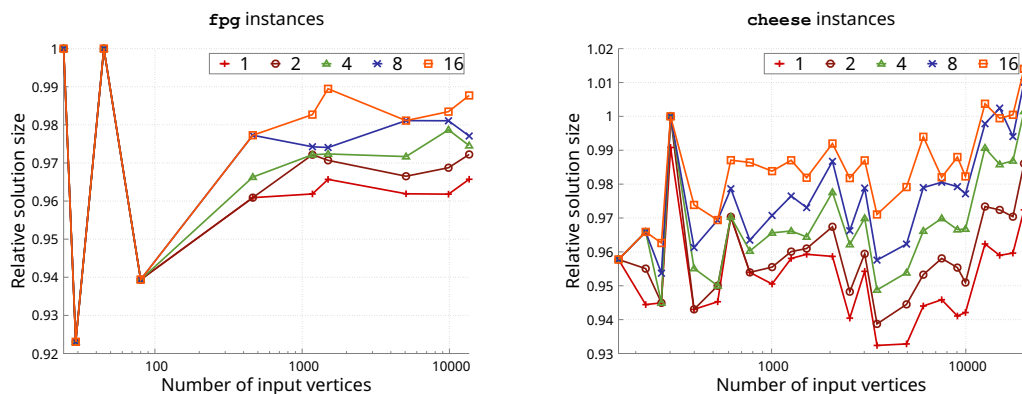


■ **Figure 9** Solution sizes relative to the best Challenge solution. A solid line is used for the IP solver and a dashed line for simulated annealing. Data is based on a triangulation randomly bloated using V (red) and then bloated again using $V \cup S_1(C)$ (green), or $V \cup S_2(C)$ (blue).

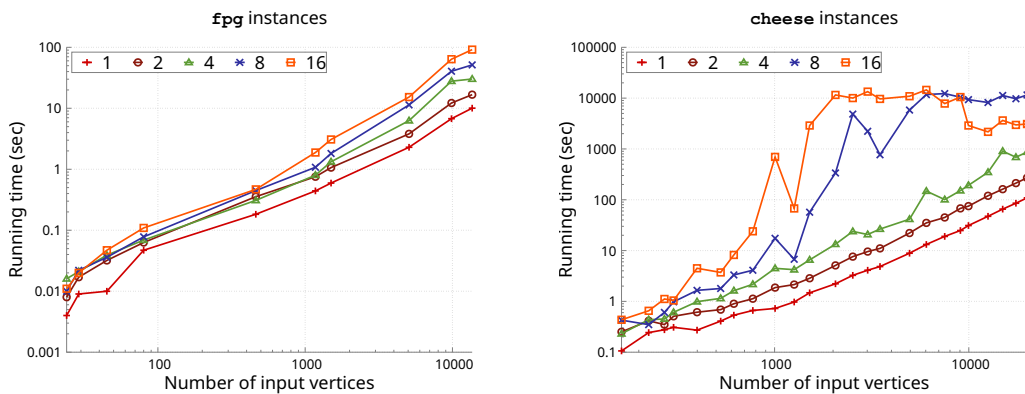
A much better collection is obtained by using the union of several high quality solutions as a collection. To produce the previous plots, we performed 3 independent runs for each settings (showing the best result found). Figures 11 and 12 shows the solution sizes and times obtained by using the best k solutions from these runs as the collection.



■ **Figure 10** Running times in seconds of the set cover solvers used to find the solutions of Figure 9.



■ **Figure 11** Relative solution sizes merging the k best solutions for different values of k .



■ **Figure 12** Running times in seconds to find the solutions of Figure 11.

References

- 1 Mikkel Abrahamsen, William B. Meyling, and André Nusser. Constructing concise convex covers via clique covers. In *39th International Symposium on Computational Geometry (SoCG 2023)*, volume 258, pages 62:1–62–9, 2023.
- 2 Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.
- 3 Kyung-Yong Chwa, Byung-Cheol Jo, Christian Knauer, Esther Moet, René Van Oostrum, and Chan-Su Shin. Guarding art galleries by guarding witnesses. *International Journal of Computational Geometry & Applications*, 16(02n03):205–226, 2006.
- 4 IBM ILOG CPLEX. V22.1: User’s manual for CPLEX. *International Business Machines Corporation*, 2023.
- 5 Sándor P. Fekete, Phillip Keldenich, Dominik Krupke, and Stefan Schirra. Minimum coverage by convex polygons: The CG:SHOP challenge 2023, 2023. [arXiv:2303.07007](https://arxiv.org/abs/2303.07007).
- 6 Ina Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science*, 250(1-2):1–30, 2001.
- 7 O. Tange. Gnu parallel - the command-line power tool. *login: The USENIX Magazine*, 36(1):42–47, February 2011. URL: <http://www.gnu.org/s/parallel>.
- 8 The CGAL Project. *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.1 edition, 2022. URL: <https://doc.cgal.org/5.5.1/Manual/packages.html>.

