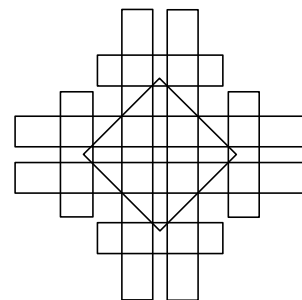


31st International Symposium on Computational Geometry

SoCG'15, June 22–25, 2015, Eindhoven, The Netherlands

Edited by

Lars Arge
János Pach



Editors

Lars Arge	János Pach
MADALGO, Aarhus University	EPFL / Rényi Institute
Aarhus, Denmark	Lausanne, Switzerland / Budapest, Hungary
large@madalgo.au.dk	pach@renyi.hu

ACM Classification 1998

F.2.2 [Analysis of Algorithms and Problem Complexity] Nonnumerical Algorithms and Problems – Geometrical problems and computations, G.2.1 [Discrete Mathematics] Combinatorics, I.3.5 [Computer Graphics] Computational Geometry and Object Modeling

ISBN 978-3-939897-83-5

Published online and open access by

Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at <http://www.dagstuhl.de/dagpub/978-3-939897-83-5>.

Publication date

June, 2015

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 <http://dnb.d-nb.de>.

License

This work is licensed under a Creative Commons Attribution 3.0 Unported license (CC-BY 3.0): <http://creativecommons.org/licenses/by/3.0/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.2015.i

ISBN 978-3-939897-83-5

ISSN 1868-8969

<http://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

- Susanne Albers (TU München)
- Chris Hankin (Imperial College London)
- Deepak Kapur (University of New Mexico)
- Michael Mitzenmacher (Harvard University)
- Madhavan Mukund (Chennai Mathematical Institute)
- Catuscia Palamidessi (INRIA)
- Wolfgang Thomas (*Chair*, RWTH Aachen)
- Pascal Weil (CNRS and University Bordeaux)
- Reinhard Wilhelm (Saarland University)

ISSN 1868-8969

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

Dedicated to the memories of Ferran Hurtado and Jiří Matoušek.

■ Contents

Foreword	xiii
Conference Organization	xv
External Reviewers	xvii
Sponsors	xix

Session 1: Best Paper

Combinatorial Discrepancy for Boxes via the γ_2 Norm <i>Jiří Matoušek and Aleksandar Nikolov</i>	1
--	---

Session 2: Multimedia Previews

Tilt: The Video – Designing Worlds to Control Robot Swarms with Only Global Signals <i>Aaron T. Becker, Erik D. Demaine, Sándor P. Fekete, Hamed Mohtasham Shad, and Rose Morris-Wright</i>	16
Automatic Proofs for Formulae Enumerating Proper Polycubes <i>Gill Barequet and Mira Shalah</i>	19
Visualizing Sparse Filtrations <i>Nicholas J. Cavanna, Mahmoodreza Jahanseir, and Donald R. Sheehy</i>	23
Visualizing Quickest Visibility Maps <i>Topi Talvitie</i>	26

Session 3a

Sylvester-Gallai for Arrangements of Subspaces <i>Zeev Dvir and Guangda Hu</i>	29
Computational Aspects of the Colorful Carathéodory Theorem <i>Wolfgang Mulzer and Yannik Stein</i>	44
Semi-algebraic Ramsey Numbers <i>Andrew Suk</i>	59
A Short Proof of a Near-Optimal Cardinality Estimate for the Product of a Sum Set <i>Oliver Roche-Newton</i>	74
A Geometric Approach for the Upper Bound Theorem for Minkowski Sums of Convex Polytopes <i>Menelaos I. Karavelas and Eleni Tzanaki</i>	81
Two Proofs for Shallow Packings <i>Kunal Dutta, Esther Ezra, and Arijit Ghosh</i>	96

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Session 3b

Shortest Path in a Polygon using Sublinear Space <i>Sariel Har-Peled</i>	111
Optimal Morphs of Convex Drawings <i>Patrizio Angelini, Giordano Da Lozzo, Fabrizio Frati, Anna Lubiw, Maurizio Patrignani, and Vincenzo Roselli</i>	126
1-String B_2 -VPG Representation of Planar Graphs <i>Therese Biedl and Martin Derka</i>	141
Spanners and Reachability Oracles for Directed Transmission Graphs <i>Haim Kaplan, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth</i>	156
Recognition and Complexity of Point Visibility Graphs <i>Jean Cardinal and Udo Hoffmann</i>	171
Geometric Spanners for Points Inside a Polygonal Domain <i>Mohammad Ali Abam, Marjan Adeli, Hamid Homapour, and Pooya Zafar Asadollahpoor</i>	186

Session 4a

An Optimal Algorithm for the Separating Common Tangents of Two Polygons <i>Mikkel Abrahamsen</i>	198
A Linear-Time Algorithm for the Geodesic Center of a Simple Polygon <i>Hee Kap Ahn, Luis Barba, Prosenjit Bose, Jean-Lou De Carufel, Matias Korman, and Eunjin Oh</i>	209
On the Smoothed Complexity of Convex Hulls <i>Olivier Devillers, Marc Glisse, Xavier Goaoc, and Rémy Thomasse</i>	224
Finding All Maximal Subsequences with Hereditary Properties <i>Drago Bokal, Sergio Cabello, and David Eppstein</i>	240

Session 4b

Riemannian Simplices and Triangulations <i>Ramsay Dyer, Gert Vegter, and Mathijs Wintraecken</i>	255
An Edge-Based Framework for Enumerating 3-Manifold Triangulations <i>Benjamin A. Burton and William Pettersson</i>	270
Order on Order Types <i>Alexander Pilz and Emo Welzl</i>	285
Limits of Order Types <i>Xavier Goaoc, Alfredo Hubard, Rémi de Joannis de Verclos, Jean-Sébastien Sereni, and Jan Volec</i>	300

Session 5a

Combinatorial Redundancy Detection
Komei Fukuda, Bernd Gärtner, and May Szedlák 315

Effectiveness of Local Search for Geometric Optimization
Vincent Cohen-Addad and Claire Mathieu 329

On the Shadow Simplex Method for Curved Polyhedra
Daniel Dadush and Nicolai Hähnle 345

Session 5b

Pattern Overlap Implies Runaway Growth in Hierarchical Tile Systems
Ho-Lin Chen, David Doty, Ján Maňuch, Arash Rafiey, and Ladislav Stacho 360

Space Exploration via Proximity Search
Sariel Har-Peled, Nirman Kumar, David M. Mount, and Benjamin Raichel 374

Star Unfolding from a Geodesic Curve
Stephen Kiazzyk and Anna Lubiw 390

Session 6: Invited Talk

The Dirac-Motzkin Problem on Ordinary Lines and the Orchard Problem
Ben J. Green 405

Session 7a

On the Beer Index of Convexity and Its Variants
Martin Balko, Vít Jelínek, Pavel Valtr, and Bartosz Walczak 406

Tight Bounds for Conflict-Free Chromatic Guarding of Orthogonal Art Galleries
Frank Hoffmann, Klaus Kriegel, Subhash Suri, Kevin Verbeek, and Max Willert .. 421

Low-Quality Dimension Reduction and High-Dimensional Approximate Nearest Neighbor
Evangelos Anagnostopoulos, Ioannis Z. Emiris, and Ioannis Psarros 436

Restricted Isometry Property for General p -Norms
Zeyuan Allen-Zhu, Rati Gelashvili, and Ilya Razenshteyn 451

Session 7b

Strong Equivalence of the Interleaving and Functional Distortion Metrics for Reeb Graphs
Ulrich Bauer, Elizabeth Munch, and Yusu Wang 461

On Generalized Heawood Inequalities for Manifolds: A Van Kampen–Flores-type
 Nonembeddability Result
*Xavier Goaoc, Isaac Mabillard, Pavel Paták, Zuzana Patáková, Martin Tancer, and Uli
 Wagner* 476

Comparing Graphs via Persistence Distortion <i>Tamal K. Dey, Dayu Shi, and Yusu Wang</i>	491
---	-----

Bounding Helly Numbers via Betti Numbers <i>Xavier Goaoc, Pavel Paták, Zuzana Patáková, Martin Tancer, and Uli Wagner</i> ...	507
--	-----

Session 8a

Polynomials Vanishing on Cartesian Products: The Elekes-Szabó Theorem Revisited <i>Orit E. Raz, Micha Sharir, and Frank de Zeeuw</i>	522
---	-----

Bisector Energy and Few Distinct Distances <i>Ben Lund, Adam Sheffer, and Frank de Zeeuw</i>	537
---	-----

Incidences between Points and Lines in Three Dimensions <i>Micha Sharir and Noam Solomon</i>	553
---	-----

The Number of Unit-Area Triangles in the Plane: Theme and Variations <i>Orit E. Raz and Micha Sharir</i>	569
---	-----

On the Number of Rich Lines in Truly High Dimensional Sets <i>Zeev Dvir and Sivakanth Gopi</i>	584
---	-----

Realization Spaces of Arrangements of Convex Bodies <i>Michael Gene Dobbins, Andreas Holmsen, and Alfredo Hubbard</i>	599
--	-----

Session 8b

Computing Teichmüller Maps between Polygons <i>Mayank Goswami, Xianfeng Gu, Vamsi P. Pingali, and Gaurish Telang</i>	615
---	-----

On-line Coloring between Two Lines <i>Stefan Felsner, Piotr Micek, and Torsten Ueckerdt</i>	630
--	-----

Building Efficient and Compact Data Structures for Simplicial Complexes <i>Jean-Daniel Boissonnat, Karthik C. S., and Sébastien Tavenas</i>	642
--	-----

Shortest Path to a Segment and Quickest Visibility Queries <i>Esther M. Arkin, Alon Efrat, Christian Knauer, Joseph S. B. Mitchell, Valentin Polishchuk, Günter Rote, Lena Schlipf, and Topi Talvitie</i>	658
--	-----

Trajectory Grouping Structure under Geodesic Distance <i>Irina Kostitsyna, Marc van Kreveld, Maarten Löffler, Bettina Speckmann, and Frank Staals</i>	674
--	-----

From Proximity to Utility: A Voronoi Partition of Pareto Optima <i>Hsien-Chih Chang, Sariel Har-Peled, and Benjamin Raichel</i>	689
--	-----

Session 9a

Faster Deterministic Volume Estimation in the Oracle Model via Thin Lattice Coverings <i>Daniel Dadush</i>	704
---	-----

Optimal Deterministic Algorithms for 2-d and 3-d Shallow Cuttings <i>Timothy M. Chan and Konstantinos Tsakalidis</i>	719
---	-----

A Simpler Linear-Time Algorithm for Intersecting Two Convex Polyhedra in Three Dimensions <i>Timothy M. Chan</i>	733
---	-----

Session 9b

Approximability of the Discrete Fréchet Distance <i>Karl Bringmann</i>	739
---	-----

The Hardness of Approximation of Euclidean k-Means <i>Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop</i>	754
---	-----

A Fire Fighter's Problem <i>Rolf Klein, Elmar Langetepe, and Christos Levcopoulos</i>	768
--	-----

Session 10a

Approximate Geometric MST Range Queries <i>Sunil Arya, David M. Mount, and Eunhui Park</i>	781
---	-----

Maintaining Contour Trees of Dynamic Terrains <i>Pankaj K. Agarwal, Thomas Mølhave, Morten Revsbæk, Issam Safa, Yusu Wang, and Jungwoo Yang</i>	796
--	-----

Hyperorthogonal Well-Folded Hilbert Curves <i>Arie Bos and Herman J. Haverkort</i>	812
---	-----

Session 10b

Topological Analysis of Scalar Fields with Outliers <i>Mickaël Buchet, Frédéric Chazal, Tamal K. Dey, Fengtao Fan, Steve Y. Oudot, and Yusu Wang</i>	827
---	-----

On Computability and Triviality of Well Groups <i>Peter Franek and Marek Krčál</i>	842
---	-----

Geometric Inference on Kernel Density Estimates <i>Jeff M. Phillips, Bei Wang, and Yan Zheng</i>	857
---	-----

Session 11: Invited Talk

Modeling Real-World Data Sets <i>Susanne Albers</i>	872
--	-----

■ Foreword

The research papers, the abstracts of invited talks, and the descriptions of videos and multimedia presentations in this volume constitute the proceedings of the 31st International Symposium on Computational Geometry (SoCG'15), which was held as part of CG Week 2015 at TU Eindhoven, the Netherlands, June 22-25, 2015.

There were 154 papers submitted to SoCG'15 of which the program committee selected 59 for presentation after a substantial review process involving 241 external reviewers. The online submission and the review were conducted using EasyChair. Several papers have been selected for the special issues of *Discrete & Computational Geometry* and the *Journal of Computational Geometry* dedicated to SoCG'15.

In addition to the technical papers, four submissions—three videos and one applet—were received in response to the Call for Video and Multimedia. All four were reviewed and accepted for presentation. The extended abstracts that describe the accepted submissions are included in this proceedings. The final versions of the videos for the accepted submissions will be archived at <http://www.computational-geometry.org>.

The Best Paper Award went to the paper “Combinatorial Discrepancy for Boxes via the γ_2 Norm” by Jiří Matoušek and Aleksandar Nikolov. The Best Student Presentation Award will be determined at the symposium based on input of the attendees. We thank all authors of submitted papers, videos and multimedia presentations. We also thank all people who gave their time for the quality and success of this conference, especially the local organizers, the external reviewers, and the program committee members.

Lars Arge

Program Committee co-chair
MADALGO, Aarhus University

János Pach

Program Committee co-chair
EPFL and Rényi Institute

Wolfgang Mulzer

Video and Multimedia Committee chair
FU Berlin



■ Conference Organization

SoCG Program Committee

Lars Arge (*co-chair, MADALGO, Aarhus University*)
János Pach (*co-chair, EPFL and Rényi Institute*)
Boris Aronov (*New York University, Polytechnic*)
Anne Driemel (*TU Eindhoven*)
John Hershberger (*Mentor Graphics*)
Akitoshi Kawamura (*University of Tokyo*)
Stefan Langerman (*Université Libre de Bruxelles*)
Kasper Green Larsen (*MADALGO, Aarhus University*)
Nabil Mustafa (*Université Paris-Est and ESIEE Paris*)
Amir Nayyeri (*Oregon State University*)
Marcus Schaefer (*DePaul University*)
Donald Sheehy (*University of Connecticut*)
Anastasios Sidiropoulos (*Ohio State University*)
Michiel Smid (*Carleton University*)
Monique Teillaud (*INRIA*)
Csaba Tóth (*California State University, Northridge*)
Antoine Vigneron (*KAUST*)
Haitao Wang (*Utah State University*)

Multimedia Program Committee

Wolfgang Mulzer (*chair, Freie Universität Berlin*)
Esther Ezra (*Georgia Institute of Technology*)
Matthias Henze (*Freie Universität Berlin*)
Matias Korman (*National Institute of Informatics, Tokyo*)
Maarten Löffler (*Utrecht University*)
Ludmila Scharf (*Freie Universität Berlin*)
Christiane Schmidt (*Hebrew University of Jerusalem*)
Stefanie Wührer (*Universität des Saarlandes*)

Workshop Program Committee

Stefan Langerman (*chair, Université Libre de Bruxelles*)
Franz Aurenhammer (*TU Graz*)
Jean Cardinal (*Université Libre de Bruxelles*)
Mark de Berg (*TU Eindhoven*)
Olivier Devillers (*INRIA*)
Vida Dujmovic (*University of Ottawa*)
John Iacono (*New York University, Polytechnic*)
Micha Sharir (*Tel Aviv University*)

31st International Symposium on Computational Geometry (SoCG'15).
Editors: Lars Arge and János Pach



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Young Researchers Forum Program Committee

Rodrigo I. Silveira (*chair, Universitat Politècnica de Catalunya*)

Benjamin Burton (*The University of Queensland*)

David Eppstein (*University of California, Irvine*)

Matthew Katz (*Ben-Gurion University*)

Matias Korman (*National Institute of Informatics, Tokyo*)

Maarten Löffler (*Utrecht University*)

Csaba Tóth (*California State University, Northridge*)

Anna Lubiw (*University of Waterloo*)

Local Organizers

Bettina Speckmann (*co-chair, TU Eindhoven*)

Marc van Kreveld (*co-chair, Utrecht University*)

Patty Koorn (*TU Eindhoven*)

Maarten Löffler (*Utrecht University*)

Arthur van Goethem (*TU Eindhoven*)

Steering Committee

Jeff Erickson (*chair, University of Illinois at Urbana-Champaign*)

David Eppstein (*secretary, University of California, Irvine*)

Mark de Berg (*TU Eindhoven*)

Joseph S. B. Mitchell (*Stony Brook University*)

Günter Rote (*Freie Universität Berlin*)

■ Additional Reviewers

Amirali Abdullah	Bernard Chazelle	Andrzej Grzesik
Karim Adiprasito	Otfried Cheong	Joachim Gudmundsson
Pankaj K. Agarwal	Tobias Christ	Larry Guth
Hee-Kap Ahn	Ken Clarkson	Dan Halperin
Oswin Aichholzer	Vincent Cohen-Addad	Thomas Dueholm Hansen
Susanne Albers	Éric Colin de Verdière	Sariel Har-Peled
Noga Alon	Atlas F. Cook IV	Herman Haverkort
Greg Aloupis	Jean Cousty	Barry Hayes
Alexandr Andoni	Artur Czumaj	Meng He
Sunil Arya	Mirela Damian	Martin Held
Dominique Attali	Mark de Berg	Yasuaki Hiraoka
Franz Aurenhammer	Jean-Lou De Carufel	Udo Hoffmann
Arturs Backurs	Frank de Zeeuw	Xiaocheng Hu
Sang Won Bae	Olivier Devillers	Ruqi Huang
Martin Balko	Tamal Dey	Stefan Huber
Boaz Barak	Michael Gene Dobbins	John Iacono
Luis Barba	Philippe Duchon	Piotr Indyk
Abdul Basit	Ingo van Duijn	Hiro Ito
Ulrich Bauer	Laurent Dupont	Takehiro Ito
Huxley Bennett	Stephane Durocher	Yoichi Iwata
Edvin Berglin	Ramsay Dyer	Justin Iwerks
Daniel Binham	José-Miguel Díaz-Báñez	Mahmoodreza Jahanseir
Johannes Blömer	Khaled Elbassioni	Minghui Jiang
Jean-Daniel Boissonnat	Amr Elmasry	Allan Grønlund Jørgensen
Nicolas Bonichon	David Eppstein	Shahin Kamali
Glencora Borradaile	Jeff Erickson	Menelaos I. Karavelas
Magnus Bakke Botnan	Thomas Erlebach	Alexander Kasprzyk
Karl Bringmann	Esther Ezra	Matthew Katz
Tobias Brunsch	Brittany Terese Fasy	Mark Keil
Kevin Buchin	Sándor Fekete	Michael Kerber
Maike Buchin	Vissarion Fisikopoulos	Masashi Kiyomi
Norbert Bus	Hervé Fournier	Rolf Klein
Lilian Buzer	Kyle Fox	Christian Knauer
Jaroslav Byrka	Fabrizio Frati	Tsvi Kopelowitz
Andreas Bärtschi	Radoslav Fulek	Matias Korman
Sergio Cabello	Shashidhara Ganjugunte	Nirman Kumar
Jean Cardinal	Jie Gao	Vitaliy Kurlin
Nicholas Cavanna	Leszek Gasieniec	Michael Lampis
Frédéric Cazals	Subir Ghosh	Sylvain Lazard
Erin Chambers	Matt Gibson	Francis Lazarus
Timothy M. Chan	Marc Glisse	Jian Li
Frédéric Chazal	Xavier Goaoc	Minming Li

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach



Leibniz International Proceedings in Informatics

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

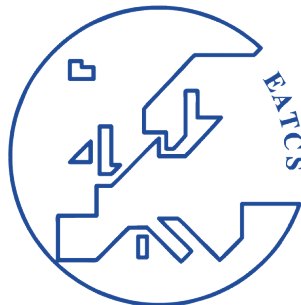
Shi Li	Salman Parsa	Bettina Speckmann
Bernard Lidicky	Amit Patel	Frank Staals
Jyh-Ming Lien	Maurizio Patrignani	William Steiger
André Lieutier	Michael Payne	Yannik Stein
Nutan Limaye	Richard Peng	Noah Stephens-Davidowitz
Chih-Hung Liu	Seth Pettie	Jian Sun
Benjamin Lund	Jeff Phillips	Kanat Tangwongsan
Maarten Löffler	Vincent Pilaud	Shin-Ichi Tanigawa
Pedro Machado	Alexander Pilz	Takahisa Toda
Manhães de Castro	Valentin Polishchuk	Takeshi Tokuyama
Takanori Maehara	Attila Por	Marc van Kreveld
Clément Maria	Marc Pouget	Kasturi Varadarajan
Liam Mencil	Eric Price	Ameya Velingker
Wouter Meulemans	Benjamin Raichel	Suresh Venkatasubramanian
Frédéric Meunier	Mathias Rav	Sander Verdonschot
David L. Millman	Saurabh Ray	Costin Vilcu
Joseph S. B. Mitchell	Orit Raz	Lukas Vokrinek
Scott Mitchell	Daniel Reem	Nicolai Vorobjov
Hiroyuki Miyata	Marcel Roeloffzen	Uli Wagner
Luis Montejano	Alfred Rossi	Bei Wang
Pat Morin	Günter Rote	Yusu Wang
Sonoko Moriyama	Joachim Rubinstein	Rephael Wenger
Guillaume Moroz	Heiko Röglin	Carola Wenk
Dmitriy Morozov	Chandan Saha	Andrew Winslow
David Mount	Toshiki Saitoh	David Woodruff
Wolfgang Mulzer	Raman Sanyal	Ge Xia
Satoshi Murai	Rik Sarkar	Jinhui Xu
Quentin Mérigot	Maria Saumell	Chee Yap
Jelani Nelson	Lars Schewe	Alper Yildirim
Huy L. Nguyen	Jean-Marc Schlenker	Yelena Yuditsky
Bengt J. Nilsson	Micha Sharir	Joshua Zahl
Joseph O'Rourke	Jonathan Shewchuk	Wuzhou Zhang
Yoshio Okamoto	Rodrigo Silveira	Gelin Zhou
Yota Otachi	Marcelo Siqueira	Hang Zhou
Steve Oudot	Rene Sitters	Binhai Zhu
Ozgur Ozkan	Arkadiy Skopenkov	Günter M. Ziegler
Andreas Paffenholz	Christian Sohler	Rade Zivaljevic
Evanthia Papadopoulou	Noam Solomon	Anastasios Zouzias
Periklis Papakonstantinou	József Solymosi	

■ Sponsors

We gratefully acknowledge the financial support received from the sponsors of CG Week 2015: TU Eindhoven, European Science Foundation (ESF), Royal Netherlands Academy of Arts and Sciences (KNAW), Netherlands Organisation for Scientific Research (NWO), NWO Gravitation Programme Networks, Center for Massive Data Algorithmics (MADALGO), European Association for Theoretical Computer Science (EATCS).



KONINKLIJKE NEDERLANDSE
AKADEMIE VAN WETENSCHAPPEN



31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach



Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Combinatorial Discrepancy for Boxes via the γ_2 Norm

Jiří Matoušek^{*1} and Aleksandar Nikolov²

- 1 Department of Applied Mathematics
Charles University
Malostranské nám. 25 118 00 Praha 1, Czech Republic, and
Department of Computer Science
ETH Zurich
8092 Zurich, Switzerland
matousek@kam.mff.cuni.cz
- 2 Microsoft Research
Redmond, WA, USA
alenik@microsoft.com

Abstract

The γ_2 norm of a real $m \times n$ matrix A is the minimum number t such that the column vectors of A are contained in a 0-centered ellipsoid $E \subseteq \mathbb{R}^m$ that in turn is contained in the hypercube $[-t, t]^m$. This classical quantity is polynomial-time computable and was proved by the second author and Talwar to approximate the *hereditary discrepancy* $\text{herdisc } A$ as follows: $\gamma_2(A)/O(\log m) \leq \text{herdisc } A \leq \gamma_2(A) \cdot O(\sqrt{\log m})$. Here we provide a simplified proof of the first inequality and show that both inequalities are asymptotically tight in the worst case.

We then demonstrate on several examples the power of the γ_2 norm as a tool for proving lower and upper bounds in discrepancy theory. Most notably, we prove a new lower bound of $\Omega(\log^{d-1} n)$ for the *d-dimensional Tusnády problem*, asking for the combinatorial discrepancy of an n -point set in \mathbb{R}^d with respect to axis-parallel boxes. For $d > 2$, this improves the previous best lower bound, which was of order approximately $\log^{(d-1)/2} n$, and it comes close to the best known upper bound of $O(\log^{d+1/2} n)$, for which we also obtain a new, very simple proof. Applications to lower bounds for dynamic range searching and lower bounds in differential privacy are given.

1998 ACM Subject Classification G.2.1 Combinatorics, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases discrepancy theory, range counting, factorization norms

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.1

1 Introduction

Discrepancy and hereditary discrepancy. Let $V = [n] := \{1, 2, \dots, n\}$ be a ground set and $\mathcal{F} = \{F_1, F_2, \dots, F_m\}$ be a system of subsets of V . The *discrepancy* of \mathcal{F} is

$$\text{disc } \mathcal{F} := \min_{x \in \{-1, 1\}^n} \text{disc}(\mathcal{F}, x),$$

where the minimum is over all choices of a vector $x \in \{-1, +1\}^n$ of signs for the points, and $\text{disc}(\mathcal{F}, x) := \max_{i=1, 2, \dots, m} \left| \sum_{j \in F_i} x_j \right|$. (A vector $x \in \{-1, 1\}^n$ is usually called a *coloring* in this context.)

* Research supported by the ERC Advanced Grant No. 267165.



This combinatorial notion of discrepancy originated in the classical theory of *irregularities of distribution*, as treated, e.g., in [8, 20, 3], and more recently it has found remarkable applications in computer science and elsewhere (see [43, 15, 31] for general introductions and, e.g., [25] for a recent use).

For the subsequent discussion, we also need the notion of discrepancy for matrices: for an $m \times n$ real matrix A we set $\text{disc } A := \min_{x \in \{-1, 1\}^n} \|Ax\|_\infty$. If A is the incidence matrix of the set system \mathcal{F} as above (with $a_{ij} = 1$ if $j \in F_i$ and $a_{ij} = 0$ otherwise), then the matrix definition coincides with the one for set systems.

A set system \mathcal{F} with small, even zero, discrepancy may contain a set system with large discrepancy. This phenomenon was exploited in [14] for showing that, assuming $P \neq NP$, no polynomial-time algorithm can distinguish systems \mathcal{F} with zero discrepancy from those with discrepancy of order \sqrt{n} in the regime $m = O(n)$, which practically means that $\text{disc } \mathcal{F}$ cannot be approximated at all in polynomial time.

A better behaved notion is the *hereditary discrepancy* of \mathcal{F} , given by

$$\text{herdisc } \mathcal{F} := \max_{J \subseteq V} \text{disc}(\mathcal{F}|_J),$$

where $\mathcal{F}|_J$ denotes the *restriction* of the set system \mathcal{F} to the ground set J , i.e., $\{F \cap J : F \in \mathcal{F}\}$. Similarly, for a matrix A , $\text{herdisc } A := \max_{J \subseteq [n]} \text{disc } A_J$ where A_J is the submatrix of A consisting of the columns indexed by the set J .

At first sight, hereditary discrepancy may seem harder to deal with than discrepancy. For example, while $\text{disc } \mathcal{F} \leq k$ has an obvious polynomial-time verifiable certificate, namely, a suitable coloring $x \in \{-1, 1\}^n$, it is not at all clear how one could certify either $\text{herdisc } \mathcal{F} \leq k$ or $\text{herdisc } \mathcal{F} > k$ in polynomial time.

However, hereditary discrepancy has turned out to have significant advantages over discrepancy. Most of the classical upper bounds for discrepancy of various set systems actually apply to hereditary discrepancy as well. A powerful tool, introduced by Lovász, Spencer and Vesztegombi [28] and called the *determinant lower bound*, works for hereditary discrepancy and *not* for discrepancy. The determinant lower bound for a matrix A is the following algebraically defined quantity:

$$\text{detlb } A = \max_k \max_B |\det B|^{1/k},$$

where B ranges over all $k \times k$ submatrices of A . Lovász et al. proved that $\text{herdisc } A \geq \frac{1}{2} \text{detlb } A$ for all A . Later it was shown in [29] that $\text{detlb } A$ also bounds $\text{herdisc } A$ from above up to a polylogarithmic factor; namely, $\text{herdisc } A = O(\text{detlb}(A) \log(mn) \sqrt{\log n})$.

While the quantity $\text{detlb } A$ enjoys some pleasant properties, there is no known polynomial-time algorithm for computing it. Bansal [4] provided a polynomial-time algorithm that, given a system \mathcal{F} with $\text{herdisc } \mathcal{F} \leq D$, computes a coloring x witnessing $\text{disc } \mathcal{F} = O(D \log(mn))$. However, this is not an approximation algorithm for the hereditary discrepancy in the usual sense, since it may find a low-discrepancy coloring even for \mathcal{F} with large hereditary discrepancy.

The γ_2 factorization norm. The first polynomial-time approximation algorithm with a polylogarithmic approximation factor for hereditary discrepancy was found by the second author, Talwar, and Zhang [37]. Their result was further strengthened and streamlined by the second author and Talwar [35], who showed that hereditary discrepancy is approximated by geometrically defined quantity which turns out to be equivalent to the γ_2 factorization norm from Banach space theory.¹ This connection was implicit in [37].

¹ This equivalence was pointed out to us by Noga Alon and Assaf Naor.

Let the ℓ_∞ norm $\|E\|_\infty$ of an ellipsoid E be defined as the largest ℓ_∞ norm of any point in E . The geometric quantity studied in [35] is the minimum ℓ_∞ norm of a 0-centered ellipsoid E that contains all column vectors of A . As noticed by several experts, this quantity is equal to the γ_2 norm of A , taken as a linear operator from ℓ_1^m to ℓ_∞^m , which is also defined as

$$\gamma_2(A) := \min\{\|B\|_{2 \rightarrow \infty} \|C\|_{1 \rightarrow 2} : A = BC\}.$$

Above, $\|\cdot\|_{p \rightarrow q}$ stands for the $\ell_p \rightarrow \ell_q$ operator norm, and B, C range over linear operators. Treating B and C as matrices, it is easy to see that $\|B\|_{2 \rightarrow \infty}$ is equal to the largest Euclidean norm of row vectors of B , and $\|C\|_{1 \rightarrow 2}$ is equal to the largest Euclidean norm of column vectors of C . We will use both the definition in terms of ellipsoids and the one in terms of a factorization of A . We use the notation $\gamma_2(\mathcal{F})$ for a set system \mathcal{F} to mean the γ_2 norm of the incidence matrix of \mathcal{F} .

In [35] it was shown that $\gamma_2(A)$ can be approximated to any desired accuracy in polynomial time, and the following two inequalities relating $\gamma_2(A)$ to herdisc A were proved: for every matrix A with m rows,

$$\text{herdisc } A \geq \frac{\gamma_2(A)}{O(\log m)}, \text{ and} \tag{1}$$

$$\text{herdisc } A \leq \gamma_2(A) \cdot O(\sqrt{\log m}) \tag{2}$$

These results together provide an $O(\log^{3/2} m)$ -approximation algorithm for herdisc A . (As we will see in Section 4.1 below, (1) is actually valid with $\log \min\{m, n\}$ instead of $\log m$.)

The upper bounds guaranteed by inequality (2) are not constructive, in the sense that we do not know of a polynomial-time algorithm that computes a coloring achieving the upper bound. Nevertheless, the algorithms of Bansal [4] or Rothvoss [40] can be used to find colorings with discrepancy $\gamma_2(A) \cdot O(\log m)$ in polynomial time.

Results on the γ_2 norm. A number of useful properties of γ_2 are known, such as the non-obvious fact that it is indeed a norm [46] (we give an example of how the triangle inequality fails for $\det\text{lb}$), and the fact that it is multiplicative under the *Kronecker product* (or tensor product) of matrices [26]. We further prove a stronger form of the triangle inequality for matrices supported on disjoint subsets of the columns.

Linial, Mendelson, Schechtman and Shraibman [27] observed that for sign matrices A , $\gamma_2(A)$ can be formulated as the optimal value of a semidefinite program. Lee, Shraibman, and Špalek generalized the semidefinite program to arbitrary real matrices, and used it to derive a dual characterization of γ_2 . We use this characterization to give a simplified proof of inequality (1). We also prove that $\gamma_2(A)$ is between $\det\text{lb } A$ and $O(\det\text{lb}(A) \log m)$.

We show that both inequalities (1) and (2) are asymptotically tight in the worst case. For (1), the asymptotic tightness is demonstrated on the following simple example: for the system \mathcal{I}_n of initial segments of $\{1, 2, \dots, n\}$, whose incidence matrix is the lower triangular matrix T_n with 1s on the main diagonal and below it, we prove that the γ_2 norm is of order $\log n$, while the hereditary discrepancy is well known to be 1.

Applications in discrepancy theory. In the second part of the paper we apply the γ_2 norm to prove new results on combinatorial discrepancy, as well as to give simple new proofs of known results.

The most significant result is a new lower bound for the d -dimensional Tusnády’s problem; before stating it, let us give some background.

The “great open problem.” Discrepancy theory started with a result conjectured by Van der Corput [18, 19] and first proved by Van Aardenne-Ehrenfest [1, 2], stating that every infinite sequence (u_1, u_2, \dots) of real numbers in $[0, 1]$ must have a significant deviation from a “perfectly uniform” distribution. Roth [39] found a simpler proof of a stronger bound, and he re-cast the problem in the following setting, dealing with finite point sets in the unit square $[0, 1]^2$ instead of infinite sequences in $[0, 1]$:

Given an n -point set $P \subset [0, 1]^2$, the *discrepancy* of P is defined as

$$D(P, \mathcal{R}_2) := \sup \left\{ \left| |P \cap R| - n\lambda^2(R \cap [0, 1]^d) \right| : R \in \mathcal{R}_2 \right\},$$

where \mathcal{R}_2 denotes the set of all 2-dimensional axis-parallel rectangles (or 2-dimensional intervals), of the form $R = [a_1, b_1] \times [a_2, b_2]$, and λ^2 is the area (2-dimensional Lebesgue measure). More precisely, $D(P, \mathcal{R}_2)$ is the *Lebesgue-measure discrepancy* of P w.r.t. axis-parallel rectangles. Further let $D(n, \mathcal{R}_2) = \inf_{P:|P|=n} D(P, \mathcal{R}_2)$ be the best possible discrepancy of an n -point set.

Roth proved that $D(n, \mathcal{R}_2) = \Omega(\sqrt{\log n})$, while earlier work of Van der Corput yields $D(n, \mathcal{R}_2) = O(\log n)$. Later Schmidt [41] improved the lower bound to $\Omega(\log n)$.

Roth’s setting immediately raises the question about a higher-dimensional analog of the problem: letting \mathcal{R}_d stand for the system of all axis-parallel boxes (or d -dimensional intervals) in $[0, 1]^d$, what is the order of magnitude of $D(n, \mathcal{R}_d)$? There are many ways of showing an upper bound of $O(\log^{d-1} n)$, the first one being the Halton–Hammersley construction [24, 23], and Roth’s lower bound method yields $D(n, \mathcal{R}_d) = \Omega(\log^{(d-1)/2} n)$. In these bounds, d is considered fixed and the implicit constants in the $O(\cdot)$ and $\Omega(\cdot)$ notation may depend on it.

Now, over 50 years later, the upper bound is still the best known, and Roth’s lower bound has been improved only a little: first for $d = 3$ by Beck [7] and by Bilyk and Lacey [10], and then for all d by Bilyk, Lacey, and Vagharyshakyan [11]. The lower bound from [11] has the form $\Omega((\log n)^{(d-1)/2+\eta(d)})$, where $\eta(d) > 0$ is a constant depending on d , with $\eta(d) \geq c/d^2$ for an absolute constant $c > 0$. Thus, the upper bound for $d \geq 3$ is still about the square of the lower bound, and closing this significant gap is called the “great open problem” in the book [8].

Tusnády’s problem. Here we essentially solve a combinatorial analog of this problem. In the 1980s Tusnády raised a question which, in our terminology, can be stated as follows. Let $P \subset \mathbb{R}^2$ be an n -point set, and let $\mathcal{R}_2(P) := \{R \cap P : R \in \mathcal{R}_2\}$ be the system of all subsets of P induced by axis-parallel rectangles $R \in \mathcal{R}_2$. What can be said about the discrepancy of such a set system for the worst possible n -point P ? In other words, what is

$$\text{disc}(n, \mathcal{R}_2) = \max \{ \text{disc } \mathcal{R}_2(P) : |P| = n \}?$$

We stress that for the Lebesgue-measure discrepancy $D(n, \mathcal{R}_d)$ we ask for the best placement of n points so that each rectangle contains approximately the right number of points, while for $\text{disc}(n, \mathcal{R}_2)$ the point set P is given by an adversary, and we seek a ± 1 coloring so that the points in each rectangle are approximately balanced.

Tusnády actually asked if $\text{disc}(n, \mathcal{R}_2)$ could be bounded by a constant independent of n . This was answered negatively by Beck [5], who also proved an upper bound of $O(\log^4 n)$. His lower bound argument uses a “transference principle,” showing that the function $\text{disc}(n, \mathcal{R}_2)$ in Tusnády’s problem cannot be asymptotically smaller than the smallest achievable Lebesgue-measure discrepancy of n points with respect to axis-aligned boxes. (This principle is actually

simple to prove and quite general; Simonovits attributes the idea to V. T. Sós.) The upper bound was improved to $O((\log n)^{3.5+\varepsilon})$ by Beck [6], to $O(\log^3 n)$ by Bohus [12], and to the current best bound of $O(\log^{2.5} n)$ by Srinivasan [44].

The obvious d -dimensional generalization of Tusnády’s problem was attacked by similar methods. All known lower bounds so far relied on the transference principle mentioned above. The current best upper bound for $d \geq 3$ is $O(\log^{d+1/2} n)$ due to Larsen [25], which is a slight strengthening of a previous bound of $O(\log^{d+1/2} n \sqrt{\log \log n})$ from [30].

Here we improve on the lower bound for the d -dimensional Tusnády’s problem significantly; while up until now the uncertainty in the exponent of $\log n$ was roughly between $(d - 1)/2$ and $d + 1/2$, we reduce it to $d - 1$ versus $d + 1/2$.

► **Theorem 1.** *For every fixed $d \geq 2$ and for infinitely many values of n , there exists an n -point set $P \subset \mathbb{R}^d$ with*

$$\text{disc } \mathcal{R}_d(P) = \Omega(\log^{d-1} n),$$

where the constant of proportionality depends only on d .

From the point of view of the “great open problem,” this result is perhaps somewhat disappointing, since it shows that, in order to determine the asymptotics of the Lebesgue-measure discrepancy $D(n, \mathcal{R}_d)$, one has to use some special properties of the Lebesgue measure—combinatorial discrepancy cannot help, at least for improving the upper bound.

Using the γ_2 norm as the main tool, our proof of Theorem 1 is surprisingly simple. In a nutshell, first we observe that, since the target bound is polylogarithmic in n , instead of estimating the discrepancy for some cleverly constructed n -point set P , we can bound from below the hereditary discrepancy of the regular d -dimensional grid $[n]^d$, where $[n] = \{1, 2, \dots, n\}$. By a standard and well known reduction, instead of all d -dimensional intervals in \mathcal{R}_d , it suffices to consider only “anchored” intervals, of the form $[0, b_1] \times \dots \times [0, b_d]$. Now the main observation is that the set system $\mathcal{G}_{d,n}$ induced on $[n]^d$ by anchored intervals is a d -fold product of the system \mathcal{I}_n of one-dimensional intervals mentioned earlier, and its incidence matrix is the d -fold Kronecker product of the matrix T_n .

Thus, by the properties of the γ_2 norm established earlier, we get that $\gamma_2(\mathcal{G}_{d,n})$ is of order $\log^d n$, and inequality (1) finishes the proof of Theorem 1.

At the same time, using the other inequality (2), we obtain a new proof of the best known upper bound $\text{disc}(n, \mathcal{R}_d) = O(\log^{d+1/2} n)$, with no extra effort. This proof is very different from the previously known ones and relatively simple.

The same method also gives a surprisingly precise upper bound on the discrepancy of the set system of all subcubes of the d -dimensional cube $\{0, 1\}^d$, where this time d is a variable parameter, not a constant as before. This discrepancy has previously been studied in [16, 17, 36], and it was known that it is between $2^{c_1 d}$ and $2^{c_2 d}$ for some constants $c_2 > c_1 > 0$. In Section 5.1 we show that it is $2^{(c_0 + o(1))d}$, for $c_0 = \log_2(2/\sqrt{3}) \approx 0.2075$.

Immediate applications in computer science. Our lower bound for Tusnády’s problem implies a lower bound of $\sqrt{t_u t_q} = \Omega(\log^d n)$ on the update time t_u and query time t_q of constant multiplicity oblivious data structures for orthogonal range searching in \mathbb{R}^d in the group model. This lower bound is tight up to a constant. The relationship between hereditary discrepancy and differential privacy from [33] and the lower bound for Tusnády’s problem imply that the necessary error for computing orthogonal range counting queries under differential privacy is $\Omega(\log^{d-1} n)$, which is best possible up to a factor of $\log n$.

Our lower and upper bounds on the discrepancy of subcubes of the Boolean cube $\{0, 1\}^d$ and the results from [37] imply that the necessary and sufficient error for computing marginal queries on d -attribute databases under differential privacy is $(2/\sqrt{3})^{d+o(d)}$.

General theorems on discrepancy. Transferring the various properties of the γ_2 norm into the setting of hereditary discrepancy via inequalities (1), (2), we obtain general results about the behavior of discrepancy under operations on set systems. In particular, we get a sharper version of a result of [29] concerning the discrepancy of the union of several set systems, and a new bound on the discrepancy of a set system \mathcal{F} in which every set $F \in \mathcal{F}$ is a disjoint union $F_1 \cup \dots \cup F_t$, where $\mathcal{F}_1, \dots, \mathcal{F}_t$ are given set systems and $F_i \in \mathcal{F}_i$, $i = 1, 2, \dots, t$. These consequences are presented in the full version of the paper.

Other problems in combinatorial discrepancy: new simple proofs. In the full version we also revisit two set systems for which discrepancy has been studied extensively: arithmetic progressions in $[n]$ and intervals in k permutations of $[n]$. In both of these cases, asymptotically tight bounds have been known. Using the γ_2 norm we recover almost tight upper bounds, up to a factor of $\sqrt{\log n}$, with very short proofs.

2 Properties of the γ_2 norm

2.1 Known properties of γ_2

The γ_2 norm has various favorable properties, which make it a very convenient and powerful tool in studying hereditary discrepancy, as we will illustrate later on. We begin by recalling some classical facts. It is clear that $\gamma_2(A)$ is monotone non-increasing under removing rows or columns of A . From the definition of $\gamma_2(A)$ in terms of factorization of matrices, we also see that $\gamma_2(A) = \gamma_2(A^T)$. Moreover, it is well-known (see e.g. [46]) that γ_2 is indeed a norm and therefore satisfies the triangle inequality, i.e. for any two $m \times n$ matrices A and B we have

$$\gamma_2(A + B) \leq \gamma_2(A) + \gamma_2(B). \quad (3)$$

Remark on the determinant lower bound. Here is an example showing that the determinant lower bound of Lovász et al. does not satisfy the (exact) triangle inequality: for

$$A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 1 & 0 \\ -1 & 1 \end{pmatrix},$$

we have $\det \text{lb } A = \det \text{lb } B = 1$, but $\det \text{lb}(A + B) = \sqrt{5}$.

It may still be that the determinant lower bound satisfies an approximate triangle inequality, say in the following sense: $\det \text{lb}(A_1 + \dots + A_t) \stackrel{?}{\leq} O(t) \cdot \max_i \det \text{lb } A_i$. However, at present we can only prove this kind of inequality with $O(t^{3/2})$ instead of $O(t)$.

On ellipsoids. An ellipsoid E in \mathbb{R}^m is often defined as $\{x \in \mathbb{R}^m : x^T A x \leq 1\}$, where A is a positive definite matrix. Here we will mostly work with the *dual matrix* $D = A^{-1}$. Using this dual matrix we have (see, e.g., [42])

$$E = E(D) = \{z \in \mathbb{R}^m : z^T x \leq \sqrt{x^T D x} \text{ for all } x \in \mathbb{R}^m\}. \quad (4)$$

This definition can also be used for D only positive semidefinite; if D is singular, then $E(D)$ is a flat (lower-dimensional) ellipsoid.

2.2 Putting matrices side-by-side

► **Lemma 2.** *Let A, B be matrices, each with m rows, and let C be a matrix in which each column is a column of A or of B . Then*

$$\gamma_2(C)^2 \leq \gamma_2(A)^2 + \gamma_2(B)^2.$$

Proof. After possibly reordering the columns of C , we can write $C = \tilde{A} + \tilde{B}$, where the first k columns of \tilde{A} are among the columns of A and the remaining ℓ columns are zeros, and the last ℓ columns of \tilde{B} are among the columns of B and the first k are zeros.

Since the γ_2 norm is, by definition, monotone under the removal of columns, we have $a := \gamma_2(\tilde{A}) \leq \gamma_2(A)$, $b := \gamma_2(\tilde{B}) \leq \gamma_2(B)$.

Let $E_1 = E(D_1)$ and $E_2 = E(D_2)$ be ellipsoids witnessing $\gamma_2(\tilde{A})$ and $\gamma_2(\tilde{B})$, respectively. We claim that the ellipsoid $E(D_1 + D_2)$ contains all columns of \tilde{A} and also all columns of \tilde{B} . This is clear from the definition of the ellipsoid $E(D) = \{z : z^T x \leq \sqrt{x^T D x} \text{ for all } x\}$, since for every x , we have $x^T (D_1 + D_2)x = x^T D_1 x + x^T D_2 x \geq x^T D_1 x$ by positive semidefiniteness of D_2 . All the diagonal entries of D_1 are bounded above by a^2 , those of D_2 are at most b^2 , and hence $\|E\|_\infty \leq \sqrt{a^2 + b^2}$. ◀

► **Lemma 3.** *If C is a block-diagonal matrix with blocks A and B on the diagonal, then $\gamma_2(C) = \max(\gamma_2(A), \gamma_2(B))$.*

Proof. If D_1 is the dual matrix of the ellipsoid witnessing $\gamma_2(A)$ and similarly for D_2 and B , then the block-diagonal matrix D with blocks D_1 and D_2 on the diagonal defines an ellipsoid containing all columns of C . This is easy to check using the formula (4) defining $E(D)$ and the fact that a sum of positive definite matrices is positive definite. ◀

2.3 Dual formulation

Let $\|A\|_*$ denote the *nuclear norm* of a matrix A , which is the sum of the singular values of A (other names for $\|A\|_*$ are *Schatten 1-norm*, *trace norm*, or *Ky Fan n -norm*; see the text by Bhatia [9] for general background on symmetric matrix norms). Using a semidefinite formulation of γ_2 , and the duality theory for semidefinite programming, Lee, Shraibman and Špalek [26] derived a dual characterization of the γ_2 norm as a maximization problem.

► **Theorem 4** ([26, Thm. 9]). *We have*

$$\gamma_2(A) = \max\{\|P^{1/2} A Q^{1/2}\|_* : P, Q \text{ diagonal, nonnegative, } \text{Tr } P = \text{Tr } Q = 1\}.$$

Several times we will use this theorem with A a square matrix and $P = Q = \frac{1}{n} I_n$, in which case it gives $\gamma_2(A) \geq \frac{1}{n} \|A\|_*$.

2.4 Kronecker product

Let A be an $m \times n$ matrix and B a $p \times q$ matrix. We recall that the *Kronecker product* $A \otimes B$ is the following $mp \times nq$ matrix, consisting of $m \times n$ blocks of size $p \times q$ each:

$$\begin{pmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}B & a_{m2}B & \dots & a_{mn}B \end{pmatrix}$$

In [26] it was shown that γ_2 is multiplicative with respect to the Kronecker product:

► **Theorem 5** ([26, Thm. 17]). *For every two matrices A, B we have*

$$\gamma_2(A \otimes B) = \gamma_2(A) \cdot \gamma_2(B).$$

3 The γ_2 norm for intervals

In this section we deal with a particular example: the system \mathcal{I}_n of all initial segments $\{1, 2, \dots, i\}$, $i = 1, 2, \dots, n$, of $\{1, 2, \dots, n\}$. Its incidence matrix is T_n , the $n \times n$ matrix with 0s above the main diagonal and 1s everywhere else.

It is well known, and easy to see, that $\text{herdisc } T_n = 1$. We will prove that $\gamma_2(T_n)$ is of order $\log n$. This shows that the γ_2 norm can be $\log n$ times larger than the hereditary discrepancy, and thus the inequality (1) is asymptotically tight.

Moreover, this example is one of the key ingredients in the proof of the lower bound on the d -dimensional Tuszny problem.

► **Proposition 6.** *We have $\gamma_2(T_n) = \Theta(\log n)$.*

The upper bound follows from the observation $\text{herdisc } T_n = 1$ and the inequality (1) relating γ_2 to herdisc . It can also be proved directly using, for example, a decomposition into dyadic intervals. In the next section we prove the lower bound.

3.1 Lower bound on $\gamma_2(T_n)$

Proof of the lower bound in Proposition 6. The nuclear norm $\|T_n\|_*$ can be computed exactly (we are indebted to Alan Edelman and Gil Strang for this fact); namely, the singular values of T_n are

$$\frac{1}{2 \sin \frac{(2j-1)\pi}{4n+2}}, \quad j = 1, 2, \dots, n.$$

Using the inequality $\sin x \leq x$ for $x \geq 0$, we get

$$\gamma_2(T_n) \geq \frac{1}{n} \|T_n\|_* \geq \frac{2n+1}{\pi n} \sum_{j=1}^n \frac{1}{2j-1} = \Omega(\log n),$$

as needed.

The singular values of T_n can be obtained from the eigenvalues of the matrix $S_n := (T_n T_n^T)^{-1}$ which, as is not difficult to check, has the following simple tridiagonal form:

$$\begin{pmatrix} 2 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & -1 & 2 & -1 \\ 0 & 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{pmatrix}$$

(the 1 in the lower right corner is exceptional; the rest of the main diagonal are 2s). By general properties of eigenvalues and singular values, if $\lambda_1, \dots, \lambda_n$ are the eigenvalues of S_n , then the singular values of T_n are $\lambda_1^{-1/2}, \dots, \lambda_n^{-1/2}$. The eigenvalues of S_n are computed, as a part of more general theory, in Strang and MacNamara [45, Sec. 9]; the calculation is not hard to verify since they also give the eigenvectors explicitly.

One can also calculate the characteristic polynomial $p_n(x)$ of S_n : it satisfies the recurrence $p_{n+1} = (2-x)p_n - p_{n-1}$ with initial conditions $p_1 = 1-x$ and $p_0 = 1$, from which one can check that $p_n(x) = U_n\left(\frac{2-x}{2}\right) - U_{n-1}\left(\frac{2-x}{2}\right)$, where U_n is the degree- n Chebyshev polynomial of the second kind. The claimed roots of p_n can then be verified using the trigonometric representation of U_n . ◀

4 Deviation of the γ_2 norm from the hereditary discrepancy

Here we consider the inequalities (1) and (2) relating γ_2 and herdisc. For the first one we provide a simplified and elementary proof, and for the second one we briefly recall the proof and prove asymptotic optimality.

We have already seen in Section 3 that (1) is asymptotically tight. Let us first mention a simple but perhaps useful observation, which gives a somewhat weaker result.

There are examples of set systems $\mathcal{F}_1, \mathcal{F}_2$ on an n -point set X such that $|\mathcal{F}_1|, |\mathcal{F}_2| = O(n)$, herdisc \mathcal{F}_1 and herdisc \mathcal{F}_2 are bounded by a constant (actually by 1), and $\text{herdisc}(\mathcal{F}_1 \cup \mathcal{F}_2) = \Omega(\log n)$ [38, 34]. Therefore, no quantity obeying the triangle inequality (possibly up to a constant), such as the γ_2 norm, can approximate herdisc with a factor better than $\log n$.

4.1 The γ_2 norm is at most $\log m$ times the determinant lower bound

We establish the following inequalities relating the γ_2 norm to the determinant lower bound.

► **Theorem 7.** *For any $m \times n$ matrix A of rank r ,*

$$\text{detlb } A \leq \gamma_2(A) \leq O(\log r) \cdot \text{detlb } A.$$

Inequality (1) is an immediate consequence of the second inequality in the theorem (and of $r \leq \min\{m, n\}$):

$$\gamma_2(A) \leq O(\log \min\{m, n\}) \cdot \text{detlb } A \leq O(\log \min\{m, n\}) \text{herdisc } A,$$

where the last inequality uses the Lovász–Spencer–Vesztergombi bound $\text{herdisc } A \geq \frac{1}{2} \text{detlb } A$. In [35], inequality (1) was proved by using a sophisticated tool, the *restricted invertibility principle* of Bourgain and Tzafriri; see [13, 47]. Our proof of Theorem 7 is based only on elementary linear algebra and the determinant lower bound.

Before we prove Theorem 7, we need a lemma similar to an argument in [29].

► **Lemma 8.** *Let A be an $k \times n$ matrix, and let W be a nonnegative diagonal unit-trace $n \times n$ matrix. Then there exists a k -element set $J \subseteq [n]$ such that*

$$|\det A_J|^{1/k} \geq \sqrt{k/e} \cdot |\det AWA^T|^{1/2k}.$$

Proof of Theorem 7. For the inequality $\text{detlb } A \leq \gamma_2(A)$, we first observe that if B is a $k \times k$ matrix, then

$$|\det B|^{1/k} \leq \frac{1}{k} \|B\|_* \tag{5}$$

Indeed, the left-hand side is the geometric mean of the singular values of B , while the right-hand side is the arithmetic mean.

Now let B be a $k \times k$ submatrix of A with $\text{detlb } A = |\det B|^{1/k}$; then

$$\text{detlb } A = |\det B|^{1/k} \leq \frac{1}{k} \|B\|_* \leq \gamma_2(B) \leq \gamma_2(A).$$

For the second inequality $\gamma_2(A) \leq O(\log m) \cdot \text{detlb } A$, we compare $\det BB^T$ and the nuclear norm of B for a carefully chosen (rectangular) matrix B . First let P_0 and Q_0 be diagonal unit-trace matrices with $\gamma_2(A) = \|P_0^{1/2} A Q_0^{1/2}\|$ as in Theorem 4. For brevity, let us write $\tilde{A} := P_0^{1/2} A Q_0^{1/2}$, and let $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > 0$ be the nonzero singular values of \tilde{A} .

By a standard bucketing argument (see, e.g., [29, Lemma 7]), there is some $t > 0$ such that if we set $K := \{i \in [m] : t \leq \sigma_i < 2t\}$, then

$$\sum_{i \in K} \sigma_i \geq \Omega\left(\frac{1}{\log r}\right) \sum_{i=1}^m \sigma_i.$$

Let us set $k := |K|$.

Next, we define a suitable $k \times n$ matrix with singular values σ_i , $i \in K$. Let $\tilde{A} = U\Sigma V^T$ be the singular-value decomposition of \tilde{A} , with U and V orthogonal and Σ having $\sigma_1, \dots, \sigma_r$ on the main diagonal.

Let Π_K be the $k \times m$ matrix corresponding to the projection on the coordinates indexed by K ; that is, Π_K has 1s in positions $(1, i_1), \dots, (k, i_k)$, where $i_1 < \dots < i_k$ are the elements of K . The matrix $\Pi_K \Sigma = \Pi_K U^T \tilde{A} V = U_K^T \tilde{A} V$ has singular values σ_i , $i \in K$, and so does the matrix $U_K^T \tilde{A}$, since right multiplication by the orthogonal matrix V^T does not change the singular values.

This $k \times m$ matrix $U_K^T \tilde{A}$ is going to be the matrix B alluded to in the sketch of the proof idea above. We have

$$|\det BB^T|^{1/2k} = \left(\prod_{i \in K} \sigma_i\right)^{1/k} \geq \frac{1}{2k} \sum_{i \in K} \sigma_i = \Omega\left(\frac{1}{k \log r}\right) \gamma_2(A).$$

It remains to relate $\det BB^T$ to the determinant of a square submatrix of A , and this is where Lemma 8 is applied—actually applied twice, once for columns, and once for rows.

First we set $C := U_K^T P_0^{1/2} A$; then $B = C Q_0^{1/2}$. Applying Lemma 8 with C in the role of A and Q_0 in the role of W , we obtain a k -element index set $J \subseteq [n]$ such that

$$|\det C_J|^{1/k} \geq \sqrt{k/e} \cdot |\det BB^T|^{1/2k}.$$

Next, we set $D := P_0^{1/2} A_J$, and we claim that $\det D^T D \geq (\det C_J)^2$. Indeed, we have $C_J = U_K^T D$, and, since U is an orthogonal transformation, $(U^T D)^T (U^T D) = D^T D$. Then, by the Binet–Cauchy formula,

$$\begin{aligned} \det D^T D &= \det(U^T D)^T (U^T D) = \sum_L (\det U_L^T D)^2 \\ &\geq (\det U_K^T D)^2 = (\det C_J)^2. \end{aligned}$$

The next (and last) step is analogous. We have $D^T = A_J^T P_0^{1/2}$, and so we apply Lemma 8 with A_J^T in the role of A and P_0 in the role of W , obtaining a k -element subset $I \subseteq [m]$ with $|\det A_{I,J}|^{1/k} \geq \sqrt{k/e} \cdot |\det D^T D|^{1/2k}$ (where $A_{I,J}$ is the submatrix of A with rows indexed by I and columns by J).

Following the chain of inequalities backwards, we have

$$\begin{aligned} \det b A &\geq |\det A_{I,J}|^{1/k} \geq \sqrt{k/e} \cdot |\det D^T D|^{1/2k} \geq \sqrt{k/e} \cdot |\det C_J|^{1/k} \\ &\geq (k/e) |\det BB^T|^{1/2k} = \Omega\left(\frac{1}{\log r}\right) \gamma_2(A), \end{aligned}$$

and the theorem is proved. \blacktriangleleft

4.2 The hereditary discrepancy can be $\sqrt{\log m}$ times larger than γ_2

Next, we show that $\sqrt{\log m}$ in inequality (2) cannot be replaced by any asymptotically smaller factor.

► **Theorem 9.** For all m , there are $m \times n$ matrices A , with $n = \Theta(\log m)$, such that

$$\text{herdisc } A \geq \Omega(\sqrt{\log m}) \cdot \gamma_2(A).$$

Proof. A very simple example is the incidence matrix A of the system of all subsets of $[n]$, with $m = 2^n$, whose discrepancy is $n/2 = \Theta(\log m)$. Indeed, the characteristic vectors of all sets fit into the ball of radius \sqrt{n} , and hence $\gamma_2(A) = \gamma_2(A^T) \leq \sqrt{n} = O(\sqrt{\log m})$, where we used the fact that γ_2 is invariant under transposition. ◀

5 On Tusnády’s problem

Proof of Theorem 1. The proof was already sketched in the introduction, so here we just present it slightly more formally. Let $\mathcal{A}_d \subseteq \mathcal{R}_d$ be the set of all *anchored* axis-parallel boxes, of the form $[0, b_1] \times \dots \times [0, b_d]$. Clearly $\text{disc}(n, \mathcal{A}_d) \leq \text{disc}(n, \mathcal{R}_d)$, and since every box $R \in \mathcal{R}_d$ can be expressed as a signed combination of at most 2^d anchored boxes, we have $\text{disc}(n, \mathcal{R}_d) \leq 2^d \text{disc}(n, \mathcal{A}_d)$.

Let us consider the d -dimensional grid $[n]^d \subset \mathbb{R}^d$ (with n^d points), and let $\mathcal{G}_{d,n} = \mathcal{A}_d([n]^d)$ be the subsets induced on it by anchored boxes. It suffices to prove that $\text{herdisc } \mathcal{G}_{d,n} = \Omega(\log^{d-1} n)$, and for this, in view of inequality (1), it is enough to show that $\gamma_2(\mathcal{G}_{d,n}) = \Omega(\log^d n)$.

Now $\mathcal{G}_{d,n}$ is (isomorphic to) the d -fold product \mathcal{I}_n^d of the system of initial segments in $\{1, 2, \dots, n\}$, and so $\gamma_2(\mathcal{G}_{d,n}) = \gamma_2(\mathcal{I}_n^d) = \Theta(\log^d n)$ (Theorem 5 and Proposition 6).

This finishes the proof of the lower bound. To prove the upper bound $\text{disc}(n, \mathcal{R}_d) = O(\log^{d+1/2} n)$, we consider an arbitrary n -point set $P \subset \mathbb{R}^d$. Since the set system $\mathcal{A}_d(P)$ is not changed by a monotone transformation of each of the coordinates, we may assume $P \subseteq [n]^d$. Hence

$$\text{disc}(\mathcal{A}_d(P)) \leq \text{herdisc } \mathcal{G}_{d,n} \leq O(\gamma_2(\mathcal{G}_{d,n}) \sqrt{\log n^d}) = O(\log^{d+1/2} n).$$

◀

5.1 Discrepancy of boxes in high dimension

Chazelle and Lvov [16, 17] investigated the hereditary discrepancy of the set system $\mathcal{C}_d := \mathcal{R}_d(\{0, 1\}^d)$, the set system induced by axis-parallel boxes on the d -dimensional Boolean cube $\{0, 1\}^d$. In other words, the sets in \mathcal{C}_d are subcubes of $\{0, 1\}^d$. Unlike for Tusnády’s problem where d was considered fixed, here one is interested in the asymptotic behavior as $d \rightarrow \infty$.

Chazelle and Lvov proved $\text{herdisc } \mathcal{C}_d = \Omega(2^{cd})$ for an absolute constant $c \approx 0.0477$, which was later improved to $c = 0.0625$ in [36] (in relation to the hereditary discrepancy of homogeneous arithmetic progressions). Here we obtain an optimal value of the constant c :

► **Theorem 10.** The system \mathcal{C}_d of subcubes of the d -dimensional Boolean cube satisfies

$$\text{herdisc } \mathcal{C}_d = 2^{c_0 d + o(d)},$$

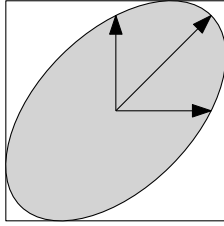
where $c_0 = \log_2(2/\sqrt{3}) \approx 0.2075$. The same bound holds for the system $\mathcal{A}_d(\{0, 1\}^d)$ of all subsets of the cube induced by anchored boxes.

Proof. The number of sets in \mathcal{C}_d is 3^d , and so in view of inequalities (1) and (2) it suffices to prove $\gamma_2(\mathcal{C}_d) = \gamma_2(\mathcal{A}_d(\{0, 1\}^d)) = 2^{c_0 d}$.

The system \mathcal{C}_d is the d -fold product \mathcal{C}_1^d , and so by Theorem 5, $\gamma_2(\mathcal{C}_d) = \gamma_2(\mathcal{C}_1)^d$. The incidence matrix of \mathcal{C}_1 is

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

To get an upper bound on $\gamma_2(A)$, we exhibit an appropriate ellipsoid; it is more convenient to do it for A^T , since this is a planar problem. The optimal ellipse containing the rows of A is $\{x \in \mathbb{R}^2 : x_1^2 + x_2^2 - x_1x_2 \leq 1\}$; here are a picture and the dual matrix:



$$D = \begin{pmatrix} \frac{4}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{4}{3} \end{pmatrix}.$$

Hence $\gamma_2(A) \leq 2/\sqrt{3}$. The same ellipse also works for the incidence matrix of the system $\mathcal{A}_1(\{0, 1\})$, which is the familiar lower triangular matrix T_2 .

There are several ways of bounding $\gamma_2(T_2) \leq \gamma_2(A)$ from below. For example, we can use Theorem 4 with

$$P = \begin{pmatrix} \frac{1}{3} & 0 \\ 0 & \frac{2}{3} \end{pmatrix}, \quad Q = \begin{pmatrix} \frac{2}{3} & 0 \\ 0 & \frac{1}{3} \end{pmatrix}.$$

With some effort (or a computer algebra system) one can check that the singular values of $P^{1/2}T_2Q^{1/2}$ are $\frac{1}{\sqrt{3}} \pm \frac{1}{3}$, and hence the nuclear norm is $2/\sqrt{3}$ as needed.

Alternatively, one can also check the optimality of the ellipse above by elementary geometry, or exhibit an optimal solution of the dual semidefinite program for $\gamma_2(T_2)$. ◀

Other set systems. In the full version of the paper we use the properties of γ_2 to give new simple proofs of other upper and lower bounds in discrepancy theory. In particular, we revisit two set systems that have been studied extensively: arithmetic progressions in $[n]$ and intervals of k permutations on $[n]$. While the bounds we get are slightly suboptimal, the proofs are very short.

6 Applications in Computer Science

Range searching in the oblivious group model. A range searching problem is defined by a system \mathcal{F} of subsets of a set $P \subseteq \mathbb{R}^d$. The input is an assignment of weights to P , where each weight is an element of a commutative group; a query is specified by a range $F \in \mathcal{F}$ may ask, for example, whether for the sum of the weights of points in F or whether it is non-zero. The goal is to maintain a data structure that supports fast queries. One of the best studied special cases is orthogonal range searching, in which \mathcal{F} is induced by axis-aligned boxes, i.e. $\mathcal{F} = \mathcal{R}_d(P)$.

Following Fredman [22] and Larsen [25], we define an oblivious data structure for a range searching problem given by \mathcal{F} as a factorization $A = BC$, where $A \in \{0, 1\}^{m \times n}$ is the incidence matrix of \mathcal{F} , and B, C are integer matrices. The update time t_u is defined as the maximum number of non-zero entries of a column of C , and the query time t_q is the

maximum number of non-zero entries of a row of B . The multiplicity Δ is the maximum absolute value of an entry in B or C . The motivation is that the actual data structure kept in memory is $y = Cx$, where x are the weights assigned to P , and queries are answered by computing the appropriate entry of By . Then, updating a single weight requires updating at most t_u cells in the data structure, and answering a query requires reading at most t_q cells.

By the factorization definition of γ_2 , we have that for any oblivious data structure for \mathcal{F} , $\gamma_2(\mathcal{F}) \leq |\Delta| \sqrt{t_u t_q}$. In the proof of Theorem 1 we showed that for $\mathcal{G}_{d,n} = \mathcal{A}_d([n]^d)$ (recall \mathcal{A}_d is the set of axis-aligned boxes anchored at 0), $\gamma_2(\mathcal{G}_{d,n}) = \Theta((\log n)^d)$. Therefore, for any oblivious data structure for orthogonal range searching on P with constant multiplicity, $t_u t_q = \Omega((\log n)^d)$. This lower bound is tight up to constants. The best previous lower bound was due to Larsen [25] and was on the order of $(\log n)^{(d-1)/2}$.

Differential Privacy. Differential privacy is a popular definition of privacy for data analysis algorithms. Informally, it states that an algorithm is private if its output distribution is almost the same when we add or remove one person's data from the input; see the book [21] for the formal definition. A class of problems of general interest in differential privacy are counting problems, in which a database is a multiset of elements of a universe U , and a family of queries is specified by a system \mathcal{F} of subsets of U . A query given by a set $F \in \mathcal{F}$ asks for the number of elements of F that are in the database D (counted with multiplicity). In [37] it was shown that, up to factors logarithmic in $|\mathcal{F}|$, the optimal worst-case error for answering the queries specified by \mathcal{F} is equal to $\gamma_2(\mathcal{F})$. A query set of special interest is the one given by the set system \mathcal{C}_d of subcubes of the d -dimensional boolean cube, which corresponds to the set of marginal queries on a d -dimensional database. For these queries, Theorem 10 shows that the optimal worst-case error is on the order of $2^{c_0 d \pm o(d)}$, where $c_0 = \log_2(2/\sqrt{3})$. The best previous upper bound was $2^{d/2 + o(d)}$.

Acknowledgments. We would like to thank Alan Edelman and Gil Strang for invaluable advice concerning the singular values of the matrix in Proposition 6, and Van Vu for recommending the right experts for this question. We would also like to thank Noga Alon and Assaf Naor for pointing out that the geometric quantity in [35, 32] is equivalent to the γ_2 norm. We also thank Imre Bárány and Vojtěch Tůma for useful discussions.

References

- 1 T. van Aardenne-Ehrenfest. Proof of the impossibility of a just distribution of an infinite sequence of points. *Nederl. Akad. Wet., Proc.*, 48:266–271, 1945. Also in *Indag. Math.* 7, 71–76 (1945).
- 2 T. van Aardenne-Ehrenfest. On the impossibility of a just distribution. *Nederl. Akad. Wet., Proc.*, 52:734–739, 1949. Also in *Indag. Math.* 11, 264–269 (1949).
- 3 J. R. Alexander, J. Beck, and W. W. L. Chen. Geometric discrepancy theory and uniform distribution. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 10, pages 185–207. CRC Press LLC, Boca Raton, FL, 1997.
- 4 N. Bansal. Constructive algorithms for discrepancy minimization. <http://arxiv.org/abs/1002.2259>, also in *FOCS'10: Proc. 51st IEEE Symposium on Foundations of Computer Science*, pages 3–10, 2010.
- 5 J. Beck. Balanced two-colorings of finite sets in the square. I. *Combinatorica*, 1:327–335, 1981.
- 6 J. Beck. Balanced two-colorings of finite sets in the cube. *Discrete Mathematics*, 73:13–25, 1989.

- 7 J. Beck. A two-dimensional van Aardenne-Ehrenfest theorem in irregularities of distribution. *Compositio Math.*, 72:269–339, 1989.
- 8 J. Beck and W. W. L. Chen. *Irregularities of Distribution*. Cambridge University Press, Cambridge, 1987.
- 9 Rajendra Bhatia. *Matrix analysis*, volume 169 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1997.
- 10 D. Bilyk and M. T. Lacey. On the small ball inequality in three dimensions. *Duke Math. J.*, 143(1):81–115, 2008.
- 11 D. Bilyk, M. T. Lacey, and A. Vagharshakyan. On the small ball inequality in all dimensions. *J. Funct. Anal.*, 254(9):2470–2502, 2008.
- 12 G. Bohus. On the discrepancy of 3 permutations. *Random Struct. Algo.*, 1:215–220, 1990.
- 13 J. Bourgain and L. Tzafriri. Invertibility of large submatrices with applications to the geometry of banach spaces and harmonic analysis. *Israel journal of mathematics*, 57(2):137–224, 1987.
- 14 M. Charikar, A. Newman, and A. Nikolov. Tight hardness results for minimizing discrepancy. In *Proc. 22nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), San Francisco, California, USA*, pages 1607–1614, 2011.
- 15 B. Chazelle. *The Discrepancy Method*. Cambridge University Press, Cambridge, 2000.
- 16 B. Chazelle and A. Lvov. A trace bound for the hereditary discrepancy. *Discrete Comput. Geom.*, 26(2):221–231, 2001.
- 17 B. Chazelle and A. Lvov. The discrepancy of boxes in higher dimension. *Discrete Comput. Geom.*, 25(4):519–524, 2001.
- 18 J. G. van der Corput. Verteilungsfunktionen I. *Akad. Wetensch. Amsterdam, Proc.*, 38:813–821, 1935.
- 19 J. G. van der Corput. Verteilungsfunktionen II. *Akad. Wetensch. Amsterdam, Proc.*, 38:1058–1066, 1935.
- 20 M. Drmota and R. F. Tichy. *Sequences, discrepancies and applications (Lecture Notes in Mathematics 1651)*. Springer-Verlag, Berlin etc., 1997.
- 21 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
- 22 Michael L. Fredman. The complexity of maintaining an array and computing its partial sums. *J. ACM*, 29(1):250–260, 1982.
- 23 J. H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numer. Math.*, 2:84–90, 1960.
- 24 J. M. Hammersley. Monte Carlo methods for solving multivariable problems. *Ann. New York Acad. Sci.*, 86:844–874, 1960.
- 25 K. G. Larsen. On range searching in the group model and combinatorial discrepancy. *SIAM Journal on Computing*, 43(2):673–686, 2014.
- 26 Troy Lee, Adi Shraibman, and Robert Špalek. A direct product theorem for discrepancy. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity, CCC 2008, 23-26 June 2008, College Park, Maryland, USA*, pages 71–80. IEEE Computer Society, 2008.
- 27 Nati Linial, Shahar Mendelson, Gideon Schechtman, and Adi Shraibman. Complexity measures of sign matrices. *Combinatorica*, 27(4):439–463, 2007.
- 28 L. Lovász, J. Spencer, and K. Vesztegombi. Discrepancy of set-systems and matrices. *European J. Combin.*, 7:151–160, 1986.
- 29 J. Matoušek. The determinant bound for discrepancy is almost tight. *Proc. Amer. Math. Soc.*, 141(2):451–460, 2013.
- 30 J. Matoušek. On the discrepancy for boxes and polytopes. *Monatsh. Math.*, 127(4):325–336, 1999.

- 31 J. Matoušek. *Geometric Discrepancy (An Illustrated Guide)*, 2nd printing. Springer-Verlag, Berlin, 2010.
- 32 Jiří Matoušek and Aleksandar Nikolov. Combinatorial discrepancy for boxes via the ellipsoid-infinity norm. To appear in SoCG 15., 2014.
- 33 S. Muthukrishnan and A. Nikolov. Optimal private halfspace counting via discrepancy. In *STOC '12: Proceedings of the 44th symposium on Theory of Computing*, pages 1285–1292, New York, NY, USA, 2012. ACM.
- 34 A. Newman, O. Neiman, and A. Nikolov. Beck's three permutations conjecture: A counterexample and some consequences. In *Proc. 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 253–262, 2012.
- 35 A. Nikolov and K. Talwar. Approximating hereditary discrepancy via small width ellipsoids. Preprint arXiv:1311.6204, 2013.
- 36 A. Nikolov and K. Talwar. On the hereditary discrepancy of homogeneous arithmetic progressions. *To Appear in Proceedings of AMS*, 2013.
- 37 A. Nikolov, K. Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *Proc. 45th ACM Symposium on Theory of Computing (STOC), Palo Alto, California, USA*, pages 351–360, 2013. Full version to appear in *SIAM Journal on Computing as The Geometry of Differential Privacy: the Small Database and Approximate Cases*.
- 38 D. Pálvölgyi. Indecomposable coverings with concave polygons. *Discrete Comput. Geom.*, 44(3):577–588, 2010.
- 39 K. F. Roth. On irregularities of distribution. *Mathematika*, 1:73–79, 1954.
- 40 Thomas Rothvoß. Constructive discrepancy minimization for convex sets. *CoRR*, abs/1404.0339, 2014. To Appear in FOCS 2014.
- 41 W. M. Schmidt. On irregularities of distribution VII. *Acta Arith.*, 21:45–50, 1972.
- 42 A. Seeger. Calculus rules for combinations of ellipsoids and applications. *Bull. Australian Math. Soc.*, 47(01):1–12, 1993.
- 43 J. Spencer. *Ten Lectures on the Probabilistic Method*. CBMS-NSF. SIAM, Philadelphia, PA, 1987.
- 44 A. Srinivasan. Improving the discrepancy bound for sparse matrices: better approximations for sparse lattice approximation problems. In *Proc. 8th ACM-SIAM Symposium on Discrete Algorithms*, pages 692–701, 1997.
- 45 G. Strang and S. MacNamara. Functions of difference matrices are Toeplitz plus Hankel. *SIAM Review*, 2014. To appear.
- 46 Nicole Tomczak-Jaegermann. *Banach-Mazur distances and finite-dimensional operator ideals*, volume 38 of *Pitman Monographs and Surveys in Pure and Applied Mathematics*. Longman Scientific & Technical, Harlow; copublished in the United States with John Wiley & Sons, Inc., New York, 1989.
- 47 R. Vershynin. John's decompositions: Selecting a large part. *Israel Journal of Mathematics*, 122(1):253–277, 2001.

Tilt: The Video – Designing Worlds to Control Robot Swarms with Only Global Signals

Aaron T. Becker¹, Erik D. Demaine², Sándor P. Fekete³,
Hamed Mohtasham Shad¹, and Rose Morris-Wright¹

- 1 Department of Electrical and Computer Engineering, University of Houston
Houston, TX 77004, USA
atbecker@uh.edu
- 2 CSAIL, MIT
Cambridge, MA 02139, USA
edemaine@mit.edu
- 3 Department of Computer Science, TU Braunschweig
38106 Braunschweig, Germany
s.fekete@tu-bs.de

Abstract

We present fundamental progress on the computational universality of swarms of micro- or nano-scale robots in complex environments, controlled not by individual navigation, but by a uniform global, external force. More specifically, we consider a 2D grid world, in which all obstacles and robots are unit squares, and for each actuation, robots move maximally until they collide with an obstacle or another robot. The objective is to control robot motion within obstacles, design obstacles in order to achieve desired permutation of robots, and establish controlled interaction that is complex enough to allow arbitrary computations. In this video, we illustrate progress on all these challenges: we demonstrate NP-hardness of parallel navigation, we describe how to construct obstacles that allow arbitrary permutations, and we establish the necessary logic gates for performing arbitrary in-system computations.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems – Geometrical problems and computations, F.1.1 Models of Computation–Bounded-action devices

Keywords and phrases Particle swarms, global control; complexity, geometric computation

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.16

1 Introduction: Global Motion Control

One of the exciting new directions of robotics is the design and development of micro- and nanorobot systems, with the goal of letting a massive swarm of robots perform complex operations in a complicated environment. Due to scaling issues, individual control of the involved robots becomes physically impossible: while energy storage capacity drops with the third power of robot size, medium resistance decreases much slower. A possible answer lies in applying a global, external force to all particles in the swarm. This is what many current micro- and nanorobot systems with many robots do: the whole swarm is steered and directed by an external force that acts as a common control signal; see our paper [8] for detailed references. These common control signals include global magnetic or electric fields, chemical gradients, and turning a light source on and off.

Clearly, having only one global signal that uniformly affects all robots at once poses a strong restriction on the ability of the swarm to perform complex operations. The only



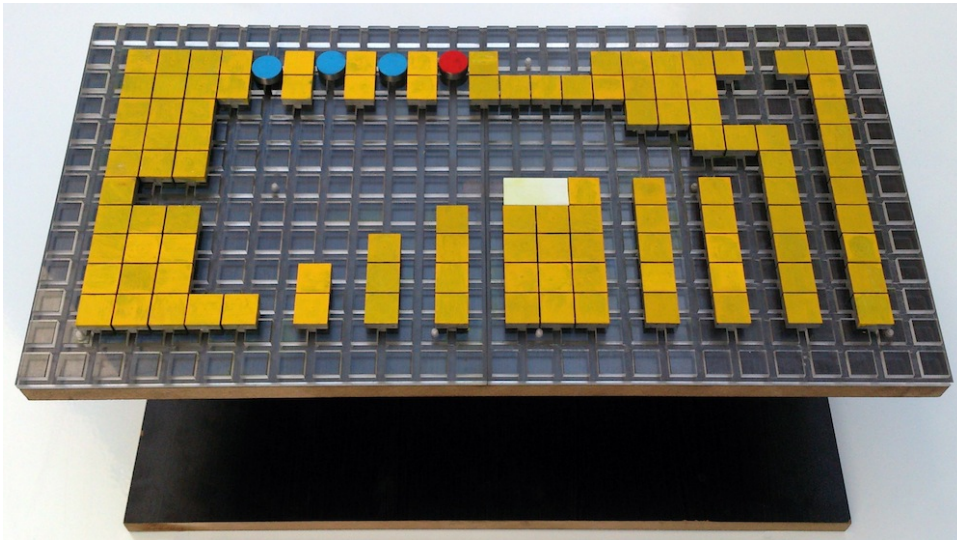
© Aaron T. Becker, Erik D. Demaine, Sándor P. Fekete, Hamed Mohtasham Shad, Rose Morris-Wright;
licensed under Creative Commons License CC-BY
31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 16–18



Leibniz International Proceedings in Informatics

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



■ **Figure 1** Gravity-fed hardware implementation of particle computation. The reconfigurable prototype is set up as a FAN-OUT gate using a 2×1 robot (white).

hope for breaking symmetry is to use interactions between the robot swarm and obstacles in the environment. The key challenge is to establish if interactions with obstacles are sufficient to perform complex operations, ideally by analyzing the complexity of possible logical operations.

This resembles the logic puzzle Tilt [9], and dexterity ball-in-a-maze puzzles such as Pigs in Clover and Labyrinth, which involve tilting a board to cause all mobile pieces to roll or slide in a desired direction. Problems of this type are also similar to sliding-block puzzles with fixed obstacles [3, 5, 6, 7], except that all particles receive the same control inputs, as in the Tilt puzzle. Another connection is Randolph’s Ricochet Robots [4], a game that allows individual and independent control of the involved particles.

2 The Problems

We consider a two-dimensional grid world, with some cells occupied and others free. Initially, the planar square grid is filled with some unit-square particles (each occupying a cell of the grid) and some fixed unit-square blocks. All particles are commanded in unison: a valid command is “Go Up” (u), “Go Right” (r), “Go Down” (d), or “Go Left” (l). All particles move in the commanded direction until they hit an obstacle or another particle. A representative command sequence is $\langle u, r, d, l, d, r, u, \dots \rangle$. We call these global commands *force-field moves*. We assume we can bound the minimum particle speed and can guarantee all particles have moved to their maximum extent.

Three of the most basic problems are as follows.

1. *Given a map of an environment, along with initial and goal positions for each particle, does there exist a sequence of inputs that will bring each particle to its goal position?*
2. *Given an initial matrix arrangement of particles, how can we design a set of obstacles, such that any permutation can be realized with a relatively simple sequence of moves?*
3. *Can we establish sets of obstacles, particles, and moves, such that the resulting motion can be used for carrying out arbitrary computation strictly within the system, i.e., without an intelligent observer?*

We have provided answers for these problems in our previous papers [2, 8, 1]. Here we present a compact visual demonstration, in part based on a real-world realization, showing that further applications and extensions are possible.

3 The Video

The video consists of a number of animation sequences, as well as several scenes demonstrating real-world model environments.

In the first part of the video, we describe the underlying model, based on a physical realization, and motivate the background from micro- and nano-robotics. We then proceed to sketch the elements and overall construction for an NP-hardness proof, resolving one aspect of the complexity of the first problem. (A separate argument shows that the problem of minimizing the number of moves for achieving a target configuration is in fact PSPACE-complete, but this is omitted.) In the third part of the video, we demonstrate how to solve the second problem: We can design relatively simple sets of obstacles that allow arbitrary matrix permutations, based on simple clockwise and counterclockwise subsequences of moves. Finally, the fourth and last part shows some of the key components for carrying out universal computation, demonstrated on a physical model for simple components, and animations for overall construction.

References

- 1 Aaron T. Becker, Erik D. Demaine, Sándor P. Fekete, Golnaz Habibi, and James McLurkin. Reconfiguring massive particle swarms with limited, global control. In *9th International Symposium on Algorithms and Experiments for Sensor Systems, Wireless Networks and Distributed Robotics (ALGOSENSORS)*, volume 8343 of *Springer LNCS*, pages 51–66, 2013.
- 2 Aaron T. Becker, Erik D. Demaine, Sándor P. Fekete, and James McLurkin. Particle computation: Designing worlds to control robot swarms with only global signals. In *2014 IEEE International Conference on Robotics and Automation, ICRA 2014, Hong Kong, China, May 31 - June 7, 2014*, pages 6751–6756, 2014.
- 3 Erik D. Demaine, Martin L. Demaine, and Joseph O’Rourke. PushPush and Push-1 are NP-hard in 2D. In *Proceedings of the 12th Annual Canadian Conference on Computational Geometry (CCCG)*, pages 211–219, August 2000.
- 4 Birgit Engels and Tom Kamphans. Randolphs robot game is NP-hard! *Electronic Notes in Discrete Mathematics*, 25:49–53, 2006.
- 5 Robert A. Hearn and Erik D. Demaine. PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *arXiv:cs/0205005*, cs.CC/0205005, 2002.
- 6 Michael Hoffmann. Motion planning amidst movable square blocks: Push-* is NP-hard. In *Canadian Conference on Computational Geometry*, pages 205–210, June 2000.
- 7 Markus Holzer and Stefan Schwoon. Assembling molecules in ATOMIX is hard. *Theoretical Computer Science*, 313(3):447–462, 2004.
- 8 Hahmed Mohtasham Shad, Rose Morris-Wright, Erik D. Demaine, Sándor P. Fekete, and Aaron T. Becker. Particle computation: Device fan-out and binary memory. In *2015 IEEE International Conference on Robotics and Automation, ICRA Seattle, USA, May 26 - 30, 2015*, page to appear, 2015.
- 9 ThinkFun. Tilt: Gravity fed logic maze. <http://www.thinkfun.com/tilt>.

Automatic Proofs for Formulae Enumerating Proper Polycubes

Gill Barequet and Mira Shalah

Department of Computer Science
The Technion – Israel Institute of Technology
Haifa 32000, Israel
{barequet,mshalah}@cs.technion.ac.il

Abstract

This video describes a general framework for computing formulae enumerating polycubes of size n which are proper in $n-k$ dimensions (i.e., spanning all $n-k$ dimensions), for a fixed value of k . (Such formulae are central in the literature of statistical physics in the study of percolation processes and collapse of branched polymers.) The implemented software re-affirmed the already-proven formulae for $k \leq 3$, and proved rigorously, for the first time, the formula enumerating polycubes of size n that are proper in $n-4$ dimensions.

1998 ACM Subject Classification G.2.1 Combinatorics, G.2.2 Graph Labeling

Keywords and phrases Polycubes, inclusion-exclusion

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.19

1 Introduction

A d -dimensional polycube of size n is a connected set of n cubes in d dimensions, where connectivity is through $(d-1)$ -dimensional faces. A polycube is said to be *proper* in d dimensions if the convex hull of the centers of its cubes is d -dimensional. Following Lunnun [8], let $\text{DX}(n, d)$ denote the number of polycubes of size n that are proper in d dimensions.

Enumeration of polycubes and computing their asymptotic growth rate are important problems in combinatorics and discrete geometry, originating in statistical physics [5]. Polycubes (*polyominoes* in 2D) play a fundamental role in statistical physics in the analysis of percolation processes and collapse of branched polymers. To-date, no formula is known for $A_d(n)$, the number of polycubes of size n in d dimensions, for any value of d , let alone in the general case. The main interest in DX stems from the formula $A_d(n) = \sum_{i=0}^d \binom{d}{i} \text{DX}(n, i)$ [8]. In a matrix listing the values of DX , the top-right triangular half and the main diagonal contain only 0s. This gives rise to the question of whether a pattern can be found in the sequences $\text{DX}(n, n-k)$, where $k < n$ is the ordinal number of the diagonal.

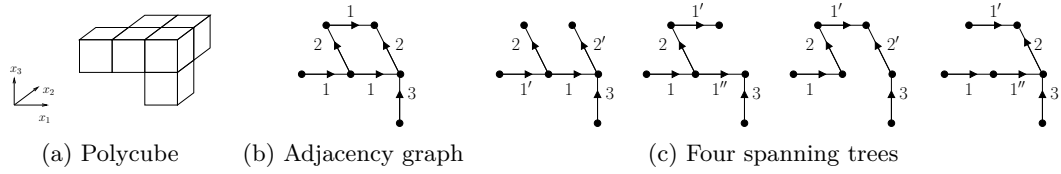
Klarner [6] showed that the limit $\lambda_2 = \lim_{n \rightarrow \infty} \sqrt[n]{A_2(n)}$ exists. Much later Madras [10] proved the convergence of the sequence $(A_2(n+1)/A_2(n))_{n=1}^{\infty}$ to λ_2 (a similar claim holds in any dimension d). Thus, λ_2 is the *growth rate* limit of polyominoes. Its exact value has remained elusive till these days. The best known lower and upper bounds on λ_2 are roughly 4.0025 [2] and 4.6496 [7], respectively. Significant progress in estimating λ_d has been obtained in statistical physics, although the computations usually relied on unproven assumptions and on formulae for $\text{DX}(n, n-k)$ interpolated empirically from known values of $A_d(n)$. Peard and Gaunt [12] predicted that for $k > 1$, the diagonal formula $\text{DX}(n, n-k)$ has the pattern $2^{n-2k+1} n^{n-2k-1} (n-k) h_k(n)$, where $h_k(n)$ is a polynomial in n , and conjectured formulae for $h_k(n)$ for $k \leq 6$. Luther and Mertens [9] conjectured a formula for $k = 7$.



© Gill Barequet and Mira Shalah;
licensed under Creative Commons License CC-BY
31st International Symposium on Computational Geometry (SoCG'15).
Editors: Lars Arge and János Pach; pp. 19–22



Leibniz International Proceedings in Informatics
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



■ **Figure 1** A polycube P , the corresponding graph $\gamma(P)$, and spanning trees of $\gamma(P)$.

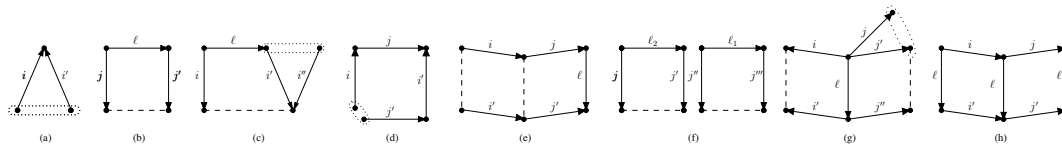
It is easy to show that $\text{DX}(n, n-1) = 2^{n-1}n^{n-3}$ (seq. A127670 in OEIS [11]). Barequet et al. [4] proved rigorously that $\text{DX}(n, n-2) = 2^{n-3}n^{n-5}(n-2)(2n^2-6n+9)$ (seq. A171860). The proof uses a case analysis of the possible structures of spanning trees of the polycubes, and the various ways in which cycles can be formed in their cell-adjacency graphs. Similarly, Asinowski et al. [1] proved that $\text{DX}(n, n-3) = 2^{n-6}n^{n-7}(n-3)(12n^5-104n^4+360n^3-679n^2+1122n-1560)/3$, again, by counting spanning trees of polycubes, yet the reasoning and the calculations were significantly more involved. The inclusion-exclusion principle was applied in order to count correctly polycubes whose cell-adjacency graphs contained certain subgraphs, so-called “distinguished structures.” In comparison with $k=2$, the number of such structures is substantially higher, and the ways in which they can appear in spanning trees are much more varied. The latter proof provided a better understanding of the difficulties that one would face in applying this technique to higher values of k . The number of distinguished structures grows rapidly, and their inclusion relations are much more complicated. As anticipated, it is impractical to achieve a similar proof manually for $k > 3$.

In this video we describe a theoretical set-up [3] for proving the formula for $\text{DX}(n, n-k)$, for a fixed k . Using our implementation of this method, we could prove the following theorem.

► **Theorem 1.** $\text{DX}(n, n-4) = 2^{n-7}n^{n-9}(n-4)(8n^8-128n^7+828n^6-2930n^5+7404n^4-17523n^3+41527n^2-114302n+204960)/6$.

2 Method

Denote by \mathcal{P}_n the set of polycubes of size n proper in $n-k$ dimensions. Let $P \in \mathcal{P}_n$, and let $\gamma(P)$ denote the directed edge-labeled graph that is constructed as follows: The vertices of $\gamma(P)$ correspond to the cells of P ; two vertices of $\gamma(P)$ are connected by an edge if the corresponding cells of P are adjacent; and an edge has label i ($1 \leq i \leq n-k$) if the corresponding cells have different i -coordinate. The direction of the edge is from the lower to the higher cell (see Figure 1). Since $P \mapsto \gamma(P)$ is an injection, it suffices to count the graphs obtained from the members of \mathcal{P}_n in this way. We count these graphs by counting their spanning trees. A spanning tree of $\gamma(P)$ has $n-1$ edges labeled by numbers from the set $\{1, 2, \dots, n-k\}$; all these labels are present, otherwise the polycube is not proper in $n-k$ dimensions. Hence, $n-k$ edges of the spanning tree are labeled with the labels $1, 2, \dots, n-k$, and the remaining $k-1$ edges are labeled with repeated labels from the same set. There is a bijection between the possibilities of repeated edge labels and the partitions of $k-1$. Specifically, each partition $p = \sum_{i=1}^h a_i \in \Pi(k-1)$ corresponds to h repeated labels in the spanning tree, such that the i th repeated label appears a_i+1 times. In such case, we say that the tree is *labeled according to p* . When we consider a spanning tree of $\gamma(P)$, we distinguish a repeated label i that appears r times by $i, i', \dots, i'^{(r-1)}$. However, when considering $\gamma(P)$, repeated labels are assumed not to be distinguished. Every repeated label must occur an even number of times in any cycle of $\gamma(P)$. In addition, the number of cycles in $\gamma(P)$ and the length of each such cycle are bounded from above due to the limited multiplicity of labels.



■ **Figure 2** (a–g) A few distinguished structures for $k = 4$ (note that (f) is disconnected); (h) A cycle structure. A dotted line is drawn between every pair of neighboring cells and around every pair of coinciding cells.

In order to compute $|\mathcal{P}_n|$, we consider all possible directed edge-labeled trees of size n with edge labels as conjectured, and count only those that represent valid polycubes. In this process two things may happen:

(a) Cells may coincide (Figures 2(a,d)). A tree with overlapping cells is invalid; and
 (b) Two cells which are not connected by a tree edge may be adjacent (Figures 2(b,e)). Such a tree corresponds to a polycube P with cycles in $\gamma(P)$, hence, its spanning tree is not unique. In order to count correctly, we consider small structures (Figure 2), contained in these trees, which cause the problems above. The counting involves a delicate inclusion-exclusion analysis of the structures. See the video and [3] for more details.

3 The Video

The video illustrates the framework described above. First, it defines polycubes and explains what “proper polycubes” are. Then, it describes the importance of polycubes in combinatorics, discrete geometry, and statistical physics. The video then turns to defining $\text{DX}(n, n - k)$ and showing how it is computed automatically, using examples from the case $k = 4$. The video displays a few lemmas and formulas, defines *distinguished structures*, shows how they are generated, and explains the inclusion-exclusion graph built to obtain the sought formula. Finally, the video presents the results obtained by our program.

The video was produced on a 2.53GHz DELL 64 processor PC with 4GB of RAM. The animations were designed using the Autodesk Maya 2015 (student version) modeling software and Microsoft PowerPoint 2010. The video was constructed by Windows Live Movie Maker.

References

- 1 A. Asinowski, G. Barequet, R. Barequet, and G. Rote, Proper n -cell polycubes in $n-3$ dimensions, *J. of Integer Sequences*, 15 (2012), #12.8.4.
- 2 G. Barequet, G. Rote, and M. Shalah, $\lambda > 4$, *30th European Workshop on Computational Geometry*, Ein-Gedi, Israel, March 2014.
- 3 G. Barequet and M. Shalah, Automatic proofs for formulae enumerating proper polycubes, *Proc. 31st European Workshop on Computational Geometry*, Ljubljana, Slovenia, 4 pp., March 2015.
- 4 R. Barequet, G. Barequet, and G. Rote, Formulae and growth rates of high-dimensional polycubes, *Combinatorica*, 30 (2010), 257–275.
- 5 S. R. Broadbent and J. M. Hammersley, Percolation processes: I. Crystals and mazes, *Proc. Cambridge Philosophical Society*, 53 (1957), 629–641.
- 6 D. A. Klarner, Cell growth problems, *Canadian J. of Mathematics*, 19 (1967), 851–863.
- 7 D. A. Klarner and R. L. Rivest, A procedure for improving the upper bound for the number of n -ominoes, *Canadian J. of Mathematics*, 25 (1973), 585–602.

- 8 W. F. Lunnon, Counting multidimensional polyominoes, *The Comp. Journal*, 18 (1975), 366–367.
- 9 S. Luther and S. Mertens, Counting lattice animals in high dimensions, *J. of Statistical Mechanics: Theory and Experiment*, 9 (2011), 546–565.
- 10 N. Madras, A pattern theorem for lattice clusters, *Annals of Combinatorics*, 3 (1999), 357–384.
- 11 *The On-Line Encyclopedia of Integer Sequences OEIS*, available at <http://oeis.org>
- 12 P. J. Peard and D. S. Gaunt, $1/d$ -expansions for the free energy of lattice animal models of a self-interacting branched polymer, *J. Physics, A: Mathematical and General*, 28 (1995), 6109–6124.

Visualizing Sparse Filtrations*

Nicholas J. Cavanna, Mahmoodreza Jahanseir, and
Donald R. Sheehy

Department of Computer Science, University of Connecticut
Storrs, CT, USA
nicholas.j.cavanna@uconn.edu, {reza,donald}@engr.uconn.edu

Abstract

Over the last few years, there have been several approaches to building sparser complexes that still give good approximations to the persistent homology [5, 4, 3, 2, 1]. In this video, we have illustrated a geometric perspective on sparse filtrations that leads to simpler proofs, more general theorems, and a more visual explanation. We hope that as these techniques become easier to understand, they will also become easier to use.

1998 ACM Subject Classification F.2.2 Geometrical problems and computations, G.2.1 Combinatorial algorithms

Keywords and phrases Topological Data Analysis, Simplicial Complexes, Persistent Homology

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.23

1 The Shape of Data

Topological data analysis is concerned with finding the underlying shape of a data set. Often, a union of balls called offsets is used to approximate the shape and fill in the space between points. The topology of the balls can be represented as a simplicial complex called a nerve. Every subset of balls with a common intersection contributes one simplex to the nerve: an edge for each pairwise intersection, a triangle for each 3-way intersection, etc. (see Fig. 1).

Instead of looking at just one radius, we can look at the offsets at all radii from zero to infinity. A growing space like this is called a filtration. The nerves give a corresponding simplicial filtration. Persistent homology is a way to study the changes in topology over the course of a filtration. The output is called a persistent barcode and marks the components, holes, and voids as well as their lifespans. The Nerve Theorem and its persistent variant guarantee that the barcode for the offsets is the same as that of the nerve filtration.

Nerve complexes get very big very fast, even when restricting to subsets of constant size. A common variant that doesn't assume Euclidean metrics is the Rips complex and it suffers similar difficulties.

At larger scales, fewer points are needed to give a good approximation. The sparser subsample of our point set at scale α is obtained by calculating an ε -net, i.e. a subset where each pair is at least ε apart and the ε -radius balls centered on the points of the net cover the input. However, removing points cause the nerve of balls to no longer be a filtration, because a filtration is, by definition a monotonely growing space. For a simplicial complex, this means that simplices appear, but never disappear. We solve this problem by viewing the offset filtration as a nerve of objects one dimension higher as illustrated below.

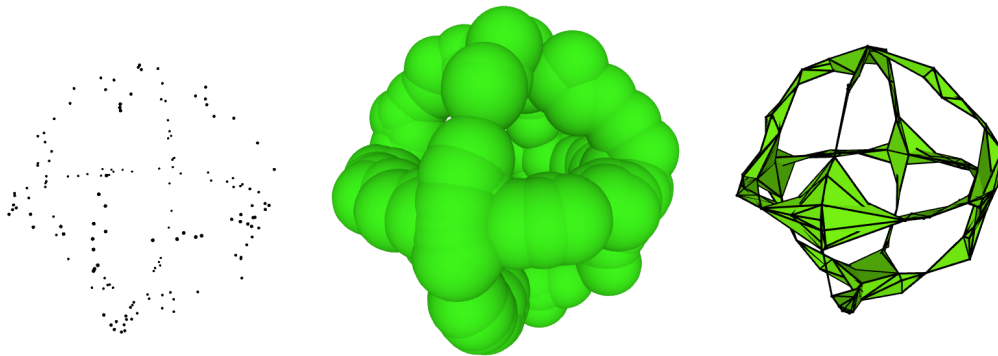
If we visualize the scale parameter as another dimension, a growing ball traces out a cone (below, left). This cone is modified in two ways. First, we assign a maximum radius to each

* Partially supported by the National Science Foundation under grant number CCF-1464379.



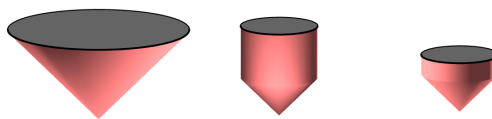


■ **Figure 1** The nerve has an edge for each pairwise intersection, a triangle for each 3-way intersection (right), etc.

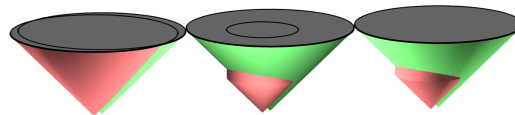


■ **Figure 2** A point set sampled on a sphere, its offsets, and its (sparsified) nerve complex.

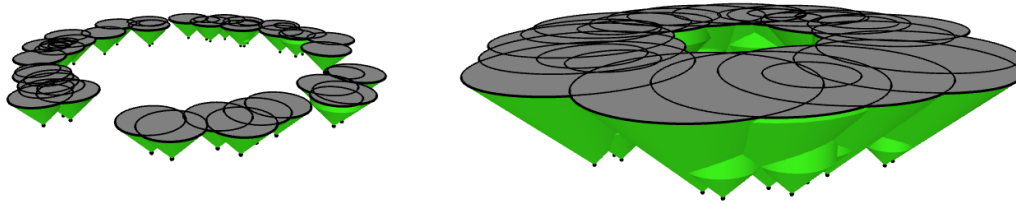
point (middle). Next, we truncate the cone at the height when the point will be removed from the filtration (right). Truncating the cone simulates the removal of the corresponding ball. The cone has not been removed, it just no longer intersects the time slices above the “removal” time.



The maximum radius and height are chosen so that the top of the cone is sure to be covered at the time it is removed.



These cones form a new filtration one dimension higher. Their nerve is the desired sparse filtration.



In this example, one can imagine flattening all the cones onto one level set, and since all the cones are stacked on each other, there won't be any loss of homological information. The sublevel sets of the cones are homotopy equivalent to the level sets, which implies they have the same homology. By the persistent nerve lemma we know that the nerve has the same persistent homology as the sublevel sets, thus we can calculate the persistent homology of our sparsified offsets by computing the persistent homology of the sparse nerve filtration.

The sparsification algorithm can cut down the asymptotic size of filtrations from polynomial to linear, while still achieving a close approximation to the persistence diagram. The video aims to provide a simple, geometric explanation for the topological guarantees of such sparse filtrations. It avoids the complexity of zig-zag inclusions maps from previous work by considering time as an extra spatial dimension. The construction and its analysis easily generalizes to Rips and other related complexes, and although the example input was two-dimensional, the construction works in any number of dimensions.

2 Production

In order to create the video, we used Processing to create the visualizations, iMovie to piece together the soundtrack and the visualizations, and Javaplex through Matlab to calculate the barcodes. We thank Julia Sheehy for supplying the narration.

References

- 1 Magnus Bakke Botnan and Gard Spreemann. Approximating persistent homology in Euclidean space through collapses. *Applicable Algebra in Engineering, Communication and Computing*, pages 1–29, 2015.
- 2 Mickaël Buchet, Frédéric Chazal, Steve Y. Oudot, and Donald Sheehy. Efficient and robust persistent homology for measures. In *ACM-SIAM Symposium on Discrete Algorithms*, 2015.
- 3 Tamal K. Dey, Fengtao Fan, and Yusu Wang. Computing topological persistence for simplicial maps. In *Proceedings of the 30th Annual Symposium on Computational Geometry*, pages 345–354, 2014.
- 4 Michael Kerber and R. Sharathkumar. Approximate Čech complexes in low and high dimensions. In *24th International Symposium on Algorithms and Computation (ISAAC 2013)*, volume LNCS 8283, pages 666–676, 2013.
- 5 Donald R. Sheehy. Linear-size approximations to the Vietoris-Rips filtration. *Discrete & Computational Geometry*, 49(4):778–796, 2013.

Visualizing Quickest Visibility Maps

Topi Talvitie

Department of Computer Science, University of Helsinki, Finland

Abstract

Consider the following modification to the shortest path query problem in polygonal domains: instead of finding shortest path to a query point q , we find the shortest path to any point that sees q . We present an interactive visualization applet visualizing these *quickest visibility paths*. The applet also visualizes *quickest visibility maps*, that is the subdivision of the domain into cells by the quickest visibility path structure.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases path planning, visibility

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.26

1 Introduction

Finding shortest paths within a polygonal domain is a classical problem in computational geometry. The query version of the shortest path problem for a fixed point s is

Shortest Path Query: Given a point q , how should we move from s to reach q ?

The problem of shortest path queries is solved by building a *shortest path map* for s , defined as the decomposition of the domain into cells such that the shortest paths to all points q within that cell is the same (the only changing vertex is the endpoint q). Consider a modification of this problem, where we only need to see the query point:

Quickest Visibility Query: Given a point q , how should we move from s to see q ?

This kind of query would be natural in applications where it is important to only see or become seen by the target point, for example for inspecting the query point or establishing communication with it. Quickest visibility queries were first studied in the case of simple polygons in [2]. This visualization accompanies the paper [1] which presents algorithms for the general case of polygons with holes and improves the results in the case of simple polygons. The visualization applet demonstrates the core concepts of quickest visibility queries: quickest visibility paths, quickest visibility maps and visibility wave propagation. These are briefly outlined below. For more formal definitions and proofs please refer to [1].

2 Quickest visibility paths

A *quickest visibility path* (QVP) from s to q is the shortest path from s to some endpoint t such that q is visible from t . It is possible that q is directly visible from s . In that case, the quickest visibility path is the path of length zero from s to s (Fig. 1a). If t lies in the interior of the domain, the path always enters t orthogonally to line tq , because otherwise we could adjust the location of t to shorten the path (Fig. 1b). Otherwise t is either in a vertex of a polygon (Fig. 1c) or on an edge of a polygon such that tq contains a polygon vertex (Fig. 1d).



© Topi Talvitie;

licensed under Creative Commons License CC-BY

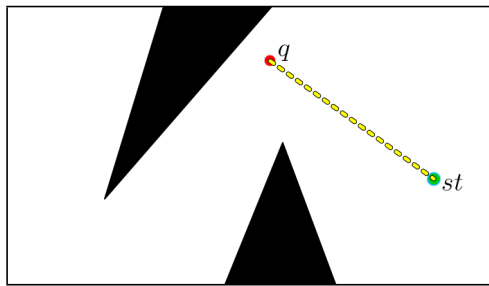
31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 26–28

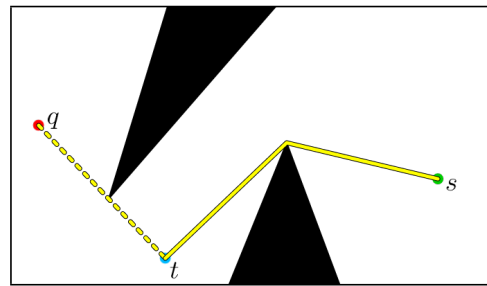


Leibniz International Proceedings in Informatics

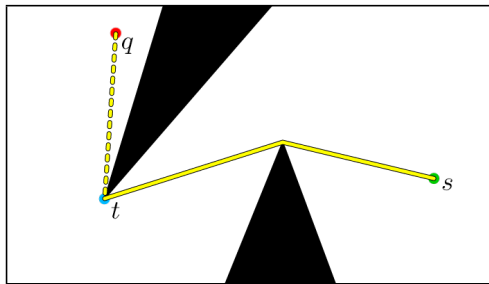
LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



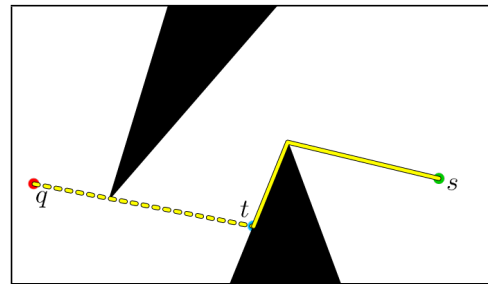
(a) q is directly visible from s . Therefore $s = t$ and the QVP has length 0.



(b) t lies in free space. The QVP and tq meet orthogonally in t .



(c) t lies on a polygon vertex.



(d) t lies on a polygon edge. Segment tq touches a polygon vertex.

■ **Figure 1** The four different types of quickest visibility paths from s to q . The quickest visibility path is drawn as a solid yellow line.

3 Quickest visibility maps

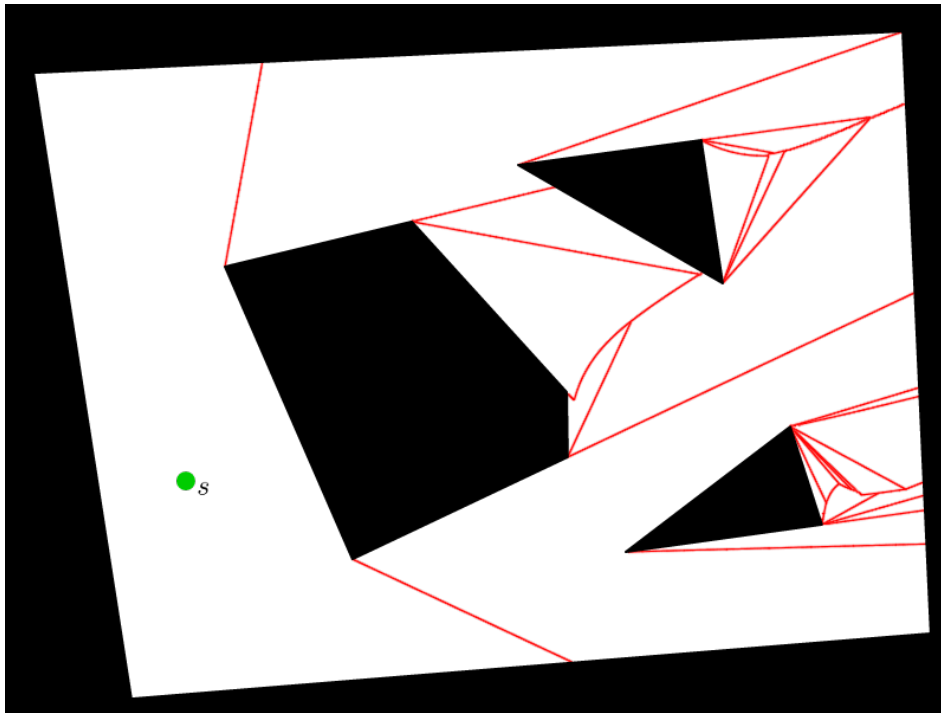
A *quickest visibility map* (QVM) for a polygonal domain and a source point s is the subdivision of the domain into cells such that for all points q within a cell, the QVP from s to q has the same structure. We say that two QVPs have the same structure if the paths differ only in the last point q , and in the case of paths of type shown in Fig. 1b, the second-to-last point t that moves when point q moves. See Fig. 2 for an example of a QVM.

Once a QVM has been built for source point s , one can query quickest visibility paths from s to any q by using point location query data structures to find the QVM cell q lies in. Therefore QVM is the analogue of shortest paths maps in the case of quickest visibility paths.

4 Implementation

The visualization applet consists of two parts: the backend library, implemented in C++, and the user interface, implemented in JavaScript.

The backend implements an algorithm for finding the quickest visibility path to a given query point. It does not use quickest visibility maps, as exact QVM construction would very complicated to implement. Instead, it handles quickest visibility paths of types shown in figures 1a, 1c and 1d by querying shortest paths using the visibility graph. That leaves only visibility paths of type where the path endpoint lies in free space (Fig. 1b), which can be found by iterating all polygon vertices v visible to q , and all points from which the extension of vg is orthogonally visible. This can be implemented simultaneously for all v as two ray



■ **Figure 2** A screenshot from the visualization applet, showing the subdivision of the polygonal domain by the red lines into the quickest visibility map for source point s .

sweeps around q to both directions in $O(n \log n)$ time with the help of precomputed shortest paths to all polygon vertices.

The user interface contains a polygon editor, in which the user can edit the polygon and set the source point s . It visualizes quickest visibility paths using the backend library. It draws the QVM by querying the quickest visibility paths to a dense grid of points in the domain, drawing points of the QVM edges in locations where adjacent grid query points have different quickest visibility paths. A local optimization algorithm is used to improve the precision of the QVM edges.

The visualization is a client-side HTML5/JavaScript applet. The backend library is compiled from C++ to JavaScript using the Emscripten compiler. The precomputation used for drawing the QVM edges is parallelized using Web Workers, which makes the loading phase faster in multi-core systems. The applet should work on all modern browsers, including the newest versions of all major web browsers.

Acknowledgments. The research was supported by the University of Helsinki Research Funds.

References

- 1 E. M. Arkin, A. Efrat, C. Knauer, J. S. B. Mitchell, V. Polishchuk, G. Rote, L. Schlipf, and T. Talvitie. Shortest path to a segment and quickest visibility queries. In *SoCG*, 2015.
- 2 R. Khosravi and M. Ghodsi. The fastest way to view a query point in simple polygons. In *European Workshop on Computational Geometry*, pages 187–190. Technische Universiteit Eindhoven, 2005.

Sylvester-Gallai for Arrangements of Subspaces

Zeev Dvir¹ and Guangda Hu²

- 1 Department of Computer Science and Department of Mathematics,
Princeton University
35 Olden Street, Princeton, NJ 08540-5233, USA
zeev.dvir@gmail.com
- 2 Department of Computer Science, Princeton University
35 Olden Street, Princeton, NJ 08540-5233, USA
guangdah@cs.princeton.edu

Abstract

In this work we study arrangements of k -dimensional subspaces $V_1, \dots, V_n \subset \mathbb{C}^\ell$. Our main result shows that, if every pair V_a, V_b of subspaces is contained in a dependent triple (a triple V_a, V_b, V_c contained in a $2k$ -dimensional space), then the entire arrangement must be contained in a subspace whose dimension depends only on k (and not on n). The theorem holds under the assumption that $V_a \cap V_b = \{0\}$ for every pair (otherwise it is false). This generalizes the Sylvester-Gallai theorem (or Kelly's theorem for complex numbers), which proves the $k = 1$ case. Our proof also handles arrangements in which we have many pairs (instead of all) appearing in dependent triples, generalizing the quantitative results of Barak et. al. [1].

One of the main ingredients in the proof is a strengthening of a theorem of Barthe [3] (from the $k = 1$ to $k > 1$ case) proving the existence of a linear map that makes the angles between pairs of subspaces large on average. Such a mapping can be found, unless there is an obstruction in the form of a low dimensional subspace intersecting many of the spaces in the arrangement (in which case one can use a different argument to prove the main theorem).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Sylvester-Gallai, Locally Correctable Codes

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.29

1 Introduction

The Sylvester-Gallai (SG) theorem states that for n points $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathbb{R}^\ell$, if for every pair $\mathbf{v}_i, \mathbf{v}_j$ there is a third point \mathbf{v}_k on the line passing through $\mathbf{v}_i, \mathbf{v}_j$, then all points must lie on a single line. This was first posed by Sylvester [14], and was solved by Melchior [13]. It was also conjectured independently by Erdős [9] and proved shortly after by Gallai. We refer the reader to the survey [4] for more information about the history and various generalizations of this theorem. The complex version of this theorem was proved by Kelly [11] (see also [8, 7] for alternative proofs) and states that if $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathbb{C}^\ell$ and for every pair $\mathbf{v}_i, \mathbf{v}_j$ there is a third \mathbf{v}_k on the same complex line, then all points are contained in some complex plane (over the complex numbers, there are planar examples and so this theorem is tight).

In [7] (based on earlier work in [1]), the following quantitative variant of the SG theorem was proved. For a set $S \subset \mathbb{C}^\ell$ we denote by $\dim(S)$ the smallest d such that S is contained in a d -dimensional subspace of \mathbb{C}^ℓ .

► **Theorem 1.1** ([7]). *Given n points $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n \in \mathbb{C}^\ell$, if for every $i \in [n]$ there exists at least δn values of $j \in [n] \setminus \{i\}$ such that the line through \mathbf{v}_i and \mathbf{v}_j contains a third point \mathbf{v}_k , then $\dim\{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n\} \leq 10/\delta$.*



© Zeev Dvir and Guangda Hu;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 29–43

Leibniz International Proceedings in Informatics



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

(The dependence on δ is asymptotically tight). From here on, we will work with homogeneous subspaces (passing through zero) instead of affine subspaces (lines/planes etc). The difference is not crucial to our results and the affine version can always be derived by intersecting with a generic hyperplane. In this setting, the above theorem will be stated for a set of one-dimensional subspaces, each spanned by some \mathbf{v}_i (and no two \mathbf{v}_i 's being a multiple of each other) and collinearity of $\mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k$ is replaced with the three vectors being linearly dependent (i.e., contained in a 2-dimensional subspace).

One natural high dimensional variant of the SG theorem, studied in [10, 1], replaces 3-wise dependencies with t -wise dependencies (e.g, every triple is in some coplanar four-tuple). In this work, we raise another natural high-dimensional variant in which the *points* themselves are replaced with k -dimensional subspaces. We consider such arrangements with many 3-wise dependencies (defined appropriately) and attempt to prove that the entire arrangement lies in some low dimensional space. We will consider arrangements $V_1, \dots, V_n \subset \mathbb{C}^\ell$ in which each V_i is k -dimensional and with each pair satisfying $V_{i_1} \cap V_{i_2} = \{\mathbf{0}\}$. A dependency can then be defined as a triple $V_{i_1}, V_{i_2}, V_{i_3}$ of k -dimensional subspaces that are contained in a single $2k$ -dimensional subspace. The pair-wise zero intersections guarantee that every pair of subspaces defines a unique $2k$ -dimensional space (their span) and so, this definition of dependency behaves in a similar way to collinearity. For example, we have that if $V_{i_1}, V_{i_2}, V_{i_3}$ are dependent and $V_{i_2}, V_{i_3}, V_{i_4}$ are dependent then also $V_{i_1}, V_{i_2}, V_{i_4}$ are dependent. This would not hold if we allowed some pairs to have non zero intersections. In fact, if we allow non-zero intersection then we can construct an arrangement of two dimensional spaces with many dependent triples and with dimension as large as \sqrt{n} (see below). We now state our main theorem, generalizing Theorem 1.1 (with slightly worse parameters) to the case $k > 1$. We use the standard $V + U$ notation to denote the subspace spanned by all vectors in $V \cup U$. We use big 'O' notation to hide absolute constants.

► **Theorem 1.2.** *Let $V_1, V_2, \dots, V_n \subset \mathbb{C}^\ell$ be k -dimensional subspaces such that $V_i \cap V_{i'} = \{\mathbf{0}\}$ for all $i \neq i' \in [n]$. Suppose that, for every $i_1 \in [n]$ there exists at least δn values of $i_2 \in [n] \setminus \{i_1\}$ such that $V_{i_1} + V_{i_2}$ contains some V_{i_3} with $i_3 \notin \{i_1, i_2\}$. Then*

$$\dim(V_1 + V_2 + \dots + V_n) = O(k^4/\delta^2).$$

The condition $V_i \cap V_{i'} = \{\mathbf{0}\}$ is needed due to the following example. Set $k = 2$ and $n = \ell(\ell - 1)/2$ and let $\{\mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_\ell\}$ be the standard basis of \mathbb{R}^ℓ . Define the n spaces to be $V_{ij} = \text{span}\{\mathbf{e}_i, \mathbf{e}_j\}$ with $1 \leq i < j \leq \ell$. Now, for each $(i, j) \neq (i', j')$ the sum $V_{ij} + V_{i'j'}$ will contain a third space (since the size of $\{i, j, i', j'\}$ is at least three). However, this arrangement has dimension $\ell > \sqrt{n}$.

The bound $O(k^4/\delta^2)$ is probably not tight and we conjecture that it could be improved to $O(k/\delta)$, possibly with a modification of our proof. One can always construct an arrangement with dimension $2k/\delta$ by partitioning the subspaces into $1/\delta$ groups, each contained in a single $2k$ dimensional space.

Overview of the proof: A preliminary observation is that it suffices to prove the theorem over \mathbb{R} . This is because an arrangement of k -dimensional complex subspaces can be translated into an arrangement of $2k$ -dimensional real subspaces (this is proved at the end of Section 2). Hence, we will now focus on real arrangements.

The proof of the theorem is considerably simpler when the arrangement of subspaces V_1, \dots, V_n satisfies an extra 'robustness' condition, namely that every two spaces have an angle bounded away from zero. More formally, if for every two unit vectors $\mathbf{v}_1 \in V_{i_1}$ and $\mathbf{v}_2 \in V_{i_2}$ we have $|\langle \mathbf{v}_1, \mathbf{v}_2 \rangle| \leq 1 - \tau$ for some absolute constant $\tau > 0$. This condition implies

that, when we have a dependency of the form $V_{i_3} \subset V_{i_1} + V_{i_2}$, every unit vector in V_{i_3} can be obtained as a linear combination *with bounded coefficients* (in absolute value) of unit vectors from V_{i_1}, V_{i_2} . Fixing an orthogonal basis for each subspace and using the conditions of the theorem, we are able to construct many local linear dependencies between the basis elements. We then show (using the bound on the coefficients in the linear combinations) that the space of linear dependencies between all basis vectors, considered as a subspace of \mathbb{R}^{kn} , contains the rows of an $nk \times nk$ matrix that has large entries on the diagonal and small entries off the diagonal. Since matrices of this form have high rank (by a simple spectral argument), we conclude that the original set of basis vectors must have small dimension.

To handle the general case, we show that, unless some low dimensional subspace W intersects many of the spaces V_i in the arrangement, we can find a change of basis that makes the angles between the spaces large on average (in which case, the previous argument works). This gives us the overall strategy of the proof: If such a W exists, we project W to zero and continue by induction. The loss in the overall dimension is bounded by the dimension of W , which can be chosen to be small enough. Otherwise (if such W does not exist) we apply the change of basis and use it to bound the dimension.

The change of basis is found by generalizing a theorem of Barthe [3] (see [6] for a more accessible treatment) from the $k = 1$ case (arrangement of points) to higher dimension. We state this result here since we believe it could be of independent interest. To state the theorem we must first introduce the following, somewhat technical, definition.

► **Definition 1.3** (admissible basis set, admissible basis vector). Given a list of vector spaces $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$), a set $H \subseteq [n]$ is called a \mathcal{V} -admissible basis set if

$$\dim\left(\sum_{i \in H} V_i\right) = \sum_{i \in H} \dim(V_i) = \dim\left(\sum_{i \in [n]} V_i\right),$$

i.e. if every space with index in H has intersection $\{0\}$ with the span of the other spaces with indices in H , and the spaces with indices in H span the entire space $\sum_{i \in [n]} V_i$.

A \mathcal{V} -admissible basis vector is any indicator vector $\mathbf{1}_H$ of some \mathcal{V} -admissible basis set H (where the i -th entry of $\mathbf{1}_H$ equals 1 if $i \in H$ and 0 otherwise).

The following theorem is proved in Section 3.

► **Theorem 1.4.** Given a list of vector spaces $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$) with $V_1 + V_2 + \dots + V_n = \mathbb{R}^\ell$ and a vector $\mathbf{p} \in \mathbb{R}^n$ in the convex hull of all \mathcal{V} -admissible basis vectors. Then there exists an invertible linear map $M : \mathbb{R}^\ell \mapsto \mathbb{R}^\ell$ such that

$$\sum_{i=1}^n p_i \text{Proj}_{M(V_i)} = I_{\ell \times \ell},$$

where $M(V_i)$ is the linear space obtained by applying M on V_i , and $\text{Proj}_{M(V_i)}$ is the orthogonal projection matrix onto $M(V_i)$.

The connection to the explanation given in the proof overview is as follows: If there is no subspace W of low dimension that intersects many of the spaces V_1, \dots, V_n then, one can show that there exists a vector \mathbf{p} in the convex hull of all \mathcal{V} -admissible basis vectors such that the entries of \mathbf{p} are not too small. This is enough to show that the average angle between pairs of spaces is large since otherwise one can derive a contradiction to the inequality which says that the sum of orthogonal projections of any unit vector must be relatively small.

The proof of the one dimensional case in [3] proceeds by defining a strictly convex function $f(t_1, \dots, t_m)$ on \mathbb{R}^m and shows that the function is bounded. This means that there must

exist a point in which all partial derivatives of f vanish. Solving the resulting equations gives an invertible matrix that defines the required change of basis. We follow a similar strategy, defining an appropriate bounded function $f(t_1, \dots, t_m, R_1, \dots, R_n)$ in more variables, where the extra variables R_1, \dots, R_n represent the action of the orthogonal group $\mathbf{O}(k)$ on each of the spaces. However, in our case, we cannot show that f is strictly convex and so a maximum might not exist. However, we are still able to show that there exists a point in which all partial derivatives are very small (smaller than any $\epsilon > 0$), which is sufficient for our purposes.

Connection to Locally Correctable Codes. A q -query Locally Correctable Code (LCC) over a field \mathbb{F} is a d -dimensional subspace $C \subset \mathbb{F}^n$ that allows for ‘local correction’ of codewords (elements of C) in the following sense. Let $\mathbf{y} \in C$ and suppose we have query access to \mathbf{y}' such that $\mathbf{y}_i = \mathbf{y}'_i$ for at least $(1 - \delta)n$ indices $i \in [n]$ (think of \mathbf{y}' as a noisy version of \mathbf{y}). Then, for every i , we can probabilistically pick q positions in \mathbf{y}' and, from their (possibly incorrect values), recover the correct value of \mathbf{y}_i with high probability (over the choice of queries). LCC’s play an important role in theoretical computer science (mostly over finite fields but recently also over the reals, see [5]) and are still poorly understood. In particular, when q is constant greater than 2, there are exponential gaps between the dimension of explicit constructions and the proven upper bounds. In [2] it was observed that q -LCCs are essentially equivalent to configurations of points with many local dependencies¹. A variant of Theorem 1.1 shows for example that the maximal dimension of a 2-LCC in \mathbb{R}^n has dimension bounded by $(1/\delta)^{O(1)}$. Our results can be interpreted in this framework as dimension upper bounds for 2-query LCC’s in which each coordinate is replaced by a ‘block’ of k coordinates. Our results then show that, even under this relaxation, the dimension still cannot increase with n . The case of 3-query LCC’s over the reals is still wide open (some modest progress was made recently in [6]) and we hope that the methods developed in this work could lead to further progress on this tough problem.

Organization. In Section 2, we define the notion of (α, δ) -systems (which generalizes the SG condition) and reduce our k -dimensional Sylvester-Gallai theorem to a more general theorem, Theorem 2.6, on the dimension of (α, δ) -systems (this part also includes the reduction from complex to real arrangements). Then, in Section 3, we prove the generalization of Barthe’s theorem (Theorem 1.4). Finally, in Section 4, we prove our main result regarding (α, δ) -systems. Due to the page limit, some of the proof are available in the full version of this paper.

Acknowledgements. We would like to thank Patrick Devlin for helpful discussions on strengthening Theorem 1.4.

2 Reduction to (α, δ) -systems

The notion of an (α, δ) -system is used to ‘organize’ the dependent triples in the arrangement in a more convenient form so that each space is in many triples and every pair of spaces is together only in a few dependent triples. We also allow dependent *pairs* as those might arise when we apply a linear map on the arrangement.

¹ One important difference is that LCC’s give rise to configurations where each point can repeat more than once.

► **Definition 2.1** ((α, δ) -system). Given a list of vector spaces $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$), we call a list of sets $\mathcal{S} = (S_1, S_2, \dots, S_w)$ an (α, δ) -system of \mathcal{V} ($\alpha \in \mathbb{Z}^+, \delta > 0$) if

1. Every S_j is a subset of $[n]$ of size either 3 or 2.
2. If S_j contains 3 elements i_1, i_2 and i_3 , then $V_{i_1} \subseteq V_{i_2} + V_{i_3}$, $V_{i_2} \subseteq V_{i_1} + V_{i_3}$ and $V_{i_3} \subseteq V_{i_1} + V_{i_2}$. If S_j contains 2 elements i_1 and i_2 , then $V_{i_1} = V_{i_2}$.
3. Every $i \in [n]$ is contained in at least δn sets of \mathcal{S} .
4. Every pair $\{i_1, i_2\}$ ($i_1 \neq i_2 \in [n]$) appears together in at most α sets of \mathcal{S} .

Note that we allow $\delta > 1$ in an (α, δ) -systems. This is different from the statement of the Sylvester-Gallai theorem where $\delta \in [0, 1]$. We have the following 3 simple observations, which are proved in the full version of this paper.

► **Lemma 2.2.** *Let $\mathcal{S} = (S_1, S_2, \dots, S_w)$ be an (α, δ) -system of some vector space list \mathcal{V} . Then $\delta n^2/3 \leq w \leq \alpha n^2/2$ and $\delta/\alpha \leq 3/2$.*

► **Lemma 2.3.** *Let $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$) be a list of vector spaces and $\mathcal{S} = (S_1, S_2, \dots, S_w)$ be a list of sets. If $w \geq \delta n^2$ and \mathcal{S} satisfies the first, second and fourth requirements in Definition 2.1, then there exists a sublist \mathcal{V}' of \mathcal{V} and a sublist \mathcal{S}' of \mathcal{S} such that $|\mathcal{V}'| \geq \delta n/(2\alpha)$ and \mathcal{S}' is an $(\alpha, \delta/2)$ -system of \mathcal{V}' .*

► **Lemma 2.4.** *Let $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$) be a list of vector spaces with an (α, δ) -system $\mathcal{S} = (S_1, S_2, \dots, S_w)$. Then for any linear map $P : \mathbb{R}^\ell \mapsto \mathbb{R}^\ell$, \mathcal{S} is also an (α, δ) -system of $\mathcal{V}' = (V'_1, V'_2, \dots, V'_n)$, where $V'_i = P(V_i)$.*

Theorem 1.2, will be derived from the following, more general statement, saying that the dimension d is small if there is a (α, δ) -system.

► **Definition 2.5** (k -bounded). A vector space $V \subseteq \mathbb{R}^\ell$ is k -bounded if $\dim V \leq k$.

► **Theorem 2.6.** *Let $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$) be a list of k -bounded vector spaces with an (α, δ) -system and $d = \dim(V_1 + V_2 + \dots + V_n)$, then $d = O(\alpha^2 k^4 / \delta^2)$.*

We can easily reduce the high dimensional Sylvester-Gallai problem in \mathbb{C}^ℓ (Theorem 1.2) to the setting of Theorem 2.6 in \mathbb{R}^ℓ as shown below.

Proof of Theorem 1.2 using Theorem 2.6. Let $B_j = \{\mathbf{v}_{j1}, \mathbf{v}_{j2}, \dots, \mathbf{v}_{jk}\}$ be a basis of V_j . Define

$$V'_j = \text{span} \{ \text{Re}(\mathbf{v}_{j1}), \text{Re}(\mathbf{v}_{j2}), \dots, \text{Re}(\mathbf{v}_{jk}), \text{Im}(\mathbf{v}_{j1}), \text{Im}(\mathbf{v}_{j2}), \dots, \text{Im}(\mathbf{v}_{jk}) \} \quad \forall j \in [n].$$

► **Claim 2.7.** $V'_j = \{ \text{Re}(\mathbf{v}) : \mathbf{v} \in V_j \}$ for every $j \in [n]$.

Proof. For every $\mathbf{v}' \in V'_j$, there exist $\lambda_1, \lambda_2, \dots, \lambda_k, \mu_1, \mu_2, \dots, \mu_k \in \mathbb{R}$ such that

$$\begin{aligned} \mathbf{v}' &= \sum_{s=1}^k \left(\lambda_s \text{Re}(\mathbf{v}_{js}) + \mu_s \text{Im}(\mathbf{v}_{js}) \right) = \sum_{s=1}^k \left(\lambda_s \text{Re}(\mathbf{v}_{js}) + \mu_s \text{Re}(-i\mathbf{v}_{js}) \right) \\ &= \text{Re} \left(\sum_{s=1}^k (\lambda_s - i\mu_s) \mathbf{v}_{js} \right). \end{aligned}$$

Since $\lambda_1, \lambda_2, \dots, \lambda_k, \mu_1, \mu_2, \dots, \mu_k$ can take all values in \mathbb{R} , we can see the claim is proved. ◀

► **Claim 2.8** ([1, Lemma 2.1]). *Given a set A with $r \geq 3$ elements, we can construct a family of $r^2 - r$ triples of elements in A with following properties: 1) Every triple contains three distinct elements; 2) Every element of A appears in exactly $3(r - 1)$ triples; 3) Every pair of two distinct elements in A is contained together in at most 6 triples.*

We call a $2k$ -dimensional subspace $U \subset \mathbb{C}^\ell$ *special* if it contains at least three of V_1, V_2, \dots, V_n . We define the *size* of a special space as the number of spaces among V_1, V_2, \dots, V_n contained in it. For a special space with size r , we take the $r^2 - r$ triples of indices of the spaces in it with the properties in Claim 2.8. Let \mathcal{S} be the family of all these triples. We claim that \mathcal{S} is a $(6, 3\delta)$ -system of $\mathcal{V} = (V'_1, V'_2, \dots, V'_n)$.

For every triple $\{j_1, j_2, j_3\} \in \mathcal{S}$, we can see that $V_{j_1}, V_{j_2}, V_{j_3}$ are contained in the same $2k$ -dimensional special space. And by $V_{j_1} \cap V_{j_2} = \{\mathbf{0}\}$, the space must be $V_{j_1} + V_{j_2}$ and hence $V_{j_3} \subseteq V_{j_1} + V_{j_2}$. By Claim 2.7,

$$V'_{j_3} = \{\operatorname{Re}(\mathbf{v}) : \mathbf{v} \in V_{j_3}\} \subseteq \{\operatorname{Re}(\mathbf{u}) + \operatorname{Re}(\mathbf{w}) : \mathbf{u} \in V_{j_1}, \mathbf{w} \in V_{j_2}\} = V'_{j_1} + V'_{j_2}.$$

Similarly, $V'_{j_1} \subseteq V'_{j_2} + V'_{j_3}$ and $V'_{j_2} \subseteq V'_{j_1} + V'_{j_3}$. One can see that every pair in $[n]$ appears in at most 6 triples because the corresponding two spaces are contained in at most one special space, and the pair appears at most 6 times in the triples constructed from this special space. For every $j \in [n]$, there are at least δn values of $j' \in [n] \setminus \{j\}$ such that there is a special space containing V_j and $V_{j'}$. This implies that the number of triples that j appears in is

$$\sum_{\substack{\text{special space } U \\ V_j \subseteq U}} 3(\operatorname{size}(U) - 1) = 3 \sum_{\substack{\text{special space } U \\ V_j \subseteq U}} |\{j' \neq j : V_{j'} \subseteq U\}| \geq 3\delta n.$$

Therefore \mathcal{S} is a $(6, 3\delta)$ -system of \mathcal{V} . By Theorem 2.6,

$$\dim(V'_1 + V'_2 + \dots + V'_n) = O(6^2(2k)^4/(3\delta)^2) = O(k^4/\delta^2).$$

Note that

$$\begin{aligned} V_1 + V_2 + \dots + V_n &\subseteq \operatorname{span} \{ \operatorname{Re}(\mathbf{v}_{js}), \operatorname{Im}(\mathbf{v}_{js}) \}_{j \in [n], s \in [k]} \quad (\text{span with complex coefficients}), \\ V'_1 + V'_2 + \dots + V'_n &= \operatorname{span} \{ \operatorname{Re}(\mathbf{v}_{js}), \operatorname{Im}(\mathbf{v}_{js}) \}_{j \in [n], s \in [k]} \quad (\text{span with real coefficients}). \end{aligned}$$

We thus have $\dim(V_1 + V_2 + \dots + V_n) \leq \dim(V'_1 + V'_2 + \dots + V'_n) = O(k^4/\delta^2)$. \blacktriangleleft

3 A generalization of Barthe's Theorem

We prove Theorem 1.4 in the following 3 subsections. In the fourth and last subsection, we state a convenient variant of the theorem (Theorem 3.8) that will be used later in the proof of our main result. The idea of the proof is similar to [3] (see also [6, Section 5]), which considers the maximum point of a function, and using the fact that all derivatives are 0 the result is proved. Here we consider a similar function f defined in Section 3.1. However, since our problem is more complicated, it is unclear whether we can find a maximum point at which all derivatives are 0. Instead we will show that there is a point with very small derivatives in Section 3.2, which is sufficient for our proof of the theorem in Section 3.3.

3.1 The function and basic properties

Let k_1, k_2, \dots, k_n be the dimensions of V_1, V_2, \dots, V_n respectively and $m = k_1 + k_2 + \dots + k_n$. Throughout our proof, we use pairs (i, j) with $i \in [n]$, $j \in [k_i]$ to denote the element of $[m]$ of position $\sum_{i' < i} k_{i'} + j$. We define a vector $\gamma \in \mathbb{R}^m$ as

$$\gamma_{ij} = p_i \quad \forall i \in [n], j \in [k_i].$$

For every $i \in [n]$, we fix $\{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \dots, \mathbf{v}_{ik_i}\}$ to be some basis of V_i (not necessarily orthonormal). A set $I \subseteq [m]$ is called a *good basis set* if

$$I = \bigcup_{i \in H} \{(i, 1), (i, 2), \dots, (i, k_i)\}$$

for some \mathcal{V} -admissible basis set H . We can see that for any good basis set I , the set $\{\mathbf{v}_{ij} : (i, j) \in I\}$ is a basis of \mathbb{R}^ℓ . For a list of vectors $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_q$ ($q \in \mathbb{Z}^+$), we use $[\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_q]$ to denote the matrix consisting of columns $\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_q$.

Let $\mathbf{O}(s)$ be the group of $s \times s$ orthogonal matrices. The function $f : \mathbb{R}^m \times \mathbf{O}(k_1) \times \mathbf{O}(k_2) \times \dots \times \mathbf{O}(k_n) \mapsto \mathbb{R}$ is defined as

$$f(\mathbf{t}, R_1, \dots, R_n) = \langle \boldsymbol{\gamma}, \mathbf{t} \rangle - \ln \det \left(\sum_{i \in [n], j \in [k_i]} e^{t_{ij}} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \right),$$

where, for every $i \in [n]$, the vectors \mathbf{x}_{ij} are given by

$$[\mathbf{x}_{i1}, \dots, \mathbf{x}_{ik_i}] = [\mathbf{v}_{i1}, \dots, \mathbf{v}_{ik_i}] R_i.$$

We note that here for every $i \in [n]$, $j \in [k_i]$, \mathbf{x}_{ij} is a function of R_i and $\{\mathbf{x}_{i1}, \dots, \mathbf{x}_{ik_i}\}$ is another basis of V_i .

The next lemma shows that the function f is bounded over its domain. The proof is similar to Proposition 3 in [3]. The proofs are given in the full version of this paper.

► **Lemma 3.1.** *There is a constant $C \in \mathbb{R}$ such that $f(\mathbf{t}, R_1, \dots, R_n) \leq C$ for all $\mathbf{t} \in \mathbb{R}^m$ and $R_i \in \mathbf{O}(k_i)$ ($i \in [n]$).*

3.2 Finding a point with small derivatives

We first define some notation. Let

$$X = \sum_{i \in [n], j \in [k_i]} e^{t_{ij}} \mathbf{x}_{ij} \mathbf{x}_{ij}^T$$

be a matrix valued function of $\mathbf{t}, R_1, R_2, \dots, R_n$. Then

$$f(\mathbf{t}, R_1, \dots, R_n) = \langle \boldsymbol{\gamma}, \mathbf{t} \rangle - \ln \det(X).$$

Note that X is always a positive definite matrix, since for any $\mathbf{w} \neq \mathbf{0}$,

$$\mathbf{w}^T X \mathbf{w} = \sum_{i \in [n], j \in [k_i]} e^{t_{ij}} \langle \mathbf{x}_{ij}, \mathbf{w} \rangle^2 > 0,$$

when $\mathbf{x}_{11}, \dots, \mathbf{x}_{nk_n}$ span the entire space (implied by $V_1 + V_2 + \dots + V_n = \mathbb{R}^\ell$). Define M to be the $\ell \times \ell$ full rank matrix satisfying $M^T M = X^{-1}$. We note that M is also a function of $\mathbf{t}, R_1, R_2, \dots, R_n$.

In a later part of the proof we will show that the linear map obtained from M satisfies the requirement in Theorem 1.4 when $\mathbf{t}, R_1, R_2, \dots, R_n$ take appropriate values. We first find an appropriate value of $(R_1, R_2, \dots, R_n) = (R_1^*(\mathbf{t}), R_2^*(\mathbf{t}), \dots, R_n^*(\mathbf{t}))$ for every $\mathbf{t} \in \mathbb{R}^m$, and then find some \mathbf{t}^* with specific properties.

► **Lemma 3.2.** *For every $\mathbf{t} \in \mathbb{R}^m$, there exists $(R_1^*(\mathbf{t}), R_2^*(\mathbf{t}), \dots, R_n^*(\mathbf{t}))$ satisfying*

1. $f(\mathbf{t}, R_1^*(\mathbf{t}), R_2^*(\mathbf{t}), \dots, R_n^*(\mathbf{t})) = \max_{R_1, R_2, \dots, R_n} \{f(\mathbf{t}, R_1, R_2, \dots, R_n)\}.$

2. For every $i \in [n]$, if $t_{ij} = t_{ij'}$ for some $j \neq j' \in [k_i]$, then

$$\langle M\mathbf{x}_{ij}, M\mathbf{x}_{ij'} \rangle = 0,$$

where $[\mathbf{x}_{i1}, \dots, \mathbf{x}_{ik_i}] = [\mathbf{v}_{i1}, \dots, \mathbf{v}_{ik_i}]R_i^*(\mathbf{t})$.

Proof. The first condition can be satisfied by the compactness of $\mathbf{O}(k_1) \times \mathbf{O}(k_2) \times \dots \times \mathbf{O}(k_n)$. We will show how to change $(R_1^*(\mathbf{t}), R_2^*(\mathbf{t}), \dots, R_n^*(\mathbf{t}))$, which already satisfies the first condition, so that it satisfies the second condition while preserving the first condition.

Fix an $i \in [n]$ and partition the indices of $(t_{i1}, t_{i2}, \dots, t_{ik_i})$ into equivalence classes $J_1, J_2, \dots, J_b \subseteq [k_i]$ such that for j, j' in the same class $t_{ij} = t_{ij'}$ and for j, j' in different classes $t_{ij} \neq t_{ij'}$. We use t_{J_r} to denote the value of t_{ij} for $j \in J_r$, and L_{J_r} to denote the matrix consisting of all columns \mathbf{x}_{ij} with $j \in J_r$. The terms in X that depend on R_i are

$$\sum_{r \in [b]} \left(e^{t_{J_r}} \sum_{j \in J_r} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \right) = \sum_{r \in [b]} (e^{t_{J_r}} \cdot L_{J_r} L_{J_r}^T) = \sum_{r \in [b]} (e^{t_{J_r}} \cdot L_{J_r} Q_r Q_r^T L_{J_r}^T),$$

where Q_r can be taken to be any $|J_r| \times |J_r|$ orthogonal matrix. This means that if we change $R_i^*(\mathbf{t})$ to $R_i^*(\mathbf{t}) \text{diag}(Q_1, \dots, Q_b)$ (here $\text{diag}(Q_1, \dots, Q_b)$ denotes the matrix in which the submatrix with row and column indices J_r is Q_r), or equivalently change L_{J_r} to $L_{J_r} Q_r$ for every $r \in [b]$, the matrix X does not change, hence M and f do not change, and the first condition is preserved as f is still the maximum for the fixed \mathbf{t} .

For every $r \in [b]$, we can find a Q_r such that the columns of $ML_{J_r} Q_r$ are orthogonal (consider the singular value decomposition of ML_{J_r}). Change $R_i^*(\mathbf{t})$ to $R_i^*(\mathbf{t}) \text{diag}(Q_1, \dots, Q_b)$ and the second condition is satisfied while preserving the first condition. Doing this for every i we can obtain an $(R_1^*(\mathbf{t}), R_2^*(\mathbf{t}), \dots, R_n^*(\mathbf{t}))$ satisfying both conditions. \blacktriangleleft

From now on we use $R_1^*(\mathbf{t}), R_2^*(\mathbf{t}), \dots, R_n^*(\mathbf{t})$ to denote the matrices satisfying the conditions in Lemma 3.2.

► **Lemma 3.3.** For any $\varepsilon > 0$, there exists $\mathbf{t}^* \in \mathbb{R}^m$ such that for every $i \in [n], j \in [k_i]$.

$$\left| \frac{\partial f}{\partial t_{ij}} \left(\mathbf{t}^*, R_1^*(\mathbf{t}^*), R_2^*(\mathbf{t}^*), \dots, R_n^*(\mathbf{t}^*) \right) \right| \leq \varepsilon.$$

This lemma follows immediately from the following, more general lemma, proved in the full version of this paper.

► **Lemma 3.4.** Let $\mathcal{A} \subseteq \mathbb{R}^h$ ($h \in \mathbb{Z}^+$) be a compact set. Let $f : \mathbb{R}^m \times \mathcal{A} \mapsto \mathbb{R}$ and $y^* : \mathbb{R}^m \mapsto \mathcal{A}$ be functions satisfying the following properties:

1. $f(\mathbf{x}, y)$ is bounded and continuous on $\mathbb{R}^m \times \mathcal{A}$.
 2. For every $\mathbf{x} \in \mathbb{R}^m$, $f(\mathbf{x}, y^*(\mathbf{x})) = \max_{y \in \mathcal{A}} \{f(\mathbf{x}, y)\}$.
 3. For every fixed $y \in \mathcal{A}$, $f(\mathbf{x}, y)$ as a function of \mathbf{x} is differentiable on \mathbb{R}^m .
- Then, for every $\varepsilon > 0$, there exists an $\mathbf{x}^* \in \mathbb{R}^m$ such that for every $i \in [m]$,

$$\left| \frac{\partial f}{\partial x_i} \left(\mathbf{x}^*, y^*(\mathbf{x}^*) \right) \right| \leq \varepsilon.$$

3.3 Proof of Theorem 1.4

Fix some $\varepsilon > 0$. We apply Lemma 3.3 and obtain a \mathbf{t}^* . In the remaining proof we will use X , M and \mathbf{x}_{ij} ($i \in [n], j \in [k_i]$) to denote their values when $\mathbf{t} = \mathbf{t}^*$ and $R_i = R_i^*(\mathbf{t}^*)$ ($i \in [n]$).

► **Lemma 3.5.** $\langle M\mathbf{x}_{ij}, M\mathbf{x}_{ij'} \rangle = 0$ for every $i \in [n]$ and $j \neq j' \in [k_i]$.

Proof. We fix $i_0 \in [n], j_0 \neq j'_0 \in [k_{i_0}]$ and prove $\langle M\mathbf{x}_{i_0j_0}, M\mathbf{x}_{i_0j'_0} \rangle = 0$. If $t_{i_0j_0}^* = t_{i_0j'_0}^*$, this is guaranteed by Lemma 3.2. We only consider the case that $t_{i_0j_0}^* \neq t_{i_0j'_0}^*$.

Let $\theta \in \mathbb{R}$ be a variable, and define \mathbf{x}'_{ij} for $i \in [n], j \in [k_i]$ as follows.

$$\mathbf{x}'_{ij} = \begin{cases} \cos \theta \cdot \mathbf{x}_{i_0j_0} - \sin \theta \cdot \mathbf{x}_{i_0j'_0} & (i, j) = (i_0, j_0), \\ \sin \theta \cdot \mathbf{x}_{i_0j_0} + \cos \theta \cdot \mathbf{x}_{i_0j'_0} & (i, j) = (i_0, j'_0), \\ \mathbf{x}_{ij} & \text{otherwise.} \end{cases}$$

We consider the following function $h : \mathbb{R} \mapsto \mathbb{R}$,

$$h(\theta) = \langle \boldsymbol{\gamma}, \mathbf{t}^* \rangle - \ln \det \left(\sum_{i \in [n], j \in [k_i]} e^{t_{ij}^*} \mathbf{x}'_{ij} \mathbf{x}'_{ij}{}^T \right).$$

► **Claim 3.6.** $h(\theta)$ has a maximum at $\theta = 0$.

Proof. Let $R(\theta)$ be the $k_{i_0} \times k_{i_0}$ orthogonal matrix obtained from the identity matrix by changing the $(j_0, j_0), (j'_0, j'_0)$ entries to $\cos \theta$, the (j_0, j'_0) entry to $\sin \theta$, and the (j'_0, j_0) entry to $-\sin \theta$. We can see $R(0)$ is the identity matrix and

$$[\mathbf{x}'_{i_01}, \dots, \mathbf{x}'_{i_0k_{i_0}}] = [\mathbf{x}_{i_01}, \dots, \mathbf{x}_{i_0k_{i_0}}]R(\theta).$$

Therefore for all $\theta \in \mathbb{R}$.

$$\begin{aligned} h(\theta) &= f\left(\mathbf{t}^*, R_1^*(\mathbf{t}^*), \dots, R_{i_0-1}^*(\mathbf{t}^*), R_{i_0}^*(\mathbf{t}^*) \cdot R(\theta), R_{i_0+1}^*(\mathbf{t}^*), \dots, R_n^*(\mathbf{t}^*)\right) \\ &\leq f\left(\mathbf{t}^*, R_1^*(\mathbf{t}^*), \dots, R_{i_0-1}^*(\mathbf{t}^*), R_{i_0}^*(\mathbf{t}^*), R_{i_0+1}^*(\mathbf{t}^*), \dots, R_n^*(\mathbf{t}^*)\right) \\ &= h(0). \end{aligned}$$

Thus the claim is proved. ◀

Using $\frac{d}{ds} \ln \det(A) = \text{tr}(A^{-1} \frac{d}{ds} A)$ for an invertible matrix A (Theorem 4 in [12, Chapter 9]), we can calculate the derivative of h .

$$\begin{aligned} \frac{dh}{d\theta}(0) &= -\text{tr} \left[X^{-1} \left(e^{t_{i_0j_0}^*} \frac{d}{d\theta} \Big|_{\theta=0} \mathbf{x}'_{i_0j_0} \mathbf{x}'_{i_0j_0}{}^T + e^{t_{i_0j'_0}^*} \frac{d}{d\theta} \Big|_{\theta=0} \mathbf{x}'_{i_0j'_0} \mathbf{x}'_{i_0j'_0}{}^T \right) \right] \\ &= -\text{tr} \left[X^{-1} \left(e^{t_{i_0j_0}^*} \frac{d}{d\theta} \Big|_{\theta=0} (\cos \theta \cdot \mathbf{x}_{i_0j_0} - \sin \theta \cdot \mathbf{x}_{i_0j'_0})(\cos \theta \cdot \mathbf{x}_{i_0j_0} - \sin \theta \cdot \mathbf{x}_{i_0j'_0})^T \right. \right. \\ &\quad \left. \left. + e^{t_{i_0j'_0}^*} \frac{d}{d\theta} \Big|_{\theta=0} (\sin \theta \cdot \mathbf{x}_{i_0j_0} + \cos \theta \cdot \mathbf{x}_{i_0j'_0})(\sin \theta \cdot \mathbf{x}_{i_0j_0} + \cos \theta \cdot \mathbf{x}_{i_0j'_0})^T \right) \right] \\ &= -e^{t_{i_0j_0}^*} \text{tr} \left[\frac{d}{d\theta} \Big|_{\theta=0} (\cos \theta \cdot M\mathbf{x}_{i_0j_0} - \sin \theta \cdot M\mathbf{x}_{i_0j'_0})(\cos \theta \cdot M\mathbf{x}_{i_0j_0} - \sin \theta \cdot M\mathbf{x}_{i_0j'_0})^T \right] \\ &\quad - e^{t_{i_0j'_0}^*} \text{tr} \left[\frac{d}{d\theta} \Big|_{\theta=0} (\sin \theta \cdot M\mathbf{x}_{i_0j_0} + \cos \theta \cdot M\mathbf{x}_{i_0j'_0})(\sin \theta \cdot M\mathbf{x}_{i_0j_0} + \cos \theta \cdot M\mathbf{x}_{i_0j'_0})^T \right] \\ &= -e^{t_{i_0j_0}^*} [-2 \cdot \langle M\mathbf{x}_{i_0j_0}, M\mathbf{x}_{i_0j'_0} \rangle] - e^{t_{i_0j'_0}^*} [2 \cdot \langle M\mathbf{x}_{i_0j_0}, M\mathbf{x}_{i_0j'_0} \rangle] \\ &= 2(e^{t_{i_0j_0}^*} - e^{t_{i_0j'_0}^*}) \cdot \langle M\mathbf{x}_{i_0j_0}, M\mathbf{x}_{i_0j'_0} \rangle. \end{aligned}$$

Since $h(0)$ is the maximum, we have $\frac{dh}{d\theta}(0) = 0$. By $t_{i_0j_0}^* \neq t_{i_0j'_0}^*$, the above equation implies $\langle M\mathbf{x}_{i_0j_0}, M\mathbf{x}_{i_0j'_0} \rangle = 0$. ◀

Finally we are able to prove Theorem 1.4.

Proof of Theorem 1.4. With a slight abuse of notation, we also use M to denote the linear map defined by the matrix M . We show that M satisfies the requirement in Theorem 1.4. Let $\mathbf{u}_{ij} = M\mathbf{x}_{ij}/\|M\mathbf{x}_{ij}\|$ ($i \in [n], j \in [k_i]$). Then $\{\mathbf{u}_{i1}, \mathbf{u}_{i2}, \dots, \mathbf{u}_{ik_i}\}$ is an orthonormal basis of $M(V_i)$, and

$$\text{Proj}_{M(V_i)} = [\mathbf{u}_{i1}, \mathbf{u}_{i2}, \dots, \mathbf{u}_{ik_i}] \begin{bmatrix} \mathbf{u}_{i1}^T \\ \vdots \\ \mathbf{u}_{ik_i}^T \end{bmatrix} = \sum_{j=1}^{k_i} \mathbf{u}_{ij} \mathbf{u}_{ij}^T. \quad (1)$$

We define

$$\varepsilon_{ij} = \frac{\partial f}{\partial t_{ij}} \left(\mathbf{t}^*, R_1^*(\mathbf{t}^*), R_2^*(\mathbf{t}^*), \dots, R_n^*(\mathbf{t}^*) \right) \in [-\varepsilon, \varepsilon].$$

Again using $\frac{d}{ds} \ln \det(A) = \text{tr}(A^{-1} \frac{d}{ds} A)$ for an invertible matrix A , we have

$$\varepsilon_{ij} = p_i - \text{tr} \left(X^{-1} e^{t_{ij}^*} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \right) = p_i - e^{t_{ij}^*} \cdot \text{tr} \left(M\mathbf{x}_{ij} \mathbf{x}_{ij}^T M^T \right) = p_i - e^{t_{ij}^*} \cdot \|M\mathbf{x}_{ij}\|^2.$$

By the definition of X and M ,

$$M^{-1}(M^T)^{-1} = X = \sum_{i \in [n], j \in [k_i]} e^{t_{ij}^*} \mathbf{x}_{ij} \mathbf{x}_{ij}^T \implies \sum_{i \in [n], j \in [k_i]} e^{t_{ij}^*} (M\mathbf{x}_{ij})(M\mathbf{x}_{ij})^T = I_{\ell \times \ell}.$$

Therefore

$$\sum_{i \in [n], j \in [k_i]} (p_i - \varepsilon_{ij}) \mathbf{u}_{ij} \mathbf{u}_{ij}^T = \sum_{i \in [n], j \in [k_i]} e^{t_{ij}^*} \|M\mathbf{x}_{ij}\|^2 \left(\frac{M\mathbf{x}_{ij}}{\|M\mathbf{x}_{ij}\|} \right) \left(\frac{M\mathbf{x}_{ij}}{\|M\mathbf{x}_{ij}\|} \right)^T = I_{\ell \times \ell}.$$

By (1),

$$\left\| \sum_{i=1}^n p_i \text{Proj}_{M(V_i)} - I_{\ell \times \ell} \right\| = \left\| \sum_{i \in [n], j \in [k_i]} \varepsilon_{ij} \mathbf{u}_{ij} \mathbf{u}_{ij}^T \right\| \leq \varepsilon \sum_{i \in [n], j \in [k_i]} \|\mathbf{u}_{ij} \mathbf{u}_{ij}^T\| \leq \varepsilon m.$$

Let $\bar{M} = M/\|M\|$, we can see that $\bar{M}(V_i)$ and $M(V_i)$ are the same linear space, hence

$$\left\| \sum_{i=1}^n p_i \text{Proj}_{\bar{M}(V_i)} - I_{\ell \times \ell} \right\| \leq \varepsilon m.$$

Take $\varepsilon \rightarrow 0$, noting that \bar{M} is contained in a compact set, there must exist a matrix M^* such that

$$\sum_{i=1}^n p_i \text{Proj}_{M^*(V_i)} = I_{\ell \times \ell}.$$

It remains to show that M^* is invertible. Assume it is not invertible, then there is a nonzero vector \mathbf{w} orthogonal to the range of M^* . We have $\text{Proj}_{M^*(V_i)}(\mathbf{w}) = \mathbf{0}$ for every $i \in [n]$. This contradicts the fact that the sum of $p_i \text{Proj}_{M^*(V_i)}$ is the identity matrix. Therefore M^* is invertible. Thus Theorem 1.4 is proved. \blacktriangleleft

3.4 A convenient form of Theorem 1.4

We give Theorem 3.8 which is implied by Theorem 1.4 and is the form that will be used in our proof. Before stating the theorem, we need to define *admissible sets* and *admissible vectors* as Definition 3.7, which have weaker requirements than admissible basis sets and admissible basis vectors (Definition 1.3) as they are not required to span the entire arrangement.

► **Definition 3.7** (admissible set, admissible vector). Given a list of vector spaces $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$), a set $H \subseteq [n]$ is called a \mathcal{V} -admissible set if $\dim(\sum_{i \in H} V_i) = \sum_{i \in H} \dim(V_i)$, i.e. if every space with index in H has intersection $\{0\}$ with the span of the other spaces with indices in H . A \mathcal{V} -admissible vector is any indicator vector $\mathbf{1}_H$ of some \mathcal{V} -admissible set H .

► **Theorem 3.8.** Given a list of vector spaces $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \subseteq \mathbb{R}^\ell$) and a vector $\mathbf{p} \in \mathbb{R}^n$ in the convex hull of all \mathcal{V} -admissible vectors. Then there exists an invertible linear map $M : \mathbb{R}^\ell \mapsto \mathbb{R}^\ell$ such that for any unit vector $\mathbf{w} \in \mathbb{R}^\ell$,

$$\sum_{i=1}^n p_i \|\text{Proj}_{M(V_i)}(\mathbf{w})\|^2 \leq 1,$$

where $\text{Proj}_{M(V_i)}(\mathbf{w})$ is the projection of \mathbf{w} onto $M(V_i)$.

The simple derivation of Theorem 3.8 from Theorem 1.4 is included in the full version of this paper.

4 Proof of the main Theorem

Theorem 2.6 will follow from the following theorem using a simple recursive argument, provided in the full version of this paper.

► **Theorem 4.1.** Let $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \in \mathbb{R}^\ell$) be a list of k -bounded vector spaces with an (α, δ) -system and $d = \dim(V_1 + V_2 + \dots + V_n)$, then for any $\beta \in (0, 1)$, at least one of these two cases holds:

1. $d \leq 40\alpha k^3 / (\beta\delta)$,
2. There is a sublist of $q \geq \delta n / (20\alpha)$ spaces $(V_{i_1}, V_{i_2}, \dots, V_{i_q})$ such that there are nonzero vectors $\mathbf{z}_1 \in V_{i_1}, \mathbf{z}_2 \in V_{i_2}, \dots, \mathbf{z}_q \in V_{i_q}$ with

$$\dim(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_q) \leq \beta d.$$

4.1 Proof of Theorem 4.1 – a special case

In this subsection, we consider the case that all vector spaces are ‘well separated’.

► **Definition 4.2.** Two vector spaces $V, V' \subseteq \mathbb{R}^\ell$ are τ -separated if $|\langle \mathbf{u}, \mathbf{u}' \rangle| \leq 1 - \tau$ for any two unit vectors $\mathbf{u} \in V$ and $\mathbf{u}' \in V'$.

We will use the following two simple lemmas about τ -separated spaces (both are proved in the full version of this paper.)

► **Lemma 4.3.** Given two vector spaces $V, V' \subseteq \mathbb{R}^\ell$ that are τ -separated and let $B = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{k_1}\}$ and $B' = \{\mathbf{u}'_1, \mathbf{u}'_2, \dots, \mathbf{u}'_{k_2}\}$ be orthonormal bases for V, V' respectively. For any unit vector $\mathbf{u} \in V + V'$, if we write \mathbf{u} as

$$\mathbf{u} = \lambda_1 \mathbf{u}_1 + \lambda_2 \mathbf{u}_2 + \dots + \lambda_{k_1} \mathbf{u}_{k_1} + \mu_1 \mathbf{u}'_1 + \mu_2 \mathbf{u}'_2 + \dots + \mu_{k_2} \mathbf{u}'_{k_2},$$

then the coefficients satisfy $\lambda_1^2 + \lambda_2^2 + \dots + \lambda_{k_1}^2 + \mu_1^2 + \mu_2^2 + \dots + \mu_{k_2}^2 \leq \frac{1}{\tau}$.

► **Lemma 4.4.** *Given two vector spaces $V, V' \subseteq \mathbb{R}^\ell$ and let $B = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{k_1}\}$ be an orthonormal basis of V . If V and V' are not τ -separated, there must exist $j \in [k_1]$ such that $\|\text{Proj}_{V'}(\mathbf{u}_j)\|^2 \geq (1 - \tau)^2/k_1$, where $\text{Proj}_{V'}(\mathbf{u}_j)$ is the projection of \mathbf{u}_j onto V' .*

We will need the following lower bound for the rank of a diagonal dominating matrix. The proof is included in the full version of this paper.

► **Lemma 4.5.** *Let $D = (d_{ij})$ be a complex $m \times m$ matrix and L, K be positive real numbers. If $d_{ii} = L$ for every $i \in [m]$ and $\sum_{i \neq j} |d_{ij}|^2 \leq K$, then $\text{rank}(D) \geq m - K/L^2$.*

The following theorem handles the ‘well separated case’ of Theorem 4.1.

► **Theorem 4.6.** *Let $\mathcal{V} = (V_1, V_2, \dots, V_n)$ ($V_i \in \mathbb{R}^\ell$) be a list of k -bounded vector spaces with an (α, δ) -system $\mathcal{S} = (S_1, S_2, \dots, S_w)$ and $d = \dim(V_1 + V_2 + \dots + V_n)$. If for every $j \in [w]$ and $\{i_1, i_2\} \subseteq S_j$, V_{i_1} and V_{i_2} are τ -separated, then $d \leq \alpha k / (\tau \delta)$.*

Proof. Let k_1, k_2, \dots, k_n be the dimensions of V_1, V_2, \dots, V_n , and $m = k_1 + k_2 + \dots + k_n$. For every $i \in [n]$, fix $B_i = \{\mathbf{u}_{i1}, \mathbf{u}_{i2}, \dots, \mathbf{u}_{ik_i}\}$ to be some orthonormal basis of V_i . We use A to denote the $m \times \ell$ matrix whose rows are $\mathbf{u}_{11}^T, \dots, \mathbf{u}_{nk_n}^T$. We will bound $d = \text{rank}(A)$ by constructing a high rank $m \times m$ matrix D satisfying $DA = 0$.

For $s \in [m]$, we use $\psi(s) \in [n]$ to denote the number satisfying

$$k_1 + k_2 + \dots + k_{\psi(s)-1} + 1 \leq s \leq k_1 + k_2 + \dots + k_{\psi(s)-1} + k_{\psi(s)}.$$

In other words, the s -th row of A is a vector in $B_{\psi(s)}$.

► **Claim 4.7.** *For every $s \in [m]$, there is a vector $\mathbf{y}_s \in \mathbb{R}^m$ satisfying $\mathbf{y}_s^T A = \mathbf{0}^T$, $y_{ss} = \lceil \delta n \rceil$, and $\sum_{t \neq s} y_{st}^2 \leq \alpha \lceil \delta n \rceil / \tau$.*

Proof. Say the s -th row of A is \mathbf{u}^T , where $\mathbf{u} \in B_{\psi(s)}$. Let $J \subseteq [w]$ be a set of size $|J| = \lceil \delta n \rceil$ such that for every $j \in J$, S_j contains $\psi(s)$. We construct a vector \mathbf{c}_j for every $j \in J$ as follows.

■ If S_j contains 3 elements $\{\psi(s), i, i'\}$, we have $\lambda_1, \lambda_2, \dots, \lambda_{k_i}, \mu_1, \mu_2, \dots, \mu_{k_{i'}} \in \mathbb{R}$ such that

$$\mathbf{u} - \lambda_1 \mathbf{u}_{i1} - \lambda_2 \mathbf{u}_{i2} - \dots - \lambda_{k_i} \mathbf{u}_{ik_i} - \mu_1 \mathbf{u}_{i'1} - \mu_2 \mathbf{u}_{i'2} - \dots - \mu_{k_{i'}} \mathbf{u}_{i'k_{i'}} = \mathbf{0}.$$

We can obtain from this equation a vector \mathbf{c}_j such that $\mathbf{c}_j^T A = \mathbf{0}^T$, $c_{js} = 1$, and by Lemma 4.3

$$\sum_{t \neq s} c_{jt}^2 = \lambda_1^2 + \lambda_2^2 + \dots + \lambda_{k_i}^2 + \mu_1^2 + \mu_2^2 + \dots + \mu_{k_{i'}}^2 \leq \frac{1}{\tau}.$$

■ If S_j contains 2 elements $\{\psi(s), i\}$, there exist $\lambda_1, \lambda_2, \dots, \lambda_{k_i}$ with $\lambda_1^2 + \lambda_2^2 + \dots + \lambda_{k_i}^2 = 1$ such that

$$\mathbf{u} - \lambda_1 \mathbf{u}_{i1} - \lambda_2 \mathbf{u}_{i2} - \dots - \lambda_{k_i} \mathbf{u}_{ik_i} = \mathbf{0}.$$

We can obtain from this equation a vector \mathbf{c}_j such that $\mathbf{c}_j^T A = \mathbf{0}^T$, $c_{js} = 1$, and

$$\sum_{t \neq s} c_{jt}^2 = \lambda_1^2 + \lambda_2^2 + \dots + \lambda_{k_i}^2 = 1 \leq 1/\tau.$$

In either case we obtain a \mathbf{c}_j such that $\mathbf{c}_j^T A = \mathbf{0}^T$, $c_{js} = 1$ and $\sum_{t \neq s} c_{jt}^2 \leq 1/\tau$. We define

$$\mathbf{y}_s = \sum_{j \in J} \mathbf{c}_j.$$

We have $\mathbf{y}_s^T A = \mathbf{0}^T$ and $y_{ss} = \lceil \delta n \rceil$. We consider $\sum_{t \neq s} y_{st}^2$. From the above construction of \mathbf{c}_j , we can see $c_{jt} \neq 0$ ($t \neq s$) only when $\psi(t) \neq \psi(s)$ and $\{\psi(s), \psi(t)\} \subseteq S_j$. Hence for every $t \neq s$, there are at most α nonzero values in $\{c_{jt}\}_{j \in J}$. It follows that

$$\sum_{t \neq s} y_{st}^2 = \sum_{t \neq s} \left(\sum_{j \in J} c_{jt} \right)^2 \leq \alpha \sum_{t \neq s} \left(\sum_{j \in J} c_{jt}^2 \right) = \alpha \sum_{j \in J} \left(\sum_{t \neq s} c_{jt}^2 \right) \leq \frac{\alpha \lceil \delta n \rceil}{\tau}.$$

Thus the claim is proved. ◀

Define D to be the matrix consists of rows $\mathbf{y}_1^T, \mathbf{y}_2^T, \dots, \mathbf{y}_m^T$. Then every entry on the diagonal of D is $\lceil \delta n \rceil$, and the sum of squares of all entries off the diagonal is at most $\alpha \lceil \delta n \rceil m / \tau$. Apply Lemma 4.5 on D , and we have

$$\text{rank}(D) \geq m - \frac{\alpha \lceil \delta n \rceil m / \tau}{\lceil \delta n \rceil^2} = m - \frac{\alpha m}{\tau \lceil \delta n \rceil} \geq m - \frac{\alpha k}{\tau \delta}.$$

By $DA = 0$, the rank of A is $d \leq \alpha k / (\tau \delta)$. ◀

4.2 Proof of Theorem 4.1 – general case

Now we prove Theorem 4.1. We assume that the first case of Theorem 4.1 does not hold, i.e. $d > 40\alpha k^3 / (\beta \delta)$. We will show the second case holds.

► **Lemma 4.8.** *If the second case of Theorem 4.1 does not hold, i.e. for any sublist of $q \geq \delta n / (20\alpha)$ spaces $(V_{i_1}, V_{i_2}, \dots, V_{i_q})$ and nonzero vectors $\mathbf{z}_1 \in V_{i_1}, \mathbf{z}_2 \in V_{i_2}, \dots, \mathbf{z}_q \in V_{i_q}$,*

$$\dim(\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_q) > \beta d,$$

then there exists a distribution \mathcal{D} on \mathcal{V} -admissible sets and an $I \subseteq [n]$ with $|I| \geq (1 - \delta / (10\alpha))n$ such that for every $i \in I$,

$$\Pr_{H \sim \mathcal{D}} [i \in H] \geq \frac{\beta d}{kn}.$$

Proof. Fix $q = \lceil \delta n / (20\alpha) \rceil$. By assumption $d > 40\alpha k^3 / (\beta \delta)$, we have $n \geq d/k > 10\alpha/\delta$. It follows that $q < \delta n / (10\alpha)$. We can also see $\delta n / (10\alpha) < n$ by $\delta/\alpha \leq 3/2$ (Lemma 2.2).

We will find a distribution using the following claim.

► **Claim 4.9.** *For a subset $E \subseteq [n]$ of size greater than q , we can find a \mathcal{V} -admissible set $H \subseteq E$ with size at least $\beta d/k$.*

Proof. Initially let $H = \emptyset$. In each step we pick an $i_0 \in E$ with $V_{i_0} \cap \sum_{i \in H} V_i = \{\mathbf{0}\}$, and add i_0 to H . If such an i_0 does not exist, the procedure terminates. If $|H| < \beta d/k$, then for every $i_0 \in E$, V_{i_0} has a nonzero vector contained in the space $\sum_{i \in H} V_i$, which has dimension at most βd . This contradicts the condition that the second case of Theorem 4.1 does not hold. Hence $|H| \geq \beta d/k$, and the claim is proved. ◀

We repeatedly find a \mathcal{V} -admissible sets H_1, H_2, \dots such that $H_i \subseteq [n] \setminus (H_1 \cup \dots \cup H_{i-1})$ and $|H_i| \geq \beta d/k$ using the above claim. We can find at most

$$\frac{n - q}{\beta d/k} \leq \frac{nk}{\beta d}$$

such \mathcal{V} -admissible sets in total. Let I be the union of these \mathcal{V} -admissible sets. We have $|I| \geq n - q \geq (1 - \delta / (10\alpha))n$. Let \mathcal{D} be the uniform distribution on these \mathcal{V} -admissible sets. We can see that the probability $\Pr_{H \sim \mathcal{D}} [i \in H] \geq \beta d / (kn)$ for every $i \in I$. Thus the lemma is proved. ◀

Assume the second case of Theorem 4.1 does not hold and apply Lemma 4.8. For $i \in [n]$, we use k_i to denote the dimension of V_i , and p_i to denote $\Pr_{H \sim \mathcal{D}}[i \in H]$. Then $p_i \geq \beta d / (kn)$ for every $i \in I$.

► **Lemma 4.10.** *The vector $\mathbf{p} = (p_1, p_2, \dots, p_n)$ is in the convex hull of \mathcal{V} -admissible vectors.*

Proof. For every \mathcal{V} -admissible set H , we use q_H to denote the probability that H is picked according to \mathcal{D} , and $\mathbf{1}_H$ to denote the \mathcal{V} -admissible vector corresponding to H . Then,

$$\mathbf{p} = (p_1, p_2, \dots, p_n) = \sum_{\mathcal{V}\text{-admissible } H} q_H \mathbf{1}_H$$

and p_i is exactly the probability that $i \in H$. ◀

We apply Theorem 3.8 with the $\mathbf{p} = (p_1, p_2, \dots, p_n)$, and obtain an invertible linear map $M : \mathbb{R}^\ell \mapsto \mathbb{R}^\ell$ such that for any unit vector $\mathbf{w} \in \mathbb{R}^\ell$,

$$\sum_{i=1}^n p_i \|\text{Proj}_{V'_i}(\mathbf{w})\|^2 \leq 1,$$

where V'_i denotes $M(V_i)$. Since $p_i \geq \beta d / (kn)$ for every $i \in I$, we have

$$\sum_{i \in I} \|\text{Proj}_{V'_i}(\mathbf{w})\|^2 \leq \frac{kn}{\beta d}. \quad (2)$$

We will reduce the problem to the special case discussed in the previous subsection. We say a pair $\{i_1, i_2\} \subseteq [n]$ is *bad* if V'_{i_1}, V'_{i_2} are not $\frac{1}{2}$ -separated. Let $\mathcal{S} = (S_1, S_2, \dots, S_w)$ be the (α, δ) -system of \mathcal{V} . By Lemma 2.4, \mathcal{S} is also an (α, δ) -system of $\mathcal{V}' = (V'_1, V'_2, \dots, V'_n)$. We estimate the number of sets among S_1, S_2, \dots, S_w containing a bad pair.

► **Lemma 4.11.** *For every $i_0 \in I$, there are at most $\delta n / (10\alpha)$ values of $i \in I$ such that V'_{i_0} and V'_i are not $\frac{1}{2}$ -separated.*

Proof. Let $\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_{k_{i_0}}\}$ be an orthonormal basis of V'_{i_0} . For any i that V'_{i_0} and V'_i are not $\frac{1}{2}$ -separated, by Lemma 4.4, there must be $j \in [k_{i_0}]$ such that

$$\|\text{Proj}_{V'_i}(\mathbf{u}_j)\|^2 \geq \frac{1}{4k_{i_0}} \geq \frac{1}{4k}.$$

For every $j_0 \in [k_{i_0}]$, we set $\mathbf{w} = \mathbf{u}_{j_0}$ in inequality (2). The number of i 's such that $\|\text{Proj}_{V'_i}(\mathbf{u}_{j_0})\| \geq 1/(4k)$ is at most

$$\frac{kn}{\beta d} \Big/ \frac{1}{4k} = \frac{4k^2 n}{\beta d}.$$

Since there are $k_{i_0} \leq k$ values of $j_0 \in [k_{i_0}]$, the number of i 's that V'_{i_0} and V'_i are not $\frac{1}{2}$ -separated is at most

$$k \cdot \frac{4k^2 n}{\beta d} \leq \frac{4k^3 n}{\beta d} \leq \frac{\delta n}{10\alpha}.$$

In the last inequality we used the assumption $d > 40\alpha k^3 / (\beta\delta)$. ◀

The number of bad pairs is at most

$$|[n] \setminus I| \cdot n + |I| \cdot \frac{\delta n}{10\alpha} \leq \frac{\delta n^2}{10\alpha} + \frac{\delta n^2}{10\alpha} = \frac{\delta n^2}{5\alpha}.$$

We remove all S_j 's that contains a bad pair and use \mathcal{S}' to denote the list of the remaining sets. Since each pair appears at most α times, we have removed at most $\delta n^2/5$ sets. Originally we have at least $\delta n^2/3$ sets by Lemma 2.2. Now we have at least $\delta n^2/3 - \delta n^2/5 \geq \delta n^2/10$ sets. By Lemma 2.3, there is a sublist $\mathcal{V}'' = (V'_{i_1}, V'_{i_2}, \dots, V'_{i_q})$ ($q \geq \delta n/(20\alpha)$) of \mathcal{V}' and a sublist \mathcal{S}'' of \mathcal{S}' such that \mathcal{S}'' is an $(\alpha, \delta/20)$ -system of \mathcal{V}'' .

Since we have removed all bad pairs, \mathcal{V}'' and \mathcal{S}'' must satisfy the conditions of Theorem 4.6. By Theorem 4.6,

$$\dim(V'_{i_1} + V'_{i_2} + \dots + V'_{i_q}) \leq \frac{\alpha k}{\frac{1}{2} \cdot \delta/20} = \frac{40\alpha k}{\delta} \leq \beta d.$$

In the last inequality we used the assumption $d > 40\alpha k^3/(\beta\delta)$. Recall that the linear map M is invertible. So the space $V_{i_1} + V_{i_2} + \dots + V_{i_q}$ has the same dimension as $V'_{i_1} + V'_{i_2} + \dots + V'_{i_q}$. Therefore there are $q \geq \delta n/(20\alpha)$ spaces $V_{i_1}, V_{i_2}, \dots, V_{i_q}$ within dimension βd . The second case of Theorem 4.1 holds.

In summary, under the assumption $d > 40\alpha k^3/(\beta\delta)$ we have shown the second case of Theorem 1.4 is always satisfied. Therefore Theorem 4.1 is proved. ◀

References

- 1 Boaz Barak, Zeev Dvir, Avi Wigderson, and Amir Yehudayoff. Fractional Sylvester-Gallai theorems. *Proceedings of the National Academy of Sciences*, 110(48):19213–19219, 2013.
- 2 Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC'11, pages 519–528, 2011.
- 3 Franck Barthe. On a reverse form of the Brascamp-Lieb inequality. *Inventiones mathematicae*, 134(2):335–361, 1998.
- 4 P. Borwein and W. O. J. Moser. A survey of Sylvester's problem and its generalizations. *Aequationes Mathematicae*, 40(1):111–135, 1990.
- 5 Zeev Dvir. On matrix rigidity and locally self-correctable codes. *computational complexity*, 20(2):367–388, 2011.
- 6 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Breaking the quadratic barrier for 3-LCC's over the reals. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC'14, pages 784–793, 2014.
- 7 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of Kelly's theorem. *Forum of Mathematics, Sigma*, 2, 10 2014.
- 8 Noam Elkies, Lou M Pretorius, and Konrad J Swanepoel. Sylvester-Gallai theorems for complex numbers and quaternions. *Discrete & Computational Geometry*, 35(3):361–373, 2006.
- 9 P. Erdős, Richard Bellman, H. S. Wall, James Singer, and V. Thébault. Problems for solution: 4065-4069. *The American Mathematical Monthly*, 50(1):65–66, 1943.
- 10 Sten Hansen. A generalization of a theorem of Sylvester on the lines determined by a finite point set. *Mathematica Scandinavica*, 16:175–180, 1965.
- 11 L. M. Kelly. A resolution of the Sylvester-Gallai problem of J.-P. Serre. *Discrete & Computational Geometry*, 1(1):101–104, 1986.
- 12 Peter D. Lax. *Linear algebra and its applications*. Pure and Applied Mathematics. Wiley-Interscience, 2007.
- 13 E. Melchior. Über vielseitige der projektive ebene. *Deutsche Math.*, 5:461–475, 1940.
- 14 J. J. Sylvester. Mathematical question 11851. *Educational Times*, 59:98, 1893.

Computational Aspects of the Colorful Carathéodory Theorem

Wolfgang Mulzer* and Yannik Stein†

Institut für Informatik, Freie Universität Berlin, Germany
{mulzer, yannikstein}@inf.fu-berlin.de

Abstract

Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be d -dimensional point sets such that the convex hull of each P_i contains the origin. We call the sets P_i *color classes*, and we think of the points in P_i as having color i . A *colorful choice* is a set with at most one point of each color. The *colorful Carathéodory theorem* guarantees the existence of a colorful choice whose convex hull contains the origin. So far, the computational complexity of finding such a colorful choice is unknown.

We approach this problem from two directions. First, we consider approximation algorithms: an m -*colorful choice* is a set that contains at most m points from each color class. We show that for any fixed $\varepsilon > 0$, an $\lceil \varepsilon d \rceil$ -colorful choice containing the origin in its convex hull can be found in polynomial time. This notion of approximation has not been studied before, and it is motivated through the applications of the colorful Carathéodory theorem in the literature. In the second part, we present a natural generalization of the colorful Carathéodory problem: in the *Nearest Colorful Polytope* problem (NCP), we are given sets $P_1, \dots, P_n \subset \mathbb{R}^d$ that do not necessarily contain the origin in their convex hulls. The goal is to find a colorful choice whose convex hull minimizes the distance to the origin. We show that computing local optima for the NCP problem is PLS-complete, while computing a global optimum is NP-hard.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems – Geometrical problems and computations

Keywords and phrases colorful Carathéodory theorem, high-dimensional approximation, PLS

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.44

1 Introduction

Let $P \subset \mathbb{R}^d$ be a point set. Carathéodory’s theorem [6, Theorem 1.2.3] states that if $\vec{0} \in \text{conv}(P)$, there is a subset $P' \subseteq P$ of at most $d + 1$ points with $\vec{0} \in \text{conv}(P')$. Bárány [3] gives a generalization to the colorful setting.

► **Theorem 1.1** (Colorful Carathéodory Theorem [3]). *Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be point sets (the color classes). If $\vec{0} \in \text{conv}(P_i)$, for $i = 1, \dots, d + 1$, there is a colorful choice C with $\vec{0} \in \text{conv}(C)$. Here, a colorful choice is a set with at most one point from each color class.*

Theorem 1.1 implies Carathéodory’s theorem by setting $P_1 = \dots = P_{d+1}$. Moreover, there are many variants with weaker assumptions [7]. While Carathéodory’s theorem can be cast as a linear system and thus be implemented in polynomial time, very little is known about the algorithmic complexity of the colorful Carathéodory theorem [4]. This question

* Supported in part by DFG Grants MU 3501/1 and MU 3501/2.

† Supported by the Deutsche Forschungsgemeinschaft within the research training group “Methods for Discrete Structures” (GRK 1408).



© Wolfgang Mulzer and Yannik Stein;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG’15).

Editors: Lars Arge and János Pach; pp. 44–58



Leibniz International Proceedings in Informatics

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

is particularly interesting because Sarkaria’s proof [13] of Tverberg’s theorem¹ [15] gives a polynomial-time reduction from computing Tverberg partitions to computing a colorful choice with the origin in its convex hull. Both problems lie in *Total Function NP* (TFNP), the complexity class of total search problems that can be solved in non-deterministic polynomial time. It is well known that no problem in TFNP is NP-hard unless $\text{NP} = \text{coNP}$ [5]. Recently, Meunier and Sarrabezolles [8] have shown that a related problem is complete for a subclass of TFNP: given $d + 1$ pairs of points $P_1, \dots, P_{d+1} \in \mathbb{Q}^d$ and a colorful choice that contains the origin in its convex hull, it is PPAD-complete [12] to find another colorful choice that contains the origin in its convex hull.

Since we have no exact polynomial-time algorithms for the colorful Carathéodory theorem, approximation algorithms are of interest. This was first considered by Bárány and Onn [4] who described how to find a colorful choice whose convex hull is “close” to the origin. Let $\varepsilon, \rho > 0$ be parameters. We call a set ε -close if its convex hull has distance at most ε to the origin. Given sets $P_1, \dots, P_{d+1} \in \mathbb{Q}^d$ s.t. (i) each P_i contains a ball of radius ρ centered at the origin in its convex hull, (ii) all points $p \in P_i$ fulfill $1 \leq \|p\| \leq 2$, and (iii) the points in all sets can be encoded using L bits, one can find a colorful choice C that is ε -close to the origin in time $\text{poly}(L, \log(1/\varepsilon), 1/\rho)$ on the WORD-RAM with logarithmic costs. If $1/\rho = O(\text{poly}(L))$, the algorithm actually finds a colorful choice with the origin in its convex hull.

However, when using the colorful Carathéodory theorem in the proof of another statement, it is often crucial that the convex hull of the colorful choice contains the origin. Being “close” is not enough. On the other hand, allowing multiple points from each color class may have a natural interpretation in the reduction. For example, this is the case in Sarkaria’s proof [13] of Tverberg’s theorem, in the proof of the First Selection Lemma² [6, Theorem 9.1.1], and in the proof of the colorful Kirchberger theorem³ [2]. This motivates a different notion of approximation: we need a “colorful” set with the origin in its convex hull, but we may take more than one point from each color. More formally, given a parameter m and sets $P_1, \dots, P_{d+1} \in \mathbb{Q}^d$, find a set C s.t. $\vec{0} \in \text{conv}(C)$ and s.t. for all P_i , we have $|C \cap P_i| \leq m$. In contrast to the setting considered by Bárány and Onn, we have no general position assumption. Surprisingly, this notion does not seem to have been studied before.

Coming from another direction, as a first step towards understanding what makes the problem hard, we consider the *Nearest Colorful Polytope (NCP) problem*, a natural generalization inspired by the proof of Theorem 1.1. Given color classes $P_1, \dots, P_n \subset \mathbb{R}^d$, not necessarily containing the origin in their convex hulls, find a colorful choice whose convex hull minimizes the distance to the origin. We study two variants: the local search problem, where we want to find a colorful choice whose convex hull cannot be brought closer to the origin by exchanging a *single* point with another point of the same color; and the global search problem, where we want to compute a colorful choice with minimum distance to the origin. We refer to these problems as L-NCP and G-NCP, respectively. L-NCP is particularly interesting since Bárány’s proof of the colorful Carathéodory theorem gives a local search algorithm. The NP-hardness proof of G-NCP settles an open problem by Bárány and Onn [4]. This question was also answered independently by Meunier and Sarrabezolles [8].

¹ Tverberg’s theorem states that a point set $P \subset \mathbb{R}^d$ can be partitioned into $\lceil |P|/(d+1) \rceil$ sets whose convex hulls have a nonempty intersection.

² Let $P \subset \mathbb{R}^d$. Then, the First Selection Lemma guarantees that there is a point contained in “many” simplices that are defined by $d+1$ points in P .

³ The colorful Kirchberger theorem says that given “many” Tverberg partitions, there is a Tverberg partition containing exactly one point from each Tverberg partition.

1.1 Our Results

Given sets $P_1, \dots, P_n \subset \mathbb{R}^d$, we call a set C containing at most m points from each set P_i an m -colorful choice. A 1-colorful choice is also called *perfect colorful choice*. All presented algorithms are analyzed on the REAL-RAM model with unit costs. We begin with an approximation algorithm based on a simple dimension reduction argument.

► **Proposition 1.2.** *Let $P_1, \dots, P_{\lfloor d/2 \rfloor + 1} \subset \mathbb{R}^d$ be $\lfloor d/2 \rfloor + 1$ sets of size at most $d + 1$ that each contain the origin in their convex hulls. Then, a $(\lfloor d/2 \rfloor + 1)$ -colorful choice containing the origin in its convex hull can be computed in $O(d^5)$ time.*

Generalizing the algorithm from Proposition 1.2, we can further improve the approximation guarantee by repeatedly combining approximations for lower dimensional linear subspaces. This can be seen as a counterpart to Mulzer and Werner’s approximation algorithm for Tverberg partitions [11].

► **Theorem 1.3.** *Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be sets of size at most $d + 1$ s.t. $\vec{0} \in \text{conv}(P_i)$ for all $i = 1, \dots, d + 1$. Then, for any $\varepsilon = \Omega(d^{-1/6})$, an $\lceil \varepsilon d \rceil$ -colorful choice containing the origin in its convex hull can be computed in $d^{O((1/\varepsilon) \ln(1/\varepsilon))}$ time.*

In particular, for any constant ε the algorithm from Theorem 1.3 runs in polynomial time. Given $\Theta(d^2 \log d)$ color classes, we can also improve the naive $d^{O(d)}$ algorithm for finding a perfect colorful choice. This algorithm follows the structure of Miller and Sheehy’s approximation algorithm for Tverberg partitions [10].

► **Proposition 1.4.** *Let $P_1, \dots, P_n \subset \mathbb{R}^d$ be $n = \Theta(d^2 \log d)$ sets of size at most $d + 1$ s.t. $\vec{0} \in \text{conv}(P_i)$, for $i = 1, \dots, n$. Then, a perfect colorful choice can be computed in $d^{O(\log d)}$ time.*

On the other hand, if we are given only two color classes, we can achieve a $d - \Theta(\sqrt{d})$ approximation guarantee. Note that a $\lceil (d + 1)/2 \rceil$ -colorful choice is the best possible in this scenario if we assume general position.

► **Proposition 1.5.** *Let $P, Q \subset \mathbb{R}^d$ be two sets of size at most $d + 1$ that contain the origin in their convex hulls. Then, a $(d - \Theta(\sqrt{d}))$ -colorful choice can be computed in $O(d^4)$ time.*

On the hardness side, we show that a generalization of the colorful Carathéodory problem, the *Local Search Nearest Colorful Polytope (L-NCP)* problem, is complete for the complexity class *polynomial-time local search (PLS)*. Using essentially the same reduction, we can also prove that finding a global optimum for NCP (G-NCP) is NP-hard and answer a question by Bárány and Onn [4].

► **Theorem 1.6.** *L-NCP is PLS-complete.*

► **Theorem 1.7.** *G-NCP is NP-hard.*

2 Approximating the Colorful Carathéodory Theorem

Throughout the paper, we denote for a given point set $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$ by

- $\text{span}(P) = \{\sum_{i=1}^n \alpha_i p_i \mid \alpha_i \in \mathbb{R}\}$ its linear span and by $\text{span}(P)^\perp = \{v \in \mathbb{R}^d \mid \forall p \in \text{span}(P) : \langle v, p \rangle = 0\}$ the subspace orthogonal to $\text{span}(P)$;
- $\text{aff}(P) = \{\sum_{i=1}^n \alpha_i p_i \mid \alpha_i \in \mathbb{R}, \sum_{i=1}^n \alpha_i = 1\}$ its affine hull;
- $\text{pos}(P) = \{\sum_{i=1}^n \mu_i p_i \mid \mu_i \geq 0\}$ all linear combinations with nonnegative coefficients;

- $\text{conv}(P) = \{\sum_{i=1}^n \lambda_i p_i \mid \lambda_i \geq 0, \sum_{i=1}^n \lambda_i = 1\}$ its convex hull; and by
- $\text{dim}(P)$ the dimension of $\text{span}(P)$.

Furthermore, we say that a set $P \subset \mathbb{R}^d$ is in *general position* if for every $k \leq d$, no $k + 2$ points lie in a k -flat *and* if no proper subset of P contains the origin in its convex hull. We also use the following constructive version of Carathéodory's theorem:

► **Lemma 2.1.** *Let $P \subset \mathbb{R}^d$ be a set of $O(d)$ points that contains the origin in its convex hull. In $O(d^4)$ time, we can find a subset $P' \subseteq P$ of at most $d + 1$ points in general position such that P' contains the origin in its convex hull.*

2.1 Simple Approximations

Since there are no known approximation algorithms for computing m -colorful choices, even simple ones are of interest to gain some intuition for the problem. It is a straightforward exercise to show that a $(d - 1)$ -colorful choice can be computed in polynomial time. However, even $m = d - 2$ seems to be nontrivial.

In this section, we present two algorithms that both compute a $(d + 1)/2$ -colorful choice in $O(d^5)$ time, but differ in the number of required color classes. The following lemma is the key ingredient of both algorithms. It enables us to replace each color class P_i by two points v_1, v_2 , so that each point represents half of the points in P_i . We call the points v_1, v_2 *representatives* for P_i . Now, a perfect colorful choice for the representatives will correspond to a $\lceil (d + 1)/2 \rceil$ -colorful choice for the original points. The presented algorithms differ only in the way the perfect colorful choice is computed for this special case of the colorful Carathéodory problem. The first one uses basic linear algebra, while the second one is based on a simple dimension reduction argument.

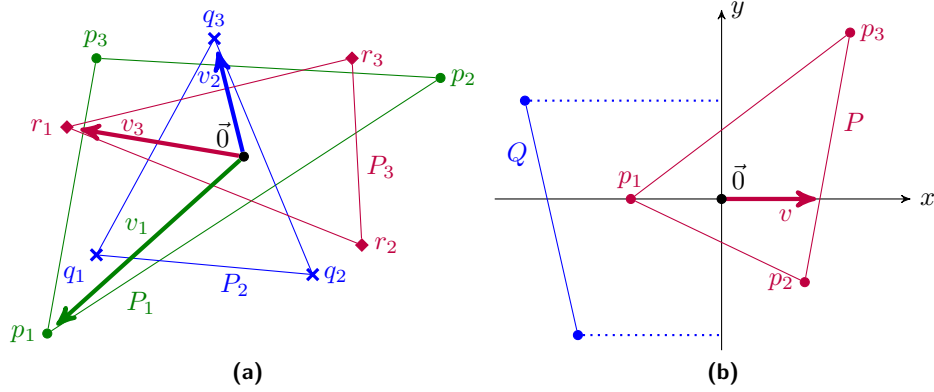
► **Lemma 2.2.** *Let $P \subset \mathbb{R}^d$, $2 \leq |P| \leq d + 1$, be a set in general position that contains the origin in its convex hull. Then, for every partition of P into two sets P_1, P_2 , there is a vector $v \neq \vec{0}$ s.t. $v \in \text{pos}(P_1)$ and $-v \in \text{pos}(P_2)$. This vector can be found in $O(d^3)$ time.*

Proof. Write $\vec{0}$ as $\vec{0} = \sum_{p \in P} \lambda_p p$, such that $\lambda_p \geq 0$ for all $p \in P$ and such that $\sum_{p \in P} \lambda_p = 1$. The coefficients λ_p can be computed in $O(d^3)$ time. Since P is in general position, we have $\lambda_p > 0$ for all $p \in P$. Set $v = \sum_{p \in P_1} \lambda_p p$. By construction, we have $v \neq \vec{0}$, $v \in \text{pos}(P_1)$, and $-v \in \text{pos}(P_2)$. ◀

In the first algorithm, we partition each set P_i into two sets $P_{i,1}, P_{i,2}$ of equal size and apply Lemma 2.2 to obtain $d + 1$ representatives v_1, \dots, v_{d+1} . The set $\{v_1, \dots, v_{d+1}\}$ must be linearly dependent. Depending on the sign of the coefficients in the nontrivial $\vec{0}$ -combination, we replace each representative v_i by either $P_{i,1}$ or $P_{i,2}$.

► **Proposition 2.3.** *Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be $d + 1$ sets s.t. $|P_i| \leq d + 1$ and s.t. P_i contains the origin in its convex hull, for $i = 1, \dots, d + 1$. Then, a $\lceil (d + 1)/2 \rceil$ -colorful choice can be computed in $O(d^5)$ time.*

Proof. First, prune each set P_i , $i = 1, \dots, d + 1$, with Lemma 2.1. This requires $O(d^5)$ time. Assume w.l.o.g. that all sets still contain at least two points (since otherwise at least one set contains the origin). Partition each set P_i arbitrarily into two sets $P_{i,1}, P_{i,2}$ of equal size and let v_1, \dots, v_{d+1} be the vectors obtained by applying Lemma 2.2 to the partitions. Since these vectors are linearly dependent, we can express $\vec{0}$ as $\vec{0} = \sum_{i=1}^{d+1} \mu_i v_i$ where $\mu_j \neq 0$ for at least one $j \in \{1, \dots, d + 1\}$. The coefficients μ_i can be computed in $O(d^3)$ time by solving a linear system of equations. For each vector v_i with $\mu_i > 0$, take $P_{i,1}$ (since $v_i \in \text{pos}(P_{i,1})$),



■ **Figure 1 (a)** Example of Proposition 2.3 in two dimensions. The color classes are partitioned into $P_1 = \{p_1\} \dot{\cup} \{p_2, p_3\}$, $P_2 = \{q_3\} \dot{\cup} \{q_1, q_2\}$, and $P_3 = \{r_1\} \dot{\cup} \{r_2, r_3\}$. The set $C = \{p_1\} \dot{\cup} \{q_3\} \dot{\cup} \{r_2, r_3\}$ is a 2-colorful choice. **(b)** Example of Proposition 1.2 in two dimensions. The representative v is computed for the partition $P = \{p_2, p_3\} \dot{\cup} \{p_1\}$. W.l.o.g. assume v lies on the x -axis. The set Q is a recursively computed approximation that contains the origin in its convex hull if projected onto the y -axis. The set $C = Q \cup \{p_2, p_3\}$ is a 2-colorful choice containing the origin in its convex hull.

otherwise $P_{i,2}$ (since $-v_i \in \text{pos}(P_{i,2})$). Figure 1(a) shows an example in two dimensions. The overall running time is dominated by the initial pruning step. ◀

Lemma 2.2 can also be used to reduce the dimension by one. We repeat this until the dimension is small enough, i.e., $\lceil d/2 \rceil$, and then simply apply Lemma 2.1 in the low dimensional space. This algorithm requires only $\lfloor d/2 \rfloor + 1$ color classes instead of $d + 1$. We will generalize it in the next section.

Proof of Proposition 1.2. We prune P_1 with Lemma 2.1. If $|P_1| = 1$, we have $P_1 = \{\vec{0}\}$, and P_1 is a valid approximation. If $|P_1| \geq 2$, we partition P_1 arbitrarily into two sets $P_{1,1}, P_{1,2}$ of equal size. We apply Lemma 2.2 to obtain a vector v . We project the remaining color classes onto the orthogonal subspace $\text{span}(v)^\perp$ and recursively compute a $(\lceil d/2 \rceil + 1)$ -colorful choice \tilde{C} for the projection. Let C' be the d -dimensional point set corresponding to \tilde{C} . If the convex hull of C' intersects $\text{pos}(v)$, we set $C = C' \cup P_{1,2}$ (since $-v \in \text{pos}(P_{i,2})$), otherwise, we set $C = C' \cup P_{i,1}$ (since $v \in \text{pos}(P_{i,1})$). In both cases, C is a $(\lceil d/2 \rceil + 1)$ -colorful choice with the origin in its convex hull. See Figure 1(b). If only one color is left, i.e., if we are in dimension $d - \lfloor d/2 \rfloor = \lceil d/2 \rceil$, we prune this color with Lemma 2.1 and we return the resulting set of size at most $\lceil d/2 \rceil + 1$.

Each invocation of Lemma 2.1 and of Lemma 2.2 takes $O(d^4)$ time. The recursion depth is bounded by $\lfloor d/2 \rfloor + 1$, which results in a total running time of $O(d^5)$, as claimed. ◀

2.2 Approximation by Rebalancing

The algorithm from Proposition 1.2 prunes half of the points from each color class in a complete run. We generalize this approach in two respects. First, we repeatedly prune points to improve the approximation guarantee. Second, we reduce the dimensionality in each step by more than one to improve the running time.

Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be the color classes and $\lceil \varepsilon d \rceil$ be the desired approximation guarantee. Throughout the execution of the algorithm, we maintain a temporary approximation $C \subset P_1 \cup \dots \cup P_{d+1}$ that contains the origin in its convex hull, but may have more than $\lceil \varepsilon d \rceil$

points of the same color. Initially, C is a complete color class. Using the following lemma, we can replace a single point in C by an approximate colorful choice for the orthogonal space $\text{span}(C)^\perp$.

► **Lemma 2.4.** *Let $C \subset \mathbb{R}^d$, $|C| = k \leq d + 1$, be a set in general position that contains the origin in its convex hull. Furthermore, let $Q \subset \mathbb{R}^d$ be a set of size $O(d)$ whose orthogonal projection onto $\text{span}(C)^\perp$ contains the origin in its convex hull. Then, there is a point $c \in C$ computable in $O(d^4)$ time s.t. $\vec{0} \in \text{conv}(Q \cup C \setminus \{c\})$.*

Proof. Write Q as $Q = \{q_1, \dots, q_l\}$. Each q_i can be expressed as $\tilde{q}_i + \hat{c}_i$, where \tilde{q}_i denotes the orthogonal projection of q_i onto $\text{span}(C)^\perp$ and $\hat{c}_i \in \text{span}(C)$. By our assumption, the origin is a convex combination of $\tilde{q}_1, \dots, \tilde{q}_l$: $\vec{0} = \sum_{i=1}^l \lambda_i \tilde{q}_i$, where $\lambda_i \geq 0$ and $\sum_{i=1}^l \lambda_i = 1$. Consider the convex combination $q = \sum_{i=1}^l \lambda_i q_i$ of points in Q with the same coefficients. Since $q = \sum_{i=1}^l \lambda_i q_i = \sum_{i=1}^l \lambda_i (\tilde{q}_i + \hat{c}_i) = \sum_{i=1}^l \lambda_i \hat{c}_i$, q is contained in $\text{span}(C)$.

By our assumption, we have $\vec{0} \in \text{conv}(C)$. Since C is in general position, this implies $\text{pos}(C) = \text{span}(C)$. Thus, there are $k - 1$ points $c_{j_1}, \dots, c_{j_{k-1}}$ in C s.t. $\text{pos}(c_{j_1}, \dots, c_{j_{k-1}})$ contains $-q$. We can take $c \in C$ as the single point that does not appear in $c_{j_1}, \dots, c_{j_{k-1}}$.

This point can be found in $O(d^4)$ time by solving $k \leq d + 1$ linear equation systems L_1, \dots, L_k , where L_j is defined as $\sum_{c_i \in C, i \neq j} \alpha_i c_i = -q$. Since C is in general position, all $(k - 1)$ -subsets of C are a basis for $\text{span}(C)$. Thus, the linear systems have unique solutions. Furthermore, because C contains the origin in its convex hull, one of the linear systems has a solution with no negative coefficients. ◀

Unfortunately, we cannot control which point is replaced when applying Lemma 2.4. We always want to replace a point whose color appears more than $\lceil \varepsilon d \rceil$ times in C . Generalizing Lemma 2.2, the next lemma enables us to compute representatives for partitions of arbitrary size. Instead of applying Lemma 2.4 to C , we replace one of the representatives for C . By choosing the partition for the representatives appropriately, we can influence the color of the removed points.

► **Lemma 2.5.** *Let $C \subset \mathbb{R}^d$, $|C| \leq d + 1$, be a set in general position that contains the origin in its convex hull and let C_1, \dots, C_m be a partition of C . Then, we can find in $O(d^3)$ time a set $C' = \{c'_1, \dots, c'_m\} \subset \mathbb{R}^d$ with the following properties:*

1. $\forall i = 1, \dots, m: c'_i \in \text{pos}(C_i) \setminus \{\vec{0}\}$
2. $\vec{0} \in \text{conv}(C')$
3. $\dim(C') = m - 1$

We call the points in C' representatives for C with respect to the partition C_1, \dots, C_m .

Proof. Since C contains the origin in its convex hull, we can write $\vec{0}$ as $\vec{0} = \sum_{c \in C} \lambda_c c$, where all $\lambda_c > 0$, since C is in general position. Define c'_j as $c'_j = \sum_{c \in C_j} \lambda_c c$ for all $i = 1, \dots, m$. Properties 1. and 2. can be easily verified for the set $C' = \{c'_1, \dots, c'_m\}$. Furthermore, c'_1 can be expressed as a linear combination of the other points in C' : $c'_1 = -(c'_2 + \dots + c'_m)$. Thus, $\dim(C') < m$. On the other hand, we have $\dim(C') \geq m - 1$ due to general position. This proves Property 3. ◀

Now, we are ready to put everything together. The algorithm repeatedly replaces points in C by a recursively computed approximate colorful choice for a linear subspace. We are given as input the color classes $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$, each containing the origin in its convex hull, a recursion depth threshold $j_{\max} \in \mathbb{N}$ and two parameter functions $\mathcal{M}, \mathcal{D} : \mathbb{N}_0 \rightarrow \mathbb{N}$ that control the dimension reduction. The first function returns for a given recursion depth the desired approximation guarantee. After completion, the algorithm outputs an $\mathcal{M}(0)$ -colorful

choice. The second function, $\mathcal{D} : \mathbb{N}_0 \rightarrow \mathbb{N}$, controls the dimension reduction. It returns for a given recursion depth j the desired dimension of the problem. We require the parameter functions to have the following properties.

► **Definition 2.6** (Feasible Parameter Functions). Let $\mathcal{M}, \mathcal{D} : \mathbb{N}_0 \rightarrow \mathbb{N}$ be two functions. We call $(\mathcal{M}, \mathcal{D})$ j_{\max} -feasible if the functions fulfill the following conditions

1. \mathcal{M} and \mathcal{D} are strictly decreasing over the interval $[0, j_{\max} - 1]$ and can be computed in $O(d^4)$ time;
2. $\mathcal{D}(0) = d$; and
3. for all $j < j_{\max}$, the following inequalities hold

$$\left\lfloor \frac{\mathcal{D}(j) + 1}{\mathcal{M}(j) - \mathcal{M}(j+1)} \right\rfloor \stackrel{(i)}{\leq} \mathcal{D}(j) - \mathcal{D}(j+1) \stackrel{(ii)}{\leq} \mathcal{M}(j).$$

Suppose we have a j_{\max} -feasible pair $(\mathcal{M}, \mathcal{D})$ of parameter functions and we are at recursion depth j . As long as the parameter functions are feasible, that is $j < j_{\max}$, we apply our dimension reduction argument. Otherwise, we compute a perfect colorful choice by brute-force.

Assume we have not yet reached the recursion depth threshold ($j < j_{\max}$). That is, the input points are $\mathcal{D}(j)$ -dimensional and we want to compute an $\mathcal{M}(j)$ -colorful choice. We initialize the temporary approximation C with a complete color class and prune it with Lemma 2.1. As long as C is not an $\mathcal{M}(j)$ -colorful choice, we repeat the following steps: we partition C into $k = \mathcal{D}(j) - \mathcal{D}(j+1) + 1$ sets C_1, \dots, C_k , where the points from each color in C are distributed evenly among the k sets. Let $n_i = |P_i \cap C|$ denote the number of points from P_i in C . Since the parameter functions are feasible, we have $k \leq \mathcal{M}(j) + 1$. Hence, each set in the partition contains at least one point from each color class P_i for which $n_i \geq \mathcal{M}(j) + 1$. Applying Lemma 2.5, we compute representatives $C' = \{c'_1, \dots, c'_k\}$ for this partition. Note that $\dim(C') = k - 1$ and that $\dim(\text{span}(C')^\perp) = \mathcal{D}(j) - k + 1 = \mathcal{D}(j+1)$.

We call a color class P_i *light* if $n_i \leq \mathcal{M}(j) - \mathcal{M}(j+1)$; otherwise we call P_i *heavy*. Light color classes can be reused in the recursion since adding an $\mathcal{M}(j+1)$ -colorful choice that consists of points from light color classes to our temporary approximation C does not increase the amount of points from any color class over the desired approximation guarantee $\mathcal{M}(j)$. We find $\mathcal{D}(j+1) + 1$ light color classes and project these orthogonally onto $\text{span}(C')^\perp$. Let $\tilde{P}_{j_1}, \dots, \tilde{P}_{j_{\mathcal{D}(j+1)+1}}$ denote the projections. Next, we recursively compute an $\mathcal{M}(j+1)$ -colorful choice \tilde{Q} for the space orthogonal to $\text{span}(C')$ with $(\tilde{P}_{j_1}, \dots, \tilde{P}_{j_{\mathcal{D}(j+1)+1}}, j+1, \mathcal{M}, \mathcal{D}, j_{\max})$ as input. Let Q be the point set whose projection gives \tilde{Q} . Using Lemma 2.4, we compute a point $c'_j \in C'$ s.t. $\text{conv}(Q \cup C' \setminus c'_j)$ contains the origin. We replace the subset C_j of C by Q and prune C again with Lemma 2.1. Since each representative c'_i is contained in the cone $\text{pos}(C_i)$, $Q \cup C \setminus C_j$ still contains the origin in its convex hull and hence the invariant is maintained. Thus, in one iteration of the algorithm, at least one point from each color class P_i for which $n_i > \mathcal{M}(j)$ is replaced by points from light color classes. This is repeated until no color class appears more than $\mathcal{M}(j)$ times in C . See Algorithm 2.1 for pseudocode.

We first prove correctness and afterwards analyze the running time for a specific pair of feasible parameter functions.

► **Lemma 2.7** (Correctness of Algorithm 2.1). *Let $P_1, \dots, P_{d+1} \subset \mathbb{R}^d$ be sets s.t. $|P_i| \leq d+1$ and s.t. $\vec{0} \in \text{conv}(P_i)$, for $i = 1, \dots, d+1$. Furthermore, let $\mathcal{M}, \mathcal{D} : \mathbb{N}_0 \rightarrow \mathbb{N}$ be a pair of j_{\max} -feasible parameter functions. On input $(P_1, \dots, P_{d+1}, 0, \mathcal{M}, \mathcal{D}, j_{\max})$, Algorithm 2.1 returns an $\mathcal{M}(0)$ -colorful choice.*

Algorithm 2.1: Approximation by Rebalancing

input: $P_1, \dots, P_{d'+1} \subset \mathbb{R}^{d'}$ s.t. $\vec{0} \in \text{conv}(P_i)$ for all $i = 1, \dots, d' + 1$, recursion depth $j \in \mathbb{N}_0$ (initially 0), approximation parameter function $\mathcal{M} : \mathbb{N}_0 \rightarrow \mathbb{N}$, dimension parameter function $\mathcal{D} : \mathbb{N}_0 \rightarrow \mathbb{N}$, recursion depth threshold j_{\max}

- 1 **if** $j = j_{\max}$ **then**
- 2 | **return** *brute force computed perfect colorful choice*
- 3 $C \leftarrow P_1$
- 4 Prune C with Lemma 2.1.
- 5 $d'' \leftarrow \mathcal{D}(j + 1)$; $k \leftarrow d' - d'' + 1$
- 6 **while** C is not an $\mathcal{M}(j)$ -colorful choice **do**
- 7 | Partition C into k sets C_1, \dots, C_k s.t. for all color classes P_i and all pairs of indices $1 \leq l_1, l_2 \leq k$, we have $|\#(P_i \cap C_{l_1}) - \#(P_i \cap C_{l_2})| \leq 1$.
- 8 | Apply Lemma 2.5 to C_1, \dots, C_k . Let $C' = \{c'_1, \dots, c'_k\}$ be the set of the representatives.
- 9 | Find $d'' + 1$ color classes $P_{j_1}, \dots, P_{j_{d''+1}}$ s.t. $|C \cap P_{j_i}| \leq \mathcal{M}(j) - \mathcal{M}(j + 1)$.
- 10 **for** $i = 1$ to $d'' + 1$ **do**
- 11 | $\tilde{P}_{j_i} \leftarrow$ orthogonal projection of P_{j_i} onto $\text{span}(C')^\perp$
- 12 | $Q \leftarrow \text{recurse}(\tilde{P}_{j_1}, \tilde{P}_{j_2}, \dots, \tilde{P}_{j_{d''+1}}, j + 1, \mathcal{M}, \mathcal{D}, j_{\max})$
- 13 | Apply Lemma 2.4 to C' and Q to find a point $c'_i \in C'$ s.t. $\vec{0} \in \text{conv}(Q \cup C' \setminus \{c'_i\})$.
- 14 | $C \leftarrow \left(\bigcup_{j=1, j \neq i}^{k+1} C_j \right) \cup Q$
- 15 | Prune C with Lemma 2.1.
- 16 **return** C

Proof. We prove correctness by showing that the algorithm respects the parameter functions \mathcal{D} and \mathcal{M} . By our discussion above it is clear that the dimension in the j th recursion is $\mathcal{D}(j)$ for $j < j_{\max}$. Next, we show that in the j th recursion, the returned colorful choice is an $\mathcal{M}(j)$ -colorful choice. The prove is by induction on the recursion depth. We have two base cases. First, if $j = j_{\max}$, a perfect colorful choice is computed in line 2. Since $\mathcal{M}(j) \geq 1$, a perfect colorful choice is always an $\mathcal{M}(j)$ -colorful choice. Second, if C pruned with Lemma 2.1 in line 4 or line 15 is already an $\mathcal{M}(j)$ -colorful choice, the algorithm terminates, too. Hence, the induction hypothesis holds in both base cases. Assume now that the current recursion depth is $j < j_{\max}$ and the induction hypothesis holds for all $j' > j$. Let $C^{(t)}$ denote the set C after t iterations of the while-loop in the j th recursion. We show the following invariant:

- (α) $\vec{0} \in \text{conv}(C^{(t)})$,
- (β) for all color classes $P_i, i = 2, \dots, d + 1$, we have $|C^{(t)} \cap P_i| \leq \mathcal{M}(j)$, and
- (γ) $|C^{(t-1)} \cap P_1| > |C^{(t)} \cap P_1|$, for $t \geq 1$.

The invariant implies that the while-loop terminates and an $\mathcal{M}(j)$ -colorful choice is returned. Before the first iteration, the invariant holds since $C^{(0)} = P_1$. Assume we are now in iteration t and the invariant holds for all previous iterations. Due to Lemmas 2.5 and 2.4, we have $\vec{0} \in \text{conv}(C^{(t)})$ and thus Property (α) holds. By the induction hypothesis, the recursively computed set Q in line 12 is an $\mathcal{M}(j + 1)$ -colorful choice. Since we use only light color classes in the recursion, adding the points from Q to $C^{(t)}$ does not violate Property (β) of the invariant. It remains to show that we can always find $\mathcal{D}(j + 1) + 1$ light color classes. Since C is pruned to at most $\mathcal{D}(j) + 1$ points at the end of each while-loop iteration, the number of heavy color classes is upper bounded by $\left\lfloor \frac{\mathcal{D}(j)+1}{\mathcal{M}(j)-\mathcal{M}(j+1)} \right\rfloor$. This is at most $\mathcal{D}(j) - \mathcal{D}(j + 1)$ since \mathcal{M}, \mathcal{D} are feasible in the current recursion depth. Therefore, there are always at least $\mathcal{D}(j + 1) + 1$ light color classes.

Finally, we need to check that the number of points from P_1 in $C^{(t)}$ is strictly less than in $C^{(t-1)}$. Again, since \mathcal{M}, \mathcal{D} are feasible in recursion depth j , we have $\mathcal{M}(j) + 1 \geq \mathcal{D}(j) - \mathcal{D}(j+1) + 1 = k$. Since $C^{(t-1)}$ was not an $\mathcal{M}(j)$ -colorful choice (otherwise the while-loop would have terminated), $C^{(t-1)}$ contains at least $\mathcal{M}(j) + 1$ points from P_1 . Hence, each set C_i in line 7 contains at least one point from P_1 . Since one of these sets is removed in line 14 and Q does not contain the color P_1 , Property (γ) of the invariant also holds. \blacktriangleleft

► **Remark.** Before the applications of Lemmas 2.4 and 2.5 in Algorithm 2.1, we ensure general position by pruning the points with Lemma 2.1. Hence although Lemmas 2.4 and 2.5 require general position, the input of Algorithm 2.1 does not need to be in general position.

Proof of Theorem 1.3. We use Algorithm 2.1 with parameter functions $\mathcal{M}(j) = \lceil \varepsilon(1 - \varepsilon/2)^{j/2} d \rceil$ and $\mathcal{D}(j) = \lceil (1 - \varepsilon/2)^j d \rceil$. In particular, we reduce the dimension by $(\varepsilon/2)d$ in each step of the recursion. However, in the j th recursion, we do not compute an $\lceil \varepsilon \mathcal{D}(j) \rceil$ -colorful choice, but a $\lceil (1 - \varepsilon)^{-j/2} \varepsilon \mathcal{D}(j) \rceil$ -colorful choice. This “slack” increases throughout the recursion. It can be shown that \mathcal{M} and \mathcal{D} are $(\frac{4}{3\varepsilon}(\ln(\varepsilon^3 d) - O(1)))$ -feasible. The proof is rather tedious and thus omitted from this extended abstract due to the space limitation. It can be found in the full version. Now, Lemma 2.7 guarantees correctness.

It remains to analyze the running time. If the dimension becomes smaller than the desired approximation guarantee, that is $\mathcal{D}(j) + 1 \leq \mathcal{M}(j)$, pruning C with Lemma 2.1 in line 4 already gives a valid approximation. For $\varepsilon = \Omega(d^{-1/5})$, it can be shown that $\mathcal{M}(j_*) \geq \mathcal{D}(j_*) + 1$ for $j_* = \lceil (4/\varepsilon) \ln(2/\varepsilon) \rceil$. Now, for $\varepsilon = \Omega(d^{-1/6})$, the parameter functions are feasible up to recursion depth j_* . Hence, the algorithm does not terminate with computing a perfect colorful choice by brute force in line 2, but always with a pruning step.

During each iteration of the while-loop, the maximum number of points from each color class is reduced by one until the desired approximation guarantee is reached. Thus, the total number of iterations is bounded by $\mathcal{D}(j) + 1 - \mathcal{M}(j) = O(d)$. Each iteration requires $O(\mathcal{D}(j)^4) = O(d^4)$ time. This results in $d^{O((1/\varepsilon) \ln(1/\varepsilon))}$ total running time as claimed. \blacktriangleleft

2.3 Varying the Number of Color Classes

First, we consider the case that we have “many” color classes: given $\Theta(d^2 \log d)$ color classes, our algorithm computes a perfect colorful choice in $d^{O(\log d)}$ time by repeatedly combining m -colorful choices (for some m) to one $\lceil m/2 \rceil$ -colorful choice. The algorithm follows the structure of the Miller-Sheehy approximation algorithm for Tverberg partitions [10] and improves the brute force $d^{O(d)}$ algorithm. Second, we present an algorithm that computes a $(d - \Theta(\sqrt{d}))$ -colorful choice given only two color classes in $O(d^4)$ time.

► **Lemma 2.8.** *Let $C_1, \dots, C_{d+1} \subset \mathbb{R}^d$ be m -colorful choices s.t. $|C_i| \leq d + 1$ and s.t. $\vec{0} \in \text{conv}(C_i)$ for $i = 1, \dots, d + 1$. Furthermore, no color appears in more than one set C_i . Then, a $\lceil m/2 \rceil$ -colorful choice C s.t. $\vec{0} \in \text{conv}(C)$ can be computed in $O(d^5)$ time.*

Proof. First, we prune each set C_i with Lemma 2.1. This requires $O(d^5)$ time. Next, we proceed as in the proof of Proposition 2.3 where we treat the sets C_i as the color classes. This time however, we do not partition a set C_i into two *arbitrary* sets $C_{i,1}, C_{i,2}$ of equal size, but we distribute the points from each color class in C_i evenly among the both sets. \blacktriangleleft

Proof of Proposition 1.4. Let A be an array of size $k = \Theta(\log d)$. We set $c_0 = d + 1$ and $c_i = \lceil c_{i-1}/2 \rceil$, for $i = 1, \dots, k - 1$. The i th cell of A stores a collection of c_i -colorful choices, such that each color class appears in exactly one colorful choice in A . Initially, $A[0]$ contains all $\Theta(d^2 \log d)$ color classes. We repeat the following steps, until we have computed a perfect

colorful choice: let i be the maximum index s.t. $A[i]$ contains some $d + 1$ sets C_1, \dots, C_{d+1} . We apply Lemma 2.8 to obtain one c_{i+1} -colorful choice C . Let C' be the set C pruned with Lemma 2.1. If C' is a perfect colorful choice, we return it. Otherwise, we add it to $A[i + 1]$. Furthermore, we add all colors that were removed during the pruning to $A[0]$. As these colors do not appear anywhere else in A , the invariant is maintained. We claim that a combination of $d + 1$ sets in $A[k]$ for $k = \lceil \log(d + 1) \rceil + 1$ results in a perfect colorful choice. We have $c_j \leq \frac{d+1}{2^k} + 2$. Thus, sets in $A[\lceil \log(d + 1) \rceil]$ are 3-colorful choices, sets in $A[\lceil \log(d + 1) \rceil + 1] = A[k]$ are 2-colorful choices and the combination of $d + 1$ sets in $A[k]$ gives a perfect colorful choice. It remains to show that we can always make progress. The array has $k = \Theta(\log d)$ levels and each colorful choice has at most d colors. Thus, for $d^2k + 1 = \Theta(d^2 \log d)$ colors, the pigeonhole principle implies that there is a cell with $d + 1$ sets.

Let us consider the running time. One combination step takes $O(d^5)$ time. To compute a set in level i , we have to compute $d + 1$ sets in level $i - 1$. Hence, computing one set in level $k + 1$ takes $d^{O(\log d)}$ time. ◀

Proof of Proposition 1.5. Let P and Q be the two color classes. Let k be a parameter to be determined later. We prune P with Lemma 2.1 and partition it into k sets P_1, \dots, P_k of equal size. We apply Lemma 2.5 to obtain representatives $P' = \{p'_1, \dots, p'_k\}$ for these sets and project Q onto the $(d - k + 1)$ -dimensional subspace $\text{span}(P')^\perp$. Again, we prune Q with Lemma 2.1 and apply Lemma 2.4 to replace one point p'_i of P' with Q . Thus, the set $C = \bigcup_{j=1, j \neq i}^k P_j \cup Q$ contains the origin its convex hull and has at most $\max\{\lceil (d + 1)(1 - 1/k) \rceil, d - k + 2\}$ points of each color. Setting $k = \Theta(\sqrt{d})$ gives the result. ◀

3 The Nearest Colorful Polytope Problem

The complexity class *Polynomial-Time Local Search* (PLS) contains local search problems for which a single improvement step can be carried out in polynomial time. In contrast to complexity classes for decision problems such as P and NP, the existence of a solution (a local optimum) to a PLS problem is always guaranteed. Instead, the difficulty lies in finding the solution. Mathematically, a PLS problem A is a relation $A \subseteq \mathcal{I} \times \mathcal{S}$, where \mathcal{I} is the set of *problem instances* and \mathcal{S} is the set of *candidate solutions*. The relation A is in PLS if

- problem instances $I \in \mathcal{I}$ and candidate solutions $s \in \mathcal{S}$ are polynomial-time verifiable and the size of the valid candidate solutions for an instance I is polynomial in the size of I ;
- there is a polynomial-time computable function $\mathcal{B} : \mathcal{I} \rightarrow \mathcal{S}$ that returns some candidate solution (the base solution) for each instance;
- there is a polynomial-time computable function $\mathcal{C} : \mathcal{I} \times \mathcal{S} \rightarrow \mathbb{N}$ that assigns *costs* to each instance-solution pair;
- there is a polynomial-time computable neighborhood function $\mathcal{N} : \mathcal{I} \times \mathcal{S} \rightarrow 2^{\mathcal{S}}$ assigning each candidate solution a set of neighboring candidate solutions; and
- for every instance $I \in \mathcal{I}$, A contains exactly the pairs (I, s) so that s is a *local optimum* for I ; i.e., all elements in $\mathcal{N}(I, s)$ have smaller costs in a maximization problem and larger costs in a minimization problem.

The computational problem modeled by A is: given $I \in \mathcal{I}$, find an $s \in \mathcal{S}$ s.t. $(I, s) \in A$. The following algorithm is called the *standard algorithm*: start with the base solution $\mathcal{B}(I)$ and use \mathcal{N} to improve until a local optimum is reached. Each iteration takes polynomial time, but the total number of iterations may be exponential. There are examples where it is PSPACE-hard to find the solution given by the standard algorithm [1, Chapter 2].

To define hardness with respect to PLS, we need an appropriate notion of reduction. A *PLS-reduction* from a PLS-problem A to a PLS-problem B is given by two polynomial-time computable functions $f : \mathcal{I}_A \rightarrow \mathcal{I}_B$ and $g : \mathcal{I}_A \times \mathcal{S}_B \rightarrow \mathcal{S}_A$ such that f maps A -instances to B -instances and g maps local optima for B to local optima for A . Thus, if A is PLS-reducible to B , we can convert any algorithm for B into an algorithm for A with polynomial-time overhead. We call B *PLS-complete* if all problems in PLS are PLS-reducible to B .

Like PPAD, PLS is a subset of the class *Total Function NP* (TFNP). TFNP contains search problems whose solution can be verified in polynomial time. No problem in TFNP can be NP-hard unless $\text{NP} = \text{coNP}$ [5]. On the other hand, it is not believed that PLS-complete problems can be solved in polynomial time, although this would not break any assumptions on complexity classes. For more information see one of the several main publications on the topic [1, 9, 14, 5]. In the language of PLS, L-NCP is defined as follows:

► **Definition 3.1** (L-NCP).

Instances \mathcal{I}_{NCP} . Set families $P = \{P_1, \dots, P_n\}$ in \mathbb{R}^d , where each $P_i \subset \mathbb{R}^d$ is a color.

Solutions \mathcal{S}_{NCP} . All perfect colorful choices, i.e., sets with exactly one point of each color.

Cost function \mathcal{C}_{NCP} . Let S_{NCP} be a colorful choice. Then, $\mathcal{C}_{\text{NCP}}(S_{\text{NCP}}) = \|\text{conv}(S_{\text{NCP}})\|_1$, where $\|\text{conv}(S_{\text{NCP}})\|_1 = \min\{\|q\|_1 \mid q \in \text{conv}(S_{\text{NCP}})\}$. We want to minimize \mathcal{C}_{NCP} .

Neighborhood \mathcal{N}_{NCP} . The neighbors $\mathcal{N}_{\text{NCP}}(S_{\text{NCP}})$ of a colorful choice S_{NCP} are all colorful choices that can be obtained by swapping one point with another point of the same color. We reduce the following PLS-complete problem [14, Corollary 5.12] to L-NCP.

► **Definition 3.2** (Max-2SAT/Flip).

Instances $\mathcal{I}_{\text{M2SAT}}$. All weighted 2-CNF formulas $\bigwedge_{i=1}^d C_i$, where each clause C_i is the disjunction of at most two literals and has weight $w_i \in \mathbb{N}_+$.

Solutions $\mathcal{S}_{\text{M2SAT}}$. Let x_1, x_2, \dots, x_n be the variables appearing in the clauses. Then, every complete assignment $\mathcal{A} : \{x_1, \dots, x_n\} \rightarrow \{0, 1\}$ of these variables is a solution.

Cost function $\mathcal{C}_{\text{M2SAT}}$. The cost of an assignment is the sum of the weights of all satisfied clauses. We want to maximize the cost function.

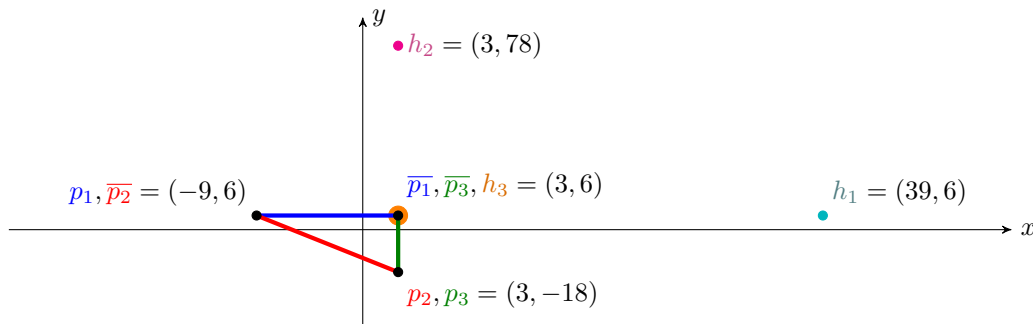
Neighborhood $\mathcal{N}_{\text{M2SAT}}$. The neighbors $\mathcal{N}_{\text{M2SAT}}(\mathcal{A})$ of an assignment \mathcal{A} are all assignments obtained by flipping (i.e., negating) a single variable in \mathcal{A} .

Proof of Theorem 1.6. Let $I_{\text{M2SAT}} = (C_1, \dots, C_d, w_1, \dots, w_d, x_1, \dots, x_n)$ be an instance of M2SAT. We construct an instance I_{NCP} of L-NCP in which each colorful choice encodes an assignment to the variables in I_{M2SAT} . Furthermore, the distance to the origin of the convex hull of a colorful choice in I_{NCP} will be the total weight of all unsatisfied clauses of the encoded assignment for I_{M2SAT} .

For each variable x_i , we introduce a color class $P_i = \{p_i, \bar{p}_i\}$ consisting of two points in \mathbb{R}^d that encode whether x_i is set to 1 or 0. We assign the j th dimension to the j th clause and set $(p_i)_j = -nw_j$, if $x_i = 1$ satisfies clause j , and $(p_i)_j = w_j$, otherwise. Similarly, $(\bar{p}_i)_j = -nw_j$, if $x_i = 0$ satisfies C_j , and $(\bar{p}_i)_j = w_j$ otherwise. A colorful choice S of P_1, \dots, P_n corresponds to the assignment in I_{M2SAT} where x_i is 1 if $p_i \in S$ and 0 if $\bar{p}_i \in S$. More formally, we define a mapping $g : \mathcal{I}_{\text{M2SAT}} \times \mathcal{S}_{\text{NCP}} \rightarrow \mathcal{S}_{\text{M2SAT}}$ between the solutions of the L-NCP instance and the M2SAT instance in the following way:

$$g(I_{\text{M2SAT}}, S_{\text{NCP}})(x_i) = \begin{cases} 1 & \text{if } p_i \in S_{\text{NCP}}, \text{ and} \\ 0 & \text{if } \bar{p}_i \in S_{\text{NCP}}. \end{cases}$$

The main idea is to construct an instance of L-NCP in which the convex hull of a colorful choice S contains the origin if projected onto the dimensions corresponding to the satisfied



■ **Figure 2** Construction of the point sets corresponding to the M2SAT instance $(x_1 \vee \overline{x_2}) \wedge (x_2 \vee x_3)$ with weights 3 and 6, respectively.

clauses. Furthermore, if projected onto the subspace corresponding to the unsatisfied clauses, the distance of $\text{conv}(S)$ to the origin will be equal to the total weight of those clauses.

We introduce additional helper color classes to decrease the distance to the origin in dimensions that correspond to satisfied clauses. In particular, we have for each clause C_j a color class $H_j = \{h_j\}$ consisting of a single point, where

$$(h_j)_k = \begin{cases} (d + 1) \left((n + 2) - \frac{d}{d+1} \right) w_j & \text{if } k = j, \text{ and} \\ w_k & \text{otherwise.} \end{cases}$$

The last helper color class $H_{d+1} = \{h_{d+1}\}$ again contains a single point, but now all coordinates are set to the clause weights, i.e., $(h_{d+1})_j = w_j$ for $j = 1, \dots, d$. See Fig. 2.

The remaining proof is divided into two parts: (i) for every colorful choice S_{NCP} of the L-NCP problem instance $\{P_1, \dots, P_n, H_1, \dots, H_{d+1}\}$, the cost $\mathcal{C}_{\text{NCP}}(S_{\text{NCP}})$ is lower-bounded by the total weight of unsatisfied clauses in $g(S_{\text{NCP}})$; and (ii) this lower bound is tight, i.e., the distance of the convex hull of any colorful choice S_{NCP} to the origin is at most the total weight of unsatisfied clauses in $g(S_{\text{NCP}})$.

Both claims together imply that $\mathcal{C}_{\text{NCP}}(S_{\text{NCP}})$ equals the total weight of unsatisfied clauses for the assignment $g(S_{\text{NCP}})$, which proves the theorem. Consider some local optimum S_{NCP}^* of the L-NCP instance. By definition, the costs of all other colorful choices that can be obtained from S_{NCP}^* by exchanging one point with another of the same color are greater or equal to $\mathcal{C}_{\text{NCP}}(S_{\text{NCP}}^*)$. That is, the total weight of unsatisfied clauses in $g(S_{\text{NCP}}^*)$ cannot be decreased by flipping a variable, which is equivalent to $g(S_{\text{NCP}}^*)$ being a local optimum of the M2SAT instance.

- (i) Let S_{NCP} be a colorful choice and assume some clause C_j is not satisfied by $g(S_{\text{NCP}})$. By construction, the j th coordinate of each point q in S_{NCP} is at least w_j . Thus, the j th coordinate of every convex combination of the points in S_{NCP} is at least w_j . This implies (i).
- (ii) Given a colorful choice S_{NCP} , we construct a convex combination of S_{NCP} that gives a point p whose distance to the origin is exactly the total weight of unsatisfied clauses in $g(S_{\text{NCP}})$. Let in the following part A_k denote the set of clauses C_j that are satisfied by exactly k literals with respect to $g(S_{\text{NCP}})$, for $k = 0, 1, 2$. As a first step towards constructing p , we show the existence of an intermediate point in the convex hull of the helper classes.

► **Lemma 3.3.** *There is a point $h \in \text{conv}(H_1, \dots, H_{d+1})$ whose j th coordinate is $(n + 2)w_j$ if $j \in A_2$ and w_j otherwise.*

Proof. Take $h = \sum_{a \in A_2} \frac{1}{d+1} h_a + \left(1 - \frac{|A_2|}{d+1}\right) h_{d+1}$. Then, for $j \in A_0 \cup A_1$, we have

$$(h)_j = \sum_{a \in A_2} \frac{1}{d+1} (h_a)_j + \left(1 - \frac{|A_2|}{d+1}\right) (h_{d+1})_j \stackrel{j \notin A_2}{=} \sum_{a \in A_2} \frac{1}{d+1} w_j + \left(1 - \frac{|A_2|}{d+1}\right) w_j = w_j.$$

And for $j \in A_2$, we have

$$\begin{aligned} (h)_j &= \sum_{a \in A_2} \frac{1}{d+1} (h_a)_j + \left(1 - \frac{|A_2|}{d+1}\right) (h_{d+1})_j \\ &= \frac{1}{d+1} h_j + \sum_{a \in A_2 \setminus \{j\}} \frac{1}{d+1} (h_a)_j + \left(1 - \frac{|A_2|}{d+1}\right) (h_{d+1})_j \\ &= \left((n+2) - \frac{d}{d+1}\right) w_j + \frac{d}{d+1} w_j = (n+2) w_j, \end{aligned}$$

as desired. \blacktriangleleft

Let $l_i \in P_i$ be the point from P_i in S_{NCP} . Consider $p = \sum_{i=1}^n \frac{1}{n+1} l_i + \frac{1}{n+1} h$. We show that $(p)_j = w_j$, for $j \in A_0$, and $(p)_j = 0$, otherwise. Let us start with $j \in A_0$. Since $g(S_{\text{NCP}})$ does not satisfy C_j , the j th coordinate of the points l_1, \dots, l_n is w_j . Also, $(h)_j = w_j$, by Lemma 3.3. Thus, $(p)_j = w_j$. Consider now some $j \in A_1$ and let b be s.t. the point l_b corresponds to the single literal that satisfies C_j .

$$\begin{aligned} (p)_j &= \sum_{i=1}^n \frac{1}{n+1} (l_i)_j + \frac{1}{n+1} (h)_j \\ &= \frac{1}{n+1} (l_b)_j + \sum_{i=1, i \neq b}^n \frac{1}{n+1} (l_i)_j + \frac{1}{n+1} (h)_j = \frac{-n}{n+1} w_j + \frac{n}{n+1} w_j = 0. \end{aligned}$$

Finally, consider some $j \in A_2$ and let b_1, b_2 be the indices of the two literals that satisfy C_j .

$$\begin{aligned} (p)_j &= \sum_{i=1}^n \frac{1}{n+1} (l_i)_j + \frac{1}{n+1} (h)_j \\ &= \frac{1}{n+1} (l_{b_1})_j + \frac{1}{n+1} (l_{b_2})_j + \sum_{i=1, i \notin \{b_1, b_2\}}^n \frac{1}{n+1} (l_i)_j + \frac{1}{n+1} (h)_j \\ &= \frac{-2n}{n+1} w_j + \frac{n-2}{n+1} w_j + \frac{n+2}{n+1} w_j = 0 \end{aligned}$$

This concludes the proof of (ii). \blacktriangleleft

Proof of Theorem 1.7. The proof of Theorem 1.6 can be adapted easily to reduce 3SAT to G-NCP. Given a set of clauses C_1, \dots, C_d , we set the weight of each clause to 1 and construct the same point sets as in the PLS reduction. Additionally, we introduce for each clause C_j a new helper color class $H'_j = \{h'_j\}$, where

$$(h'_i)_j = \begin{cases} (d+1) \left((2n+2) - \frac{d}{d+1} \right) & \text{if } i = j, \text{ and} \\ 1 & \text{otherwise.} \end{cases}$$

Let S now be any colorful choice and $A = g(S)$ the corresponding assignment. As in the PLS-reduction, we define the sets A_k , $k = 0, \dots, 3$, to contain all clauses that are satisfied

by exactly k literals in the assignment A . Then, the following point h is contained in the convex hull of the helper points:

$$h = \sum_{a \in A_2} \frac{h_a}{d+1} + \sum_{a' \in A_3} \frac{h_{a'}}{d+1} + \left(1 - \frac{|A_2|}{d+1}\right) h_{d+1}.$$

Again, the convex combination $p = \sum_{i=1}^n \frac{1}{n+1} l_i + \frac{1}{n+1} h$ results in a point in the convex hull of S whose distance to the origin is the number of unsatisfied clauses, where $l_i \in P_i$ denotes the point from P_i that is contained in S . Together with Claim (i) from the proof of Theorem 1.6, 3SAT can be decided by knowing a global optimum S^* to the NCP problem: if the distance from $\text{conv}(S^*)$ to the origin is 0, $g(S^*)$ is a satisfying assignment. If not, there exists no satisfying assignment at all. ◀

As mentioned in the introduction, we can adapt the proof of Theorem 1.7 to answer a question by Bárány and Onn [4]. Again, this result was obtained independently by Meunier and Sarrabezolles [8].

► **Corollary 3.4.** *Let $P_1, \dots, P_n \subset \mathbb{R}^d$ be an input for G -NCP. Then, G -NCP is still NP-hard if we require $n = d + 1$.*

Proof. Let F be a 3SAT formula with d clauses and n variables. As in the proof of Theorem 1.7, we construct $n + 2d + 1 =: d' + 1$ point sets in \mathbb{R}^d s.t. there is a colorful choice containing the origin in its convex hull if and only if F is satisfiable. Since $d' > d$, we can lift the point sets to $\mathbb{R}^{d'}$ by appending 0-coordinates. Then, we have $d' + 1$ point sets s.t. there is a colorful choice containing the origin in its convex hull if and only if F is satisfiable. ◀

4 Conclusion

We have proposed a new notion of approximation for the colorful Carathéodory theorem and presented an abstract approximation scheme. By choosing the parameters carefully, we obtain a polynomial-time algorithm that computes $\lceil \varepsilon d \rceil$ -colorful choices for any constant $\varepsilon > 0$. One of the key motivations for studying this kind of approximation was the tight connection to approximating Tverberg's theorem. Here, approximation means computing a Tverberg partition of smaller size than guaranteed by Tverberg's theorem. Unfortunately, if we convert the algorithm from Theorem 1.3 to an approximation algorithm for Tverberg's theorem using Sarkaria's proof, we obtain an algorithm with a trivial approximation guarantee. However, the approximation guarantee of the algorithm from Theorem 1.3 is right at the threshold: any efficient algorithm computing an d^μ -colorful choice for some $\mu < 1$ results in a nontrivial efficient approximation algorithm for Tverberg's theorem. This is particularly interesting as no deterministic nontrivial efficient approximating algorithm for Tverberg's theorem is known. The existence of such an algorithm was conjectured by Miller and Sheehy [10].

In the second part, we have studied the complexity of a natural generalization of the colorful Carathéodory theorem, the Nearest Colorful Polytope problem, in two settings. First, we proved that the corresponding local search problem L-NCP is PLS-complete by a reduction to Max2SAT. Using an adaptation of this reduction, we proved that the problem becomes NP-hard if we restrict the solutions to global optima. Although the PLS-completeness of L-NCP together with Bárány's proof indicate that PLS is the right complexity class to show hardness of the colorful Carathéodory problem, there is a striking difference between the colorful Carathéodory problem and any known PLS-complete problem: the costs of local optima are known a-priori. While a PLS-complete problem with this property would not lead to a contradiction, this creates a major stumbling block in the construction of a reduction.

We conclude with open problems.

- The algorithm from Theorem 1.3 computes in polynomial time an $\lceil \varepsilon d \rceil$ -colorful choice for any fixed ε . A more careful analysis shows that the algorithm needs only c_ε color classes, where $c_\varepsilon > 0$ is a constant depending on ε . Hence, the algorithm does not use its complete input. Can this be used to further improve the approximation guarantee?
- Is it possible to compute an $o(d)$ -colorful choice in polynomial time and in particular, is it possible to compute an $O(1)$ -colorful choice in polynomial time?
- On the other hand, can it be shown that computing an $O(1)$ -colorful choice is as hard as computing a perfect colorful choice?
- In Section 2.3, we show that many color classes help to find a perfect colorful choice. Can a perfect colorful choice be computed in polynomial time if we have $\text{poly}(d)$ color classes?

Acknowledgements. We would like to thank Frédéric Meunier and Pauline Sarrabezolles for interesting discussions on the colorful Carathéodory problem and for hosting us during a research stay at the École Nationale des Ponts et Chaussées. Furthermore, we would like to thank the anonymous reviewers for their helpful and encouraging comments.

References

- 1 Emile Aarts and Jan Karel Lenstra, editors. *Local search in combinatorial optimization*. Princeton University Press, 2003.
- 2 Jorge L. Arocha, Imre Bárány, Javier Bracho, Ruy Fabila, and Luis Montejano. Very colorful theorems. *Discrete Comput. Geom.*, 42(2):142–154, 2009.
- 3 Imre Bárány. A generalization of Carathéodory’s theorem. *Discrete Math.*, 40(2–3):141–152, 1982.
- 4 Imre Bárány and Shmuel Onn. Colourful linear programming and its relatives. *Math. Oper. Res.*, 22(3):550–567, 1997.
- 5 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *J. Comput. System Sci.*, 37(1):79–100, 1988.
- 6 Jiří Matoušek. *Lectures on discrete geometry*. Springer, 2002.
- 7 Frédéric Meunier and Antoine Deza. A further generalization of the colourful Carathéodory theorem. In *Discrete geometry and optimization*, volume 69 of *Fields Inst. Commun.*, pages 179–190. Springer, New York, 2013.
- 8 Frédéric Meunier and Pauline Sarrabezolles. Colorful linear programming, Nash equilibrium, and pivots. *arxiv:1409.3436*, 2014.
- 9 Wil Michiels, Emile Aarts, and Jan Korst. *Theoretical aspects of local search*. Monographs in Theoretical Computer Science. Springer, Berlin, 2007.
- 10 Gary L. Miller and Donald R. Sheehy. Approximate centerpoints with proofs. *Comput. Geom.*, 43(8):647–654, 2010.
- 11 Wolfgang Mulzer and Daniel Werner. Approximating Tverberg points in linear time for any fixed dimension. *Discrete Comput. Geom.*, 50(2):520–535, 2013.
- 12 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. System Sci.*, 48(3):498–532, 1994.
- 13 Karanbir S. Sarkaria. Tverberg’s theorem via number fields. *Israel J. Math.*, 79(2–3):317–320, 1992.
- 14 Alejandro A. Schäffer and Mihalis Yannakakis. Simple local search problems that are hard to solve. *SIAM J. Comput.*, 20(1):56–87, 1991.
- 15 Helge Tverberg. Further generalization of Radon’s theorem. *J. London Math. Soc.*, 43:352–354, 1968.

Semi-algebraic Ramsey Numbers

Andrew Suk

University of Illinois at Chicago
851 S. Morgan St., Chicago, IL 60607, USA
suk@uic.edu

Abstract

Given a finite point set $P \subset \mathbb{R}^d$, a k -ary semi-algebraic relation E on P is the set of k -tuples of points in P , which is determined by a finite number of polynomial equations and inequalities in kd real variables. The description complexity of such a relation is at most t if the number of polynomials and their degrees are all bounded by t . The Ramsey number $R_k^{d,t}(s, n)$ is the minimum N such that any N -element point set P in \mathbb{R}^d equipped with a k -ary semi-algebraic relation E such that E has complexity at most t , contains s members such that every k -tuple induced by them is in E or n members such that every k -tuple induced by them is not in E .

We give a new upper bound for $R_k^{d,t}(s, n)$ for $k \geq 3$ and s fixed. In particular, we show that for fixed integers d, t, s

$$R_3^{d,t}(s, n) \leq 2^{n^{o(1)}},$$

establishing a subexponential upper bound on $R_3^{d,t}(s, n)$. This improves the previous bound of $2^{n^{C_1}}$ due to Conlon, Fox, Pach, Sudakov, and Suk where C_1 depends on d and t , and improves upon the trivial bound of $2^{n^{C_2}}$ which can be obtained by applying classical Ramsey numbers where C_2 depends on s . As an application, we give new estimates for a recently studied Ramsey-type problem on hyperplane arrangements in \mathbb{R}^d . We also study multi-color Ramsey numbers for triangles in our semi-algebraic setting, achieving some partial results.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases Ramsey theory, semi-algebraic relation, one-sided hyperplanes, Schur numbers

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.59

1 Introduction

Classical Ramsey numbers. A k -uniform hypergraph $H = (P, E)$ consists of a vertex set P and an edge set $E \subset \binom{P}{k}$, which is a collection of subsets of P of size k . The Ramsey number $R_k(s, n)$ is the minimum integer N such that every k -uniform hypergraph on N vertices contains either s vertices such that every k -tuple induced by them is an edge, or contains n vertices such that every k -tuple induced by them is not an edge.

Due to its wide range of applications in logic, number theory, analysis, and geometry, estimating Ramsey numbers has become one of the most central problems in combinatorics. For *diagonal* Ramsey numbers, i.e. when $s = n$, the best known lower and upper bounds for $R_k(n, n)$ are of the form¹ $R_2(n, n) = 2^{\Theta(n)}$, and for $k \geq 3$,

$$\text{twr}_{k-1}(\Omega(n^2)) \leq R_k(n, n) \leq \text{twr}_k(O(n)),$$

¹ We write $f(n) = O(g(n))$ if $|f(n)| \leq c|g(n)|$ for some fixed constant c and for all $n \geq 1$; $f(n) = \Omega(g(n))$ if $g(n) = O(f(n))$; and $f(n) = \Theta(g(n))$ if both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$ hold. We write $f(n) = o(g(n))$ if for every positive $\epsilon > 0$ there exists a constant n_0 such that $f(n) \leq \epsilon|g(n)|$ for all $n \geq n_0$.



© Andrew Suk;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 59–73



Leibniz International Proceedings in Informatics

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

where the tower function $\text{twr}_k(x)$ is defined by $\text{twr}_1(x) = x$ and $\text{twr}_{i+1} = 2^{\text{twr}_i(x)}$ (see [21, 18, 19, 20]). Erdős, Hajnal, and Rado [19] conjectured that $R_k(n, n) = \text{twr}_k(\Theta(n))$, and Erdős offered a \$500 reward for a proof. Despite much attention over the last 50 years, the exponential gap between the lower and upper bounds for $R_k(n, n)$, when $k \geq 3$, remains unchanged.

The *off-diagonal* Ramsey numbers, i.e. $R_k(s, n)$ with s fixed and n tending to infinity, has also been extensively studied. Unlike $R_k(n, n)$, the lower and upper bounds for $R_k(s, n)$ are much more comparable. It is known [5, 25, 8, 9] that $R_2(3, n) = \Theta(n^2/\log n)$ and, for fixed $s > 3$

$$\Omega\left(n^{\frac{s+1}{2}-\epsilon}\right) \leq R_2(s, n) \leq O\left(n^{s-1}\right), \quad (1)$$

where $\epsilon > 0$ is an arbitrarily small constant. Combining the upper bound in (1) with the results of Erdős, Hajnal, and Rado [20, 19] demonstrates that

$$\text{twr}_{k-1}(\Omega(n)) \leq R_k(s, n) \leq \text{twr}_{k-1}(O(n^{2s-4})), \quad (2)$$

for $k \geq 3$ and $s \geq 2^k$. See Conlon, Fox, and Sudakov [14] for a recent improvement.

Semi-algebraic setting. In this paper, we continue a sequence of recent works on Ramsey numbers for k -ary semi-algebraic relations E on \mathbb{R}^d (see [10, 17, 13, 33]). Before we give its precise definition, let us recall two classic Ramsey-type theorems of Erdős and Szekeres.

► **Theorem 1** ([21]). *For $N = (s-1)(n-1) + 1$, let $P = (p_1, \dots, p_N) \subset \mathbb{R}$ be a sequence of N distinct real numbers. Then P contains either an increasing subsequence of length s , or a decreasing subsequence on length n .*

In fact, there are now at least 6 different proofs of Theorem 1 (see [32]). The other well-known result from [21] is the following theorem, which is often referred to as the Erdős-Szekeres cups-caps Theorem. Let X be a finite point set in the plane in general position.² We say that $X = (p_{i_1}, \dots, p_{i_s})$ forms an s -cup (s -cap) if X is in convex position and its convex hull is bounded above (below) by a single edge.

► **Theorem 2** ([21]). *For $N = \binom{n+s-4}{s-2} + 1$, let $P = (p_1, \dots, p_N)$ be a sequence of N points in the plane in general position. Then P contains either an s -cup or an n -cap.*

Theorems 1 and 2 can be generalized using the following semi-algebraic framework. Let $P = \{p_1, \dots, p_N\}$ be a sequence of N points in \mathbb{R}^d . Then we say that $E \subset \binom{P}{k}$ is a *semi-algebraic relation* on P with *complexity* at most t , if there are t polynomials $f_1, \dots, f_t \in \mathbb{R}[x_1, \dots, x_{kd}]$ of degree at most t , and a Boolean function Φ such that for $1 \leq i_1 < \dots < i_k \leq N$,

$$(p_{i_1}, \dots, p_{i_k}) \in E \quad \Leftrightarrow \quad \Phi(f_1(p_{i_1}, \dots, p_{i_k}) \geq 0, \dots, f_t(p_{i_1}, \dots, p_{i_k}) \geq 0) = 1.$$

We say that the relation $E \subset \binom{P}{k}$ is *symmetric* if $(p_{i_1}, \dots, p_{i_k}) \in E$ iff for all permutation π ,

$$\Phi(f_1(p_{\pi(i_1)}, \dots, p_{\pi(i_k)}) \geq 0, \dots, f_t(p_{\pi(i_1)}, \dots, p_{\pi(i_k)}) \geq 0) = 1.$$

Point sets $P \subset \mathbb{R}^d$ equipped with a k -ary semi-algebraic relation $E \subset \binom{P}{k}$ are often used to model problems in discrete geometry, where the dimension d , uniformity k , and complexity

² No two members share the same x -coordinate, and no three members are collinear.

t are considered fixed but arbitrarily large constants. Since we can always make any relation E symmetric by increasing its complexity to $t' = t'(k, d, t)$, we can therefore simplify our presentation by only considering symmetric relations.

Let $R_k^{d,t}(s, n)$ be the minimum integer N such that every N -element point set P in \mathbb{R}^d equipped with a k -ary (symmetric) semi-algebraic relation $E \subset \binom{P}{k}$, which has complexity at most t , contains s points such that every k -tuple induced by them is in E , or contains n points such that no k -tuple induced by them is in E . Alon, Pach, Pinchasi, Radoičić, and Sharir [6] showed that for $k = 2$, we have

$$R_2^{d,t}(n, n) \leq n^C, \tag{3}$$

where $C = C(d, t)$. Roughly speaking, $C \approx t^{\binom{d+t}{t}}$. Conlon, Fox, Pach, Sudakov, and Suk showed that one can adapt the Erdős-Rado argument in [20] and establish the following recursive formula for $R_k^{d,t}(s, n)$.

► **Theorem 3** ([13]). *Set $M = R_{k-1}^{d,t}(s-1, n-1)$. Then for every $k \geq 3$,*

$$R_k^{d,t}(s, n) \leq 2^{C_1 M \log M},$$

where $C_1 = C_1(k, d, t)$.

Together with (3) we have $R_k^{d,t}(n, n) \leq \text{twr}_{k-1}(n^C)$, giving an exponential improvement over the Ramsey numbers for general k -uniform hypergraphs. Conlon et al. [13] also gave a construction of a geometric example that provides a $\text{twr}_{k-1}(\Omega(n))$ lower bound, demonstrating that $R_k^{d,t}(n, n)$ does indeed grow as a $(k-1)$ -fold exponential tower in n .

However, off-diagonal Ramsey numbers for semi-algebraic relations are much less understood. The best known upper bound for $R_k^{d,t}(s, n)$ is essentially the trivial bound

$$R_k^{d,t}(s, n) \leq \min \left\{ R_k^{d,t}(n, n), R_k(s, n) \right\}.$$

The crucial case is when $k = 3$, since any significant improvement on estimating $R_3^{d,t}(s, n)$ could be used in combination with Theorem 3 to obtain a better bound for $R_k^{d,t}(s, n)$, for $k \geq 4$. The trivial bound implies that

$$R_3^{d,t}(s, n) \leq 2^{n^C}, \tag{4}$$

where C is a large constant depending on d, t , and s .

The main difficulty in improving (4) is that the Erdős-Rado upper bound argument [20] will not be effective. Roughly speaking, the Erdős-Rado argument reduces the problem from 3-uniform hypergraphs to graphs, producing a recursive formula similar to Theorem 3. This approach has been used repeatedly by many researchers to give upper bounds on Ramsey-type problems arising in triple systems [14, 13, 33, 30]. However, it is very unlikely that any variant of the Erdős-Rados upper bound argument will establish a subexponential upper bound for $R_3^{d,t}(s, n)$.

With a more novel approach, our main result establishes the following improved upper bound for $R_3^{d,t}(s, n)$, showing that the function $R_3^{d,t}(s, n)$ is indeed subexponential in n .

► **Theorem 4.** *For fixed integers $d, t \geq 1$ and $s \geq 4$, we have $R_3^{d,t}(s, n) \leq 2^{n^{o(1)}}$. More precisely,*

$$R_3^{d,t}(s, n) \leq 2^{2^{c\sqrt{(\log n)(\log \log n)}}},$$

where $c = c(d, t, s)$.

Combining Theorems 4 and 3 we have the following.

► **Corollary 5.** *For fixed integers $d, t \geq 1$, $k \geq 3$, and $s \geq k + 1$, we have*

$$R_k^{d,t}(s, n) \leq \text{twr}_{k-1}(n^{o(1)}).$$

For $d \geq 2$ and $t \geq 1$, the classic cups-caps construction of Erdős and Szekeres [21] shows that $R_3^{d,t}(s, n) \geq \Omega(n^{s-2})$, and together with the semi-algebraic stepping-up lemma proven in [13] (see also [16]) we have $R_k^{d,t}(s, n) \geq \text{twr}_{k-2}(\Omega(n^{s-2}))$ for $s, d \geq 2^k$.

In Section 5, we give an application of Theorem 4 to a recently studied problem on hyperplane arrangements in \mathbb{R}^d .

Monochromatic triangles. Let $R_2(s; m) = R_2(\underbrace{s, \dots, s}_m)$ denote the smallest integer N such that any m -coloring on the edges of the complete N -vertex graph contains a monochromatic clique of size s , that is, a set of s vertices such that every pair from this set has the same color. For the case $s = 3$, the Ramsey number $R_2(3; m)$ has received a lot of attention over the last 100 years due to its application in additive number theory [31] (more details are given in Section 6.1). It is known (see [24, 31]) that

$$\Omega(3.19^m) \leq R_2(3; m) \leq O(m!).$$

Our next result states that we can improve the upper bound on $R_2(3; m)$ in our semi-algebraic setting. More precisely, let $R_2^{d,t}(3; m)$ be the minimum integer N such that every N -element point set P in \mathbb{R}^d equipped with symmetric semi-algebraic relations $E_1, \dots, E_m \subset \binom{P}{2}$, such that each E_i has complexity at most t and $\binom{P}{2} = E_1 \cup \dots \cup E_m$, contains three points such that every pair induced by them belongs to E_i for some fixed i .

► **Theorem 6.** *For fixed $d, t \geq 1$ we have*

$$R_2^{d,t}(3; m) < 2^{O(m \log \log m)}.$$

We also show that for fixed $d \geq 1$ and $t \geq 5000$, the function $R_2^{d,t}(3; m)$ does indeed grow exponentially in m .

► **Theorem 7.** *For $d \geq 1$ and $t \geq 5000$ we have*

$$R_2^{d,t}(3; m) \geq c(1681)^{m/7} \geq c(2.889)^m,$$

where c is an absolute constant.

Organization. In the next two sections, we recall several old theorems on the arrangement of surfaces in \mathbb{R}^d and establish a result on point sets equipped with multiple binary relations. In Section 4, we combine the results from Sections 2 and 3 to prove our main result, Theorem 4. We discuss a short proof of our application in Section 5, and our results on monochromatic triangles in Section 6.

We systemically omit floor and ceiling signs whenever they are not crucial for the sake of clarity of our presentation. All logarithms are assumed to be base 2.

2 Arrangement of surfaces in \mathbb{R}^d

In this section, we recall several old results on the arrangement of surfaces in \mathbb{R}^d . Let f_1, \dots, f_m be d -variate real polynomials of degree at most t , with zero sets Z_1, \dots, Z_m , that is, $Z_i = \{x \in \mathbb{R}^d : f_i(x) = 0\}$. Set $\Sigma = \{Z_1, \dots, Z_m\}$. We will assume that d and t are fixed, and m is some number tending to infinity. A *cell* in the arrangement $\mathcal{A}(\Sigma) = \bigcup_i Z_i$ is a relatively open connected set defined as follows. Let \approx be an equivalence relation on \mathbb{R}^d , where $x \approx y$ if $\{i : x \in Z_i\} = \{i : y \in Z_i\}$. Then the cells of the arrangement $\mathcal{A}(\Sigma)$ are the connected components of the equivalence classes. A vector $\sigma \in \{-1, 0, +1\}^m$ is a *sign pattern* of f_1, \dots, f_m if there exists an $x \in \mathbb{R}^d$ such that the sign of $f_j(x)$ is σ_j for all $j = 1, \dots, m$. The Milnor-Thom theorem (see [7, 29, 34]) bounds the number of cells in the arrangement of the zero sets Z_1, \dots, Z_m and, consequently, the number of possible sign patterns (see all [35]).

► **Theorem 8** (Milnor-Thom). *Let f_1, \dots, f_m be d -variate real polynomials of degree at most t . The number of cells in the arrangement of their zero sets $Z_1, \dots, Z_m \subset \mathbb{R}^d$ and, consequently, the number of sign patterns of f_1, \dots, f_m is at most*

$$\left(\frac{50mt}{d}\right)^d,$$

for $m \geq d \geq 1$.

While the Milnor-Thom Theorem bounds the number of cells in the arrangement $\mathcal{A}(\Sigma)$, the complexity of these cells may be very large (depending on m). A long standing open problem is whether each cell can be further decomposed into semi-algebraic sets³ with bounded description complexity (which depends only on d and t), such that the total number of cells for the whole arrangement is still $O(m^d)$. This can be done easily in dimension 2 by a result of Chazelle et al. [11]. Unfortunately in higher dimensions, the current bounds for this problem are not tight. In dimension 3, Chazelle et al. [11] established a near tight bound of $O(m^3\beta(m))$, where $\beta(m)$ is an extremal slowly growing function of m related to the inverse Ackermann function. For dimensions $d \geq 4$, Koltun [26] established a general bound of $O(m^{2d-4+\epsilon})$ for arbitrarily small constant ϵ , which is nearly tight in dimension 4. By combining these bounds with the standard theory of random sampling [4, 12, 6], one can obtain the following result which is often referred to as the Cutting Lemma. We say that the surface $Z_i = \{x \in \mathbb{R}^d : f_i(x) = 0\}$ *crosses* the cell $\Delta \subset \mathbb{R}^d$ if $Z_i \cap \Delta \neq \emptyset$ and Z_i does not fully contain Δ .

► **Lemma 9** (Cutting Lemma). *For $d, t \geq 1$, let Σ be a family of m algebraic surfaces (zero sets) in \mathbb{R}^d of degree at most t . Then for any integer $r \geq 1$, there exists a decomposition of \mathbb{R}^d into at most $c_1 r^{2d}$ relatively open connected sets (cells), where $c_1 = c_1(d, t)$, such that each cell is crossed by at most m/r surfaces from Σ .*

As an application, we prove the following lemma (see [27, 3] for a similar result when Σ is a collection of hyperplanes).

► **Lemma 10**. *For $d, t \geq 1$, let P be an N -element point set in \mathbb{R}^d and let Σ be a family of m surfaces of degree at most t . Then for any integer $\ell > \log m$, we can find ℓ disjoint subsets*

³ A real semi-algebraic set in \mathbb{R}^d is the locus of all points that satisfy a given finite Boolean combination of polynomial equations and inequalities in the d coordinates.

P_i and ℓ cells Δ_i , with $\Delta_i \supset P_i$, such that each subset P_i contains at least $N/(4\ell)$ points from P , and every surface in Σ crosses at most $c_2\ell^{1-1/(2d)}$ cells Δ_i , where $c_2 = c_2(d, t)$.

Proof. We first find Δ_1 and P_1 as follows. Let $\ell > \log m$ and let c_1 be as defined in Lemma 9. Given a family Σ of m surfaces in \mathbb{R}^d , we apply Lemma 9 with parameter $r = (\ell/c_1)^{1/(2d)}$, and decompose \mathbb{R}^d into at most ℓ cells, such that each cell is crossed by at most $\frac{m}{(\ell/c_1)^{1/(2d)}}$ surfaces from Σ . By the pigeonhole principle, there is a cell Δ_1 that contains at least N/ℓ points from P . Let P_1 be a subset of exactly $\lfloor N/\ell \rfloor$ points in $\Delta_1 \cap P$. Now for each surface from Σ that crosses Δ_1 , we "double it" by adding another copy of that surface to our collection. This gives us a new family of surfaces Σ_1 such that

$$|\Sigma_1| \leq m + \frac{m}{(\ell/c_1)^{1/(2d)}} = m \left(1 + \frac{1}{(\ell/c_1)^{1/(2d)}} \right).$$

After obtaining subsets P_1, \dots, P_i such that $|P_j| = \lfloor \frac{N}{\ell}(1 - \frac{1}{\ell})^{j-1} \rfloor$ for $1 \leq j \leq i$, cells $\Delta_1, \dots, \Delta_i$, and the family of surfaces Σ_i such that

$$|\Sigma_i| \leq m \left(1 + \frac{1}{(\ell/c_1)^{1/(2d)}} \right)^i,$$

we obtain $P_{i+1}, \Delta_{i+1}, \Sigma_{i+1}$ as follows. Given Σ_i , we apply Lemma 9 with the same parameter $r = (\ell/c_1)^{1/(2d)}$, and decompose \mathbb{R}^d into at most ℓ cells, such that each cell is crossed by at most $\frac{|\Sigma_i|}{(\ell/c_1)^{1/(2d)}}$ surfaces from Σ_i . Let $P' = P \setminus (P_1 \cup \dots \cup P_i)$. By the pigeonhole principle, there is a cell Δ_{i+1} that contains at least

$$\begin{aligned} \frac{|P'|}{\ell} &\geq \left(N - \sum_{j=1}^i \frac{N}{\ell} (1 - \frac{1}{\ell})^{j-1} \right) / \ell \\ &= \frac{N}{\ell} \left(1 - \frac{1}{\ell} \sum_{j=1}^i (1 - \frac{1}{\ell})^{j-1} \right) \\ &= \frac{N}{\ell} (1 - \frac{1}{\ell}) \left(1 - \frac{1}{\ell} - \frac{1}{\ell} \sum_{j=1}^{i-1} (1 - \frac{1}{\ell})^{j-1} \right) \\ &= \frac{N}{\ell} (1 - \frac{1}{\ell})^i \end{aligned}$$

points from P' . Let P_{i+1} be a subset of exactly $\lfloor \frac{N}{\ell} (1 - 1/\ell)^i \rfloor$ points in $\Delta_{i+1} \cap P'$. Finally, for each surface from Σ_i that crosses Δ_{i+1} , we "double it" by adding another copy of that surface to our collection, giving us a new family of surfaces Σ_{i+1} such that

$$\begin{aligned} |\Sigma_{i+1}| &\leq |\Sigma_i| + \frac{|\Sigma_i|}{(\ell/c_1)^{1/(2d)}} \\ &= |\Sigma_i| \left(1 + \frac{1}{(\ell/c_1)^{1/(2d)}} \right) \\ &\leq m \left(1 + \frac{1}{(\ell/c_1)^{1/(2d)}} \right)^{i+1}. \end{aligned}$$

Notice that $|P_i| \geq N/(4\ell)$ for $i \leq \ell$. Once we have obtained subsets P_1, \dots, P_ℓ and cell $\Delta_1, \dots, \Delta_\ell$, it is easy to see that each surface in Σ crosses at most $O(r^{1-1/(2d)})$ cells Δ_i . Indeed suppose $Z \in \Sigma$ crosses κ cells. Then by the arguments above, there must be 2^κ copies of Z in Σ_ℓ . Hence we have

$$2^\kappa \leq m \left(1 + \frac{1}{(\ell/c_1)^{1/(2d)}} \right)^\ell \leq m e^{c_1 \ell^{1-1/(2d)}}.$$

Since $\ell \geq \log m$, we have

$$\kappa \leq c_2 \ell^{1-1/(2d)},$$

for sufficiently large $c_2 = c_2(d, t)$. ◀

3 Multiple binary relations

Let P be a set of N points in \mathbb{R}^d , and let $E_1, \dots, E_m \subset \binom{P}{2}$ be binary (symmetric) semi-algebraic relations on P such that E_i has complexity at most t . The goal of this section is to find a large subset $P' \subset P$ such that $\binom{P'}{2} \cap E_i = \emptyset$ for all i , given that the clique number in the graphs $G_i = (P, E_i)$ are small.

First we recall a classic theorem of Dilworth (see also [23]). Let $G = (V, E)$ be a graph whose vertices are ordered $V = \{v_1, \dots, v_N\}$. We say that E is *transitive* on V if for $1 \leq i_1 < i_2 < i_3 \leq N$, $(v_{i_1}, v_{i_2}), (v_{i_2}, v_{i_3}) \in E$ implies that $(v_{i_1}, v_{i_3}) \in E$.

► **Theorem 11** (Dilworth). *Let $G = (V, E)$ be an N -vertex graph whose vertices are ordered $V = \{v_1, \dots, v_N\}$, such that E is transitive on V . If G has clique number ω , then G contains an independent set of size N/ω .*

► **Lemma 12.** *For integers $m \geq 2$ and $d, t \geq 1$, let P be a set of N points in \mathbb{R}^d equipped with (symmetric) semi-algebraic relations $E_1, \dots, E_m \subset \binom{P}{2}$, where each E_i has complexity at most t . Then there is a subset $P' \subset P$ of size $N^{1/(c_3 \log m)}$, where $c_3 = c_3(d, t)$, and a fixed ordering on P' such that each relation E_i is transitive on P' .*

Proof. We proceed by induction on N . Let c_3 be a sufficiently large number depending only on d and t that will be determined later. For each relation $E_i \subset \binom{P}{2}$, let $f_{i,1}, \dots, f_{i,t}$ be polynomials of degree at most t and let Φ_i be a boolean function such that

$$(p, q) \in E_i \iff \Phi_i(f_{i,1}(p, q) \geq 0, \dots, f_{i,t}(p, q) \geq 0) = 1.$$

For each $p \in P$, $i \in \{1, \dots, m\}$, and $j \in \{1, \dots, t\}$, we define the surface $Z_{p,i,j} = \{x \in \mathbb{R}^d : f_{i,j}(p, x) = 0\}$. Then let Σ be the family of Nmt surfaces in \mathbb{R}^d defined by

$$\Sigma = \{Z_{p,i,j} : p \in P, 1 \leq i \leq m, 1 \leq j \leq t\}.$$

By applying Lemma 9 to Σ with parameter $r = (mt)^2$, there is a decomposition of \mathbb{R}^d into at most $c_1(mt)^{4d}$ cells such that each cell has the property that at most $N/(mt)$ surfaces from Σ crosses it. We note that $c_1 = c_1(d, t)$ is defined in Lemma 9. By the pigeonhole principle, there is a cell Δ in the decomposition such that $|\Delta \cap P| \geq N/(c_1(mt)^{4d})$. Set $P_1 = \Delta \cap P$.

Let $P_2 \subset P \setminus P_1$ such that each point in P_2 gives rise to mt surfaces that do not cross Δ . More precisely,

$$P_2 = \{p \in P \setminus P_1 : Z_{p,i,j} \text{ does not cross } \Delta, \forall i, j\}.$$

Notice that

$$|P_2| \geq N - \frac{N}{mt} - \frac{N}{c_1(mt)^{4d}} \geq \frac{N}{4}.$$

We fix a point $p_0 \in P_1$. Then for each $q \in P_2$, let $\sigma(q) \in \{-1, 0, +1\}^{mt}$ be the sign pattern of the (mt) -tuple $(f_{1,1}(p_0, q), f_{1,2}(p_0, q), \dots, f_{m,t}(p_0, q))$. By Theorem 8, there are at most

$\left(\frac{50mt^2}{d}\right)^d$ distinct sign vectors σ . By the pigeonhole principle, there is a subset $P_3 \subset P_2$ such that

$$|P_3| \geq \frac{|P_2|}{(50/d)^d m^d t^{2d}},$$

and for any two points $q, q' \in P_3$, we have $\sigma(q) = \sigma(q')$. That is, q and q' give rise to vectors with the same sign pattern. Therefore, for any $p, p' \in P_1$ and $q, q' \in P_3$, we have $(p, q) \in E_i$ if and only if $(p', q') \in E_i$, for all $i \in \{1, \dots, m\}$.

Let $c_4 = c_4(d, t)$ be sufficiently large such that $|P_1|, |P_3| \geq \frac{N}{c_4 m^{4d}}$. By the induction hypothesis, we can find subsets $P_4 \subset P_1, P_5 \subset P_3$, such that

$$|P_4|, |P_5| \geq \left(\frac{N}{c_4 m^{4d}}\right)^{\frac{1}{c_3 \log m}} \geq \frac{N^{\frac{1}{c_3 \log m}}}{2},$$

where $c_3 = c_3(d, t)$ is sufficiently large, and there is an ordering on P_4 (and on P_5) such that each E_i is transitive on P_4 (and on P_5). Set $P' = P_4 \cup P_5$, which implies $|P'| \geq N^{\frac{1}{c_3 \log m}}$. We will show that P' has the desired properties. Let π and π' be the orderings on P_4 and P_5 respectively, such that E_i is transitive on P_4 and on P_5 , for every $i \in \{1, \dots, m\}$. We order the elements in $P' = \{p_1, \dots, p_{|P'|}\}$ by using π and π' , such that all elements in P_5 comes after all elements in P_4 .

In order to show that E_i is transitive on P' , it suffices to examine triples going across P_4 and P_5 . Let $p_{j_1}, p_{j_2} \in P_4$ and $p_{j_3} \in P_5$ such that $j_1 < j_2 < j_3$. By construction of P_4 and P_5 , if $(p_{j_1}, p_{j_2}), (p_{j_2}, p_{j_3}) \in E_i$, then we have $(p_{j_1}, p_{j_3}) \in E_i$. Likewise, suppose $p_{j_1} \in P_4$ and $p_{j_2}, p_{j_3} \in P_5$. Then again by construction of P_4 and P_5 , if $(p_{j_1}, p_{j_2}), (p_{j_2}, p_{j_3}) \in E_i$, then we have $(p_{j_1}, p_{j_3}) \in E_i$. Hence E_i is transitive on P' , for all $i \in \{1, \dots, m\}$, and this completes the proof. \blacktriangleleft

By combining the two previous results, we have the following.

► **Lemma 13.** For $m \geq 2$ and $d, t \geq 1$, let P be a set of N points in \mathbb{R}^d equipped with (symmetric) semi-algebraic relations $E_1, \dots, E_m \subset \binom{P}{2}$, where each E_i has complexity at most t . If graph $G_i = (P, E_i)$ has clique number ω_i , then there is a subset $P' \subset P$ of size $\frac{N^{1/(c_3 \log m)}}{\omega_1 \cdots \omega_m}$, where $c_3 = c_3(d, t)$ is defined above, such that $\binom{P'}{2} \cap E_i = \emptyset$ for all i .

Proof. By applying Lemma 12, we obtain a subset $P_1 \subset P$ of size $N^{\frac{1}{c_3 \log m}}$, and an ordering on P_1 such that E_i is transitive on P_1 for all i . Then by an m -fold application of Theorem 11, the statement follows. \blacktriangleleft

4 Proof of Theorem 4

Let P be a point set in \mathbb{R}^d and let $E \subset \binom{P}{3}$ be a semi-algebraic relation on P . We say that (P, E) is $K_s^{(3)}$ -free if every collection of s points in P contains a triple not in E . Suppose we have ℓ disjoint subsets $P_1, \dots, P_\ell \subset P$. For $1 \leq i_1 < i_2 < i_3 \leq \ell$, we say that the triple $(P_{i_1}, P_{i_2}, P_{i_3})$ is *homogeneous* if $(p_1, p_2, p_3) \in E$ for all $p_1 \in P_{i_1}, p_2 \in P_{i_2}, p_3 \in P_{i_3}$, or $(p_1, p_2, p_3) \notin E$ for all $p_1 \in P_{i_1}, p_2 \in P_{i_2}, p_3 \in P_{i_3}$. For $p_1, p_2 \in P_1 \cup \dots \cup P_\ell$ and $i \in \{1, \dots, \ell\}$, we say that the triple (p_1, p_2, i) is *good*, if $(p_1, p_2, p_3) \in E$ for all $p_3 \in P_i$, or $(p_1, p_2, p_3) \notin E$ for all $p_3 \in P_i$. We say that the triple (p_1, p_2, i) is *bad* if (p_1, p_2, i) is not good and $p_1, p_2 \notin P_i$.

► **Lemma 14.** Let P be a set of N points in \mathbb{R}^d and let $E \subset \binom{P}{3}$ be a (symmetric) semi-algebraic relation on P such that E has complexity at most t . Then for $r = \frac{N^{1/(30d)}}{tc_2}$, where c_2 is defined in Lemma 10, there are disjoint subsets $P_1, \dots, P_r \subset P$ such that

1. $|P_i| \geq \frac{N^{1/(30d)}}{tc_2}$,
2. all triples $(P_{i_1}, P_{i_2}, P_{i_3})$, $1 \leq i_1 < i_2 < i_3 \leq r$, are homogeneous, and
3. all triples (p, q, i) , where $i \in \{1, \dots, r\}$ and $p, q \in (P_1 \cup \dots \cup P_r) \setminus P_i$, are good.

Proof. We can assume that $N > (tc_2)^{30d}$, since otherwise the statement is trivial. Since E is semi-algebraic with complexity t , there are polynomials f_1, \dots, f_t of degree at most t , and a Boolean function Φ such that

$$(p_1, p_2, p_3) \in E \iff \Phi(f_1(p_1, p_2, p_3) \geq 0, \dots, f_t(p_1, p_2, p_3) \geq 0) = 1.$$

For each $p, q \in P$ and $i \in \{1, \dots, t\}$, we define the surface $Z_{p,q,i} = \{x \in \mathbb{R}^d : f_i(p, q, x) = 0\}$. Then we set

$$\Sigma = \{Z_{p,q,i} : p, q \in P, 1 \leq i \leq t\}.$$

Thus we have $|\Sigma| = N^2t$. Next we apply Lemma 10 to P and Σ with parameter $\ell = \sqrt{N}$, and obtain subsets Q_1, \dots, Q_ℓ and cells $\Delta_1, \dots, \Delta_\ell$, such that $Q_i \subset \Delta_i$, $|Q_i| = \lfloor \sqrt{N}/4 \rfloor$, and each surface in Σ crosses at most $c_2N^{1/2-1/(4d)}$ cells Δ_i . We note that $c_2 = c_2(d, t)$ is defined in Lemma 10 and $\sqrt{N} \geq \log(tN^2)$. Set $Q = Q_1 \cup \dots \cup Q_\ell$. Each pair $(p, q) \in \binom{Q}{2}$ gives rise to $2t$ surfaces in Σ . By Lemma 10, these $2t$ surfaces cross in total at most $2tc_2N^{1/2-1/(4d)}$ cells Δ_i . Hence there are at most $2tc_2N^{5/2-1/(4d)}$ bad triples of the form (p, q, i) , where $i \in \{1, \dots, \sqrt{N}\}$ and $p, q \in Q \setminus Q_i$. Moreover, there are at most $2tc_2N^{2-1/(4d)}$ bad triples (p, q, i) , where both p and q lie in the same part Q_j and $j \neq i$.

We uniformly at random pick $r = \frac{N^{1/(30d)}}{tc_2}$ subsets (parts) from the collection $\{Q_1, \dots, Q_\ell\}$, and r vertices from each of the subsets that were picked. For a bad triple (p, q, i) with p and q in distinct subsets, the probability that (p, q, i) survives is at most

$$\left(\frac{r}{\sqrt{N}}\right)^3 \left(\frac{r}{\sqrt{N}/4}\right)^2 = \frac{16}{(tc_2)^5} N^{1/(6d)-5/2}.$$

For a bad triple (p, q, i) with p, q in the same subset Q_j , where $j \neq i$, the probability that the triple (p, q, i) survives is at most

$$\left(\frac{r}{\sqrt{N}}\right)^2 \left(\frac{r}{\sqrt{N}/4}\right)^2 = \frac{16}{(tc_2)^4} N^{2/(15d)-2}.$$

Therefore, the expected number of bad triples in our random subset is at most

$$\left(\frac{16}{(tc_2)^5} N^{1/(6d)-5/2}\right) (tc_2N^{5/2-1/(4d)}) + \left(\frac{16}{(tc_2)^4} N^{2/(15d)-2}\right) (tc_2N^{2-1/(4d)}) < 1.$$

Hence we can find disjoint subsets P_1, \dots, P_r , such that $|P_i| \geq r = \frac{N^{1/(30d)}}{tc_2}$, and there are no bad triples (p, q, i) , where $i \in \{1, \dots, r\}$ and $p, q \in (P_1 \cup \dots \cup P_r) \setminus P_i$.

It remains to show that every triple $(P_{i_1}, P_{i_2}, P_{i_3})$ is homogeneous for $1 \leq i_1 < i_2 < i_3 \leq r$. Let $p_1 \in P_{i_1}, p_2 \in P_{i_2}, p_3 \in P_{i_3}$ and suppose $(p_1, p_2, p_3) \in E$. Then for any choice $q_1 \in P_{i_1}, q_2 \in P_{i_2}, q_3 \in P_{i_3}$, we also have $(q_1, q_2, q_3) \in E$. Indeed, since the triple (p_1, p_2, i_3) is good, this implies that $(p_1, p_2, q_3) \in E$. Since the triple (p_1, q_3, i_2) is also good, we have $(p_1, q_2, q_3) \in E$. Finally since (q_2, q_3, i_1) is good, we have $(q_1, q_2, q_3) \in E$. Likewise, if $(p_1, p_2, p_3) \notin E$, then $(q_1, q_2, q_3) \notin E$ for any $q_1 \in P_{i_1}, q_2 \in P_{i_2}, q_3 \in P_{i_3}$. ◀

We are finally ready to prove Theorem 4, which follows immediately from the following theorem.

► **Theorem 15.** *Let P be a set of N points in \mathbb{R}^d and let $E \subset \binom{P}{3}$ be a (symmetric) semi-algebraic relation on P such that E has complexity at most t . If (P, E) is $K_s^{(3)}$ -free, then there exists a subset $P' \subset P$ such that $\binom{P'}{3} \cap E = \emptyset$ and*

$$|P'| \geq 2^{\frac{(\log \log N)^2}{c^s \log \log \log N}},$$

where $c = c(d, t)$.

Proof. The proof is by induction on N and s . The base cases are $s = 3$ or $N \leq (tc_2)^{30d}$, where c_2 is defined in Lemma 10. When $N \leq (tc_2)^{30d}$, the statement holds trivially for sufficiently large $c = c(d, t)$. If $s = 3$, then again the statement follows immediately by taking $P' = P$.

Now assume that the statement holds if $s' \leq s, N' \leq N$ and not both inequalities are equalities. We apply Lemma 14 to (P, E) and obtain disjoint subsets P_1, \dots, P_r , where $r = \frac{N^{1/(30d)}}{tc_2}$, such that $|P_i| \geq \frac{N^{1/(30d)}}{tc_2}$, every triple of parts $(P_{i_1}, P_{i_2}, P_{i_3})$ is homogeneous, and every triple (p, q, i) is good where $i \in \{1, \dots, r\}$ and $p, q \in (P_1 \cup \dots \cup P_r) \setminus P_i$.

Let P_0 be the set of $\frac{N^{1/(30d)}}{tc_2}$ points obtained by selecting one point from each P_i . Since (P_0, E) is $K_s^{(3)}$ -free, we can apply the induction hypothesis on P_0 , and find a set of indices $I = \{i_1, \dots, i_m\}$ such that

$$\log |I| \geq \frac{\left(\log \log \frac{N^{1/(30d)}}{tc_2}\right)^2}{c^s \log \log \log \frac{N^{1/(30d)}}{tc_2}} \geq (1/2) \log \log N,$$

and for every triple $i_1 < i_2 < i_3$ in I all triples with one point in each P_{i_j} does not satisfy E . Hence we have $m = \sqrt{\log N}$, and let $Q_j = P_{i_j}$ for $1 \leq j \leq m$.

For each subset Q_i , we define binary semi-algebraic relations $E_{i,j} \subset \binom{Q_i}{2}$, where $j \neq i$, as follows. Since $E \subset \binom{P}{3}$ is semi-algebraic with complexity t , there are t polynomials f_1, \dots, f_t of degree at most t , and a Boolean function Φ such that $(p_1, p_2, p_3) \in E$ if and only if

$$\Phi(f_1(p_1, p_2, p_3) \geq 0, \dots, f_t(p_1, p_2, p_3) \geq 0) = 1.$$

Fix a point $q_0 \in Q_j$, where $j \neq i$. Then for $p_1, p_2 \in Q_i$, we have $(p_1, p_2) \in E_{i,j}$ if and only if

$$\Phi(f_1(p_1, p_2, q_0) \geq 0, \dots, f_t(p_1, p_2, q_0) \geq 0) = 1.$$

Suppose there are $2^{(\log N)^{1/4}}$ vertices in Q_i that induces a clique in the graph $G_{i,j} = (Q_i, E_{i,j})$. Then these vertices would induce a $K_{s-1}^{(3)}$ -free subset in the original (hypergraph) (P, E) . By the induction hypothesis, we can find a subset $Q'_i \subset Q_i$ such that

$$|Q'_i| \geq 2^{\frac{((1/4) \log \log N)^2}{c^{s-1} \log \log \log N}} \geq 2^{\frac{(\log \log N)^2}{c^s \log \log \log N}},$$

for sufficiently large c , such that $\binom{Q'_i}{3} \cap E = \emptyset$ and we are done. Hence we can assume that each graph $G_{i,j} = (Q_i, E_{i,j})$ has clique number at most $2^{(\log n)^{1/4}}$. By applying Lemma 13 to each Q_i , where Q_i is equipped with $m - 1$ semi-algebraic relations $E_{i,j}$, $j \neq i$, we can find subsets $T_i \subset Q_i$ such that

$$|T_i| \geq \frac{|Q_i|^{1/(c_3 \log m)}}{2^{(\log N)^{1/4} \sqrt{\log N}}} = \frac{2^{\frac{\log N}{30dc_3 \log(\sqrt{\log N})}}}{2^{(\log N)^{3/4}}} \geq 2^{\frac{\log N}{c_5 \log \log N}},$$

where $c_5 = c_5(d, t)$, and $\binom{T_i}{2} \cap E_j = \emptyset$ for all $j \neq i$. Therefore, we now have subsets T_1, \dots, T_m , such that

1. $m = \sqrt{\log N}$,
2. for any triple $(T_{i_1}, T_{i_2}, T_{i_3})$, $1 \leq i_1 < i_2 < i_3 \leq m$, every triple with one vertex in each T_{i_j} is not in E ,
3. for any pair (T_{i_1}, T_{i_2}) , $1 \leq i_1 < i_2 \leq m$, every triple with two vertices T_{i_1} and one vertex in T_{i_2} is not in E , and every triple with two vertices T_{i_2} and one vertex in T_{i_1} is also not in E .

By applying the induction hypothesis to each (T_i, E) , we obtain a collection of subsets $U_i \subset T_i$ such that

$$\log |U_i| \geq \frac{\left(\log\left(\frac{\log N}{c_5 \log \log N}\right)\right)^2}{c^s \log \log\left(\frac{\log N}{c_5 \log \log N}\right)} \geq \frac{(\log \log N - \log(c_5 \log \log N))^2}{c^s \log \log \log N},$$

and $\left(\bigcup_{i=1}^m U_i\right) \cap E = \emptyset$. Let $P' = \bigcup_{i=1}^m U_i$. Then by above we have $\binom{P'}{3} \cap E = \emptyset$ and

$$\begin{aligned} \log |P'| &\geq \frac{(\log \log N - \log(c_5 \log \log N))^2}{c^s \log \log \log N} + \frac{1}{2} \log \log N \\ &\geq \frac{(\log \log N)^2 - 2(\log \log N) \log(c_5 \log \log N) + (\log(c_5 \log \log N))^2}{c^s \log \log \log N} + \frac{1}{2} \log \log N \\ &\geq \frac{(\log \log N)^2}{c^s \log \log \log N}, \end{aligned}$$

for sufficiently large $c = c(d, t)$. ◀

5 Application: One-sided hyperplanes

Let us consider a finite set H of hyperplanes in \mathbb{R}^d in general position, that is, every d members in H intersect at a distinct point. Let $OSH_d(s, n)$ denote the smallest integer N such that every set H of N hyperplanes in \mathbb{R}^d in general position contains s members H_1 such that the vertex set of the arrangement of H_1 lies above the $x_d = 0$ hyperplane, or contains n members H_2 such that the vertex set of the arrangement of H_2 lies below the $x_d = 0$ hyperplane.

In 1992, Matoušek and Welzl [28] observed that $OSH_2(s, n) = (s-1)(n-1)+1$. Dujmović and Langerman [15] used the existence of $OSH_d(n, n)$ to prove a ham-sandwich cut theorem for hyperplanes. Again by adapting the Erdős-Rado argument, Conlon et al. [13] showed that for $d \geq 3$,

$$OSH_d(s, n) \leq \text{twr}_{d-1}(c_6 s n \log n), \tag{5}$$

where c_6 is a constant that depends only on d . See Eliáš and Matoušek [17] for more related results, including lower bound constructions.

Since each hyperplane $h_i \in H$ is specified by the linear equation

$$a_{i,1}x_1 + \dots + a_{i,d}x_d = b_i,$$

we can represent $h_i \in H$ by the point $h_i^* \in \mathbb{R}^{d+1}$ where $h_i^* = (a_{i,1}, \dots, a_{i,d}, b_i)$ and let $P = \{h_i^* : h_i \in H\}$. Then we define a relation $E \subset \binom{P}{d}$ such that $(h_{i_1}^*, \dots, h_{i_d}^*) \in E$ if and only if $h_{i_1} \cap \dots \cap h_{i_d}$ lies above the hyperplane $x_d = 0$ (i.e. the d -th coordinate of the intersection point is positive). Clearly, E is a semi-algebraic relation with complexity at most $t = t(d)$. Therefore, as an application of Theorem 4 and Corollary 5, we make the following improvement on (5).

► **Theorem 16.** For fixed $s \geq 4$, we have $OSH_3(s, n) \leq 2^{n^{o(1)}}$. For fixed $d \geq 4$ and $s \geq d+1$, we have

$$OSH_d(s, n) \leq \text{twr}_{d-1}(n^{o(1)}).$$

6 Monochromatic triangles

In this section, we will prove Theorem 6.

Proof of Theorem 6. We proceed by induction on m . The base case when $m = 1$ is trivial. Now assume that the statement holds for $m' < m$. Set $N = 2^{cm \log \log m}$, where $c = c(d, t)$ will be determined later, and let $E_1, \dots, E_m \subset \binom{P}{2}$ be (symmetric) semi-algebraic relations on P such that $\binom{P}{2} = E_1 \cup \dots \cup E_m$, and each E_i has complexity at most t . For sake of contradiction, suppose P does not contain three points such that every pair of them is in E_i for some fixed i .

For each relation E_i , there are t polynomials $f_{i,1}, \dots, f_{i,t}$ of degree at most t , and a Boolean function Φ_i such that

$$(p, q) \in E_i \quad \Leftrightarrow \quad \Phi_i(f_{i,1}(p, q) \geq 0, \dots, f_{i,t}(p, q) \geq 0) = 1.$$

For $1 \leq i \leq m, 1 \leq j \leq t, p \in P$, we define the surface $Z_{i,j,p} = \{x \in \mathbb{R}^d : f_{i,j}(p, x) = 0\}$, and let

$$\Sigma = \{Z_{i,j,p} : 1 \leq i \leq m, 1 \leq j \leq t, p \in P\}.$$

Hence $|\Sigma| = mtN$. We apply Lemma 9 to Σ with parameter $r = 2tm$, and decompose \mathbb{R}^d into $c_1(2tm)^{2d}$ regions Δ_i , where $c_1 = c_1(t, d)$ is defined in Lemma 9, such that each region Δ_i is crossed by at most $tmN/r = N/2$ members in Σ . By the pigeonhole principle, there is a region $\Delta \subset \mathbb{R}^d$, such that $|\Delta \cap P| \geq \frac{N}{c_1(2tm)^{2d}}$, and at most $N/2$ members in Σ crosses Δ . Let P_1 be a set of exactly $\lfloor \frac{N}{c_1(2tm)^{2d}} \rfloor$ points in $P \cap \Delta$, and let P_2 be the set of points in $P \setminus P_1$ that does not give rise to a surface that crosses Δ . Hence

$$|P_2| \geq N - \frac{N}{c_1(2tm)^{2d}} - \frac{N}{2} \geq \frac{N}{4}.$$

Therefore, each point $p \in P_2$ has the property that $p \times P_1 \subset E_i$ for some fixed i . We define the function $\chi : P_2 \rightarrow \{1, \dots, m\}$, such that $\chi(p) = i$ if and only if $p \times P_1 \subset E_i$. Set $I = \{\chi(p) : p \in P_2\}$ and $m_0 = |I|$, that is, m_0 is the number of distinct relations (colors) between the sets P_1 and P_2 . Now the proof falls into 2 cases.

Case 1. Suppose $m_0 > \log m$. By the assumption, every pair of points in P_1 is in E_i for some $i \in \{1, \dots, m\} \setminus I$. By the induction hypothesis, we have

$$\frac{2^{cm \log \log m}}{c_1(2tm)^{2d}} \leq |P_1| \leq 2^{c(m-m_0) \log \log m}.$$

Hence

$$cm_0 \log \log m \leq \log(c_1(2tm)^{2d}) \leq 2d \log(c_1 2tm),$$

which implies

$$m_0 \leq \frac{2d \log(c_1 2tm)}{c \log \log m},$$

and we have a contradiction for sufficiently large $c = c(d, t)$.

Case 2. Suppose $m_0 \leq \log m$. By the pigeonhole principle, there is a subset $P_3 \subset P_2$, such that $|P_3| \geq \frac{N}{4m_0}$ and $P_1 \times P_3 \subset E_i$ for some fixed i . Hence every pair of points $p, q \in P_3$ satisfies $(p, q) \notin E_i$, for some fixed i . By the induction hypothesis, we have

$$\frac{2^{cm \log \log m}}{4m_0} \leq |P_3| \leq 2^{c(m-1) \log \log m}.$$

Therefore

$$c \log \log m \leq \log(4m_0) \leq \log(4 \log(m)),$$

which is a contradiction since c is sufficiently large. This completes the proof of Theorem 6. ◀

6.1 Lower bound construction and Schur numbers

Before we prove Theorem 7, let us recall a classic Theorem of Schur [31] which is considered to be one of the earliest applications of Ramsey Theory. A subset of numbers $P \subset \mathbb{R}$ is said to be *sum-free* if for any two (not necessarily distinct) elements $x, y \in P$, their sum $x + y$ is not in P . The Schur number $S(m)$ is defined to be the maximum integer N for which the integers $\{1, \dots, N\}$ can be partitioned into m sum-free sets.

Given a partition $\{1, \dots, N\} = P_1 \cup \dots \cup P_m$ into m parts such that P_i is sum-free, we can define an m -coloring on the edges on a complete $(N + 1)$ -vertex graph which does not contain a monochromatic triangle as follows. Let $V = \{1, \dots, N + 1\}$ be the vertex set, and we define the coloring $\chi : \binom{V}{2} \rightarrow m$ by $\chi(x, y) = i$ iff $|x - y| \in P_i$. Now suppose for sake of contradiction there are vertices x, y, z that induces a monochromatic triangle, say with color i , such that $x < y < z$. Then we have $y - x, z - y, z - x \in P_i$ and $(y - x) + (z - y) = (z - x)$, which is a contradiction since P_i is sum free. Therefore $S(m) < R_2(3; m)$.

Since Schur’s original 1916 paper, the lower bound on $S(m)$ has been improved by several authors [2, 1, 22], and the current record of $S(m) \geq \Omega(3.19^m)$ is due to Fredricksen and Sweet [24]. Their lower bound follows by computing $S(6) \geq 538$, and using the recursive formula

$$S(m) \geq c_\ell(2S(\ell) + 1)^{m/\ell},$$

which was established by Abbott and Hanson [1]. Fredricksen and Sweet also computed $S(7) \geq 1680$, which we will use to prove Theorem 7.

► **Lemma 17.** *For each integer $\ell \geq 1$, there is a set P_ℓ of $(1681)^\ell$ points in \mathbb{R} equipped with semi-algebraic relations $E_1, \dots, E_{7\ell} \subset \binom{P_\ell}{2}$, such that*

1. $E_1 \cup \dots \cup E_{7\ell} = \binom{P_\ell}{2}$,
2. E_i has complexity at most 5000,
3. E_i is translation invariant, that is, $(x, y) \in E_i$ iff $(x + C, y + C) \in E_i$, and
4. the graph $G_{\ell,i} = (P_\ell, E_i)$ is triangle free for all i .

Proof. We start by setting $P_1 = \{1, 2, \dots, 1681\}$. By [24], there is a partition on $\{1, \dots, 1680\} = A_1 \cup \dots \cup A_7$ into seven parts, such that each A_i is sum-free. For $i \in \{1, \dots, 7\}$, we define the binary relation E_i on P_1 by

$$(x, y) \in E_i \iff (1 \leq |x - y| \leq 1680) \wedge (|x - y| \in A_i).$$

Since $|A_i| \leq 1680$, E_i has complexity at most 5000. By the arguments above, the graph $G_{1,i} = (P_1, E_i)$ is triangle free for all $i \in \{1, \dots, 7\}$. In what follows, we blow-up this construction so that the statement holds.

Having defined $P_{\ell-1}$ and $E_1, \dots, E_{7\ell-7}$, we define P_ℓ and $E_{\ell-6}, \dots, E_\ell$ as follows. Let $C = C(\ell)$ be a very large constant, say $C > (5000 \cdot \max\{P_{\ell-1}\})^2$. We construct 1681

translated copies of $P_{\ell-1}$, $Q_i = P_{\ell-1} + iC$ for $1 \leq i \leq 1681$, and set $P_\ell = Q_1 \cup \dots \cup Q_{1681}$. For $1 \leq j \leq 7$, we define the relation $E_{\ell-7+j}$ by

$$(x, y) \in E_{\ell-7+j} \quad \Leftrightarrow \quad (C/2 \leq |x - y| \leq 1682C) \wedge (\exists z \in A_j : ||x - y|/C - z| < 1/1000).$$

Clearly $E_1, \dots, E_{7\ell}$ satisfy properties (1), (2), and (3). The fact that $G_{\ell,i} = (P_\ell, E_i)$ is triangle follows from the same argument as above. \blacktriangleleft

Theorem 7 immediately follows from Lemma 17.

References

- 1 H. L. Abbott and D. Hanson. A problem of schur and its generalizations. *Acta Arith.*, 20:175–187, 1972.
- 2 H. L. Abbott and L. Moser. Sum-free sets of integers. *Acta Arith.*, 11:392–396, 1966.
- 3 P. K. Agarwal and J. Erickson. Optimal partition trees. In *In Proc. 26th Ann. ACM Sympos. Comput. Geom.*, pages 1–10, 2010.
- 4 P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In J. E. Goodman B. Chazelle and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, pages 1–56, 1998.
- 5 M. Ajtai, J. Komlós, and E. Szemerédi. A note on ramsey numbers. *J. Combin. Theory Ser. A*, 29:354–360, 1980.
- 6 N. Alon, J. Pach, R. Pinchasi, R. Radoičić, and M. Sharir. Crossing patterns of semi-algebraic sets. *J. Combin. Theory Ser. A*, 111:310–326, 2005.
- 7 S. Basu, R. Pollack, and M. F. Roy. *Algorithms in Real Algebraic Geometry*. Springer-Verlag, Berlin, 2nd edition edition, 2006.
- 8 T. Bohman. The triangle-free process. *Adv. Math.*, 221:1653–1677, 2009.
- 9 T. Bohman and P. Keevash. The early evolution of the h -free process. *Invent. Math.*, 181:291–336, 2010.
- 10 B. Bukh and M. Matoušek. Erdős-Szekeres-type statements: Ramsey function and decidability in dimension 1. *Duke Math. Journal*, 63:2243–2270, 2014.
- 11 B. Chazelle, H. Edelsbrunner, L. Guibas, and M. Sharir. A singly exponential stratification scheme for real semi-algebraic varieties and its applications. *Theor. Comput. Sci.*, 84:77–105, 1991.
- 12 K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, ii. *Discrete Comput. Geom.*, 4:387–421, 1989.
- 13 D. Conlon, J. Fox, J. Pach, B. Sudakov, and A. Suk. Ramsey-type results for semi-algebraic relations. *Trans. Amer. Math. Soc.*, 366:5043–5065, 2014.
- 14 D. Conlon, J. Fox, and B. Sudakov. Hypergraph ramsey numbers. *J. Amer. Math. Soc.*, 23:247–266, 2010.
- 15 V. Dujmović and S. Langerman. A center transversal theorem for hyperplanes and applications to graph drawing. In *In Proc. 27th Ann. ACM Sympos. Comput. Geom.*, pages 117–124, 2011.
- 16 M. Eliáš, J. Matoušek, E. Roldán-Pensado, and Z. Safernová. Lower bounds on geometric Ramsey functions. *SIAM J. Discrete Math*, 28:1960–1970, 2014.
- 17 M. Eliáš and J. Matoušek. Higher-order Erdős-Szekeres theorems. *Advances in Mathematics*, 244:1–15, 2013.
- 18 P. Erdős. Some remarks on the theory of graphs. *Bull. Amer. Math. Soc.*, 53:292–294, 1947.
- 19 P. Erdős, A. Hajnal, and R. Rado. Partition relations for cardinal numbers. *Acta Math. Acad. Sci. Hungar.*, 16:93–196, 1965.

- 20 P. Erdős and R. Rado. Combinatorial theorems on classifications of subsets of a given set. *Proc. London Math. Soc.*, 3:417–439, 1952.
- 21 P. Erdős and G. Szekeres. A combinatorial problem in geometry. *Compos. Math.*, 2:463–470, 1935.
- 22 G. Exoo. A Lower Bound for Schur numbers and multicolor Ramsey numbers of K_3 . *Electronic J. Combinatorics*, 1:1–3, 1994.
- 23 J. Fox, J. Pach, B. Sudakov, and A. Suk. Erdős-Szekeres-type theorems for monotone paths and convex bodies. *Proceedings of the London Mathematical Society*, 105:953–982, 2012.
- 24 H. Fredricksen and M. Sweet. Symmetric sum-free partitions and lower bounds for schur numbers. *Electronic J. Combinatorics*, 7:1–9, 2000.
- 25 J. H. Kim. The ramsey number $r(3, t)$ has order of magnitude $t^2/\log t$. *Random Structures Algorithms*, 7:173–207, 1995.
- 26 V. Koltun. Almost tight upper bounds for vertical decompositions in four dimensions. *J. ACM*, 51:699–730, 2004.
- 27 J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.*, 8:315–334, 1992.
- 28 J. Matoušek and E. Welzl. Good splitters for counting points in triangles. *J. Algorithms*, 13:307–319, 1992.
- 29 J. Milnor. On the betti numbers of real varieties. *Proc. Amer. Math. Soc.*, 15:275–280, 1964.
- 30 D. Mubayi and A. Suk. A ramsey-type result for geometric ℓ -hypergraphs. *European Journal of Combinatorics*, 41:232–241, 2014.
- 31 I. Schur. Über die Kongruenz $x^m + y^m = z^m \pmod p$. *Jahresber. Deutch. Math. Verein.*, 25:114–117, 1916.
- 32 M. J. Steele. Variations on the monotone subsequence theme of Erdős and Szekeres. In D. Aldous, editor, *Discrete Probability and Algorithms, IMA Volumes in Mathematics and its Applications*, pages 111–131, Berlin, 1995. Springer.
- 33 A. Suk. A note on order-type homogeneous point sets. *Mathematika*, 60:37–42, 2014.
- 34 R. Thom. Sur l’homologie des variétés algébriques réelles. In *Differential and Combinatorial Topology (A Symposium in Honor of Marston Morse)*, pages 255–265, Princeton, N.J., 1965. Princeton University.
- 35 H. E. Warren. Lower bounds for approximation by nonlinear manifold. *Trans. Amer. Math. Soc.*, 133:167–178, 1968.

A Short Proof of a Near-Optimal Cardinality Estimate for the Product of a Sum Set*

Oliver Roche-Newton

Johann Radon Institute for Computational and Applied Mathematics (RICAM)
69 Altenberger Straße, Linz, Austria
o.rochenewton@gmail.com

Abstract

In this note it is established that, for any finite set A of real numbers, there exist two elements $a, b \in A$ such that

$$|(a + A)(b + A)| \gg \frac{|A|^2}{\log |A|}.$$

In particular, it follows that $|(A + A)(A + A)| \gg \frac{|A|^2}{\log |A|}$. The latter inequality had in fact already been established in an earlier work of the author and Rudnev [8], which built upon the recent developments of Guth and Katz [2] in their work on the Erdős distinct distance problem. Here, we do not use those relatively deep methods, and instead we need just a single application of the Szemerédi-Trotter Theorem. The result is also qualitatively stronger than the corresponding sum-product estimate from [8], since the set $(a + A)(b + A)$ is defined by only two variables, rather than four. One can view this as a solution for the pinned distance problem, under an alternative notion of distance, in the special case when the point set is a direct product $A \times A$. Another advantage of this more elementary approach is that these results can now be extended for the first time to the case when $A \subset \mathbb{C}$.

1998 ACM Subject Classification G.2.1 Combinatorics

Keywords and phrases Szemerédi-Trotter Theorem, pinned distances, sum-product estimates

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.74

1 Introduction

In this note, we consider a variation on the sum-product problem, in which the aim is to show that certain sets defined by a combination of additive and multiplicative operations will always be large. For example, given a finite set A of real numbers, define

$$(A - A)(A - A) := \{(a - b)(c - d) : a, b, c, d \in A\}.$$

By the same heuristic arguments that support the Erdős-Szemerédi sum-product conjecture, one expects that $(A - A)(A - A)$ will always be large in comparison to the input set A . In [8], the following¹ bound was established which showed that this is indeed the case:

$$|(A - A)(A - A)| \gg \frac{|A|^2}{\log |A|}. \tag{1}$$

* The author was supported by the Austrian Science Fund (FWF): Project F5511-N26, which is part of the Special Research Program “Quasi-Monte Carlo Methods: Theory and Applications”.

¹ Here and throughout this paper, for positive values X and Y the notation $X \gg Y$ is used as a shorthand for $X \geq cY$, for some absolute constant $c > 0$. If both $X \gg Y$ and $X \ll Y$ hold, we may write $X \approx Y$.



© Oliver Roche-Newton;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 74–80



Leibniz International Proceedings in Informatics

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

The same argument in [8] yields the same lower bound for $|(A + A)(A + A)|$. Some other interesting results in this direction can be found in [1], [3], [4], [6], [7] and [11], amongst others.

In all of the aforementioned works, incidence geometry plays a central role. An extremely influential result in this area is the Szemerédi-Trotter Theorem, which says that, given finite sets P and L of points and lines respectively in \mathbb{R}^2 , the number of incidences between P and L satisfies the upper bound

$$|\{(p, l) \in P \times L : p \in l\}| \ll |P|^{2/3}|L|^{2/3} + |P| + |L|. \tag{2}$$

The quantity on the left hand side of the above inequality is usually denoted by $I(P, L)$. Incidence geometry also played a central role in the recent landmark work of Guth and Katz [2] on the Erdős distinct distances problem. Guth and Katz established an incidence bound for points and lines in \mathbb{R}^3 , which was then used to prove that for any finite set P of points in \mathbb{R}^2 , the set of distinct distances determined by P has near-linear size. To be precise, they proved that

$$|\{d(p, q) : p, q \in P\}| \gg \frac{|P|}{\log |P|}, \tag{3}$$

where $d(p, q)$ denotes the Euclidean distance between p and q . Note that the example $P = [N] \times [N]$, where $[N] = \{1, 2, \dots, N\}$, illustrates that this bound is close to best possible.

One of the tools that Guth and Katz use in their analysis is the Szemerédi-Trotter Theorem. They also introduced polynomial partitioning, and utilise some non-trivial facts from algebraic geometry.

In [8] the authors considered the pseudo-distance $R(p, q)$ in place of $d(p, q)$, where $R(p, q)$ denotes the (signed) area of the axis-parallel rectangle with p and q at opposite corners. To be precise, for two points $p = (p_1, p_2)$ and $q = (q_1, q_2)$ in the plane, we define

$$R(p, q) := (p_1 - q_1)(p_2 - q_2)$$

It was then possible to apply the incidence result of Guth and Katz to establish that

$$|\{R(p, q) : p, q \in P\}| \gg \frac{|P|}{\log |P|}, \tag{4}$$

and (1) followed as a corollary after taking $P = A \times A$. Once again, the example $P = [N] \times [N]$ shows that this bound is close to best possible.

In this note, we prove the following result which strengthens (1):

► **Theorem 1.** *For any set $A \subset \mathbb{R}$, there exist elements $a, a' \in A$ such that*

$$|(A - a)(A - a')| \gg \frac{|A|^2}{\log |A|}.$$

Here, we obtain quadratic growth for a set which depends on only two variables. There are similarities here with the Erdős pinned distance problem, where the aim is to show that, for any finite set $P \subset \mathbb{R}^2$, there exists $p \in P$ such that

$$|\{d(p, q) : q \in P\}| \gg \frac{|P|}{\sqrt{\log |P|}}.$$

This harder version of the Erdős distinct distance problem remains open, with the current best-known result, due to Katz and Tardos [5], stating that there exists $p \in P$ such that

$$|\{d(p, q) : q \in P\}| \gg |P|^\alpha,$$

where $\alpha \approx 0.864$. However, Theorem 1 shows that, if we instead consider the pseudo-distance $R(p, q)$ then we have a near-optimal bound for the corresponding pinned distance problem, in the special case when $P = A \times A$ is a direct product. Such a result, even with the additional direct product restriction, is not currently known for Euclidean distance.

Another advantage of the approach in this paper is that the proof is relatively straightforward. In particular, we obtain a new proof of (1), and in fact a stronger result, without utilising the Guth-Katz machinery.

This paper is closely related to work contained in the PhD thesis of Jones [3] on the growth of sets of real numbers. In fact, the main lemma here, the forthcoming Lemma 3, forms part of the proof of [3, Theorem 5.2], although it is expressed rather differently there in terms of the notion of the cross-ratio. Consequently, we are able to give a new proof of Theorem 5.2 from [3]; that is we establish the following three-variable expander bound

$$\left| \left\{ \frac{a-b}{a-c} : a, b, c \in A \right\} \right| \gg \frac{|A|^2}{\log |A|}.$$

It appears that the proof here is more straightforward than the one originally given by Jones [3].

The only major tool needed in this paper is the Szemerédi-Trotter Theorem. In particular, we use the following standard corollary of (2) which bounds the number of rich lines in an incidence configuration:

► **Corollary 2** (Szemerédi-Trotter Theorem). *Let P be a set of points in \mathbb{R}^2 and let $k \geq 2$ be a real number. Define L_k to be the set of lines containing at least k points from P . Then*

$$|L_k| \ll \frac{|P|^2}{k^3} + \frac{|P|}{k}. \quad (5)$$

In particular, if $k \leq |P|^{1/2}$, then

$$|L_k| \ll \frac{|P|^2}{k^3}. \quad (6)$$

2 Energy bound

► **Lemma 3.** *Let Q denote the number of solutions to the equation*

$$(a-b)(a'-c') = (a-c)(a'-b') \quad (7)$$

such that $a, a', b, b', c, c' \in A$. Then

$$Q \ll |A|^4 \log |A|.$$

Proof. First of all, the number solutions to (7) of the form

$$(a-b)(a'-c') = (a-c)(a'-b') = 0,$$

is at most $4|A|^4$. Also, there are at most $|A|^4$ trivial solutions whereby $b = c$. Now, let Q^* denote the number of solutions to

$$(a-b)(a'-c') = (a-c)(a'-b') \neq 0, \quad b \neq c. \quad (8)$$

This is the same as the number of solutions to

$$\frac{a-b}{a'-b'} = \frac{a-c}{a'-c'} \neq 0, \quad b \neq c. \quad (9)$$

Let $P = A \times A$ and let $L(P)$ denote the set of lines determined by P . That is, $L(P)$ is the set of lines supporting 2 or more points from the set. Note that a, a', b, b', c and c' satisfy (9) only if the points $(a', a), (b', b)$ and (c', c) from P are collinear and distinct. Therefore,

$$Q^* \leq \sum_{l \in L(P)} |l \cap P|^3 \ll \sum_j \sum_{2^j \leq |l \cap P| < 2^{j+1}} |l \cap P|^3,$$

where j ranges over all positive integers such that $2^j \leq |A|$. Note that there are no lines in $L(P)$ which contain more than $|A|$ points from P , which is why this sum does not need to include any larger values of j .

For the aforementioned range of values for j , it follows from Corollary 2, and in particular bound (6), that

$$|\{l : |l \cap P| \geq 2^j\}| \ll \frac{|P|^2}{(2^j)^3}.$$

Therefore,

$$Q^* \ll \sum_j |P|^2 \ll |A|^4 \log |A|.$$

Finally, $Q \ll |A|^4 + Q^* \ll |A|^4 \log |A|$, as required. ◀

► **Corollary 4.** *For any finite set $A \subset \mathbb{R}$,*

$$\left| \left\{ \frac{a-b}{a-c} : a, b, c \in A \right\} \right| \gg \frac{|A|^2}{\log |A|}.$$

Proof. Let

$$n(x) := \left| \left\{ (a, b, c) \in A^3 : \frac{a-b}{a-c} = x \right\} \right|$$

denote the number of representations of x as an element of the set in question. We know that

$$|A|^3 \ll |A|^3 - |A|^2 = \sum_x n(x).$$

Also, the quantity $\sum_x n^2(x)$ is strictly² less than the number of solutions to (7). Therefore, it follows from the Cauchy-Schwarz inequality and Lemma 3 that

$$\begin{aligned} |A|^6 &\ll \left(\sum_x n(x) \right)^2 \leq \left| \left\{ \frac{a-b}{a-c} : a, b, c \in A \right\} \right| \sum_x n^2(x) \\ &\ll \left| \left\{ \frac{a-b}{a-c} : a, b, c \in A \right\} \right| |A|^4 \log |A|, \end{aligned}$$

and the result follows after rearranging this inequality. ◀

² The quantity $\sum_x n^2(x)$ is the number of solutions to (7), minus the number of solutions for which $a = c$ or $a' = c'$.

2.1 Remarks

Let $E^*(A, B)$ be the *multiplicative energy* of A and B ; that is, the number of solutions to

$$ab = a'b'$$

such that $a, a' \in A$ and $b, b' \in B$. Using this notation, Lemma 3 can be expressed in the form of the following bound:

$$\sum_{a, a' \in A} E^*(a - A, a' - A) \ll |A|^4 \log |A|. \quad (10)$$

See [6, Lemma 2.4] for a similar bound on the sum of multiplicative energies after different additive shifts.

The proof of Lemma 3 can undergo a number of small modifications in order to deduce slightly different results involving multiple sets $A, B, C, \dots \in \mathbb{R}$ of approximately the same size. For example, if we instead take $P = (A \cup B) \times (A \cup B)$, where $|B| \approx |A|$, then the number of solutions to (9) such that $a, a' \in A$ and $b, b', c, c' \in B$ is less than the number of collinear triples in the point set P . After repeating the argument of Lemma 3, it follows that

$$\sum_{a, a' \in A} E^*(a - B, a' - B) \ll |A|^4 \log |A|. \quad (11)$$

In particular, if $B = -A$, this yields

$$\sum_{a, a' \in A} E^*(a + A, a' + A) \ll |A|^4 \log |A|. \quad (12)$$

3 Proof of Theorem 1

It follows from (10) that there exist $a, a' \in A$ such that

$$E^*(a - A, a' - A) \ll |A|^2 \log |A|. \quad (13)$$

We also have the following well-known bound for the multiplicative energy, which follows from an application of the Cauchy-Schwarz inequality:

$$E^*(A, B) \geq \frac{|A|^2 |B|^2}{|AB|}. \quad (14)$$

After comparing (13) and (14), it follows that

$$|(A - a)(A - a')| \gg \frac{|A|^2}{\log |A|},$$

as required. ◀

3.1 Remark

By the same argument, but utilising (12) in place of (10), it also follows that there exist $a, a' \in A$ such that

$$|(A + a)(A + a')| \gg \frac{|A|^2}{\log |A|}.$$

4 The complex setting

As stated in the abstract, an advantage of this more straightforward approach is that it allows for results that were previously only known for sets of real numbers to be extended to the complex setting. The only tool used in the proofs of Theorem 1 and Corollary 4 is the Szemerédi-Trotter Theorem. It is now known that this theorem holds for sets of points and lines in \mathbb{C}^2 (this was first proven by [10], with a more modern proof given by Zahl [12]; see also Solymosi and Tao [9]).

One can therefore repeat the analysis of this paper verbatim in the complex setting, applying the complex Szemerédi-Trotter Theorem in place of the real version, and deduce exactly the same results for a set A of complex numbers.

In particular, we deduce that for any finite set $A \subset \mathbb{C}$, there exist $a, a' \in A$ such that

$$|(A - a)(A - a')| \gg \frac{|A|^2}{\log |A|} \quad (15)$$

and it follows that

$$|(A - A)(A - A)| \gg \frac{|A|^2}{\log |A|}. \quad (16)$$

Since the earlier proof of (16) for real A in [8] was based on the three dimensional incidence bounds in [2], it was not previously known that this bound extended to the complex setting. Similarly, the approach in this paper can be used to show that for any finite set $A \subset \mathbb{C}$, we have

$$\left| \left\{ \frac{a-b}{a-c} : a, b, c \in A \right\} \right| \gg \frac{|A|^2}{\log |A|}.$$

Acknowledgements. I am grateful to Brendan Murphy for his helpful feedback. I am also grateful to Ilya Shkredov for helpful conversations and especially for his interpretation of the work of Jones [3].

References

- 1 G. Elekes, M. Nathanson and I. Ruzsa, “Convexity and sumsets”, *J Number Theory*. **83** (1999), 194–201.
- 2 L. Guth and N.H. Katz, “On the Erdős distinct distance problem in the plane”, *Ann. of Math.* **181** (2015), no. 1, 155–190.
- 3 T. G. F. Jones, “New quantitative estimates on the incidence geometry and growth of finite sets”, Ph.D thesis, *arXiv:1301.4853* (2013).
- 4 T. G. F. Jones, “New results for the growth of sets of real numbers”, *arXiv:1202.4972* (2012).
- 5 N.H. Katz and G. Tardos, “A new entropy inequality for the Erdős distance problem”, *Towards a theory of geometric graphs*, *Contemp. Math.* **342** (2004), 119–126.
- 6 B. Murphy, O. Roche-Newton and I. Shkredov, “Variations on the sum-product problem”, to appear in *SIAM J. Discrete Math.*, preprint *arxiv:1312.6438* (2013).
- 7 O. Raz, M. Sharir and J. Solymosi, “Polynomials vanishing on grids: The Elekes-Rónyai problem revisited”, to appear in *Amer. J. Math.*, preprint *arxiv:1401.7419* (2014).
- 8 O. Roche-Newton and M. Rudnev, “On the Minkowski distances and products of sum sets”, to appear in *Israel J. Math.*, preprint *arxiv:1203.6237* (2012).

- 9 J. Solymosi and T. Tao, “An incidence theorem in higher dimensions”, *Discrete Comput. Geom.* **48** (2012), 255–280.
- 10 C. Tóth, “The Szemerédi-Trotter theorem in the complex plane”, *arXiv:math/0305283*.
- 11 P. Ungar, “ $2N$ non collinear points determine at least $2N$ directions”, *J. Combin. Theory Ser. A* **33** (1982), 343–347.
- 12 J. Zahl, “A Szemerédi-Trotter type theorem in \mathbb{R}^4 ”, *arXiv:1203.4600*.

A Geometric Approach for the Upper Bound Theorem for Minkowski Sums of Convex Polytopes*

Menelaos I. Karavelas^{1,2} and Eleni Tzanaki²

1 Department of Mathematics and Applied Mathematics
University of Crete, Heraklion, Greece
mkaravel@uoc.gr

2 Institute of Applied and Computational Mathematics
Foundation for Research and Technology – Hellas, Heraklion, Greece
etzanaki@uoc.gr

Abstract

We derive tight expressions for the maximum number of k -faces, $0 \leq k \leq d - 1$, of the Minkowski sum, $P_1 + \dots + P_r$, of r convex d -polytopes P_1, \dots, P_r in \mathbb{R}^d , where $d \geq 2$ and $r < d$, as a (recursively defined) function on the number of vertices of the polytopes. Our results coincide with those recently proved by Adiprasito and Sanyal [1]. In contrast to Adiprasito and Sanyal’s approach, which uses tools from Combinatorial Commutative Algebra, our approach is purely geometric and uses basic notions such as f - and h -vector calculus, stellar subdivisions and shellings, and generalizes the methodology used in [10] and [9] for proving upper bounds on the f -vector of the Minkowski sum of two and three convex polytopes, respectively. The key idea behind our approach is to express the Minkowski sum $P_1 + \dots + P_r$ as a section of the Cayley polytope \mathcal{C} of the summands; bounding the k -faces of $P_1 + \dots + P_r$ reduces to bounding the subset of the $(k + r - 1)$ -faces of \mathcal{C} that contain vertices from each of the r polytopes. We end our paper with a sketch of an explicit construction that establishes the tightness of the upper bounds.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems – Geometrical problems and computations, G.2.1 Combinatorics

Keywords and phrases Convex polytopes, Minkowski sum, upper bound

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.81

1 Introduction

Given two sets A and B in \mathbb{R}^d , $d \geq 2$, their Minkowski sum $A + B$ is the set $\{a + b \mid a \in A, b \in B\}$. The Minkowski sum definition can be extended naturally to any number of summands: $A_{[r]} := A_1 + A_2 + \dots + A_r = \{a_1 + a_2 + \dots + a_r \mid a_i \in A_i, 1 \leq i \leq r\}$. Minkowski sums have a wide range of applications, including algebraic geometry, computational commutative algebra, collision detection, computer-aided design, graphics, robot motion planning and game theory, just to name a few (see also [1], [9] and the references therein).

In this paper we focus on convex polytopes, and we are interested in computing the worst-case complexity of their Minkowski sum. More precisely, given r d -polytopes P_1, \dots, P_r

* The work in this paper has been partially supported by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: THALIS – UOA (MIS 375891).



in \mathbb{R}^d , we seek tight bounds on the number of k -faces $f_k(P_{[r]})$, $0 \leq k \leq d-1$, of their Minkowski sum $P_{[r]} := P_1 + P_2 + \dots + P_r$. This problem, which can be seen as a generalization of the Upper Bound Theorem (UBT) for polytopes [14], has a history of more than 20 years. Gritzmann and Sturmfels [7] were the first to consider the problem, and gave a complete answer for it, for any number of d -polytopes in \mathbb{R}^d , in terms of the number of non-parallel edges of the r polytopes. More than 10 years later, Fukuda and Weibel [5] proved tight upper bounds on the number of k -faces of the Minkowski sum of two 3-polytopes, expressed either in terms of the number of vertices or number of facets of the summands. Fogel, Halperin, and Weibel [4] extended one of the results in [5], and expressed the number of facets of the Minkowski sum of r 3-polytopes in terms of the number of facets of the summands. Quite recently Weibel [16] provided a relation for the number of k -faces of the Minkowski sum of $r \geq d$ summands in terms of the k -faces of the Minkowski sums of subsets of size at most $d-1$ of these summands. This result should be viewed in conjunction with a result by Sanyal [15] stating that the number of vertices of the Minkowski sum of r d -polytopes, where $r \geq d$, is strictly less than the product of the vertices of the summands (whereas for $r < d$ this is indeed possible). About 3 years ago, the authors of this paper proved the first tight upper bound on the number of k -faces for the Minkowski sum of two d -polytopes in \mathbb{R}^d , for any $d \geq 2$ and for all $0 \leq k \leq d-1$ (cf. [10]), a result which was subsequently extended to three summands in collaboration with Konaxis (cf. [9]).

In a recent paper, Adiprasito and Sanyal [1] provide the complete resolution of the *Upper Bound Theorem for Minkowski sums (UBTM)*. In particular, they show that there exists, what they call, a *Minkowski-neighborly* family of r d -polytopes N_1, \dots, N_r , with $f_0(N_i) = n_i$, $1 \leq i \leq r$, such that for any r d -polytopes $P_1, P_2, \dots, P_r \subset \mathbb{R}^d$ with $f_0(P_i) = n_i$, $1 \leq i \leq r$, $f_k(P_{[r]})$ is bounded by above by $f_k(N_{[r]})$, for all $0 \leq k \leq d-1$. The majority of the arguments in the UBTM proof by Adiprasito and Sanyal make use of powerful tools from Combinatorial Commutative Algebra. The high-level layout of the proof is analogous to McMullen's proof of the UBT, as well as the proofs of the UBTM in [10] and [9] for two and three summands, respectively:

1. Consider the Cayley polytope $\mathcal{C} \subset \mathbb{R}^{d+r-1}$ of the r polytopes P_1, P_2, \dots, P_r , and identify their Minkowski sum as a section of \mathcal{C} with an appropriately defined d -flat \overline{W} . Let $\mathcal{F} \subset \mathbb{R}^{d+r-1}$ be the faces of \mathcal{C} that intersect \overline{W} , and let \mathcal{K} be the closure of \mathcal{F} under subface inclusion (\mathcal{K} is a $(d+r-1)$ -polytopal complex). By the Cayley trick, there is a bijection between the faces of \mathcal{F} and the faces of $P_{[r]}$; as a result, to bound the number of faces of $P_{[r]}$ it suffices to bound the number of faces of \mathcal{F} .
2. Define the h -vector $\mathbf{h}(\mathcal{F})$ of \mathcal{F} , and prove the Dehn-Sommerville equations for $\mathbf{h}(\mathcal{F})$, relating its elements to the elements of $\mathbf{h}(\mathcal{K})$.
3. Prove a recurrence relation for the elements of $\mathbf{h}(\mathcal{F})$.
4. Use the recurrence relation above to prove upper bounds for $h_k(\mathcal{F})$, for all $0 \leq k \leq \lfloor \frac{d+r-1}{2} \rfloor$.
5. Prove upper bounds for $h_k(\mathcal{K})$, for all $0 \leq k \leq \lfloor \frac{d+r-1}{2} \rfloor$.
6. Provide necessary and sufficient conditions under which the elements of both $\mathbf{h}(\mathcal{F})$ and $\mathbf{h}(\mathcal{K})$ are maximized for all k . These conditions are conditions on the *lower half* of the h -vector of \mathcal{F} . Due to the relation between the f - and h -vectors of \mathcal{F} , these are also conditions for the maximality of the elements of $\mathbf{f}(\mathcal{F})$.
7. Describe a family of polytopes for which the necessary and sufficient conditions hold; clearly, such a family establishes the tightness of the upper bounds.

In Adiprasito and Sanyal's proof steps 2, 3 and 4 are proved by introducing a powerful new theory that they call the *relative Stanley-Reisner theory* for simplicial complexes. The focus of this theory is on relative simplicial complexes, and is able to reveal properties of such complexes not only under topological restrictions, but also account for their combinatorial

and geometric structure. To apply their theory, Adiprasito and Sanyal consider the simplicial complex \mathcal{K} and then define \mathcal{F} as a relative simplicial complex (they call them the Cayley and *relative Cayley* complex, respectively). They then apply their relative Stanley-Reisner theory to \mathcal{F} to establish the Dehn-Sommerville equations of step 2, the recurrence relation of step 3 and finally the upper bounds for $\mathbf{h}(\mathcal{F})$ in 4. Steps 5 and 6 are done by clever algebraic manipulation of the h -vectors of \mathcal{F} and \mathcal{K} , by exploiting the geometric properties of \mathcal{K} , and by making use of the recurrence relation in step 3. Step 7 is reduced to results by Matschke, Pfeifle, and Pilaud [13] and Weibel [16].

Our contribution. In what follows, we provide a completely geometric proof of the UBTM, that generalizes the technique we used in [10] and [9] for two and three summands to the case of r summands, when $r < d$. Instead of relying on algebraic tools, we use basic notions from combinatorial geometry, such as stellar subdivisions and shellings. Our proof, in essence, differs from that of Adiprasito and Sanyal in steps 2, 3, 4 and 5 of the layout above (the remaining steps do not use tools from Combinatorial Commutative Algebra anyway).

In more detail, to prove the various intermediate results, towards the UBTM, we consider the Cayley polytope \mathcal{C} and we perform a series of stellar subdivisions to get a simplicial polytope \mathcal{Q} . From the analysis of the combinatorial structure of \mathcal{Q} , we derive the Dehn-Sommerville equations of step 2 (see Sections 3 and 4), as well as the recurrence relation of step 3 (see Section 5). This recurrence relation is then used for establishing the upper bounds for the elements of $\mathbf{h}(\mathcal{F})$ and $\mathbf{h}(\mathcal{K})$ (see Section 6). We end with a construction similar to the one presented in [13, Theorem 2.6], that establishes the tightness of the upper bounds (see Section 7). Due to space limitations, the majority of the proofs have been omitted; the interested reader may refer to the full version of the paper [11].

2 Preliminaries

Let P be a d -dimensional polytope, or d -polytope for short. Its dimension is the dimension of its affine span. The faces of P are \emptyset, P , and the intersections of P with its supporting hyperplanes. The \emptyset and P faces are called *improper*, while the remaining faces are called *proper*. Each face of P is itself a polytope, and a face of dimension k is called a k -face. Faces of P of dimension 0, 1, $d-2$ and $d-1$ are called vertices, edges, ridges, and facets, respectively.

A d -dimensional *polytopal complex* or, simply, *d -complex*, \mathcal{C} is a finite collection of polytopes in \mathbb{R}^d such that (i) $\emptyset \in \mathcal{C}$, (ii) if $P \in \mathcal{C}$ then all the faces of P are also in \mathcal{C} and (iii) the intersection $P \cap Q$ for two polytopes P and Q in \mathcal{C} is a face of both. The dimension $\dim(\mathcal{C})$ of \mathcal{C} is the largest dimension of a polytope in \mathcal{C} . A polytopal complex is called *pure* if all its maximal (with respect to inclusion) faces have the same dimension. In this case the maximal faces are called the *facets* of \mathcal{C} . A polytopal complex is *simplicial* if all its faces are simplices. A polytopal complex \mathcal{C}' is called a *subcomplex* of a polytopal complex \mathcal{C} if all faces of \mathcal{C}' are also faces of \mathcal{C} . For a polytopal complex \mathcal{C} , the *star* of v in \mathcal{C} , denoted by $\text{star}(v, \mathcal{C})$, is the subcomplex of \mathcal{C} consisting of all faces that contain v , and their faces. The *link* of v , denoted by \mathcal{C}/v , is the subcomplex of $\text{star}(v, \mathcal{C})$ consisting of all the faces of $\text{star}(v, \mathcal{C})$ that do not contain v .

A d -polytope P , together with all its faces, forms a d -complex, denoted by $\mathcal{C}(P)$. The polytope P itself is the only maximal face of $\mathcal{C}(P)$, i.e., the only facet of $\mathcal{C}(P)$, and is called the *trivial* face of $\mathcal{C}(P)$. Moreover, all proper faces of P form a pure $(d-1)$ -complex, called the *boundary complex* $\mathcal{C}(\partial P)$, or simply ∂P , of P . The facets of ∂P are just the facets of P .

For a $(d-1)$ -complex \mathcal{C} , its f -vector is defined as $\mathbf{f}(\mathcal{C}) = (f_{-1}, f_0, f_1, \dots, f_{d-1})$, where

$f_k = f_k(\mathcal{C})$ denotes the number of k -faces of P and $f_{-1}(\mathcal{C}) := 1$ corresponds to the empty face of \mathcal{C} . From the f -vector of \mathcal{C} we define its h -vector as the vector $\mathbf{h}(\mathcal{C}) = (h_0, h_1, \dots, h_d)$, where $h_k = h_k(\mathcal{C}) := \sum_{i=0}^k (-1)^{k-i} \binom{d-i}{d-k} f_{i-1}(\mathcal{C})$, $0 \leq k \leq d$.

Denote by \mathcal{Y} a generic subset of faces of a polytopal complex \mathcal{C} , and define its dimension $\dim(\mathcal{Y})$ as the maximum of the dimensions of its faces. Let $\dim(\mathcal{Y}) = \delta - 1$; then we may define (if not already properly defined), the h -vector $\mathbf{h}(\mathcal{Y})$ of \mathcal{Y} as:

$$h_k(\mathcal{Y}) = \sum_{i=0}^{\delta} (-1)^{k-i} \binom{\delta-i}{\delta-k} f_{i-1}(\mathcal{Y}). \quad (2.1)$$

We can further define the m -order g -vector of \mathcal{Y} according to the following recursive formula:

$$g_k^{(m)}(\mathcal{Y}) = \begin{cases} h_k(\mathcal{Y}), & m = 0, \\ g_k^{(m-1)}(\mathcal{Y}) - g_{k-1}^{(m-1)}(\mathcal{Y}), & m > 0. \end{cases} \quad (2.2)$$

Clearly, $\mathbf{g}^{(m)}(\mathcal{Y})$ is nothing but the backward m -order finite difference of $\mathbf{h}(\mathcal{Y})$; therefore:

$$g_k^{(m)}(\mathcal{Y}) = \sum_{i=0}^m (-1)^i \binom{m}{i} h_{k-i}(\mathcal{Y}), \quad k, m \geq 0. \quad (2.3)$$

Observe that for $m = 0$ we get the h -vector of \mathcal{Y} , while for $m = 1$ we get what is typically defined as the g -vector.

The relation between the f - and h -vector of \mathcal{Y} is better manipulated using generating functions. We define the f -polynomial and h -polynomial of \mathcal{Y} as follows:

$$\mathbf{f}(\mathcal{Y}; t) = \sum_{i=0}^{\delta} f_{i-1} t^{\delta-i} = f_{\delta-1} + f_{\delta-2} t + \dots + f_{-1} t^{\delta}, \quad \mathbf{h}(\mathcal{Y}; t) = \sum_{i=0}^{\delta} h_i t^{\delta-i} = h_{\delta} + h_{\delta-1} t + \dots + h_0 t^{\delta},$$

where, we simplified $f_i(\mathcal{Y})$ and $h_i(\mathcal{Y})$ to f_i and h_i . In this set-up, the relation between the f -vector and h -vector (cf. (2.1)) can be expressed as:

$$\mathbf{f}(\mathcal{Y}; t) = \mathbf{h}(\mathcal{Y}; t+1), \quad \text{or, equivalently, as} \quad \mathbf{h}(\mathcal{Y}; t) = \mathbf{f}(\mathcal{Y}; t-1). \quad (2.4)$$

2.1 The Cayley embedding, the Cayley polytope and the Cayley trick

Let P_1, P_2, \dots, P_r be r d -polytopes with vertex sets $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r$, respectively. Let $\mathbf{e}_0, \mathbf{e}_1, \dots, \mathbf{e}_{r-1}$ be an affine basis of \mathbb{R}^{r-1} and call $\mu_i : \mathbb{R}^d \rightarrow \mathbb{R}^{r-1} \times \mathbb{R}^d$ the affine inclusion given by $\mu_i(\mathbf{x}) = (\mathbf{e}_{i-1}, \mathbf{x})$, $1 \leq i \leq r$. The *Cayley embedding* $\mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r)$ of the point sets $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r$ is defined as $\mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r) = \bigcup_{i=1}^r \mu_i(\mathcal{V}_i)$. The polytope corresponding to the convex hull $\text{conv}(\mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r))$ of the Cayley embedding $\mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r)$ of $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r$ is typically referred to as the *Cayley polytope* of P_1, P_2, \dots, P_r .

The following lemma, known as *the Cayley trick for Minkowski sums*, relates the Minkowski sum of the polytopes P_1, P_2, \dots, P_r with their Cayley polytope.

► **Lemma 2.1** ([8, Lemma 3.2]). *Let P_1, P_2, \dots, P_r be r d -polytopes with vertex sets $\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r \subset \mathbb{R}^d$. Moreover, let \overline{W} be the d -flat defined as $\{\frac{1}{r}\mathbf{e}_0 + \dots + \frac{1}{r}\mathbf{e}_{r-1}\} \times \mathbb{R}^d \subset \mathbb{R}^{r-1} \times \mathbb{R}^d$. Then, the Minkowski sum $P_{[r]}$ has the following representation as a section of the Cayley embedding $\mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r)$ in $\mathbb{R}^{r-1} \times \mathbb{R}^d$:*

$$\begin{aligned} P_{[r]} &\cong \mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r) \cap \overline{W} \\ &:= \left\{ \text{conv}\{(\mathbf{e}_{i-1}, \mathbf{v}) \mid 1 \leq i \leq r\} \cap \overline{W} : (\mathbf{e}_{i-1}, \mathbf{v}) \in \mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r), 1 \leq i \leq r \right\}. \end{aligned}$$

Moreover, F is a facet of $P_{[r]}$ if and only if it is of the form $F = F' \cap \overline{W}$ for a facet F' of $\mathcal{C}(\mathcal{V}_1, \mathcal{V}_2, \dots, \mathcal{V}_r)$ containing at least one point $(\mathbf{e}_{i-1}, \mathbf{v})$ for all $1 \leq i \leq r$.

Let $\mathcal{C}_{[r]}$ be the Cayley polytope of P_1, P_2, \dots, P_r , and call $\mathcal{F}_{[r]}$ the set of faces of $\mathcal{C}_{[r]}$ that have non-empty intersection with the d -flat \overline{W} . A direct consequence of Lemma 2.1 is a bijection between the $(k - 1)$ -faces of \overline{W} and the $(k - r)$ -faces of $\mathcal{F}_{[r]}$, for $r \leq k \leq d + r - 1$. This further implies that:

$$f_{k-1}(\mathcal{F}_{[r]}) = f_{k-r}(P_{[r]}), \quad \text{for all } r \leq k \leq d + r - 1. \tag{2.5}$$

In what follows, to keep the notation lean, we identify $V_i := \mu_i(\mathcal{V}_i)$ with its pre-image \mathcal{V}_i . For any $\emptyset \subset R \subseteq [r]$, we denote by \mathcal{C}_R the Cayley polytope of the polytopes P_i where $i \in R$. In particular, if $R = \{i\}$ for some $i \in [r]$, then $\mathcal{C}_{\{i\}} \equiv P_i$. We shall assume below that $\mathcal{C}_{[r]}$ is “*as simplicial as possible*”. This means that we consider all faces of $\mathcal{C}_{[r]}$ to be simplicial, except possibly for the trivial faces $\{\mathcal{C}_R\}^1$, $\emptyset \subset R \subseteq [r]$. Otherwise, we can employ the so called *bottom-vertex triangulation* [12, Section 6.5, pp. 160–161] to triangulate all proper faces of $\mathcal{C}_{[r]}$ except for the trivial ones, i.e., $\{\mathcal{C}_R\}$, $\emptyset \subset R \subseteq [r]$. The resulting complex is polytopal (cf. [2]) with all its faces being simplices, except possibly for the trivial ones. Moreover, it has the same number of vertices as $\mathcal{C}_{[r]}$, while the number of its k -faces is never less than the number of k -faces of $\mathcal{C}_{[r]}$.

For each $\emptyset \subset R \subseteq [r]$, we denote by \mathcal{F}_R the set of faces of \mathcal{C}_R having at least one vertex from each V_i , $i \in R$, and we call it the set of *mixed faces of \mathcal{C}_R* . We trivially have that $\mathcal{F}_{\{i\}} \equiv \partial P_i$. We define the dimension of \mathcal{F}_R to be the maximum dimension of the faces in \mathcal{F}_R , i.e., $\dim(\mathcal{F}_R) = \max_{F \in \mathcal{F}_R} \dim(F) = d + |R| - 2$. Under the “*as simplicial as possible*” assumption above, the faces in \mathcal{F}_R are simplices. We denote by \mathcal{K}_R the *closure*, under subface inclusion, of \mathcal{F}_R . By construction, \mathcal{K}_R contains: (1) all faces in \mathcal{F}_R , (2) all faces that are subfaces of faces in \mathcal{F}_R , and (3) the empty set. It is easy to see that \mathcal{K}_R does not contain any of the trivial faces $\{\mathcal{C}_S\}$, $\emptyset \subset S \subseteq R$, and thus, \mathcal{K}_R is a pure simplicial $(d + |R| - 2)$ -complex. It is also easy to verify that:

$$f_k(\mathcal{K}_R) = \sum_{\emptyset \subset S \subseteq R} f_k(\mathcal{F}_S), \quad -1 \leq k \leq d + |R| - 2, \tag{2.6}$$

where in order for the above equation to hold for $k = -1$, we set $f_{-1}(\mathcal{F}_S) = (-1)^{|S|-1}$ for all $\emptyset \subset S \subseteq R$. In what follows we use the convention that $f_k(\mathcal{F}_R) = 0$, for any $k < -1$ or $k > d + |R| - 2$.

A general form of the Inclusion-Exclusion Principle states that if f and g are two functions defined over the subsets of a finite set A , such that $f(A) = \sum_{\emptyset \subset B \subseteq A} g(B)$, then $g(A) = \sum_{\emptyset \subset B \subseteq A} (-1)^{|A|-|B|} f(B)$ [6, Theorem 12.1]. Applying this principle to (2.6), we deduce that:

$$f_k(\mathcal{F}_R) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} f_k(\mathcal{K}_S), \quad -1 \leq k \leq d + |R| - 2. \tag{2.7}$$

In the majority of our proofs that involve evaluation of f - and h -vectors, we use generating functions as they significantly simplify calculations. The starting point is to evaluate $\mathbf{f}(\mathcal{K}_R; t)$ (resp., $\mathbf{f}(\mathcal{F}_R; t)$) in terms of the generating functions $\mathbf{f}(\mathcal{F}_S; t)$ (resp., $\mathbf{f}(\mathcal{K}_S; t)$), $\emptyset \subset S \subseteq R$, for each fixed choice of $\emptyset \subset R \subseteq [r]$. Then, using (2.4) we derive the analogous relations between their h -vectors.

¹ We denote by $\{\mathcal{C}_R\}$ the polytope \mathcal{C}_R as a trivial face itself (without its non-trivial faces).

Recalling that $\dim(\mathcal{K}_R) = d + |R| - 2$ and $\dim(\mathcal{F}_S) = d + |S| - 2$ we have:

$$\begin{aligned} \mathbf{f}(\mathcal{K}_R; t) &= \sum_{k=0}^{d+|R|-1} f_{k-1}(\mathcal{K}_R) t^{d+|R|-1-k} \stackrel{(2.6)}{=} \sum_{k=0}^{d+|R|-1} \sum_{\emptyset \subset S \subseteq R} f_{k-1}(\mathcal{F}_S) t^{d+|R|-1-k} \\ &= \sum_{\emptyset \subset S \subseteq R} t^{|R|-|S|} \sum_{k=0}^{d+|R|-1} f_{k-1}(\mathcal{F}_S) t^{d+|S|-1-k} = \sum_{\emptyset \subset S \subseteq R} t^{|R|-|S|} \mathbf{f}(\mathcal{F}_S; t). \end{aligned} \quad (2.8)$$

Rewriting the above relation as $t^{-|R|} \mathbf{f}(\mathcal{K}_R; t) = \sum_{\emptyset \subset S \subseteq R} t^{-|S|} \mathbf{f}(\mathcal{F}_S; t)$ and using Möbius inversion, we get:

$$\mathbf{f}(\mathcal{F}_R; t) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} t^{|R|-|S|} \mathbf{f}(\mathcal{K}_S; t). \quad (2.9)$$

Setting $t := t - 1$ in (2.8) we have:

$$\begin{aligned} \mathbf{h}(\mathcal{K}_R; t) &= \mathbf{f}(\mathcal{K}_R; t - 1) = \sum_{\emptyset \subset S \subseteq R} (t - 1)^{|R|-|S|} \mathbf{f}(\mathcal{F}_S; t - 1) \\ &= \sum_{\emptyset \subset S \subseteq R} (t - 1)^{|R|-|S|} \mathbf{h}(\mathcal{F}_S; t) = \sum_{\emptyset \subset S \subseteq R} \mathbf{g}^{(|R|-|S|)}(\mathcal{F}_S; t). \end{aligned} \quad (2.10)$$

Similarly, from (2.9) we obtain:

$$\mathbf{h}(\mathcal{F}_R; t) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \mathbf{g}^{(|R|-|S|)}(\mathcal{K}_S; t). \quad (2.11)$$

Comparing coefficients in the above generating functions, we deduce that:

$$h_k(\mathcal{K}_R) = \sum_{\emptyset \subset S \subseteq R} g_k^{(|R|-|S|)}(\mathcal{F}_S), \quad \text{for all } 0 \leq k \leq d + |R| - 1, \quad \text{and} \quad (2.12)$$

$$h_k(\mathcal{F}_R) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} g_k^{(|R|-|S|)}(\mathcal{K}_S), \quad \text{for all } 0 \leq k \leq d + |R| - 1. \quad (2.13)$$

3 The construction of the auxiliary simplicial polytope $\mathcal{Q}_{[r]}$

The proper faces of the Cayley polytope $\mathcal{C}_{[r]}$ of P_1, \dots, P_r are the faces in each \mathcal{F}_R , $\emptyset \subset R \subseteq [r]$ as well as all trivial faces $\{\mathcal{C}_R\}$ with $\emptyset \subset R \subseteq [r]$. Since the latter are not necessarily simplices, the Cayley polytope $\mathcal{C}_{[r]}$ may not be simplicial. In order to exploit the combinatorial structure of $\mathcal{C}_{[r]}$, we add auxiliary points on $\mathcal{C}_{[r]}$ so that the resulting polytope, denoted by $\mathcal{Q}_{[r]}$, is simplicial.

The main tool for describing our construction is *stellar subdivisions*. Let $P \subset \mathbb{R}^d$ be a d -polytope, and consider a point y_F in the relative interior of a face F of ∂P . The *stellar subdivision* $\text{st}(y_F, \partial P)$ of ∂P over F , replaces F by the set of faces $\{y_F, F'\}$ where F' is a non-trivial face of F . It is a well-known fact that stellar subdivisions preserve polytopality (cf. [3, pp. 70–73]), in the sense that the newly constructed complex is combinatorially equivalent to a polytope each facet of which lies on a distinct supporting hyperplane.

Our goal is to triangulate each face $\{\mathcal{C}_R\}$, $\emptyset \subset R \subseteq [r]$, of $\mathcal{C}_{[r]}$ so that the boundaries of the resulting complexes, denoted by \mathcal{Q}_S , $\emptyset \subset S \subseteq [r]$, are simplicial polytopes. We obtain this by performing a series of stellar subdivisions. First set $\mathcal{Q}_S := \mathcal{C}_S$, for all $\emptyset \subset S \subseteq [r]$. Then, we add auxiliary vertices as follows:

$$\begin{aligned} &\text{for } s \text{ from } 1 \text{ to } r - 1 \\ &\quad \text{for all } S \subseteq [r] \text{ with } |S| = s \\ &\quad \quad \text{choose } y_S \in \text{relint}(\mathcal{Q}_S) \\ &\quad \quad \text{for all } T \text{ with } S \subset T \subseteq [r] \\ &\quad \quad \quad \mathcal{Q}_T := \text{st}(y_S, \mathcal{Q}_T) \end{aligned} \quad (3.1)$$

The recursive step of the previous definition is well defined due to the fact that for any fixed s , the order in which we add the auxiliary points y_S is independent of the S chosen, since the relative interiors of all \mathcal{Q}_S with $|S| = s$ are pairwise disjoint. At the end of the s -th iteration, the faces of each \mathcal{Q}_T of dimension less than $d + s - 1$ are simplices. At the end of the iterative procedure above, and in view of the fact that stellar subdivisions preserve polytopality, the above construction results in simplicial $(d + |R| - 1)$ -polytopes \mathcal{Q}_R , for all $\emptyset \subset R \subseteq [r]$.

The next lemma shows how the iterated stellar subdivisions performed in (3.1) are captured in the enumerative structure of \mathcal{Q}_R .

► **Lemma 3.1.** *For all $\emptyset \subset R \subseteq [r]$ we have:*

$$\mathbf{f}(\partial\mathcal{Q}_R; t) = \mathbf{f}(\mathcal{F}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{i=0}^{|R|-|S|} i! S_{|R|-|S|+1}^{i+1} t^{|R|-|S|-i} \mathbf{f}(\mathcal{F}_S; t), \tag{3.2}$$

$$\mathbf{f}(\partial\mathcal{Q}_R; t) = \mathbf{f}(\mathcal{K}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{i=0}^{|R|-|S|-2} (i+1)! S_{|R|-|S|}^{i+1} t^{|R|-|S|-i} \mathbf{f}(\mathcal{K}_S; t), \tag{3.3}$$

where $S_m^k = \frac{1}{k!} \sum_{i=0}^k (-1)^{k-i} \binom{k}{i} i^m$, $m \geq k \geq 0$, are the Stirling numbers of the second kind.

The h -vector relations stemming from the f -vector relations above are the subject of the following lemma.

► **Lemma 3.2.** *For all $\emptyset \subset R \subseteq [r]$ we have:*

$$\mathbf{h}(\partial\mathcal{Q}_R; t) = \mathbf{h}(\mathcal{F}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{j=0}^{|R|-|S|-1} E_{|R|-|S|}^j t^{j+1} \mathbf{h}(\mathcal{F}_S; t), \tag{3.4}$$

$$\mathbf{h}(\partial\mathcal{Q}_R; t) = \mathbf{h}(\mathcal{K}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{j=0}^{|R|-|S|-1} E_{|R|-|S|}^j t^j \mathbf{h}(\mathcal{K}_S; t), \tag{3.5}$$

where $E_m^k = \sum_{i=0}^k (-1)^i \binom{m+1}{i} (k+1-i)^m$, $m \geq k+1 > 0$, are the Eulerian numbers.

4 The Dehn-Sommerville equations

A very important structural property of the Cayley polytope \mathcal{C}_R is, what we call, the *Dehn-Sommerville equations*. For a single polytope they reduce to the well-known Dehn-Sommerville equations, whereas for two or more summands they relate the h -vectors of the sets \mathcal{F}_R and \mathcal{K}_R . The Dehn-Sommerville equations for \mathcal{C}_R are one of the major key ingredients for establishing our upper bounds, as they permit us to reason for the maximality of the elements of $\mathbf{h}(\mathcal{F}_R)$ and $\mathbf{h}(\mathcal{K}_R)$ by considering only the lower halves of these vectors.

► **Theorem 4.1** (Dehn-Sommerville equations). *Let \mathcal{C}_R be the Cayley polytope of the d -polytopes $P_i, i \in R$. Then, the following relations hold:*

$$t^{d+|R|-1} \mathbf{h}(\mathcal{F}_R; \frac{1}{t}) = \mathbf{h}(\mathcal{K}_R; t) \tag{4.1}$$

or, equivalently,

$$h_{d+|R|-1-k}(\mathcal{F}_R) = h_k(\mathcal{K}_R), \quad 0 \leq k \leq d + |R| - 1. \tag{4.2}$$

Proof. We prove our claim by induction on the size of R , the case $|R| = 1$ being the Dehn-Sommerville equations for a d -polytope. We next assume that our claim holds for all $\emptyset \subset S \subset R$

and prove it for R . The ordinary Dehn-Sommerville relations, written in generating function form, for the (simplicial) $(d + |R| - 1)$ -polytope \mathcal{Q}_R imply that:

$$\mathbf{h}(\partial\mathcal{Q}_R; t) = t^{d+|R|-1} \mathbf{h}(\partial\mathcal{Q}_R; \frac{1}{t}). \quad (4.3)$$

In view of relation (3.4) of Lemma 3.2, the right-hand side of (4.3) becomes:

$$t^{d+|R|-1} \mathbf{h}(\mathcal{F}_R; \frac{1}{t}) + t^{d+|R|-1} \sum_{\emptyset \subset S \subset R} \sum_{j=0}^{|R|-|S|-1} E_{|R|-|S|}^j t^{-j-1} \mathbf{h}(\mathcal{F}_S; \frac{1}{t}). \quad (4.4)$$

Using relation (3.5), along with the induction hypothesis, the left-hand side of (4.3) becomes:

$$\mathbf{h}(\mathcal{K}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{j=0}^{|R|-|S|-1} E_{|R|-|S|}^j t^j \mathbf{h}(\mathcal{K}_S; t) \quad (4.5)$$

$$= \mathbf{h}(\mathcal{K}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{j=0}^{|R|-|S|-1} E_{|R|-|S|}^j t^{|R|-|S|-j-1} \mathbf{h}(\mathcal{K}_S; t) \quad (4.6)$$

$$= \mathbf{h}(\mathcal{K}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{j=0}^{|R|-|S|-1} E_{|R|-|S|}^j t^{|R|-|S|-j-1} t^{d+|S|-1} \mathbf{h}(\mathcal{F}_S; \frac{1}{t})$$

$$= \mathbf{h}(\mathcal{K}_R; t) + \sum_{\emptyset \subset S \subset R} \sum_{j=0}^{|R|-|S|-1} E_{|R|-|S|}^j t^{d+|R|-j-2} \mathbf{h}(\mathcal{F}_S; \frac{1}{t}), \quad (4.7)$$

where to go from (4.5) to (4.6) we changed variables and used the well-known symmetry of the Eulerian numbers, namely, $E_m^k = E_m^{m-k-1}$, for all $m \geq k + 1 > 0$.

Now, substituting (4.4) and (4.7) in (4.3), we deduce that $t^{d+|R|-1} \mathbf{h}(\mathcal{F}_R; \frac{1}{t}) = \mathbf{h}(\mathcal{K}_R; t)$, which is, coefficient-wise, equivalent to (4.2). ◀

5 The recurrence relation for $h(\mathcal{F}_R)$

The subject of this section is the generalization, for the h -vector of \mathcal{F}_R , $\emptyset \subset R \subseteq [r]$, of the recurrence relation

$$(k+1)h_{k+1}(\partial P) + (d-k)h_k(\partial P) \leq n h_k(\partial P), \quad 0 \leq k \leq d-1, \quad (5.1)$$

that holds true for any simplicial d -polytope $P \subset \mathbb{R}^d$. This is the content of the next theorem.

► **Theorem 5.1** (Recurrence inequality). *For any $\emptyset \subset R \subseteq [r]$ we have:*

$$h_{k+1}(\mathcal{F}_R) \leq \frac{n_R - d - |R| + 1 + k}{k+1} h_k(\mathcal{F}_R) + \sum_{i \in R} \frac{n_i}{k+1} g_k(\mathcal{F}_{R \setminus \{i\}}), \quad 0 \leq k \leq d + |R| - 2, \quad (5.2)$$

where: (1) $n_R = \sum_{i \in R} n_i$, and, (2) $g_k(\mathcal{F}_\emptyset) = g_k(\emptyset) = 0$, for all k .

Sketch of proof. To prove the inequality in the statement of the theorem, we generalize McMullen's steps in the proof of his Upper Bound theorem [14].

Our starting point is relation (5.1) applied to the simplicial $(d + |R| - 1)$ -polytope \mathcal{Q}_R , expressed in terms of generating functions:

$$(d + |R| - 1) \mathbf{h}(\partial\mathcal{Q}_R; t) + (1-t) \mathbf{h}'(\partial\mathcal{Q}_R; t) = \sum_{v \in \text{vert}(\partial\mathcal{Q}_R)} \mathbf{h}(\partial\mathcal{Q}_R/v; t). \quad (5.3)$$

Exploiting the combinatorial structure of \mathcal{Q}_R in order to express: (1) $\mathbf{h}(\partial\mathcal{Q}_R)$ in terms of $\mathbf{h}(\mathcal{F}_S)$, $\emptyset \subset S \subseteq R$, and (2) $\mathbf{h}(\partial\mathcal{Q}_R/v)$ in terms of $\mathbf{h}(\mathcal{F}_S/v)$, $\emptyset \subset S \subseteq R$, and $\mathbf{h}(\mathcal{F}_S)$, $\emptyset \subset S \subset R$, relation (5.3) yields:

$$(d + |R| - 1)\mathbf{h}(\mathcal{F}_R; t) + (1 - t)\mathbf{h}'(\mathcal{F}_R; t) = \sum_{v \in V_R} \mathbf{h}(\mathcal{F}_R/v; t),$$

the element-wise form of which is:

$$(k + 1)h_{k+1}(\mathcal{F}_R) + (d + |R| - 1 - k)h_k(\mathcal{F}_R) = \sum_{v \in V_R} h_k(\mathcal{F}_R/v), \quad 0 \leq k \leq d + |R| - 2.$$

Noticing that $h_k(\mathcal{F}_R/v)$ is equal to $\sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \sum_{v \in V_S} g_k^{(|R|-|S|)}(\mathcal{K}_S/v)$ (by the Inclusion-Exclusion Principle; see also relations (2.12) and (2.13)), and using a particular shelling of $\partial\mathcal{Q}_R$, we show that:

$$\sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \sum_{v \in V_S} g_k^{(|R|-|S|)}(\mathcal{K}_S/v) \leq \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \sum_{v \in V_S} g_k^{(|R|-|S|)}(\mathcal{K}_S).$$

The right-hand side of the above relation simplifies to $n_R h_k(\mathcal{F}_R) + \sum_{i \in R} n_i g_k(\mathcal{F}_{R \setminus \{i\}})$, which in turn suggests the following inequality:

$$(k + 1)h_{k+1}(\mathcal{F}_R) + (d + |R| - 1 - k)h_k(\mathcal{F}_R) \leq n_R h_k(\mathcal{F}_R) + \sum_{i \in R} n_i g_k(\mathcal{F}_{R \setminus \{i\}}) \tag{5.4}$$

that holds true for all $0 \leq k \leq d + |R| - 2$. Solving in terms of $h_{k+1}(\mathcal{F}_R)$ results in (5.2). ◀

6 Upper bounds

Let S_1, \dots, S_r be a partition of a set S into r sets. We say that $A \subseteq \bigcup_{1 \leq i \leq r} S_i$ is a *spanning subset* of S if $A \cap S_i \neq \emptyset$ for all $1 \leq i \leq r$.

► **Definition 6.1.** Let $P_i, i \in R$, be d -polytopes with vertex sets $V_i, i \in R$. We say that their Cayley polytope \mathcal{C}_R is *R -neighborly* if every spanning subset of $\bigcup_{i \in R} V_i$ of size $|R| \leq \ell \leq \lfloor \frac{d+|R|-1}{2} \rfloor$ is a face of \mathcal{C}_R (or, equivalently, a face of \mathcal{F}_R). We say that the Cayley polytope \mathcal{C}_R is *Minkowski-neighborly* if, for every $\emptyset \subset S \subseteq R$, the Cayley polytope \mathcal{C}_S is S -neighborly.

The following lemma characterizes R -neighborly Cayley polytopes in terms of the f - and h -vector of \mathcal{F}_R .

► **Lemma 6.2.** *The following are equivalent:*

- (i) \mathcal{C}_R is R -neighborly,
 - (ii) $f_{\ell-1}(\mathcal{F}_R) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \binom{n_S}{\ell}$, for all $0 \leq \ell \leq \lfloor \frac{d+|R|-1}{2} \rfloor$,
 - (iii) $h_\ell(\mathcal{F}_R) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \binom{n_S - d - |R| + \ell}{\ell}$, for all $0 \leq \ell \leq \lfloor \frac{d+|R|-1}{2} \rfloor$,
- where n_i is the number of vertices of P_i and $n_S = \sum_{i \in S} n_i$.

From the recurrence relation in Theorem 5.1 we arrive at the following theorem. The proof is by induction on k .

► **Theorem 6.3.** *For any $\emptyset \subset R \subseteq [r]$ and $0 \leq k \leq d + |R| - 1$, we have:*

$$g_k(\mathcal{F}_R) \leq \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \binom{n_S - d - |R| - 1 + k}{k}, \quad \text{and} \tag{6.1}$$

$$h_k(\mathcal{F}_R) \leq \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \binom{n_S - d - |R| + k}{k}, \tag{6.2}$$

where $n_S = \sum_{i \in S} n_i$. Equalities hold for all $0 \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$ if and only if the Cayley polytope \mathcal{C}_R is R -neighborly.

Before proceeding with proving upper bounds for the h -vectors of \mathcal{F}_R and \mathcal{K}_R we need to define the following functions.

► **Definition 6.4.** Let $d \geq 2$, $\emptyset \subset R \subseteq [r]$, $m \geq 0$, $0 \leq k \leq d + |R| - 1$, and $n_i \in \mathbb{N}$, $i \in R$, with $n_i \geq d + 1$. We define the functions $\Phi_{k,d}^{(m)}(\mathbf{n}_R)$ and $\Psi_{k,d}(\mathbf{n}_R)$ via the following conditions:

1. $\Phi_{k,d}^{(0)}(\mathbf{n}_R) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \binom{n_S - d - |R| + k}{k}$, $0 \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$,
2. $\Phi_{k,d}^{(m)}(\mathbf{n}_R) = \Phi_{k,d}^{(m-1)}(\mathbf{n}_R) - \Phi_{k-1,d}^{(m-1)}(\mathbf{n}_R)$, $m > 0$,
3. $\Psi_{k,d}(\mathbf{n}_R) = \sum_{\emptyset \subset S \subseteq R} \Phi_{k,d}^{(|R|-|S|)}(\mathbf{n}_S)$,
4. $\Phi_{k,d}^{(0)}(\mathbf{n}_R) = \Psi_{d+|R|-1-k,d}(\mathbf{n}_R)$,

where \mathbf{n}_R stands for the $|R|$ -dimensional vector whose elements are the values n_i , $i \in R$.

Notice that $\Phi_{k,d}^{(0)}(\mathbf{n}_R)$ and $\Psi_{k,d}(\mathbf{n}_R)$ are well defined, though in a recursive manner (in the size of R), since for any $k > \lfloor \frac{d+|R|-1}{2} \rfloor$, we have:

$$\begin{aligned} \Phi_{k,d}^{(0)}(\mathbf{n}_R) &= \Psi_{d+|R|-1-k,d}(\mathbf{n}_R) = \sum_{\emptyset \subset S \subseteq R} \Phi_{d+|R|-1-k,d}^{(|R|-|S|)}(\mathbf{n}_S) \\ &= \Phi_{d+|R|-1-k,d}^{(0)}(\mathbf{n}_R) + \sum_{\emptyset \subset S \subseteq R} \Phi_{d+|R|-1-k,d}^{(|R|-|S|)}(\mathbf{n}_S) \\ &= \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \binom{n_S - k - 1}{d+|R|-1-k} + \sum_{\emptyset \subset S \subseteq R} \Phi_{d+|R|-1-k,d}^{(|R|-|S|)}(\mathbf{n}_S), \end{aligned} \quad (6.3)$$

where the second sum in (6.3) is to be understood as 0 when $|R| = 1$. In other words, $\Phi_{k,d}^{(0)}(\mathbf{n}_R)$, and, thus, also $\Phi_{k,d}^{(m)}(\mathbf{n}_R)$ for any $m > 0$, is fully defined for some R and any k , once we know the values $\Phi_{k,d}^{(\ell)}(\mathbf{n}_S)$ for all $\emptyset \subset S \subseteq R$, for all $0 \leq k \leq d + |S| - 1$, and for all $1 \leq \ell \leq |R| - 1$. Moreover, it is easy to verify that $\Phi_{k,d}^{(0)}(\mathbf{n}_R)$ satisfies the following recurrence relation:

$$\Phi_{k+1,d}^{(0)}(\mathbf{n}_R) = \frac{n_R - d - |R| + k + 1}{k + 1} \Phi_{k,d}^{(0)}(\mathbf{n}_R) + \sum_{i \in R} \frac{n_i}{k + 1} \Phi_{k,d}^{(1)}(\mathbf{n}_{R \setminus \{i\}}), \quad 0 \leq k < \lfloor \frac{d+|R|-1}{2} \rfloor. \quad (6.4)$$

The next theorem provides upper bounds for the h -vectors of \mathcal{F}_R and \mathcal{K}_R , as well as necessary and sufficient conditions for these upper bounds to be attained.

► **Theorem 6.5.** For all $0 \leq k \leq d + |R| - 1$, we have:

- (i) $h_k(\mathcal{F}_R) \leq \Phi_{k,d}^{(0)}(\mathbf{n}_R)$,
- (ii) $h_k(\mathcal{K}_R) \leq \Psi_{k,d}(\mathbf{n}_R)$.

Equalities hold for all k if and only if the Cayley polytope \mathcal{C}_R is Minkowski-neighborly.

Proof. To prove the upper bounds we use recursion on the size of $|R|$. For $|R| = 1$, the result for both $h_k(\mathcal{F}_R)$ and $h_k(\mathcal{K}_R)$ comes from the UBT for d -polytopes. For $|R| > 1$, we assume that the bounds hold for all S with $\emptyset \subset S \subseteq R$, and for all k with $0 \leq k \leq d + |S| - 1$. Furthermore, the upper bound for $h_k(\mathcal{F}_R)$ for $k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$ is immediate from Theorem 6.3. To prove the upper bound for $h_k(\mathcal{K}_R)$, $0 \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$, we use the following expansion for $h_k(\mathcal{K}_R)$ (cf. [1, Lemma 5.14]):

$$\begin{aligned} h_k(\mathcal{K}_R) &= \sum_{j=0}^{\lfloor \frac{|R|}{2} \rfloor} \sum_{s=c-2j-1}^{|R|-2j} \sum_{\substack{S \subseteq R \\ |S|=s}} \binom{|R|-s}{2j} \left(h_{k-2j}(\mathcal{F}_S) - \frac{1}{2j+1} \sum_{i \in S} h_{k-2j-1}(\mathcal{F}_{S \setminus \{i\}}) \right) \\ &\quad + \sum_{j=0}^{\lfloor \frac{|R|}{2} \rfloor} \sum_{\substack{S \subseteq R \\ |S|=c-2j+1}} \binom{|R|-|S|}{2j} \left(h_{k-2j}(\mathcal{F}_S) - \frac{1}{2j+1} \sum_{i \in S} h_{k-2j-1}(\mathcal{F}_{S \setminus \{i\}}) \right), \end{aligned} \quad (6.5)$$

where c depends on k, d and $|R|$. Under the assumption that $r < d$, it is easy to show that:

$$h_{k-2j}(\mathcal{F}_S) - \frac{1}{2j+1} \sum_{i \in S} h_{k-2j-1}(\mathcal{F}_{S \setminus \{i\}}) \leq \Phi_{k-2j,d}^{(0)}(\mathbf{n}_S) - \frac{1}{2j+1} \sum_{i \in S} \Phi_{k-2j-1,d}^{(0)}(\mathbf{n}_{S \setminus \{i\}}). \quad (6.6)$$

Substituting the upper bound from (6.6) in (6.5), and reversing the derivation logic for (6.5), we deduce that $h_k(\mathcal{K}_R) \leq \Psi_{k,d}(\mathbf{n}_R)$.

For $k > \lfloor \frac{d+|R|-1}{2} \rfloor$ we have:

$$h_k(\mathcal{F}_R) = h_{d+|R|-1-k}(\mathcal{K}_R) \leq \Psi_{d+|R|-1-k,d}(\mathbf{n}_R) = \Phi_{k,d}^{(0)}(\mathbf{n}_R), \quad \text{and,}$$

$$h_k(\mathcal{K}_R) = h_{d+|R|-1-k}(\mathcal{F}_R) \leq \Phi_{d+|R|-1-k,d}^{(0)}(\mathbf{n}_R) = \Psi_{k,d}(\mathbf{n}_R).$$

The necessary and sufficient conditions are easy consequences of the equality claim in Theorem 6.3. ◀

For any $d \geq 2, \emptyset \subset R \subseteq [r], 0 \leq k \leq d + |R| - 1$, and $n_i \in \mathbb{N}, i \in R$, with $n_i \geq d + 1$, let

$$\Xi_{k,d}(\mathbf{n}_R) = \sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} f_k(C_{d+|R|-1}(n_S)) + \sum_{i=0}^{\lfloor \frac{d+|R|-2}{2} \rfloor} \binom{i}{k-d-|R|+1+i} \sum_{\emptyset \subset S \subseteq R} \Phi_{i,d}^{(|R|-|S|)}(\mathbf{n}_S),$$

where $C_\delta(n)$ stands for the cyclic δ -polytope with n vertices. It is straightforward to verify that for $0 \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$, $\Xi_{k,d}(\mathbf{n}_R)$ simplifies to $\sum_{\emptyset \subset S \subseteq R} (-1)^{|R|-|S|} \binom{n_S}{k}$. We are finally ready to state and prove the main result of the paper.

► **Theorem 6.6.** *Let P_1, \dots, P_r be r d -polytopes, $r < d$, with n_1, \dots, n_r vertices respectively. Then, for all $1 \leq k \leq d$, we have:*

$$f_{k-1}(P_{[r]}) \leq \Xi_{k+r,d}(\mathbf{n}_{[r]}).$$

Equality holds for all $0 \leq k \leq d$ if and only if the Cayley polytope $\mathcal{C}_{[r]}$ of P_1, \dots, P_r is Minkowski-neighborly.

Proof. We start by recalling that:

$$f_{k-1}(\mathcal{F}_{[r]}) = \sum_{i=0}^{d+r-1} \binom{d+r-1-i}{k-i} h_i(\mathcal{F}_{[r]}).$$

In view of Theorem 6.5, the above expression is bounded from above by:

$$\sum_{i=0}^{\lfloor \frac{d+r-1}{2} \rfloor} \binom{d+r-1-i}{k-i} \Phi_{i,d}^{(0)}(\mathbf{n}_{[r]}) + \sum_{i=\lfloor \frac{d+r-1}{2} \rfloor+1}^{d+r-1} \binom{d+r-1-i}{k-i} \Phi_{i,d}^{(0)}(\mathbf{n}_{[r]}) \quad (6.7)$$

$$= \sum_{i=0}^{\lfloor \frac{d+r-1}{2} \rfloor} \binom{d+r-1-i}{k-i} \Phi_{i,d}^{(0)}(\mathbf{n}_{[r]}) + \sum_{i=0}^{\lfloor \frac{d+r-2}{2} \rfloor} \binom{i}{k-d-r+1+i} \sum_{\emptyset \subset R \subseteq [r]} \Phi_{i,d}^{(r-|R|)}(\mathbf{n}_R) \quad (6.8)$$

$$= \sum_{i=0}^{\frac{d+r-1}{2}} \left(\binom{d+r-1-i}{k-i} + \binom{i}{k-d-r+1+i} \right) \sum_{\emptyset \subset R \subseteq [r]} (-1)^{r-|R|} \binom{n_R-d-r+i}{i} + \sum_{i=0}^{\lfloor \frac{d+r-2}{2} \rfloor} \binom{i}{k-d-r+1+i} \sum_{\emptyset \subset R \subseteq [r]} \Phi_{i,d}^{(r-|R|)}(\mathbf{n}_R) \quad (6.9)$$

$$= \sum_{\emptyset \subset R \subseteq [r]} (-1)^{r-|R|} f_k(C_{d+r-1}(n_R)) + \sum_{i=0}^{\lfloor \frac{d+r-2}{2} \rfloor} \binom{j}{k-d-r+1+i} \sum_{\emptyset \subset R \subseteq [r]} \Phi_{i,d}^{(r-|R|)}(\mathbf{n}_R), \quad (6.10)$$

where to go:

- from (6.7) to (6.8) we changed the variable of the second sum from i to $d+r-1-i$, and used conditions 3 and 4 of Definition 6.4,
- from (6.8) to (6.9) we wrote the explicit expression of $\Phi_{i,d}^{(0)}(\mathbf{n}_{[r]})$ from relation (6.3),
- from (6.9) to (6.10) we used that the number of $(k-1)$ -faces of a cyclic δ -polytope with n vertices is $\sum_{i=0}^{\frac{\delta}{2}} \binom{\delta-i}{k-i} + \binom{i}{k-\delta+i} \binom{n-\delta-1+i}{i}$, where $\sum_{i=0}^{\frac{\delta}{2}} T_i$ denotes the sum of the elements $T_0, T_1, \dots, T_{\lfloor \frac{\delta}{2} \rfloor}$ where the last term is halved if δ is even.

Finally, observing that the expression in (6.10) is nothing but $\Xi_{k,d}(\mathbf{n}_{[r]})$, and recalling that $f_{k-1}(\mathcal{F}_{[r]}) = f_{k-r}(P_{[r]})$, we arrive at the upper bound in the statement of the theorem. The equality claim is immediate from Theorem 6.5. \blacktriangleleft

7 Tight bound construction

In this section we show that the bounds in Theorem 6.6 are tight. Before getting into the technical details, we outline our approach. We start by considering the $(d-r+1)$ -dimensional moment curve, which we embed in r distinct subspaces of \mathbb{R}^d . We consider the r copies of the $(d-r+1)$ -dimensional moment curve as different curves, and we perturb them appropriately, so that they become d -dimensional moment-like curves. The perturbation is controlled via a non-negative parameter ζ , which will be chosen appropriately. We then choose points on these r moment-like curves, all parameterized by a positive parameter τ , which will again be chosen appropriately. These points are the vertices of r d -polytopes P_1, P_2, \dots, P_r , and we show that, for all $\emptyset \subset R \subseteq [r]$, the number of $(k-1)$ -faces of \mathcal{F}_R , where $|R| \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$, becomes equal to $\Xi_{k,d}(\mathbf{n}_R)$ for small enough positive values of ζ and τ . Our construction produces *projected prodsimplicial-neighborly* polytopes (cf. [13]). For $\zeta = 0$ our polytopes are essentially the same as those in [13, Theorem 2.6], while for $\zeta > 0$ we get *deformed* versions of those polytopes. The positivity of ζ allows us to ensure the tightness of the upper bound on $f_k(P_{[r]})$, not only for small, but also for large values of k .

At a more technical level, the proof that $f_{k-1}(\mathcal{F}_R) = \Xi_{k,d}(\mathbf{n}_R)$, for all $|R| \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$, is performed in two steps. We first consider the cyclic $(d-r+1)$ -polytopes $\hat{P}_1, \dots, \hat{P}_r$, embedded in appropriate subspaces of \mathbb{R}^d . The \hat{P}_i 's are the *unperturbed*, with respect to ζ , versions of the d -polytopes P_1, P_2, \dots, P_r (i.e., the polytope \hat{P}_i is the polytope we get from P_i , when we set ζ equal to zero). For each $\emptyset \subset R \subseteq [r]$ we denote by $\hat{\mathcal{C}}_R$ the Cayley polytope of $\hat{P}_i, i \in R$, seen as a polytope in \mathbb{R}^d , and we focus on the set $\hat{\mathcal{F}}_R$ of its mixed faces. Recall that the polytopes $\hat{P}_i, i \in R$, are parameterized by the parameter τ ; we show that there exists a sufficiently small positive value τ^* for τ , for which the number of $(k-1)$ -faces of $\hat{\mathcal{F}}_R$ is equal to $\Xi_{k,d}(\mathbf{n}_R)$ for all $|R| \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor$. For τ equal to τ^* , we consider the polytopes P_1, P_2, \dots, P_r (with τ set to τ^*), and show that for sufficiently small ζ (denoted by ζ^\diamond), $f_{k-1}(\mathcal{F}_R)$ is equal to $\Xi_{k,d}(\mathbf{n}_R)$.

In the remainder of this section we describe our construction in detail. For each $1 \leq i \leq r$, we define the d -dimensional moment-like curve²:

$$\gamma_i(t; \zeta) = (\zeta t^{d-r+2}, \dots, \zeta t^{d-r+i}, \overset{i\text{-th coordinate}}{\downarrow} t, \zeta t^{d-r+i+2}, \dots, \zeta t^{d+1}, t^2, \dots, t^{d-r+1}),$$

and the d -polytope

² The curve $\gamma_i(t; \zeta)$, $\zeta > 0$, is the image under an invertible linear transformation, of the curve $\hat{\gamma}_i(t) = (t, t^2, \dots, t^{d-r+i}, t^{d-r+i+2}, \dots, t^{d+1})$. Polytopes whose vertices are n distinct points on this curve are combinatorially equivalent to the cyclic d -polytope with n vertices.

$$P_i := \text{conv}(\{\gamma_i(y_{i,1}; \zeta), \dots, \gamma_i(y_{i,n_i}; \zeta)\}), \tag{7.1}$$

where the parameters $y_{i,j}$ belong to the sets $Y_i = \{y_{i,1}, \dots, y_{i,n_i}\}$, $1 \leq i \leq r$, whose elements are determined as follows. Choose

- $n_{[r]} + d + r$ arbitrary real numbers $x_{i,j}$ and M_s , such that:
 - $0 < x_{i,1} < x_{i,1} + \epsilon < x_{i,2} < x_{i,2} + \epsilon < \dots < x_{i,n_i} + \epsilon$, for $1 \leq i \leq r - 1$,
 - $0 < x_{r,1} < x_{r,1} + \epsilon < x_{r,2} < x_{r,2} + \epsilon < \dots < x_{r,n_r} + \epsilon < M'_1 < \dots < M'_{d+r}$,
 where $\epsilon > 0$ is sufficiently small and $x_{i,n_i} < x_{i+1,1}$ for all i , and
- r non-negative integers $\beta_1, \beta_2, \dots, \beta_r$, such that $\beta_1 > \beta_2 > \dots > \beta_{r-1} > \beta_r \geq 0$.

We then set $y_{i,j} := x_{i,j}\tau^{\beta_i}$, $\tilde{y}_{i,j} := (x_{i,j} + \epsilon)\tau^{\beta_i}$ and $M_i := M'_i\tau^{\beta_r}$, where τ is a positive parameter. The $y_{i,j}$'s, $\tilde{y}_{i,j}$'s and M_i 's are used to define determinants whose value is positive for a small enough value of τ . The positivity of these determinants is crucial in defining supporting hyperplanes for the Cayley polytopes \hat{C}_R and C_R in Lemmas 7.1 and 7.2 below.

Next, for each $1 \leq i \leq r$, we define $\hat{P}_i := \lim_{\zeta \rightarrow 0^+} P_i$. Clearly, each \hat{P}_i is a cyclic $(d - r + 1)$ -polytope embedded in the $(d - r + 1)$ -flat F_i of \mathbb{R}^d , where $F_i = \{x_j = 0 \mid 1 \leq j \leq r \text{ and } j \neq i\}$. The following lemma establishes the first step towards our construction.

► **Lemma 7.1.** *There exists a sufficiently small positive value τ^* for τ , such that, for any $\emptyset \subset R \subseteq [r]$, the set of mixed faces $\hat{\mathcal{F}}_R$ of the Cayley polytope of the polytopes $\hat{P}_1, \dots, \hat{P}_r$ constructed above, has*

$$f_{k-1}(\hat{\mathcal{F}}_R) = \Xi_{k,d}(\mathbf{n}_R), \quad |R| \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor.$$

Proof. Let \mathcal{U}_i be the set of vertices of \hat{P}_i for $1 \leq i \leq r$ and set $\mathcal{U} := \cup_{i \in R} \mathcal{U}_i$. The objective in the proof is, for each $\emptyset \subset R \subseteq [r]$ and each spanning subset U of the partition $U = \cup_{i \in R} \mathcal{U}_i$, to exhibit a supporting hyperplane of the $(d + |R| - 1)$ -dimensional Cayley polytope \hat{C}_R , containing exactly the vertices in U . In that respect, our approach is similar in spirit to the proof showing, by defining supporting hyperplanes constructed from Vandermonde determinants, that the cyclic n -vertex d -polytope $C_d(n)$ is neighborly (see, e.g., [17, Corollary 0.8]).

In our proof we need to involve the parameter ζ before taking the limit $\zeta \rightarrow 0^+$. This is due to the fact that, when $\emptyset \subset R \subseteq [r]$, the information of the relative position of the polytopes \hat{P}_i , $i \in R$, is lost if we set $\zeta = 0$ from the very first step. To describe our construction, we write each spanning subset U of U as the disjoint union of non-empty sets U_i , $i \in R$, where $U_i = U \cap \mathcal{U}_i$ and $|U_i| = \kappa_i \leq n_i$. For this particular U , we define the linear equation:

$$H_U(\mathbf{x}) = \lim_{\zeta \rightarrow 0^+} (-1)^{\frac{|R|(|R|-1)}{2} + \sigma(R)} \zeta^{|R|-r} D_U(\mathbf{x}; \zeta), \tag{7.2}$$

where $\mathbf{x} = (x_1, x_2, \dots, x_{d+|R|-1})$, and $D_U(\mathbf{x}; \zeta)$ is the $(d + |R|) \times (d + |R|)$ determinant:

- whose first column is $(1, \mathbf{x})^\top$,
- the next κ_i , $i \in R$, pairs of columns are $(1, \mathbf{e}_{i-1}, \gamma_i(y_{i,j}; \zeta))^\top$ and $(1, \mathbf{e}_{i-1}, \gamma_i(\tilde{y}_{i,j}; \zeta))^\top$ where $\mathbf{e}_0, \dots, \mathbf{e}_{|R|-1}$ is the standard affine basis of $\mathbb{R}^{|R|-1}$, $y_{i,j} \in \{y \in Y_i \mid \gamma_i(y; 0) \in U_i\}$, and
- the last $s := d + |R| - 1 - \sum_{i \in R} \kappa_i$ columns are $(1, \mathbf{e}_{|R|-1}, \gamma_{|R|-1}(M_i; \zeta))^\top$, $1 \leq i \leq s$; these columns exist only if $s > 0$.

The quantity $\sigma(R)$ above is a non-negative integer counting the total number of row swaps required to shift, for all $j \in [r] \setminus R$, the $(|R| + j)$ -th row of $D_U(\mathbf{x}; \zeta)$ to the bottom of the determinant, so that the powers of $y_{i,j}$ in each column are in increasing order (notice that if $R \equiv [r]$ no such row swaps are required). Moreover, $\sigma(R)$ depends only on R and not on the choice of the spanning subset U of U .

The equation $H_U(\mathbf{x}) = 0$ is the equation of a hyperplane in $\mathbb{R}^{d+|R|-1}$ that passes through the points in U . We claim that, for any choice of U , and for all vertices \mathbf{u} in $\mathcal{U} \setminus U$, we have

$H_U(\mathbf{u}) > 0$. To prove our claim, notice first that, for each $j \in [r] \setminus R$, the $(|R| + j)$ -th row of the determinant $D_U(\mathbf{u}; \zeta)$ will contain the parameters $y_{i,j}^{d-r+1+j}$, $\tilde{y}_{i,j}^{d-r+1+j}$ and $M_i^{d-r+1+j}$, multiplied by ζ . After extracting ζ from each of these rows and shifting them to their *proper* position (i.e., the position where the powers along each column increase), we will have a term $\zeta^{r-|R|}$ and a sign $(-1)^{\sigma(R)}$ (induced from the $\sigma(R)$ row swaps required altogether). These terms cancel out with the term $(-1)^{\sigma(R)} \zeta^{|R|-r}$ in (7.2). We can, therefore, transform $H_U(\mathbf{u})$ in the form of the determinant $D_N(\mathbf{Z}; \alpha_1, \dots, \alpha_m)$, $\mathbf{Z} = \{z_{i,j} \mid 1 \leq i \leq \rho, 1 \leq j \leq \nu_i\}$, $N = (\nu_1, \nu_2, \dots, \nu_m)$, $0 \leq \alpha_1 < \alpha_2 < \dots < \alpha_m$, shown below:

$$D_N(\mathbf{Z}; \alpha_1, \dots, \alpha_m) := (-1)^{\frac{\rho(\rho-1)}{2}} \begin{vmatrix} z_{1,1}^{\alpha_1} & \cdots & z_{1,\nu_1}^{\alpha_1} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & z_{2,1}^{\alpha_1} & \cdots & z_{2,\nu_2}^{\alpha_1} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & z_{\rho,1}^{\alpha_1} & \cdots & z_{\rho,\nu_\rho}^{\alpha_1} \\ z_{1,1}^{\alpha_2} & \cdots & z_{1,\nu_1}^{\alpha_2} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & z_{2,1}^{\alpha_2} & \cdots & z_{2,\nu_2}^{\alpha_2} & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & z_{\rho,1}^{\alpha_2} & \cdots & z_{\rho,\nu_\rho}^{\alpha_2} \\ z_{1,1}^{\alpha_3} & \cdots & z_{1,\nu_1}^{\alpha_3} & z_{2,1}^{\alpha_3} & \cdots & z_{2,\nu_2}^{\alpha_3} & \cdots & z_{n,1}^{\alpha_3} & \cdots & z_{n,\nu_n}^{\alpha_3} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \cdots & \vdots & \ddots & \vdots \\ z_{1,1}^{\alpha_m} & \cdots & z_{1,\nu_1}^{\alpha_m} & z_{2,1}^{\alpha_m} & \cdots & z_{2,\nu_2}^{\alpha_m} & \cdots & z_{\rho,1}^{\alpha_m} & \cdots & z_{\rho,\nu_\rho}^{\alpha_m} \end{vmatrix},$$

by means of the following determinant transformations:

1. By subtracting rows 2 to $|R|$ of $H_U(\mathbf{u})$ from its first row.
2. By shifting the first column of $H_U(\mathbf{u})$ to the right, so that all columns of $H_U(\mathbf{u})$ are arranged in increasing order with respect to their parameters $z_{i,j}$. Clearly, this can be done with an *even* number of column swaps.

The determinant $D_N(\mathbf{Z}; \alpha_1, \dots, \alpha_m)$ is strictly positive for all τ between 0 and some value $\hat{\tau}(R, U, \mathbf{u})$, that, depends (only) on the choice of R , U and \mathbf{u} . Since there is a finite number of possible such determinants, the value $\hat{\tau}^* := \min_{R,U,\mathbf{u}} \hat{\tau}(R, U, \mathbf{u})$ is necessarily positive. Choosing some $\tau^* \in (0, \hat{\tau}^*)$ makes all these determinants simultaneously positive; this completes our proof. \blacktriangleleft

The following lemma establishes the second (and last) step of our construction.

► **Lemma 7.2.** *There exists a sufficiently small positive value ζ^\diamond for ζ , such that, for any $\emptyset \subset R \subseteq [r]$, the set \mathcal{F}_R of mixed faces of the Cayley polytope \mathcal{C}_R of the polytopes P_1, \dots, P_r in (7.1) has*

$$f_{k-1}(\mathcal{F}_R) = \Xi_{k,d}(\mathbf{n}_R), \quad \text{for all } |R| \leq k \leq \lfloor \frac{d+|R|-1}{2} \rfloor.$$

Proof. Briefly speaking, the value ζ^\diamond is determined by replacing the limit $\zeta \rightarrow 0^+$ in the previous proof, by a specific value of ζ for which the determinants we consider are positive.

More precisely, let \mathcal{U}_i be the set of vertices of P_i , $1 \leq i \leq r$, and set $\mathcal{U} := \cup_{i=1}^r \mathcal{U}_i$. Our goal is, for each $\emptyset \subset R \subseteq [r]$ and each spanning subset U of the partition $\mathbf{U} = \cup_{i \in R} \mathcal{U}_i$, to exhibit a supporting hyperplane of the Cayley polytope \mathcal{C}_R , containing exactly the vertices in U . To this end, we define the hyperplane $\tilde{H}_U(\mathbf{x}; \zeta) = 0$, $\mathbf{x} = (x_1, x_2, \dots, x_{d+|R|-1})$, with

$$\tilde{H}_U(\mathbf{x}; \zeta) = (-1)^{\frac{|R|(|R|-1)}{2} + \sigma(R)} \zeta^{|R|-r} D_U(\mathbf{x}; \zeta), \quad \zeta > 0, \quad (7.3)$$

where $D_U(\mathbf{x}; \zeta)$ is the determinant in the proof of Lemma 7.1, where we have set τ to τ^* . Clearly, for each $\mathbf{u} \in \mathcal{U} \setminus U$, we have $\lim_{\zeta \rightarrow 0^+} \tilde{H}_U(\mathbf{u}; \zeta) = H_U(\mathbf{u}) > 0$. This immediately implies that for each combination of R , U and \mathbf{u} there exists a value $\hat{\zeta}(R, U, \mathbf{u})$ such that,

for all $\zeta \in (0, \hat{\zeta}(R, U, \mathbf{u}))$, $\tilde{H}_U(\mathbf{u}; \zeta) > 0$. Since the number of possible combinations for R , U and \mathbf{u} is finite, the minimum $\hat{\zeta}^\diamond := \min_{R, U, \mathbf{u}} \{\hat{\zeta}(R, U, \mathbf{u})\}$ is well defined and positive. Taking ζ^\diamond to be any value in $(0, \hat{\zeta}^\diamond)$, satisfies our demands. ◀

Acknowledgments. The authors would like to thank Christos Konaxis for useful discussions and comments on earlier versions of this paper, as well as Vincent Pilaud for discussions related to the tightness construction presented in the paper.

References

- 1 Karim A. Adiprasito and Raman Sanyal. Relative Stanley-Reisner theory and Upper Bound Theorems for Minkowski sums, 2014. [arXiv:1405.7368v3 \[math.CO\]](#).
- 2 G. Ewald and G. C. Shephard. Stellar Subdivisions of Boundary Complexes of Convex Polytopes. *Mathematische Annalen*, 210:7–16, 1974.
- 3 Günter Ewald. *Combinatorial Convexity and Algebraic Geometry*. Graduate Texts in Mathematics. Springer, 1996.
- 4 Efi Fogel, Dan Halperin, and Christophe Weibel. On the Exact Maximum Complexity of Minkowski Sums of Polytopes. *Discrete Comput. Geom.*, 42:654–669, 2009.
- 5 Komei Fukuda and Christophe Weibel. f -vectors of Minkowski Additions of Convex Polytopes. *Discrete Comput. Geom.*, 37(4):503–516, 2007.
- 6 R. L. Graham, M. Grötschel, and L. Lovász. *Handbook of Combinatorics*, volume 2. MIT Press, North Holland, 1995.
- 7 Peter Gritzmann and Bernd Sturmfels. Minkowski Addition of Polytopes: Computational Complexity and Applications to Gröbner bases. *SIAM J. Disc. Math.*, 6(2):246–269, 1993.
- 8 Birkett Huber, Jörg Rambau, and Francisco Santos. The Cayley Trick, lifting subdivisions and the Bohne-Dress theorem on zonotopal tilings. *J. Eur. Math. Soc.*, 2(2):179–198, 2000.
- 9 Menelaos I. Karavelas, Christos Konaxis, and Eleni Tzanaki. The maximum number of faces of the Minkowski sum of three convex polytopes. *J. Comput. Geom.*, 6(1):21–74, 2015.
- 10 Menelaos I. Karavelas and Eleni Tzanaki. The maximum number of faces of the Minkowski sum of two convex polytopes. In *Proceedings of the 23rd ACM-SIAM Symposium on Discrete Algorithms (SODA '12)*, pages 11–28, 2012.
- 11 Menelaos I. Karavelas and Eleni Tzanaki. A geometric approach for the upper bound theorem for Minkowski sums of convex polytopes, 2015. [arXiv:1502.02265v2 \[cs.CG\]](#).
- 12 Jiří Matoušek. *Lectures on Discrete Geometry*. Graduate Texts in Mathematics. Springer-Verlag New York, Inc., New York, 2002.
- 13 B. Matschke, J. Pfeifle, and V. Pilaud. Prodsimplicial-neighborly polytopes. *Discrete Comput. Geom.*, 46(1):100–131, 2011.
- 14 P. McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17:179–184, 1970.
- 15 Raman Sanyal. Topological obstructions for vertex numbers of Minkowski sums. *J. Comb. Theory, Ser. A*, 116(1):168–179, 2009.
- 16 Christophe Weibel. Maximal f -vectors of Minkowski Sums of Large Numbers of Polytopes. *Discrete Comput. Geom.*, 47(3):519–537, 2012.
- 17 Günter M. Ziegler. *Lectures on Polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995.

Two Proofs for Shallow Packings

Kunal Dutta¹, Esther Ezra², and Arijit Ghosh¹

1 D1: Algorithms & Complexity

Max-Planck-Institut für Informatik, 66123 Saarbrücken, Germany
{kdutta,agosh}@mpi-inf.mpg.de

2 Department of Computer Science and Engineering

Polytechnic Institute of NYU, Brooklyn, NY 11201-3840, USA; and

School of Mathematics

Georgia Institute of Technology, Atlanta, Georgia 30332, USA

esther@courant.nyu.edu

Abstract

We refine the bound on the packing number, originally shown by Haussler, for shallow geometric set systems. Specifically, let \mathcal{V} be a finite set system defined over an n -point set X ; we view \mathcal{V} as a set of indicator vectors over the n -dimensional unit cube. A δ -separated set of \mathcal{V} is a subcollection \mathcal{W} , s.t. the Hamming distance between each pair $\mathbf{u}, \mathbf{v} \in \mathcal{W}$ is greater than δ , where $\delta > 0$ is an integer parameter. The δ -packing number is then defined as the cardinality of the largest δ -separated subcollection of \mathcal{V} . Haussler showed an asymptotically tight bound of $\Theta((n/\delta)^d)$ on the δ -packing number if \mathcal{V} has VC-dimension (or *primal shatter dimension*) d . We refine this bound for the scenario where, for any subset, $X' \subseteq X$ of size $m \leq n$ and for any parameter $1 \leq k \leq m$, the number of vectors of length at most k in the restriction of \mathcal{V} to X' is only $O(m^{d_1} k^{d-d_1})$, for a fixed integer $d > 0$ and a real parameter $1 \leq d_1 \leq d$ (this generalizes the standard notion of *bounded primal shatter dimension* when $d_1 = d$). In this case when \mathcal{V} is “ k -shallow” (all vector lengths are at most k), we show that its δ -packing number is $O(n^{d_1} k^{d-d_1} / \delta^d)$, matching Haussler’s bound for the special cases where $d_1 = d$ or $k = n$. We present two proofs, the first is an extension of Haussler’s approach, and the second extends the proof of Chazelle, originally presented as a simplification for Haussler’s proof.

1998 ACM Subject Classification F.2.2 [Nonnumerical Algorithms and Problems] Computations on discrete structures, Geometrical problems and computations, F.1.2 [Modes of Computation] Probabilistic computation

Keywords and phrases Set systems of bounded primal shatter dimension, δ -packing and Haussler’s approach, relative approximations, Clarkson-Shor random sampling approach


Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.96

1 Introduction

Let \mathcal{V} be a set system defined over an n -point set X . We follow the notation in [19], and view \mathcal{V} as a set of indicator vectors in \mathbb{R}^n , that is, $\mathcal{V} \subseteq \{0, 1\}^n$. Given a subsequence of indices (coordinates) $I = (i_1, \dots, i_k)$, $1 \leq i_j \leq n$, $k \leq n$, the *projection* $\mathcal{V}|_I$ of \mathcal{V} onto I (also referred to as the *restriction* of \mathcal{V} to I) is defined as

$$\mathcal{V}|_I = \{(\mathbf{v}_{i_1}, \dots, \mathbf{v}_{i_k}) \mid \mathbf{v} = (\mathbf{v}_1, \dots, \mathbf{v}_n) \in \mathcal{V}\}.$$

With a slight abuse of notation we write $I \subseteq [n]$ to state the fact that I is a subsequence of indices as above. We now recall the definition of the primal shatter function of \mathcal{V} :

 © Kunal Dutta, Esther Ezra, and Arijit Ghosh;
licensed under Creative Commons License CC-BY
31st International Symposium on Computational Geometry (SoCG’15).
Editors: Lars Arge and János Pach; pp. 96–110



Leibniz International Proceedings in Informatics
LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

► **Definition 1** (Primal Shatter Function [21, 27]). The *primal shatter function* of $\mathcal{V} \subseteq \{0, 1\}^n$ is a function, denoted by $\pi_{\mathcal{V}}$, whose value at m is defined by $\pi_{\mathcal{V}}(m) = \max_{I \subseteq [n], |I|=m} |\mathcal{V}|_I$. In other words, $\pi_{\mathcal{V}}(m)$ is the maximum possible number of distinct vectors of \mathcal{V} when projected onto a subsequence of m indices.

From now on we say that $\mathcal{V} \subseteq \{0, 1\}^n$ has *primal shatter dimension* d if $\pi_{\mathcal{V}}(m) \leq Cm^d$, for all $m \leq n$, where $d > 1$ and $C > 0$ are constants. A notion closely related to the primal shatter dimension is that of the *VC-dimension*:

► **Definition 2** (VC-dimension [19, 32]). An index sequence $I = (i_1, \dots, i_k)$ is *shattered* by \mathcal{V} if $\mathcal{V}|_I = \{0, 1\}^k$. The *VC-dimension* of \mathcal{V} , denoted by d_0 is the size of the longest sequence I shattered by \mathcal{V} . That is, $d_0 = \max\{k \mid \exists I = (i_1, i_2, \dots, i_k), 1 \leq i_j \leq n, \text{ with } \mathcal{V}|_I = \{0, 1\}^k\}$.

The notions of primal shatter dimension and VC-dimension are interrelated. By the Sauer-Shelah Lemma (see [29, 31] and the discussion below) the VC-dimension of a set system \mathcal{V} always bounds its primal shatter dimension, that is, $d \leq d_0$. On the other hand, when the primal shatter dimension is bounded by d , the VC-dimension d_0 does not exceed $O(d \log d)$ (which is straightforward by definition, see, e.g., [16]).

A typical family of set systems that arise in geometry with bounded primal shatter (resp., VC-) dimension consists of set systems defined over points in some low-dimensional space \mathbb{R}^d , where \mathcal{V} represents a collection of certain simply-shaped regions, e.g., halfspaces, balls, or simplices in \mathbb{R}^d . In such cases, the primal shatter (and VC-) dimension is a function of d ; see, e.g., [16] for more details. When we flip the roles of points and regions, we obtain the so-called *dual set systems* (where we refer to the former as *primal set systems*). In this case, the ground set is a collection \mathcal{S} of algebraic surfaces in \mathbb{R}^d , and \mathcal{V} corresponds to faces of all dimensions in the *arrangement* $\mathcal{A}(\mathcal{S})$ of \mathcal{S} , that is, this is the decomposition of \mathbb{R}^d into connected open *cells* of dimensions $0, 1, \dots, d$ induced by \mathcal{S} . Each cell is a maximal connected region that is contained in the intersection of a fixed number of the surfaces and avoids all other surfaces; in particular, the 0-dimensional cells of $\mathcal{A}(\mathcal{S})$ are called “vertices”, and d -dimensional cells are simply referred to as “cells”; see [30] for more details. The distinction between primal and dual set systems in geometry is essential, and set systems of both kinds appear in numerous geometric applications, see, once again [16] and the references therein.

δ -packing

The *length* $\|\mathbf{v}\|$ of a vector $\mathbf{v} \in \mathcal{V}$ under the L^1 norm is defined as $\sum_{i=1}^n |\mathbf{v}_i|$, where \mathbf{v}_i is the i th coordinate of \mathbf{v} , $i = 1, \dots, n$. The *distance* $\rho(\mathbf{u}, \mathbf{v})$ between a pair of vectors $\mathbf{u}, \mathbf{v} \in \mathcal{V}$ is defined as the L^1 norm of the difference $\mathbf{u} - \mathbf{v}$, that is, $\rho(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^n |\mathbf{u}_i - \mathbf{v}_i|$. In other words, it is the *symmetric difference distance* between the corresponding sets represented by \mathbf{u}, \mathbf{v} .

Let $\delta > 0$ be an integer parameter. We say that a subset of vectors $\mathcal{W} \subseteq \{0, 1\}^n$ is *δ -separated* if for each pair $\mathbf{u}, \mathbf{v} \in \mathcal{W}$, $\rho(\mathbf{u}, \mathbf{v}) > \delta$. The *δ -packing number* for \mathcal{V} , denote it by $\mathcal{M}(\delta, \mathcal{V})$, is then defined as the cardinality of the largest δ -separated subset $\mathcal{W} \subseteq \mathcal{V}$. A key property, originally shown by Haussler [19] (see also [8, 9, 11, 27, 33]), is that set systems of bounded primal shatter dimension admit small δ -packing numbers. That is:

► **Theorem 3** (Packing Lemma [19, 27]). *Let $\mathcal{V} \subseteq \{0, 1\}^n$ be a set of indicator vectors of primal shatter dimension d , and let $1 \leq \delta \leq n$ be an integer parameter. Then $\mathcal{M}(\delta, \mathcal{V}) = O((n/\delta)^d)$, where the constant of proportionality depends on d .*

We note that in the original formulation in [19] the assumption is that the set system has a finite VC-dimension. However, its formulation in [27], which is based on a simplification of

the analysis of Haussler by Chazelle [8], relies on the assumption that the primal shatter dimension is d , which is the actual bound that we state in Theorem 3. We also comment that a closer inspection of the analysis in [19] shows that this assumption can be replaced with that of having bounded primal shatter dimension (independent of the analysis in [8]). We describe these considerations in Section 2.1.

Previous work. In his seminal work, Dudley [11] presented the first application of *chaining*, a proof technique due to Kolmogorov, to empirical process theory, where he showed the bound $O((n/\delta)^{d_0} \log^{d_0}(n/\delta))$ on $\mathcal{M}(\delta, \mathcal{V})$, with a constant of proportionality depending on the VC-dimension d_0 (see also previous work by Haussler [18] and Pollard [28] for an alternative proof and a specification of the constant of proportionality). This bound was later improved by Haussler [19], who showed $\mathcal{M}(\delta, \mathcal{V}) \leq e(d_0 + 1) \left(\frac{2en}{\delta}\right)^{d_0}$ (see also Theorem 3), and presented a matching lower bound, which leaves only a constant factor gap, which depends exponentially in d_0 . In fact, the aforementioned bounds are more general, and can also be applied to classes of real-valued functions of finite “pseudo-dimension” (the special case of set systems corresponds to Boolean functions), see, e.g., [18], however, we do not discuss this generalization in this paper and focus merely on set systems \mathcal{V} of finite primal shatter (resp., VC-) dimension.

The bound of Haussler [19] (Theorem 3) is in fact a generalization of the so-called Sauer-Shelah Lemma [29, 31], asserting that $|\mathcal{V}| \leq (en/d_0)^{d_0}$, where e is the base of the natural logarithm, and thus this bound is $O(n^{d_0})$. Indeed, when $\delta = 1$, the corresponding δ -separated set should include all vectors in \mathcal{V} , and then the bound of Haussler [19] becomes $O(n^{d_0})$, matching the Sauer-Shelah bound up to a constant factor that depends on d_0 .

There have been several studies extending Haussler’s bound or improving it in some special scenarios. We name only a few of them. Gottlieb *et al.* [15] presented a sharpening of this bound when δ is relatively large, i.e., δ is close to $n/2$, in which case the vectors are “nearly orthogonal”. They also presented a tighter lower bound, which considerably simplifies the analysis of Bshouty *et al.* [6], who achieved the same tightening.

A major application of packing is in obtaining improved bounds on the *sample complexity* in machine learning. This was studied by Li *et al.* [22] (see also [18]), who presented an asymptotically tight bound on the sample complexity, in order to guarantee a small “relative error.” This problem has been revisited by Har-Peled and Sharir [17] in the context of geometric set systems, where they referred to a sample of the above kind as a “relative approximation”, and showed how to integrate it into an *approximate range counting* machinery, which is a central application in computational geometry. The packing number has also been used by Welzl [33] in order to construct spanning trees of low crossing number (see also [27]) and by Matoušek [26, 27] in order to obtain asymptotically tight bounds in geometric discrepancy.

Our result

In the sequel, we refine the bound in the Packing Lemma (Theorem 3) so that it becomes sensitive to the length of the vectors $\mathbf{v} \in \mathcal{V}$, based on an appropriate refinement of the underlying primal shatter function. This refinement has several geometric realizations. Our ultimate goal is to show that when the set system is “shallow” (that is, the underlying vectors are short), the packing number becomes much smaller than the bound in Theorem 3.

Nevertheless, we cannot always enforce such an improvement, as in some settings the worst-case asymptotic bound on the packing number is $\Omega((n/\delta)^d)$ even when the set system is shallow; see [14] for an example.

Therefore, in order to obtain an improvement on the packing number of shallow set systems, we may need further assumptions on the primal shatter function. Such assumptions stem from the random sampling technique of Clarkson and Shor [10], which we define as follows. Let \mathcal{V} be our set system. We assume that for any sequence I of $m \leq n$ indices, and for any parameter $1 \leq k \leq m$, the number of vectors in $\mathcal{V}|_I$ of length at most k is only $O(m^{d_1} k^{d-d_1})$, where d is the primal shatter dimension and $1 \leq d_1 \leq d$ is a real parameter.¹ When $k = m$ we obtain $O(m^d)$ vectors in total, in accordance with the assumption that the primal shatter dimension is d , but the above bound is also sensitive to the length of the vectors as long as $d_1 < d$. From now on, we say that a primal shatter function of this kind has the (d, d_1) *Clarkson-Shor property*.

Let us now denote by $\mathcal{M}(\delta, k, \mathcal{V})$ the δ -packing number of \mathcal{V} , where the vector length of each element in \mathcal{V} is at most k , for some integer parameter $1 \leq k \leq n$. By these assumptions, we can assume, without loss of generality, that $k \geq \delta/2$, as otherwise the distance between any two elements in \mathcal{V} must be strictly less than δ , in which case the packing is empty. In Sections 2–3 we present two proofs for our main result, stated below:

► **Theorem 4 (Shallow Packing Lemma).** *Let $\mathcal{V} \subseteq \{0, 1\}^n$ be a set of indicator vectors, whose primal shatter function has a (d, d_1) Clarkson-Shor property, and whose VC-dim is d_0 . Let $\delta \geq 1$ be an integer parameter, and k an integer parameter between 1 and n , and suppose that $k \geq \delta/2$. Then:*

$$\mathcal{M}(\delta, k, \mathcal{V}) = O\left(\frac{n^{d_1} k^{d-d_1}}{\delta^d}\right),$$

where the constant of proportionality depends on d (and d_0).

This problem has initially been addressed by the second author in [13] as a major tool to obtain size-sensitive discrepancy bounds in set systems of this kind, where it has been shown $\mathcal{M}(\delta, k, \mathcal{V}) = O\left(\frac{n^{d_1} k^{d-d_1} \log^d(n/\delta)}{\delta^d}\right)$. The analysis in [13] is a refinement over the technique of Dudley [11] combined with the existence of small-size *relative approximations* (see [13] for more details). In the current analysis we completely remove the extra $\log^d(n/\delta)$ factor appearing in the previous bound. In particular, when $d_1 = d$ (where we just have the original assumption on the primal shatter function) or $k = n$ (in which case each vector in \mathcal{V} has an arbitrary length), our bound matches the tight bound of Haussler, and thus appears as a generalization of the Packing Lemma (when replacing VC-dimension by primal shatter dimension). We present two proofs for Theorem 4, the first is an extension of Haussler’s approach (Section 2), and the second is an extension of Chazelle’s proof [8] to the Packing Lemma (Section 3).

2 First Proof: Refining Haussler’s Approach

2.1 Preliminaries

Overview of Haussler’s Approach

For the sake of completeness, we repeat some of the details in the analysis of Haussler [19] and use similar notation for ease of presentation.

Let $\mathcal{V} \subseteq \{0, 1\}^n$ be a collection of indicator vectors of bounded primal shatter dimension d , and denote its VC-dimension by d_0 . By the discussion above, $d_0 = O(d \log d)$. From now

¹ We ignore the cases where $d_1 < 1$, as it does not seem to appear in natural set systems – see below.

on we assume that \mathcal{V} is δ -separated, and thus a bound on $|\mathcal{V}|$ is also a bound on the packing number of \mathcal{V} . The analysis in [19] exploits the method of “conditional variance” in order to conclude

$$|\mathcal{V}| \leq (d_0 + 1) \mathbf{Exp}_I [|\mathcal{V}_{|I}|] = O(d \log d \mathbf{Exp}_I [|\mathcal{V}_{|I}|]), \quad (1)$$

where $\mathbf{Exp}_I [|\mathcal{V}_{|I}|]$ is the expected size of \mathcal{V} when projected onto a subset $I = \{i_1, \dots, i_{m-1}\}$ of $m - 1$ indices chosen uniformly at random without replacements from $[n]$, and

$$m := \left\lceil \frac{(2d_0 + 2)(n + 1)}{\delta + 2d_0 + 2} \right\rceil = O\left(\frac{d_0 n}{\delta}\right) = O\left(\frac{nd \log d}{\delta}\right). \quad (2)$$

See a preliminary version of this paper for details, as well as the facts that $m \leq n$ and I consists of precisely $m - 1$ indices [14, Appendix B].

Moreover, we refine Haussler’s analysis to include two natural extensions (see [14, Appendix B] for details): (i) *Obtain a refined bound on $\mathbf{Exp}_I [|\mathcal{V}_{|I}|]$* : This extension is a direct consequence of Inequality (1). In the analysis of Haussler $\mathbf{Exp}_I [|\mathcal{V}_{|I}|]$ is replaced by its upper bound $O(m^d)$, resulting from the fact that the primal shatter dimension of \mathcal{V} (and thus of $\mathcal{V}_{|I}$) is d , from which we obtain that for any choice of I , $|\mathcal{V}_{|I}| = O((m - 1)^d) = O(m^d)$, with a constant of proportionality that depends on d , and thus the packing number is $O((n/\delta)^d)$, as asserted in Theorem 3.² However, in our analysis we would like to have a more subtle bound on the actual expected value of $|\mathcal{V}_{|I}|$. In fact, the scenario imposed by our assumptions on the set system eventually yields a much smaller bound on the expectation of $|\mathcal{V}_{|I}|$, and thus on $|\mathcal{V}|$. We review this in more detail below. (ii) *Relaxing the bound on m* . We show that Inequality (1) is still applicable when the sample I is slightly larger than the bound in (2), as a stand alone relation, this may result in a suboptimal bound on $|\mathcal{V}|$, however, this property will assist us to obtain local improvements over the bound on $|\mathcal{V}|$, eventually yielding the bound in Theorem 4. Specifically, in our analysis we proceed in iterations, where at the first iteration we obtain a preliminary bound on $|\mathcal{V}|$ (Corollary 6), and then, at each subsequent iteration $j > 1$, we draw a sample I_j of $m_j - 1$ indices where

$$m_j := m \log^{(j)}(n/\delta) = O\left(\frac{d_0 n \log^{(j)}(n/\delta)}{\delta}\right), \quad (3)$$

m is our choice in (2), and $\log^{(j)}(\cdot)$ is the j th iterated logarithm function. Then, by a straightforward generalization of Haussler’s analysis (described in [14, Appendix B]), we obtain, for each $j = 2, \dots, \log^*(n/\delta)$:

$$|\mathcal{V}| \leq (d_0 + 1) \mathbf{Exp}_{I_j} [|\mathcal{V}_{|I_j}|]. \quad (4)$$

We note that since the bounds (1)–(4) involve a dependency on the VC-dimension d_0 , we will sometimes need to explicitly refer to this parameter

in addition to the primal shatter dimension d . Nevertheless, throughout the analysis we exploit the relation $d \leq d_0 = O(d \log d)$, mentioned in Section 1.

² We note, however, that the original analysis of Haussler [19] does not rely on the primal shatter dimension, and the bound on $\mathbf{Exp}_I [|\mathcal{V}_{|I}|]$ is just $O(m^{d_0})$ due to the Sauer-Shelah Lemma.

2.2 Overview of the approach.

We next present the proof of Theorem 4. In what follows, we assume that \mathcal{V} is δ -separated. We first recall the assumption that the primal shatter function of \mathcal{V} has a (d, d_1) Clarkson-Shor property, and that the length of each vector $\mathbf{v} \in \mathcal{V}$ under the L^1 norm is most k . This implies that \mathcal{V} consists of at most $O(n^{d_1} k^{d-d_1})$ vectors.

Since the Clarkson-Shor property is hereditary, then this also applies to any projection of \mathcal{V} onto a subset of indices, implying that the bound on $|\mathcal{V}_{|I}|$ is at most $O(m^{d_1} k^{d-d_1})$, where I is a subset of $m - 1$ indices as above. However, due to our sampling scheme we expect that the length of each vector in $\mathcal{V}_{|I}$ should be much smaller than k , (e.g., in expectation this value should not exceed $k(m - 1)/n$), from which we may conclude that the actual bound on $|\mathcal{V}_{|I}|$ is smaller than the trivial bound $O(m^{d_1} k^{d-d_1})$. Ideally, we would like to show that this bound is $O(m^{d_1} (km/n)^{d-d_1}) = O(n^{d_1} k^{d-d_1} / \delta^d)$, which matches our asymptotic bound in Theorem 4 (recall that $m = O(n/\delta)$). However, this is likely to happen only in case where the length of each vector in $\mathcal{V}_{|I}$ does not exceed its expected value, or that there are only a few vectors whose length deviates from its expected value by far, whereas, in the worst case there might be many leftover “long” vectors in $\mathcal{V}_{|I}$. Nevertheless, our goal is to show that, with some carefulness one can proceed in iterations, where initially I is a slightly larger sample, and then at each iteration we reduce its size, until eventually it becomes $O(m)$ and we remain with only a few long vectors. At each such iteration $\mathcal{V}_{|I}$ is a random structure that depends on the choice of I and may thus contain long vectors, however, in expectation they will be scarce!

Specifically, we proceed over at most $\log^*(n/\delta)$ iterations, where we perform local improvements over the bound on $|\mathcal{V}|$, as follows. Let $|\mathcal{V}|^{(j)}$ be the bound on $|\mathcal{V}|$ after the j th iteration is completed, $1 \leq j \leq \log^*(n/\delta)$. We first show in Corollary 6 that for the first iteration, $|\mathcal{V}| \leq |\mathcal{V}|^{(1)} = O\left(\frac{n^{d_1} k^{d-d_1} \log^d(n/\delta)}{\delta^d}\right)$, with a constant of proportionality that depends on d . Then, at each further iteration $j \geq 2$, we select a set I_j of $m_j - 1 = O(n \log^{(j)}(n/\delta)/\delta)$ indices uniformly at random without replacements from $[n]$ (see (3) for the bound on m_j). Our goal is to bound $\mathbf{Exp}_{I_j} [|\mathcal{V}_{|I_j}|]$ using the bound $|\mathcal{V}|^{(j-1)}$, obtained at the previous iteration, which, we assume by induction to be $O\left(\frac{n^{d_1} k^{d-d_1} (\log^{(j-1)}(n/\delta))^d}{\delta^d}\right)$ (note that the actual constant of proportionality in our recursive scheme is 1, see Lemma 8), where the base case $j = 2$ is shown in Corollary 6.

A key property in the analysis is then to show that the probability that the length of a vector $\mathbf{v} \in \mathcal{V}_{|I_j}$ (after the projection of \mathcal{V} onto I_j) deviates from its expectation decays exponentially (Lemma 7). Note that in our case this expectation is at most $k(m_j - 1)/n$. This, in particular, enables us to claim that *in expectation* the overall majority of the vectors in $\mathcal{V}_{|I_j}$ have length at most $O(k(m_j - 1)/n)$, whereas the remaining longer vectors are scarce. Specifically, since the Clarkson-Shor property is hereditary, we apply it to $\mathcal{V}_{|I_j}$ and conclude that the number of its vectors of length at most $O(k(m_j - 1)/n)$ is only $O\left(\frac{n^{d_1} k^{d-d_1} (\log^{(j)}(n/\delta))^d}{\delta^d}\right)$, with a constant of proportionality that depends on d . On the other hand, due to Lemma 7 and our inductive hypothesis, the number of longer vectors does not exceed $O\left(\frac{n^{d_1} k^{d-d_1}}{\delta^d}\right)$, which is dominated by the first bound. We thus conclude $\mathbf{Exp}_{I_j} [|\mathcal{V}_{|I_j}|] = O\left(\frac{n^{d_1} k^{d-d_1} (\log^{(j)}(n/\delta))^d}{\delta^d}\right)$. Then we apply Inequality (4) in order to complete the inductive step, whence we obtain the bound on $|\mathcal{V}|^{(j)}$, and thus on $|\mathcal{V}|$. These properties are described more rigorously in Lemma 8, where derive a recursive inequality for $|\mathcal{V}|^{(j)}$ using the bound on $\mathbf{Exp}_{I_j} [|\mathcal{V}_{|I_j}|]$. We emphasize the fact that the sample I_j is

always chosen from the *original* ground set $[n]$, and thus, at each iteration we construct a new sample *from scratch*, and then exploit our observation in (4).

In what follows, we also assume that $\delta \leq n/2^{(d_0+1)}$ (where d_0 is the VC-dim), as otherwise the bound on the packing number is a constant that depends on d and d_0 by the Packing Lemma (Theorem 3). This assumption is crucial for the recursive analysis presented in this section – see below.

2.3 The First Iteration

In order to show our bound on $|\mathcal{V}^{(1)}|$, we form a subset $I_1 = (i_1, \dots, i_{m_1})$ of $m_1 = |I_1| = O\left(\frac{dn \log(n/\delta)}{\delta}\right)$ indices³ with the following two properties: (i) each vector in \mathcal{V} is mapped to a distinct vector in $\mathcal{V}_{|I_1}$, and (ii) the length of each vector in $\mathcal{V}_{|I_1}$ does not exceed $O(k \cdot m_1/n)$.

► **Lemma 5.** *A sample I_1 as above satisfies properties (i)–(ii), with probability at least $1/2$.*

A set I_1 as above exists by the considerations in [13]. See also a preliminary version of this paper for further details [14, Appendix C].

We next apply Lemma 5 in order to bound $|\mathcal{V}_{|I_1}|$. We first recall that the (d, d_1) Clarkson-Shor property of the primal shatter function of \mathcal{V} is hereditary. Incorporating the bound on m_1 and property (ii), we conclude that

$$|\mathcal{V}_{|I_1}| = O\left(m_1^{d_1} \left(\frac{km_1}{n}\right)^{d-d_1}\right) = O\left(\frac{n^{d_1} k^{d-d_1} \log^d(n/\delta)}{\delta^d}\right),$$

with a constant of proportionality that depends on d . Now, due to property (i), $|\mathcal{V}| \leq |\mathcal{V}_{|I_1}|$, we thus conclude:

► **Corollary 6.** *After the first iteration we have: $|\mathcal{V}| \leq |\mathcal{V}^{(1)}| = O\left(\frac{n^{d_1} k^{d-d_1} \log^d(n/\delta)}{\delta^d}\right)$, with a constant of proportionality that depends on d .*

► **Remark.** We note that the preliminary bound given in Corollary 6 is crucial for the analysis, as it constitutes the base for the iterative process described in Section 2.4. In fact, this step of the analysis alone bypasses our refinement to Haussler’s approach, and instead exploits the approach of Dudley [11].

2.4 The Subsequent Iterations: Applying the Inductive Step

Let us now fix an iteration $j \geq 2$. As noted above, we assume by induction on j that the bound $|\mathcal{V}^{(j-1)}|$ on $|\mathcal{V}|$ after the $(j-1)$ th iteration is $O\left(\frac{n^{d_1} k^{d-d_1} (\log^{(j-1)}(n/\delta))^d}{\delta^d}\right)$. Let I_j be a subset of $m_j - 1$ indices, chosen uniformly at random without replacements from $[n]$, with m_j given by (3). Let $\mathbf{v} \in \mathcal{V}$, and denote by $\mathbf{v}_{|I_j}$ its projection onto I_j . The expected length $\mathbf{Exp}[\|\mathbf{v}_{|I_j}\|]$ of $\mathbf{v}_{|I_j}$ is at most $k(m_j - 1)/n = O(d_0 k \log^{(j)}(n/\delta)/\delta)$. We next show (see a preliminary version of this paper [14, Appendix D] for the proof):

³ In this particular step we use a different machinery than that of Haussler [19]; see the proof of Lemma 5 and our remark after Corollary 6. Therefore, $|I_1| = m_1$, rather than $m_1 - 1$. Furthermore, the constant of proportionality in the bound on m_1 depends just on the primal shatter dimension d instead of the VC-dimension d_0 as in (3).

► **Lemma 7** (Exponential Decay Lemma).

$$\mathbf{Prob} \left[\|\mathbf{v}_{|I_j}\| \geq t \cdot \frac{k(m_j - 1)}{n} \right] < 2^{-tk(m_j - 1)/n},$$

where $t \geq 2e$ is a real parameter and e is the base of the natural logarithm.

We now proceed as follows. Recall that we assume $k \geq \delta/2$, and by (3) we have $m_j = O\left(\frac{d_0 n \log^{(j)}(n/\delta)}{\delta}\right)$. It follows from Lemma 7 that

$$\mathbf{Prob} \left[\|\mathbf{v}_{|I_j}\| \geq C \cdot \frac{k(m_j - 1)}{n} \right] < \frac{1}{(\log^{(j-1)}(n/\delta))^D}, \tag{5}$$

where $C \geq 2e$ is a sufficiently large constant, and $D > d_0$ is another constant whose choice depends on C and d_0 , and can be made arbitrarily large. Since $d_0 \geq d$ we obviously have $D > d$. We next show:

► **Lemma 8.** *Under the assumption that $k \geq \delta/2$, we have, at any iteration $j \geq 2$:*

$$|\mathcal{V}|^{(j)} \leq A(d_0 + 1) \cdot \frac{n^{d_1} k^{d-d_1} (\log^{(j)}(n/\delta))^d}{\delta^d} + (d_0 + 1) \cdot \frac{|\mathcal{V}|^{(j-1)}}{(\log^{(j-1)}(n/\delta))^D}, \tag{6}$$

where $|\mathcal{V}|^{(l)}$ is the bound on $|\mathcal{V}|$ after the l th iteration, and $A > 0$ is a constant that depends on d (and d_0) and the constant of proportionality determined by the Clarkson-Shor property of \mathcal{V} .

Proof. We in fact show:

$$\mathbf{Exp}_{I_j} \left[|\mathcal{V}_{|I_j}| \right] \leq A \cdot \frac{n^{d_1} k^{d-d_1} (\log^{(j)}(n/\delta))^d}{\delta^d} + \frac{|\mathcal{V}|^{(j-1)}}{(\log^{(j-1)}(n/\delta))^D},$$

and then exploit the relation $|\mathcal{V}| \leq (d_0 + 1) \mathbf{Exp}_{I_j} \left[|\mathcal{V}_{|I_j}| \right]$ (Inequality (4)), in order to prove (6).

In order to obtain the first term in the bound on $\mathbf{Exp}_{I_j} \left[|\mathcal{V}_{|I_j}| \right]$, we consider all vectors of length at most $C \cdot \frac{k(m_j - 1)}{n}$ (where $C \geq 2e$ is a sufficiently large constant as above) in the projection of \mathcal{V} onto a subset I_j of $m_j - 1$ indices (in this part of the analysis I_j can be arbitrary). Since the primal shatter function of \mathcal{V} has a (d, d_1) Clarkson-Shor property, which is hereditary, we obtain at most

$$O(m_j^{d_1} (k(m_j - 1)/n)^{d-d_1}) = O\left(\frac{n^{d_1} k^{d-d_1} (\log^{(j)}(n/\delta))^d}{\delta^d}\right)$$

vectors in $\mathcal{V}_{|I_j}$ of length smaller than $C \cdot \frac{k(m_j - 1)}{n} = O\left(\frac{k \log^{(j)}(n/\delta)}{\delta}\right)$. It is easy to verify that the constant of proportionality A in the bound just obtained depends on d , d_0 , and the constant of proportionality determined by the Clarkson-Shor property of \mathcal{V} .

Next, in order to obtain the second term, we consider the vectors $\mathbf{v} \in \mathcal{V}$ that are mapped to vectors $\mathbf{v}_{|I_j} \in \mathcal{V}_{|I_j}$ with $\|\mathbf{v}_{|I_j}\| > C \cdot \frac{k(m_j - 1)}{n}$. By Inequality (5):

$$\mathbf{Exp} \left[\left| \left\{ \mathbf{v} \in \mathcal{V} \mid \|\mathbf{v}_{|I_j}\| > C \cdot \frac{k(m_j - 1)}{n} \right\} \right| \right] < \frac{|\mathcal{V}|}{(\log^{(j-1)}(n/\delta))^D},$$

and recall that $|\mathcal{V}|^{(j-1)}$ is the bound on $|\mathcal{V}|$ after the previous iteration $j - 1$. This completes the proof of the lemma. ◀

► **Remark.** We note that the bound on $\mathbf{Exp}_{I_j} \left[|\mathcal{V}_{I_j}| \right]$ consists of the *worst-case* bound on the number of short vectors of length at most $C \cdot k(m_j - 1)/n$, obtained by the Clarkson-Shor property, plus the *expected* number of long vectors.

Wrapping up. We now complete the analysis and solve Inequality (6). Our initial assumption that $\delta \leq n/2^{(d_0+1)}$, and the fact that $D > d$ is sufficiently large, imply that the coefficient of the recursive term is smaller than 1, for any $2 \leq j \leq 1 + \log^*(n/\delta) - \log^*(d_0 + 1)$.⁴ Then, using induction on j , one can verify that the solution is

$$|\mathcal{V}|^{(j)} \leq 2A(d_0 + 1) \frac{n^{d_1} k^{d-d_1} (\log^{(j)}(n/\delta))^d}{\delta^d}, \quad (7)$$

for any $2 \leq j \leq 1 + \log^*(n/\delta) - \log^*(d_0 + 1)$.

We thus conclude $|\mathcal{V}|^{(j)} = O\left(\frac{n^{d_1} k^{d-d_1} (\log^{(j)}(n/\delta))^d}{\delta^d}\right)$. In particular, at the termination of the last iteration $j^* = 1 + \log^*(n/\delta) - \log^*(d_0 + 1)$, we obtain:

$$|\mathcal{V}| \leq |\mathcal{V}|^{(j^*)} = O\left(\frac{n^{d_1} k^{d-d_1}}{\delta^d}\right),$$

with a constant of proportionality that depends on d (and d_0). This at last completes the proof of Theorem 4.

3 Second Proof: Refining Chazelle's Approach

In this section, we shall prove a size-sensitive version of Haussler's upper bound for δ -separated systems in set-systems of bounded primal shatter dimension building on Chazelle's presentation of Haussler's proof, (which has been described by Matoušek as "a magician's trick") as explained in [27]. By Haussler's result [19], we know that $M = O(n/\delta)^d = (n/\delta)^{d_1} (l/\delta)^{d_2} \cdot g(n, l, \delta)^d$, where $g(n, l, \delta) = O((n/l)^{d_2})$. We would like to show the optimum upper bound for g is independent of n, l . We shall show that the optimal bound (up to constants) is in fact, $g = c^*$, where c^* is the fixed point of $f(x) = c' \log x$, with $c' > 1$ independent of n, l, δ .

Intuition

We provide some intuition for our extension of the Haussler/Chazelle proof below (at least to the reader familiar with it). A naïve attempt to extend Chazelle's proof to shallow packings, fails, because (as in the previous proof), one chooses a random subsequence I , and estimates the number of projections on I , caused by δ -packed vectors of bounded size. For a given vector, its projection on I can be much larger than expected. However, we shall choose A' in a way that the *number* of such "bad" vectors, is at most a constant times their expected *number*. This allows us to get the final bound in a single iteration.

Details

Before we give the details of the second proof, we will need the definition of *unit distance graph* of a set system which will play central role in the proof of the theorem.

⁴ We observe that $2 \leq 1 + \log^*(n/\delta) - \log^*(d_0 + 1) \leq \log^*(n/\delta)$, due to our assumption that $\delta \leq n/2^{(d_0+1)}$, and the fact that $d_0 \geq 1$.

► **Definition 9** (Unit distance graph). For a set system \mathcal{V} , unit distance graph $\mathcal{UD}(\mathcal{V})$ is a graph with vertex set \mathcal{V} and a pair $\{\mathbf{v}_1, \mathbf{v}_2\}$ is an edge if $\rho(\mathbf{v}_1, \mathbf{v}_2) = 1$.

Consider a random subsequence of indices $I = (i_1, \dots, i_s)$ where each $i \in [n]$ is selected with probability $p = \frac{36d_0K}{\delta}$, where $K \geq 1$ is a parameter to be fixed later. Define $\mathcal{V}_1 := \mathcal{V}_{|I}$. Consider the unit distance graph $\mathcal{UD}(\mathcal{V}_1)$. For each set $\mathbf{v}_1 \in \mathcal{V}_1$, define the weight of \mathbf{v}_1 as: $w(\mathbf{v}_1) := \#\{\mathbf{v} \in \mathcal{V} : \mathbf{v}_{|I} = \mathbf{v}_1\}$. Observe that

$$\sum_{\mathbf{v}_1 \in \mathcal{V}_1} w(\mathbf{v}_1) = \mathcal{M}(\delta, k, \mathcal{V}).$$

Let E be the edge set of $\mathcal{UD}(\mathcal{V}_1)$. Now define the weight of an edge $e = \{\mathbf{v}_1, \mathbf{v}'_1\} \in E$ as $w(e) := \min(w(\mathbf{v}_1), w(\mathbf{v}'_1))$. Let $W := \sum_{e \in E} w(e)$. We claim that

► **Lemma 10.** $W \leq 2d_0 \sum_{\mathbf{v}_1 \in \mathcal{V}_1} w(\mathbf{v}_1) = 2d_0 \mathcal{M}(\delta, k, \mathcal{V})$.

Proof. The proof is based on the following lemma, proved by Haussler [19] for set systems with bounded VC-dimension. The following version appears in Matoušek’s book [27]:

► **Lemma 11** ([19]). *Let \mathcal{V} be a set-system with VC-dimension d_0 . Then the unit-distance graph $\mathcal{UD}(\mathcal{V})$ has at most $d_0|\mathcal{V}|$ edges.*

Since the VC-dimension of \mathcal{V}_1 is bounded by d_0 from the hereditary property of VC-dimension, the lemma implies that there exists a vertex $\mathbf{v}_1 \in \mathcal{V}_1$, whose degree is at most $2d_0$. Removing \mathbf{v}_1 , the total vertex weight drops by $w(\mathbf{v}_1)$, and the total edge weight drops by at most $2d_0w(\mathbf{v}_1)$. Continuing the argument until all vertices are removed, we get the claim. ◀

Next, we shall prove a lower bound on the expectation $\mathbf{Exp}[W]$. Choose a random element $i_j \in \{i_1, \dots, i_s\}$. Let $\mathcal{V}_2 := \mathcal{V}_{|I'}$, where $I' = (i_1, \dots, i_{j-1}, i_{j+1}, \dots, i_s)$, i.e., by abuse of notation $I' = I \setminus \{i_j\}$. Note that I' is a random subsequence where each $i \in [n]$ was chosen with probability $p' = p - 1/n$. Crucially, one can consider the above process equivalent to first choosing I' by selecting each element of $[n]$ with probability p' , and then selecting a uniformly random element $i_j \in [n] \setminus I'$ with probability $1/n$.

Let $E_1 \subset E$ be those edges $(\mathbf{v}_1, \mathbf{v}'_1)$ of E where vectors \mathbf{v}_1 and \mathbf{v}'_1 differ in the coordinate i_j , and let

$$W_1 := \sum_{e \in E_1} w(e).$$

We need to lower bound $\mathbf{Exp}[W_1]$. Given I' , let

$$Y = Y(I') := \#\{\mathbf{v} \in \mathcal{V} : \|\mathbf{v}_{|I'}\| > c(k/\delta)\},$$

i.e., the number of vectors in \mathcal{V} , each of whose norm after projecting onto I' is more than $c(l/\delta)$, (where c shall be chosen appropriately). Let *Nice* denote the event

$$(Y \leq 8 \mathbf{Exp}[Y]) \cap \left(\frac{np}{2} \leq s \leq \frac{3np}{2} \right) = N_Y \cap N_S.$$

Conditioning W on *Nice*, we get:

$$\begin{aligned} \mathbf{Exp}[W] &= \mathbf{Prob}[Nice] \mathbf{Exp}[W|Nice] + \mathbf{Prob}[\overline{Nice}] \mathbf{Exp}[W|\overline{Nice}] \\ &> \mathbf{Prob}[Nice] \mathbf{Exp}[W|Nice] \end{aligned}$$

By Markov’s Inequality, see [4, App. A], we have

$$\mathbf{Prob}[\overline{N_Y}] = \mathbf{Prob}[Y \geq 8 \mathbf{Exp}[Y]] \leq 1/8,$$

and using Chernoff Bounds, see [4, App. A], with the fact that $n/\delta \geq 1$, we get

$$\mathbf{Prob}[\overline{N_S}] = \mathbf{Prob}[|s - np| > np/2] \leq 2e^{(-36d_0Kn/3.2^2\delta)} \ll 1/4.$$

This implies

$$\mathbf{Prob}[Nice = N_Y \cap N_S] \geq 1 - \mathbf{Prob}[\overline{N_S}] - \mathbf{Prob}[\overline{N_Y}] \geq 7/8 - e^{(-4d_0K)} \geq 3/4,$$

where the last inequality follows from the fact that $d_0K \geq 1$.

Hence,

$$\mathbf{Exp}[W] \geq (3/4) \mathbf{Exp}[W|Nice] \geq \frac{3(np/2)}{4} \mathbf{Exp}[W_1|Nice], \quad (8)$$

where the last inequality follows by symmetry of the choice of i_j from I , and the lower bound on s when the event *Nice* holds.

Hence, $\mathbf{Exp}[W] \geq (\frac{3np}{8}) \mathbf{Exp}[W_1|Nice]$. So to lower bound $\mathbf{Exp}[W]$ up to constants, it suffices just to lower bound $\mathbf{Exp}[W_1|Nice]$. Let W_2 denote $W_1|Nice$. Consider now $\mathbf{Exp}[W_2|A']$. That is, consider a fixed subsequence I' whose length is between $np/2$ and $3np/2$, and which is such that the number of vectors $\mathbf{v} \in \mathcal{V}$ whose norm after projection onto I' in more than ck/δ , is at most $8 \mathbf{Exp}[Y]$. We shall lower bound $\mathbf{Exp}[W_2|I']$ for this choice of I' .

By definition, $W_1 = \sum_{e \in E_1} w(e)$. Consider the equivalence classes of \mathcal{V} formed by their projection onto I' :

$$\mathcal{V} = \mathcal{V}'_1 \cup \dots \cup \mathcal{V}'_r.$$

Define $Bad \subset [r]$ to be those indices j for which \mathcal{V}'_j is such that

$$\forall \mathbf{v} \in \mathcal{V}'_j : \|\mathbf{v}_{I'}\| > 8c(k/\delta).$$

Further, let $Good$ be $[r] \setminus Bad$. Since *Nice* holds, we have:

$$\sum_{j \in Bad} |\mathcal{V}'_j| \leq 8 \mathbf{Exp}[Y].$$

We first estimate the contribution of the classes in *Good*, to the total weight. Consider a class \mathcal{V}'_i such that $i \in Good$. Let $\mathcal{V}''_1 \subset \mathcal{V}'_i$ be those vectors in \mathcal{V}'_i which contains 1 in the i_j -th coordinate, and let $\mathcal{V}''_2 = \mathcal{V}'_i \setminus \mathcal{V}''_1$. Let $b = |\mathcal{V}'_i|$, $b_1 = |\mathcal{V}''_1|$ and $b_2 = |\mathcal{V}''_2|$. Then the edge $e \in E_1$ formed by the projection of \mathcal{V}'_i onto I , has weight

$$w(e) = \min(b_1, b_2) \geq \frac{b_1 b_2}{b}. \quad (9)$$

Observe that in Inequality (9), b is a constant as the subsequence I' is fixed and the product $b_1 b_2$ is the random variable that depends on the choice of i_j . The product $b_1 b_2$ is the number of ordered pairs of vectors $(\mathbf{v}, \mathbf{v}')$, with \mathbf{v} and \mathbf{v}' in \mathcal{V}'_i , such that \mathbf{v} and \mathbf{v}' differs only in the i_j -th coordinate. For a given ordered pair $(\mathbf{v}, \mathbf{v}')$ of distinct vectors $\mathbf{v}, \mathbf{v}' \in \mathcal{V}'_i$, the probability \mathbf{v} and \mathbf{v}' differ in the i_j -th coordinate is $\frac{\delta}{n-s+1}$, which is at least $\frac{\delta}{n}$. Therefore, the expected contribution of $(\mathbf{v}, \mathbf{v}')$ to $b_1 b_2$ is at least $\frac{\delta}{n}$ and this implies

$$\mathbf{Exp}[b_1 b_2] \geq \frac{b(b-1)\delta}{n}.$$

And this further implies the together with Inequality (9) that the weight of e (conditioned on *Nice* and I') is at least:

$$\mathbf{Exp}[w(e)|Nice \cap I'] \geq \frac{1}{b} \mathbf{Exp}[b_1 b_2] \geq \frac{b(b-1)}{b} \cdot \frac{\delta}{n} = (b-1) \frac{\delta}{n} = (|\mathcal{V}'_i| - 1) \frac{\delta}{n}.$$

Hence, the expected weight of $\mathbf{Exp}[W_2|I']$ is:

$$\mathbf{Exp}[W_2|I'] \geq \sum_{e \in E_1} \mathbf{Exp}[w(e)|Nice \cap I'] \geq \sum_{i \in Good} (|\mathcal{V}'_i| - 1) \frac{\delta}{n}$$

But by (d, d_1) Clarkson-Shor property, we have that

$$\forall i \in Good, |\mathcal{V}'_i|_{I'} \leq Cs^{d_1} (ckp)^{d-d_1}.$$

Substituting in the lower bound for $\mathbf{Exp}[W_2]$, we get:

$$\begin{aligned} \mathbf{Exp}[W_2|A'] &\geq \left(\left(\sum_{i \in Good} |\mathcal{V}'_i| \right) - C(1.5np)^{d_1} (ckp)^{d-d_1} \right) \frac{\delta}{n} \\ &\geq \left(|\mathcal{V}| - 8 \mathbf{Exp}[Y] - C(6dK)^d \cdot (1.5)^{d_1} c^{d-d_1} \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} \right) \frac{\delta}{n} \\ &\geq \left(\mathcal{M}(\delta, k, \mathcal{V}) - 8 \mathbf{Exp}[Y] - C_1 K^d \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} \right) \frac{\delta}{n} \end{aligned}$$

where in the first inequality, we used the fact that the event $N_S \subset Nice$ holds, and in the last line, $C_1 = C \cdot (6d)^{d_2 d_1} c^{d-d_1}$. Since the above holds for each I' which satisfies $Nice$, we get that

$$\mathbf{Exp}[W_2] \geq \left(\mathcal{M}(\delta, k, \mathcal{V}) - 8 \mathbf{Exp}[Y] - C_1 K^d \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} \right) \frac{\delta}{n},$$

Using equation (8), and comparing with the upper bound on W ,

$$(3np/8) \mathbf{Exp}[W_1|Nice] \leq \mathbf{Exp}[W] \leq 2d_0 \mathcal{M}(\delta, k, \mathcal{V}),$$

and substituting the lower bound $\mathbf{Exp}[W_1|Nice]$, and solving for $\mathcal{M}(\delta, k, \mathcal{V})$, we get

$$\mathcal{M}(\delta, k, \mathcal{V}) \leq \frac{(27K/4) \left(8 \mathbf{Exp}[Y] + C_1 K^d \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} \right)}{(27K/4 - 1)}.$$

The following lemma therefore, completes the proof:

► **Lemma 12.** For $K = \max\{1, (\ln g)/36\}$, $\mathbf{Exp}[Y] \leq C_2 \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d_2}$.

Indeed, substituting the choice of K and the value of $\mathbf{Exp}[Y]$ from Lemma 12, we get that

$$\begin{aligned} g^d \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} &= \mathcal{M}(\delta, k, \mathcal{V}) \\ &\leq \frac{C_1 K^d \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} + 8C_2 \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1}}{1 - 4/27K} \\ &\leq C_3 K^d \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} \leq C_4 (\max\{1, \log g\})^d \left(\frac{n}{\delta} \right)^{d_1} \left(\frac{k}{\delta} \right)^{d-d_1} \end{aligned}$$

where the shorthand $g = g(n, l, \delta)$. This implies that $g^d \leq C_4 (\max\{1, \log g\})^d$, or $g \leq C_5 \max\{1, \log g\}$. Since for any g growing with n, l , or δ , we would have $g \gg C_5 \log g$ for sufficiently large n, k or δ , this inequality is only satisfiable when g is a constant function of n, l, δ . i.e. $g \leq c^*$, where c^* is independent of n, k, δ .

It only remains to prove Claim 12:

Proof of Lemma 12. The proof follows easily from Chernoff Bounds. Fix $\mathbf{v} \in \mathcal{V}$. Let $Z = \|\mathbf{v}_{|I'}\|$. Then $\mathbf{Exp}[Z] = \|\mathbf{v}\|p' = kp'$. Since I' is a random subsequence chosen with probability $p' = p - 1/n$, the probability that

$$Z \geq ckp' = \frac{36cdKk}{\delta} - \frac{ck}{n}$$

is upper bounded using Chernoff bounds, see [4, App. A], as:

$$\mathbf{Prob}[Z - \mathbf{Exp}[Z] > (c - 1)\mathbf{Exp}[Z]] \leq e^{(-\mathbf{Exp}[Z])} \leq e^{(-36dKk/\delta)},$$

for $c = 1.01e$ and $n \geq 100$, say. Hence the expected number $\mathbf{Exp}[Y]$ of vectors, each of whose norm when projected onto I' in more than $36cdKk/\delta$ elements, is at most:

$$\mathbf{Exp}[Y] \leq \mathcal{M}(\delta, k, \mathcal{V})e^{(-36dKk/\delta)} \leq \mathcal{M}(\delta, k, \mathcal{V})e^{(-18dK)},$$

since $k \geq \delta/2$. Substituting the value of $\mathcal{M}(\delta, k, \mathcal{V})$ and also K in terms of f , we have

$$\mathbf{Exp}[Y] \leq g^d \left(\frac{n}{\delta}\right)^{d_1} \left(\frac{k}{\delta}\right)^{d-d_1} e^{(-18dK)} \leq \left(\frac{n}{\delta}\right)^{d_1} \left(\frac{k}{\delta}\right)^{d-d_1} e^{d(\ln g - 18K)} \leq \left(\frac{n}{\delta}\right)^{d_1} \left(\frac{k}{\delta}\right)^{d-d_1}$$

for $K \geq (\ln g)/18$. ◀

This completes the proof of Theorem 4.

4 Concluding Remarks and Further Research

We briefly mention a few applications of Theorem 4:

- (i) Smaller packing numbers for several natural geometric set systems under the shallowness assumption. Letting $d > 1$ be an integer parameter, this includes set systems of points and halfspaces in d -dimensions, balls in d -dimensions, parallel slabs of arbitrary width in d -dimensions, as well as dual set systems defined over $(d - 1)$ -variate (not necessarily continuous or totally defined) functions F of *constant description complexity*. These results are described in detail in a preliminary version of this paper [14, Appendix B].
- (ii) Spanning trees with low total conflict number. This is based on the machinery of Welzl [33] to construct spanning trees of low crossing number (see also [27]). Here the tree spans \mathcal{V} (representing, say, a set of regions defined over n points in d -space), and the “conflict number” of an edge (u, v) is the symmetric difference distance between u and v . See [14, Appendix B] for further details.
- (iii) Geometric discrepancy. Following the previous work of the second author [13], the new bound in Theorem 4 leads to an improved discrepancy bound that is sensitive to the size of the sets in various geometric set systems, including point and halfspaces in d -dimensions, this is mentioned in [14] and described in detail in the preliminary work of the first and the third author [12]. As a consequence, it is shown in [12] how to derive an improved bound on relative (ε, δ) -approximations by adapting the approach in [13]. Last, but not least, it is shown in [12] that the bound in Theorem 4 leads to better bounds on the discrepancy of geometric set systems of low degree, as long as $d_1 = 1$.

References

- 1 P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29(2000):912–953.
- 2 P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. *Discrete Comput. Geom.* (1997).
- 3 P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of Ppoints. *J. ACM*, 51(4):606–635 (2004).
- 4 N. Alon and J. H. Spencer. *The Probabilistic Method*. 2nd Edition, Wiley-Interscience, New York, USA, 2000.
- 5 A. Auger and B. Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*, World Scientific Publishing, 2011.
- 6 N. H. Bshouty, Y. Li, and P. M. Long. Using the doubling dimension to analyze the generalization of learning algorithms. *J. Comput. System Sci.*, 75(6):323–335 (2009).
- 7 T. M. Chan. Dynamic coresets. *Discrete Comput. Geom.*, 42: 469–488 (2009).
- 8 B. Chazelle. A note on Haussler’s packing lemma. Unpublished manuscript, Princeton (1992).
- 9 B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of finite VC-dimension. *Discrete Comput. Geom.*, 4:467–489 (1989).
- 10 K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421 (1989).
- 11 R. M. Dudley. Central limit theorems for empirical measures. *Ann. Probab.*, 6(6):899–1049 (1978).
- 12 K. Dutta and A. Ghosh. Size sensitive packing number for Hamming cube and its consequences. CoRR abs/1412.3922 (2014).
- 13 E. Ezra. A size-sensitive discrepancy bound for set systems of bounded primal shatter dimension. In *Proc. Twenty-Fifth Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 1378–1388 (2014).
- 14 E. Ezra. Shallow Packings in Geometry. CoRR abs/1412.5215 (2014).
- 15 L. Gottlieb, A. Kontorovich, and E. Mossel. VC bounds on the cardinality of nearly orthogonal function classes. *Discrete Math.*, 312(10):1766–1775 (2012).
- 16 S. Har-Peled. *Geometric Approximation Algorithms*, Mathematical Surveys and Monographs, Vol. 173 (2011).
- 17 S. Har-Peled and M. Sharir, Relative (p, ε) -approximations in geometry, *Discrete Comput. Geom.*, 45(3):462–496 (2011).
- 18 D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. In *Information and Computation*, 100(1):78–150 (1992).
- 19 D. Haussler. Sphere packing numbers for subsets of the Boolean n -cube with bounded Vapnik-Chervonenkis dimension. *J. Combinatorial Theory Ser. A*, 69:217–232 (1995).
- 20 D. Haussler, N. Littlestone, M. K. Warmuth. Predicting $\{0, 1\}$ -functions on randomly drawn points. *Information and Computation*, 115(2), 248–292 (1994).
- 21 D. Haussler and E. Welzl. ε -nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151 (1987).
- 22 Y. Li, P. M. Long, and A. Srinivasan. Improved bounds on the sample complexity of learning. *J. Comput. Sys. Sci.*, 62(3):516–527 (2001).
- 23 S. Lovett and R. Meka. Constructive discrepancy minimization by walking on the edges. In *Proc. 53th Annu. IEEE Symp. Found. Comput. Sci.*, 61–67, (2012).
- 24 J. Matoušek, *Lectures on Discrete Geometry*, Springer-Verlag New York (2002).
- 25 J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2:169–186 (1992).

- 26 J. Matoušek. Tight upper bounds for the discrepancy of halfspaces. *Discrete Comput. Geom.*, 13:593–601 (1995).
- 27 J. Matoušek. *Geometric Discrepancy*, Algorithms and Combinatorics, Vol. 18, Springer Verlag, Heidelberg (1999).
- 28 D. Pollard. *Convergence of Stochastic Processes*, Springer-Verlag (1984).
- 29 N. Sauer. On the density of families of sets. *J. Combin. Theory, Ser A*, 13(1): 145–147 (1972).
- 30 M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York (1995).
- 31 S. Shelah. A combinatorial problem, stability and order for models and theories in infinitary languages. *Pacific J. Math.*, 41:247–261 (1972).
- 32 V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Prob. Appl.*, 16(2):264–280 (1971).
- 33 E. Welzl. On spanning trees with low crossing numbers. In *Data Structures and Efficient Algorithms, Final Report on the DFG Special Joint Initiative*, volume 594 of *Lect. Notes in Comp. Sci.*, Springer-Verlag, Heidelberg, pp. 233–249 (1992).

Shortest Path in a Polygon using Sublinear Space*

Sariel Har-Peled

Department of Computer Science, University of Illinois
201 N. Goodwin Avenue, Urbana, IL, 61801, USA
sariel@illinois.edu

Abstract

We resolve an open problem due to Tetsuo Asano, showing how to compute the shortest path in a polygon, given in a read only memory, using sublinear space and subquadratic time. Specifically, given a simple polygon P with n vertices in a read only memory, and additional working memory of size m , the new algorithm computes the shortest path (in P) in $O(n^2/m)$ expected time, assuming $m = O(n/\log^2 n)$. This requires several new tools, which we believe to be of independent interest.

Specifically, we show that violator space problems, an abstraction of low dimensional linear-programming (and LP-type problems), can be solved using constant space and expected linear time, by modifying Seidel’s linear programming algorithm and using pseudo-random sequences.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, I.1.2 Algorithm, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Shortest path, violator spaces, limited space.

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.111

1 Introduction

Space might not be the final frontier in the design of algorithms but it is an important constraint. Of special interest are algorithms that use sublinear space. Such algorithms arise naturally in streaming settings, or when the data set is massive, and only a few passes on the data are desirable. Another such setting is when one has a relatively weak embedded processor with limited high quality memory. For example, flash memory can withstand around 100,000 rewrites before starting to deteriorate. Specifically, imagine a hybrid system having a relatively large flash memory, with significantly smaller RAM. That is to a limited extent the setting in a typical smart-phone¹.

The model. The input is provided in a read only memory, and it is of size n . We have $O(m)$ available space which is a read/write space (i.e., the *work space*). We assume, as usual, that every memory cell is a word, and such a word is large enough to store a number or a pointer. We also assume that the input is given in a reasonable representation². A survey of this computational model and related work is provided in the introduction of Asano *et al.* [1, 2].

The problem. We are given a simple polygon P with n vertices in the plane, and two points $s, t \in P$ – all provided in a read-only memory. We also have $O(m)$ additional read-write memory (i.e., work space). The task is to compute the shortest path from s to t inside P .

* Work on this paper was partially supported by a NSF AF awards CCF-1421231, and CCF-1217462.

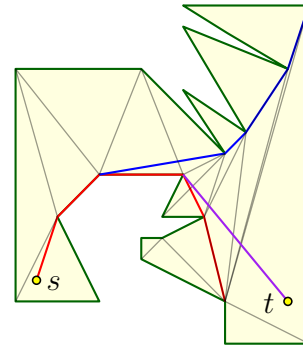
¹ For example, a typical smart-phone in 2014 have 2GB of RAM and 16GB of flash memory. I am sure these numbers would be laughable in a few years. So it goes.

² In some rare cases, the “right” input representation can lead directly to sublinear time algorithms. See the work by Chazelle *et al.* [5].



Asano *et al.* [1] showed how to solve this problem, in $O(n^2/m)$ time, using $O(m)$ space. The catch is that their solution requires quadratic time preprocessing. In a talk by Tetsuo Asano, given in a workshop in honor of his 65th birthday (during SoCG 2014), he posed the open problem of whether this quadratic preprocessing penalty can be avoided. This work provides a positive answer to this question.

If linear space is available. The standard algorithm [17] for computing the shortest path in a polygon, triangulates the polygon, (conceptually) computes the dual graph of the triangulation, which yields a tree, with a unique path between the triangles that contains the source s , and the target t . This path specifies the sequence of diagonals crossed by the shortest path, and it is now relatively easy to walk through this sequence of triangles and maintain the shortest paths from the source to the two endpoints of each diagonal. These paths share a prefix path, and then diverge into two concave chains (known together as a *funnel*). Once arriving to the destination, one computes the unique tangent from the destination t to one of these chains, and the (unique) path, formed by the prefix together with the tangent, defines the shortest path, which can be now extracted in linear time. See figure on the right for illustration.



Sketch of the new algorithm. The basic idea is to decompose the polygon into bigger pieces than triangles. Specifically, we break the polygon into canonical pieces each of size $O(m)$. To this end, we break the given polygon P into $\lceil n/m \rceil$ polygonal chains, each with at most m edges. We refer to such a chain as a *curve*. We next use the notion of *corridor decomposition*, introduced³ by the author [14], to (conceptually) decompose the polygon into canonical pieces (i.e., corridors). Oversimplifying somewhat, each corridor is a polygon having portions of two of the input curves as floor and ceiling, and additional two diagonals of P as gates. It is relatively easy, using constant space and linear time, to figure out for such a diagonal if it separates the source from the destination. Now, start from the corridor containing the source, and figure out which of its two gates the shortest path goes through. We follow this gate to the next corridor and continue in this fashion till we reach the destination. Assuming that computing the next piece can be done in roughly linear time, this algorithm solves the shortest path problem in $O(n^2/m)$ time, as walking through a piece takes (roughly) linear time, and there are $O(n/m)$ pieces the shortest path might go through. (One also needs to keep track of the funnel being constructed during this walk, and prune parts of it away because of space considerations.)

Point-location queries in a canonical decomposition. To implement the above, we need a way to perform a point-location query in the corridor decomposition, without computing it explicitly. Here we are interested in any canonical decomposition that partition the underlying space into cells. Such a partition is induced by a set of objects, and every cell

³ Many somewhat similar decomposition ideas can be found in the literature (for example, the decomposition of a polygon into monotone polygons so that the pieces, and thus the original polygon, can be triangulated [3]). Nevertheless, this specific decomposition scheme [14] is right for our nefarious purposes, but the author would not be surprised if it was known before. Well, at least this footnote is new!

is defined by a constant number of objects. Standard examples of such partitions are (i) vertical decomposition of segments in the plane, or (ii) bottom vertex triangulation of the Voronoi diagram of points in \mathbb{R}^3 . Roughly speaking, any partition that complies with the Clarkson-Shor framework [8] is such a canonical decomposition.

If space and time were not a constraint, we could build the decomposition explicitly. Then a standard point-location query in the history DAG would yield the desired cell. Alternatively, one can perform this point-location query in the history DAG implicitly, without building the DAG before hand, but it is not obvious how to do so with limited space. Surprisingly, at least for the author, this task can be solved using techniques related to low-dimensional linear programming.

Violator spaces. Low dimensional linear programming can be solved in linear time [18]. Sharir and Welzl [23] introduced *LP-type* problems, which are an extension of linear programming. Intuitively, but somewhat incorrectly, one can think about LP-Type algorithms as solving low-dimensional convex programming, although Sharir and Welzl [23] used it to decide in linear time if a set of axis-parallel rectangles can be stabbed by three points (this is quite surprising as this problem has no convex programming flavor). LP-type problems have the same notions as linear programming of bases, and an objective function. The function scores such bases, and the purpose is to find the basis that does not violate any constraint and minimizes (or maximizes) this objective. A natural question is how to solve such problems if there is no scoring function of the bases.

This is captured by the notion of *violator spaces* [20, 21, 10, 11, 4]. The basic idea is that every subset of constraints is mapped to a unique basis, every basis has size at most δ (δ is the dimension of the problem, and is conceptually a constant), and certain conditions on consistency and monotonicity hold. Computing the basis of a violator space is not as easy as solving LP-type problems, because without a clear notion of progress, one can cycle through bases (which is not possible for LP-type problems). See Šavroň [21] for an example of such cycling. Nevertheless, Clarkson's algorithm [7] works for violator spaces [4].

We revisit the violator space framework, and show the following:

- (A) Because of the cycling mentioned above, the standard version of Seidel's linear programming algorithm [22] does not work for violator spaces. However, it turns out that a variant of Seidel's algorithm does work for violator spaces.
- (B) We demonstrate that violator spaces can be used to solve the problem of point-location in canonical decomposition. While in some cases this point-location problem can be stated as LP-type problem, stating it as a violator space problem seems to be more natural and elegant.
- (C) The advantages of Seidel's algorithm is that except for constant work space, the only additional space it needs is to store the random permutations it uses. We show that one can use pseudo-random generators (PRGs) to generate the random permutation, so that there is no need to store it explicitly. This is of course well known – but the previous analysis [19] for linear programming implied only that the expected running time is $O(n \log^{\delta-1} n)$, where δ is the combinatorial dimension. Building on Mulmuley's work [19], we do a somewhat more careful analysis, showing that in this case one can use backward analysis on the random ordering of constraints generated, and as such the expected running time remains linear.

This implies that one can solve violator space problems (and thus, LP and LP-type problems) in constant dimension, using constant space, in expected linear time.

Paper organization. We present the new algorithm for computing the basis of violator spaces in Section 2. The adaptation of the algorithm to work with constant space is described in Section 2.3. We described corridor decomposition and its adaptation to our setting in Section 3. We present the shortest path algorithm in Section 4.

2 Violator spaces and constant space algorithms

First, we review the formal definition of *violator spaces* [20, 21, 10, 11, 4]. We then show that a variant of Seidel’s algorithm for linear programming works for this abstract settings, and show how to adapt it to work with constant space and in expected linear time.

2.1 Formal definition of violator space

Before dwelling into the abstract framework, let us consider the following concrete example – hopefully it would help the reader in keeping track of the abstraction.

► **Example 1.** We have a set H of n segments in the plane, and we would like to compute the vertical trapezoid of $\mathcal{A}^1(H)$ that contains, say, the origin, where $\mathcal{A}^1(H)$ denote the vertical decomposition of the arrangement formed by the segments of H . Specifically, for a subset $X \subseteq H$, let $\tau(X)$ be the vertical trapezoid in $\mathcal{A}^1(X)$ that contains the origin. The vertical trapezoid $\tau(X)$ is defined by at most four segments, which are the *basis* of X . A segment $f \in H$ *violates* $\tau = \tau(X)$, if it intersects the interior of $\tau(X)$. The set of segments of H that intersects the interior of τ , denoted by $\text{cl}(\tau)$ or $\text{cl}(X)$, is the *conflict list* of τ .

Somewhat informally, violator space identifies a vertical trapezoid $\tau = \tau(X)$, by its conflict list $\text{cl}(X)$, and not by its geometric realization (i.e., τ).

► **Definition 2.** A *violator space* is a pair $\mathcal{V} = (H, \text{cl})$, where H is a finite set of *constraints*, and $\text{cl} : 2^H \rightarrow 2^H$ is a function, such that:

- **Consistency:** For all $X \subseteq H$, we have that $\text{cl}(X) \cap X = \emptyset$.
- **Locality:** For all $X \subseteq Y \subseteq H$, if $\text{cl}(X) \cap Y = \emptyset$ then $\text{cl}(X) = \text{cl}(Y)$.
- **Monotonicity:** For all $X \subseteq Y \subseteq Z \subseteq H$, if $\text{cl}(X) = \text{cl}(Z)$ then $\text{cl}(X) = \text{cl}(Y) = \text{cl}(Z)$.

A set $B \subseteq X \subseteq H$ is a *basis* of X , if $\text{cl}(B) = \text{cl}(X)$, and for any proper subset $B' \subset B$, we have that $\text{cl}(B') \neq \text{cl}(B)$. The *combinatorial dimension*, denoted by δ , is the maximum size of a basis.

Note that consistency and locality implies monotonicity. For the sake of concreteness, it would also be convenient to assume the following (this is strictly speaking not necessary for the algorithm).

► **Definition 3.** For any $X \subseteq H$ there is a unique *cell* $\tau(X)$ associated with it, where for any $X, Y \subseteq H$, we have that if $\text{cl}(X) \neq \text{cl}(Y)$ then $\tau(X) \neq \tau(Y)$. Consider any $X \subseteq H$, and any $f \in H$. For $\tau = \tau(X)$, the constraint f *violates* τ if $f \in \text{cl}(X)$ (or alternatively, f *violates* X).

Finally, we assume that the following two basic operations are available:

- **violate**(f, B): Given a basis B (or its cell $\tau = \tau(B)$) and a constraint f , it returns true if f violates τ .

- **compBasis(X)**: Given a set X with at most $(\delta + 1)^2$ constraints, this procedure computes $\text{basis}(X)$, where δ is the combinatorial dimension of the violator space⁴. For δ a constant, we assume that this takes constant time.

2.1.1 Examples

Linear programming as a violator space. Consider an instance I of linear programming in \mathbb{R}^d . Our interpretation is somewhat convoluted, but serves as a preparation for the next example. The instance I induces a polytope P in \mathbb{R}^d , which is the *feasible domain*. The vertices V of the polytope P induce a triangulation (assuming general position) of the sphere of directions, where a direction v belongs to a vertex p , if and only if p is an extreme vertex of P in the direction of v . Now, the objective function of I specifies a direction v_I , and in solving the LP, we are looking for the extreme vertex of P in this direction.

Put differently, every subset H of the constraints of I , defines a triangulation $\mathcal{T}(H)$ of the sphere of directions. So, let the cell of H , denoted by $\tau = \tau(H)$, be the spherical triangle in this decomposition that contains v_I . The basis of H is the subset of constraints that define $\tau(H)$. A constraint f of the LP violates τ if the vertex induced by the basis $\text{basis}(H)$ (in the original space), is on the wrong side of f .

Thus solving the LP instance $I = (H, v_I)$ is no more than performing a point-location query in the spherical triangulation $\mathcal{T}(H)$, for the spherical triangle that contains v_I .

Doing point-location via violator spaces. Example 1 hints to a more general setup. So consider a space decomposition into canonical cells induced by a set of objects. For example, segments in the plane, with the canonical cells being the vertical trapezoids. More generally, consider any decomposition of a domain into simple canonical cells induced by objects, which complies with the Clarkson-Shor framework [8]. Examples of this include point-location in a (i) Delaunay triangulation, (ii) bottom vertex triangulation in an arrangement of hyperplanes, and (iii) many others.

► **Lemma 4.** *Consider a canonical decomposition of a domain into simple cells, induced by a set of objects, that complies with the Clarkson-Shor framework [8]. Then, performing a point-location query in such a domain is equivalent to computing a basis of a violator space.*

Proof. This follows readily from definition, see the full version [15] for details. ◀

It seems that for all of these point-location problems, one can solve them directly as LP-type problems. However, stating these problems as violator space problems is more natural as it avoids the need to explicitly define an artificial ordering over the bases, which can be quite tedious and not immediate.

2.2 The algorithm for computing the basis of a violator space

The input is a violator space $\mathcal{V} = (H, \text{cl})$ with $n = |H|$ constraints, having combinatorial dimension δ .

⁴ We consider $\text{basis}(X)$ to be unique (that is, we assume implicitly that the input is in general position). This can be enforced by using lexicographical ordering, if necessary, among the defining bases always using the lexicographical minimum one.

```

solveVS( $W, X$ ):
   $\langle f_1, \dots, f_m \rangle$ : A random permutation of the constraints of  $X$ .
   $B_0 \leftarrow \text{compBasis}(W)$ 
  for  $i = 1$  to  $m$  do
    if violate( $f_i, B_{i-1}$ ) then
       $B_i \leftarrow \text{solveVS}(W \cup B_{i-1} \cup \{f_i\}, \{f_1, \dots, f_i\})$ 
    else
       $B_i \leftarrow B_{i-1}$ 
  return  $B_m$ 

```

■ **Figure 2.1** The algorithm for solving violator space problems. The parameter W is a set of $O(\delta^2)$ witness constraints, and X is a set of m constraints. The function return $\text{basis}(W \cup X)$. To solve a given violator space, defined implicitly by the set of constraints H , and the functions **violate** and **compBasis**, one calls **solveVS**($\{\}, H$).

2.2.1 Description of the algorithm

The algorithm is a variant of Seidel’s algorithm [22] – it picks a random permutation of the constraints, and computes recursively in a randomized incremental fashion the basis of the solution for the first i constraints. Specifically, if the i th constraint violates the basis B_{i-1} computed for the first $i - 1$ constraints, it calls recursively, adding the constraints of B_{i-1} and the i th constraint to the set of constraints that must be included whenever computing a basis (in the recursive calls). The resulting code is depicted in Figure 2.1.

The only difference with the original algorithm of Seidel, is that the recursive call gets the set $W \cup B_{i-1} \cup \{f_i\}$ instead of $\text{basis}(B_{i-1} \cup \{f_i\})$ (which is a smaller set). This modification is required because of the potential cycling between bases in a violator space.

2.2.2 The analysis

The key observation is that the depth of the recursion of **solveVS** is bounded by δ , where δ is the combinatorial dimension of the violator space. Indeed, if f_i violates a basis, the constraints added to the witness set W guarantee that any subsequent basis computed in the recursive call contains f_i , as testified by the following lemma.

► **Lemma 5.** *Consider any set $X \subseteq H$. Let $B = \text{basis}(X)$, and let f be a constraint in $H \setminus X$ that violates B . Then, for any subset Y such that $B \cup \{f\} \subseteq Y \subseteq X \cup \{f\}$, we have that $f \in \text{basis}(Y)$.*

Proof. Assume that this is false, and let Y be the bad set with $B' = \text{basis}(Y)$, such that $f \notin B'$. Since $f \in Y$, by consistency, $f \notin \text{cl}(Y)$, see Definition 2. By definition $\text{cl}(Y) = \text{cl}(B')$, which implies that $f \notin \text{cl}(B')$; that is, f does not violate B' .

Now, by monotonicity, we have $\text{cl}(Y) = \text{cl}(Y \setminus \{f\}) = \text{cl}(B')$. By assumption, $B \subseteq Y \setminus \{f\}$, which implies, again by monotonicity, as $B \subseteq Y \setminus \{f\} \subseteq X$, that $\text{cl}(X) = \text{cl}(Y \setminus \{f\}) = \text{cl}(B)$, as $B = \text{basis}(X)$. But that implies that $\text{cl}(B) = \text{cl}(Y) = \text{cl}(B')$. As $f \notin \text{cl}(Y)$, this implies that f does not violate B , which is a contradiction. ◀

► **Lemma 6.** *The depth of the recursion of **solveVS**, see Figure 2.1_{p116}, is at most δ , where δ is the combinatorial dimension of the given instance.*

Proof. Consider a sequence of k recursive calls, with $W_0 \subseteq W_1 \subseteq W_2 \subseteq \dots \subseteq W_k$ as the different values of the parameter W of **solveVS**, where $W_0 = \emptyset$ is the value in the top-level call. Let f'_i , for $i = 1, \dots, k$, be the constraint whose violation triggered the i th level call.

Observe that $f'_i \in W_i$, and as such all these constraints must be distinct (by consistency). Furthermore, we also included the basis B'_i , that f'_i violates, in the witness set W_i , which implies, by Lemma 5, that in any basis computation done inside this recursive call, it must be that $f'_i \in \text{basis}(W_j)$, for any $j \geq i$. As such, we have $f'_1, \dots, f'_k \in \text{basis}(W_k)$. Since a basis can have at most δ elements, this is possible only if $k \leq \delta$, as claimed. \blacktriangleleft

► **Theorem 7.** *Given an instance of violator space $\mathcal{V} = (H, \text{cl})$ with n constraints, and combinatorial dimension δ , the algorithm $\text{solveVS}(\emptyset, H)$, see Figure 2.1, computes $\text{basis}(H)$. The expected number of violation tests performed is bounded by $O(\delta^\delta n)$. Furthermore, the algorithm performs in expectation $O((\delta \ln n)^\delta)$ basis computations (on sets of constraints that contain at most $\delta(\delta + 1)$ constraints).*

In particular, for constant combinatorial dimension δ , with violation test and basis computation that takes constant time, this algorithm runs in $O(n)$ expected time.

See the full version [15] for the proof of the above theorem.

2.3 Solving violator space problem with constant space and linear time

The key observation for turning solveVS into an algorithm that uses little space, is observing that the only thing we need to store (implicitly) is the random permutation used by solveVS .

2.3.1 Generating a random permutation using pseudo-random generators

To avoid storing the permutation, one can use pseudo-random techniques to compute the permutation on the fly. For our algorithm, we do not need a permutation - any random sequence that has uniform distribution over the constraints and is sufficiently long, would work.

► **Lemma 8.** *For any integer $\phi > 0$, a prime integer n , and an integer constant $c' \geq 12$, one can compute a random sequence of numbers $X_1, \dots, X_{c'n} \in \llbracket n \rrbracket = \{1, \dots, n\}$, such that:*

(A) *The probability of $X_i = j$ is $1/n$, for any $i \in \llbracket n \rrbracket$ and $j \in \llbracket c'n \rrbracket$.*

(B) *The sequence is ϕ -wise independent.*

(C) *Using $O(c'\phi)$ space, given an index i , one can compute X_i in $O(\phi)$ time.*

Proof. This is a standard pseudo-random generator (PRG) technique, described in detail by Mulmuley [19, p. 399]. We outline the idea. Randomly pick ϕ coefficients $\alpha_0, \dots, \alpha_\phi \in \{0, \dots, n-1\}$ (uniformly and independently), and consider the random polynomial $f_1(x) = \sum_{i=0}^{\phi} \alpha_i x^i$, and set $p(x) = (f(x) \bmod n)$. Now, set $X_i = 1 + p(i)$, for $i = 1, \dots, n$. It is easy to verify that the desired properties hold. To extent this sequence to be of the desired length, pick randomly c' such polynomials, and append their sequence together to get the desired longer sequence. It is easy to verify that the longer sequence is still ϕ -wise independent. \blacktriangleleft

The following lemma testifies that this PRG sequence, with good probability, contains the desired basis (as such, conceptually, we can think about it as being a permutation of $\llbracket n \rrbracket$).

► **Lemma 9.** *Let $B \subseteq \llbracket n \rrbracket$ be a specific set of δ numbers. For any integer $\phi \geq 8 + 2\delta$ and consider ϕ -wise independent random sequence of numbers $\mathcal{X} = \langle X_1, \dots, X_{c'n} \rangle$, each uniformly distributed in $\llbracket n \rrbracket$, where c' is any constant $\geq 4(5 + \lceil \ln \delta \rceil)^2$. Then, the probability that the elements of B do not appear in \mathcal{X} is bounded by, say, $1/20$.*

See the full version [15] for the proof of the above lemma.

► **Remark.** There are several low level technicalities that one needs to address in using such a PRG sequence instead of a truly random permutation:

- (A) *Repeated numbers are not a problem:* the algorithm `solveVS` (see Figure 2.1_{p116}) ignores a constraint that is being inserted for the second time, since it can not violate the current basis.
- (B) *Verifying the solution:* The sequence (of the indices) of the constraints used by the algorithm would be first $X_1, \dots, X_{c'n}$. This sequence might miss some constraints that violates the computed solution.
As such, in the second stage, the algorithm check if any of the constraints $1, 2, \dots, n$ violates the basis computed. If a violation was found, then the sequence generated failed, and the algorithm restart from scratch – resetting the PRG used in this level, regenerating the random keys used to initialize it, and rerun it to generate a new sequence.
- (C) *Independence between levels:* We will use a different PRG for each level of the recursion of `solveVS`. Specifically, we generate the keys used in the PRG in the beginning of each recursive call. Since the depth of the recursion is δ , that would increase the space requirement by a factor of δ .
- (D) *If the subproblem size is not a prime:* In a recursive call, the number of constraints given (i.e., m) might not be a prime. To this end, the algorithm can store (non-uniformly), a list of primes, such that for any m , there is a prime $m' \geq m$ that is at most twice bigger than m ⁵. Then the algorithm generates the sequence modulo m' , and ignores numbers that are larger than m . This implies that the sequence might contain invalid numbers, but such numbers are only a constant fraction of the sequence, so ignoring them does not change the running time analysis of our algorithm. (More precisely, this might cause the running time of the algorithm to deteriorate by a factor of $\exp(O(\delta))$, but as we consider δ to be a constant, this does not effect our analysis.)

One needs now to prove that backward analysis still works for our algorithm for violator spaces. The proof of the following lemma is implied by a careful tweaking of Mulmuley's analysis – we provide the details in the full version of the paper [15].

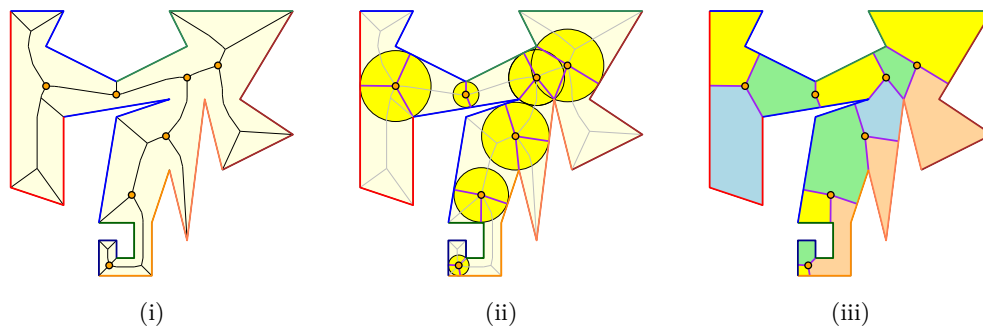
► **Lemma 10** (See [15]). *Consider a violator space $\mathcal{V} = (H, \text{cl})$ with $n = |H|$, and combinatorial dimension δ . Let $i > 2\delta$, and let $\mathcal{X} = X_1, \dots, X_{c'n}$ be a random sequence of constraints of H generated by ϕ -wise independent distribution (with each X_i having a uniform distribution), where $\phi > 6\delta + 9$ and $c' \geq 4(5 + \lceil \ln \delta \rceil)^2$ are constants. Then, for $i > 2\delta$, the probability that X_i violates $B = \text{basis}(X_1, \dots, X_{i-1})$ is $O(1/i)$.*

2.3.2 The result

► **Theorem 11.** *Given an instance of violator space $\mathcal{V} = (H, \text{cl})$ with n constraints, and combinatorial dimension δ , one can compute $\text{basis}(H)$ using $O(\delta^2 \log^2 \delta)$ space. For some constant $\zeta = O(\delta \log^2 \delta)$, we have that:*

- (A) *The expected number of basis computations is $O((\zeta \ln n)^\delta)$, each done over $O(\delta^2)$ constraints.*
- (B) *The expected number of violation tests performed is $O(\zeta^\delta n)$.*
- (C) *The expected running time (ignoring the time to do the above operations) is $O(\zeta^\delta n)$.*

⁵ That is, the program hard codes a list of such primes. The author wrote a program to compute such a list of primes, and used it to compute 50 primes that cover the range all the way to 10^{15} (the program run in a few seconds). However, it seems a bit redundant to include a list of such primes here. The interested reader can have a look here: <http://sarielhp.org/blog/?p=8700>.



■ **Figure 3.1** An example of a corridor decomposition for a polygon: (i) Input curves, medial-axis and active vertices, (ii) their critical circles, and their spokes, and (iii) the resulting corridor decomposition.

Proof. The algorithm is described above. As for the analysis, it follows readily by plugging Lemma 10 into the proof of Theorem 7.

The only non-trivial technicality is to handle the case that the PRG sequence fails to contain the basis. Formally, abusing notations somewhat, consider a recursive call on the constraints indexed by $\llbracket n \rrbracket$, and let B be the desired basis of the given subproblem. By Lemma 9, the probability that B is not contained in the generated PRG is bounded by $1/20$ – and in such a case the sequence has to be regenerated till success. As such, in expectation, this has a penalty factor of (say) 2 on the running time in each level. Overall, the analysis holds with the constants deteriorating by a factor of (at most) 2^δ . ◀

► **Remark.** Note, that the above pseudo-random generator technique is well known, but using it for linear programming by itself does not make too much sense. Indeed, pseudo-random generators are sometimes used as a way to reduce the randomness consumed by an algorithm. That in turn is used to derandomize the algorithm. However, for linear programming Megiddo’s original algorithm was already linear time deterministic. Furthermore, Chazelle and Matoušek [6], using different techniques showed that one can even derandomize Clarkson’s algorithm and get a linear running time with a better constant.

Similarly, using PRGs to reduce space of algorithms is by now a standard technique in streaming, see for example the work by Indyk [16], and references therein.

3 Corridor decomposition

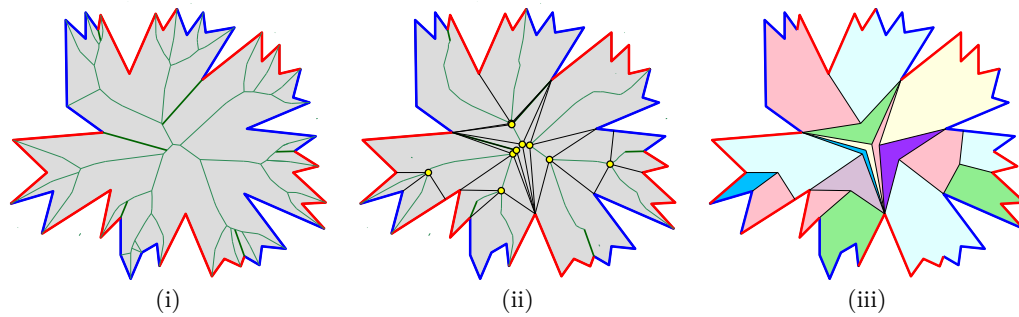
3.1 Construction

The decomposition here is similar to the decomposition described by the author in a recent work [14].

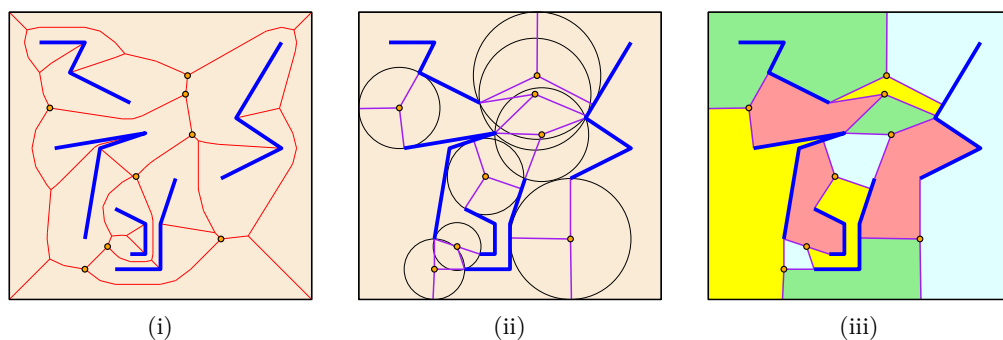
► **Definition 12** (Breaking a polygon into curves). Let the polygon P have the vertices v_1, \dots, v_n in counterclockwise order along its boundary. Let σ_i be the polygonal curve having the vertices $v_{(i-1)m+1}, v_{(i-1)m+2}, v_{(i-1)m+3}, \dots, v_{im+1}$, for $i = 1, \dots, \bar{n} - 1$, where

$$\bar{n} = \lfloor (n-1)/m \rfloor + 1.$$

The last polygonal curve is $\sigma_{\bar{n}} = v_{(\bar{n}-1)m+1}, v_{(\bar{n}-1)m+2}, \dots, v_n, v_1$. Note, that given P in a read only memory, one can encode any curve σ_i using $O(1)$ space. Let $\Gamma = \{\sigma_1, \dots, \sigma_{\bar{n}}\}$ be the resulting set of polygonal curves. From this point on, a *curve* refers to a polygonal curve generated by this process.



■ **Figure 3.2** Another example of a corridor decomposition for a polygon: (i) Input polygon and its curves and its medial-axis (the thick lines are the angle bisectors for the obtuse angles where two curves meet), (ii) active vertices and their spokes (with a reduced medial axis), and (iii) the resulting corridor decomposition.



■ **Figure 3.3** Corridor decomposition for disjoint curves: (i) Input curves, the medial-axis, and active vertices, (ii) the critical circles, and their spokes, and (iii) the resulting corridor decomposition.

Corridor decomposition for the whole polygon. Next, consider the medial axis of P restricted to the interior of P . A vertex v of the medial axis corresponds to a disk D , that touches the boundary of P in two or three points (by general position assumption, not in any larger number of points). The medial axis has the topological structure of a tree.

To make things somewhat cleaner, we pretend that there is a little hole centered at every vertex of the polygon if it is the common endpoint of two curves. This results in a medial axis edge that comes out of the vertex as an angle bisector, both for an acute angle (where a medial-axis edge already exists), and for obtuse angles, see Figure 3.1 and Figure 3.2.

A vertex of the medial axis is *active* if its disk touches three different curves of Γ . It is easy to verify that there are $O(\bar{n})$ active vertices. The segments connecting an active vertex to the three (or two) points of tangency of its empty disk with the boundary of P are its *spokes*. Introducing these spokes breaks the polygon into the desired *corridors*.

Corridor decomposition for a subset of the curves. For a subset $\Psi \subseteq \Gamma$, of total complexity t , one can apply a similar construction. Again, compute the medial axis of the curves of Ψ , by computing, in $O(t \log t)$ time, the Voronoi diagram of the segments used by the curves [9], and extracting the medial axis (it is now a planar graph instead of a tree). Again, by considering the active vertices, building their associated spokes, results in a decomposition into corridors. For technical reasons, it is convenient to add a large bounding box, and restrict the construction to this domain, treating this *frame* as yet another input curve. Figure 3.3 depicts one such corridor decomposition.

Let $\mathcal{C}(\Psi)$ denote this resulting decomposition into corridors.

3.1.1 Properties of the resulting decomposition

Every corridor in the resulting decomposition $\mathcal{C}(\Psi)$ is defined by a constant number of input curves. Specifically, consider the set of all possible corridors; that is $\mathcal{F} = \bigcup_{\Upsilon \subseteq \Gamma} \mathcal{C}(\Upsilon)$. Next, consider any corridor $C \in \mathcal{F}$, then there is a unique *defining set* $D(C) \subseteq \Psi$ (of at most 4 curves). Similarly, such a corridor has *stopping set* (or *conflict list*) of C , denoted by $K(C)$.

Consider any subset $\mathcal{S} \subseteq \Gamma$. It is easy to verify that the following two conditions hold:

- (i) For any $C \in \mathcal{C}(\mathcal{S})$, we have $D(C) \subseteq \mathcal{S}$ and $\mathcal{S} \cap K(C) = \emptyset$.
- (ii) If $D(C) \subseteq \mathcal{S}$ and $K(C) \cap \mathcal{S} = \emptyset$, then $C \in \mathcal{C}(\mathcal{S})$.

Namely, the corridor decomposition complies with the technique of Clarkson-Shor [8] (see also [13, Chapter 8]).

3.2 Computing a specific corridor

Let \mathbf{p} be a point in the plane, and let Γ be a set of \bar{n} interior disjoint curves (stored in a read only memory), where each curve is of complexity m . Let n be the total complexity of these curves (we assume that $n = \Theta(m\bar{n})$). Our purpose here is to compute the corridor $C \in \mathcal{C}(\Gamma)$ that contains \mathbf{p} . Formally, for a subset $\Psi \subseteq \Gamma$, we define the function $w(\Psi)$, to be the defining set of the corridor $C \in \mathcal{C}(\Psi)$ that contains \mathbf{p} . Note, that such a defining set has cardinality at most $\delta = 4$.

Basic operations. We need to specify how to implement the two basic operations:

- (A) **(Basis computation)** Given a set of $O(1)$ curves, we compute their medial axis, and extract the corridor containing \mathbf{p} . This takes $O(m \log m)$ time.
- (B) **(Violation test)** Given a corridor C , and a curve σ , both of complexity $O(m)$, we can check if σ violates the corridor by checking if an arbitrary vertex of σ is contained in C (this takes $O(m)$ time to check), and then check in $O(m)$ time, if any segment of σ intersects the doors of the corridor on its two sides. This takes $O(m)$ time.

► **Lemma 13.** *Given a polygon P with n vertices, stored in read only memory, and let m be a parameter. Let Γ be the set of \bar{n} curves resulting from breaking P into polygonal curves each with m vertices, as described in Definition 12. Then, given a query point \mathbf{p} inside P , one can compute, in $O(n + m \log m \log^4 \bar{n})$ expected time, the corridor of $\mathcal{C}(\Gamma)$ that contains \mathbf{p} . This algorithm uses $O(1)$ additional space.*

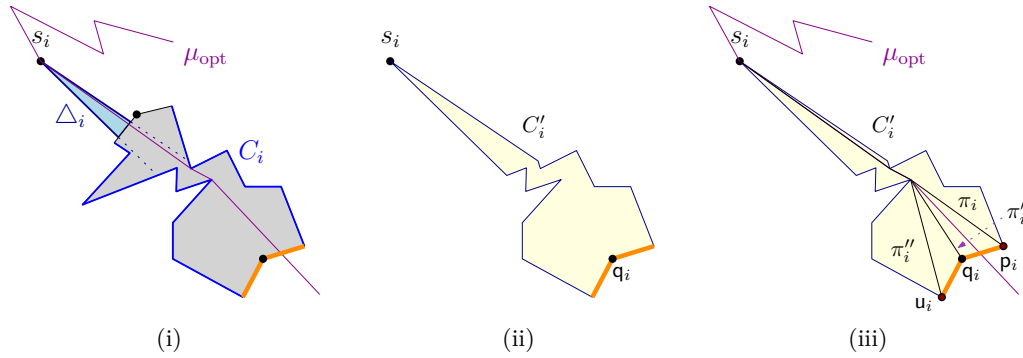
See the full version [15] for the proof of the above lemma.

4 Shortest path in a polygon in sublinear space

Let P be a simple polygon with n edges in the plane, and let s and t be two points in P , where s is the *source*, and t is the *target*. Our purpose here is to compute the shortest path between s and t inside P . The vertices of P are stored in (say) counterclockwise order in an array stored in a read only memory. Let m be a prespecified parameter that is (roughly) the amount of additional space available for the algorithm.

4.1 Updating the shortest path through a corridor

A corridor has two *doors* – a door is made of two segments, with a middle endpoint in the interior of the polygon, and the other endpoints on the boundary of the polygon. The rest of the boundary of the corridor is made out two chains from the original polygon.



■ **Figure 4.1** (i) The state in the beginning of the i th iteration. (ii) The clipped polygon C'_i . (iii) The funnel created by the shortest paths from s_i to the two spoke endpoints.

Given two rays σ and σ' , that share their source vertex v (which lies inside P), consider the polygon Q that starts at v , follows the ray σ till it hits the boundary of P , then trace the boundary of P in a counterclockwise direction till the first intersection of σ' with ∂P , and then back to v . The polygon $Q = P\langle\sigma, \sigma'\rangle$ is the *clipped* polygon. See Figure 4.1.

A *geodesic* is the shortest path between two points (restricted to lie inside P). Two geodesics might have a common intersection, but they can cross at most once. Locally, inside a polygon, a geodesic is a straight segment. For our algorithm, we need some basic operations:

- (A) **isPntIn**(p): Given a query point p , it decides if p is inside P . This is done by scanning the edges of P one by one, and counting how many edges crosses the vertical ray shooting from p downward. This operation takes linear time (in the number of vertices of P).
- (B) **isInSubPoly**(p, σ, σ'): returns true if p is in the clipped polygon $P\langle\sigma, \sigma'\rangle$. It is easy to verify that this can be implemented to work in linear time and constant space.

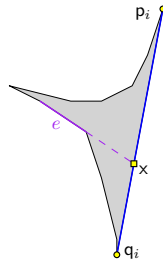
Using vertical and horizontal rays shot from s , one can decide, in $O(n)$ time, which quadrant around s is locally used by the shortest path from s to t . Assume that this path is in the positive quadrant. It would be useful to think about geodesics starting at s as being sorted angularly. Specifically, if τ and τ' are two geodesic starting at s , then τ is to the *left* of τ' , if the first edge of τ is counterclockwise to the first edge of τ' . If the prefix of τ and τ' is non-empty, we apply the same test to the last common point of the two paths. Let $\tau \prec \tau'$ denote that τ is to the left of τ' .

In particular, if the endpoint of the rays σ, σ' is the source vertex s , and the geodesic between s and t lies in $P\langle\sigma, \sigma'\rangle$, then given a third ray π lying between σ and σ' , the shortest path between s and t in P must lie completely either in $P\langle\sigma, \pi\rangle$ or $P\langle\pi, \sigma'\rangle$, and this can be tested by a single call to **isInSubPoly** for checking if t is in $P\langle\pi, \sigma'\rangle$.

4.1.1 Limiting the search space

► **Lemma 14.** *Let P , s and t be as above, and μ be the shortest path from s to t in P . Let pq be the last edge in the shortest path τ from s to q , where q is in P . Then, one can decide in $O(n)$ time, and using $O(1)$ space, if $\mu \prec \tau$, where n is the number of vertices of P .*

See the full version [15] for the proof of the above lemma.



■ **Figure 4.2** Funnel reduction.

4.1.2 Walking through a corridor

In the beginning of the i th iteration of the algorithm it would maintain the following quantities (depicted in Figure 4.1 (i)):

- (A) s_i : the current source (it lies on the optimal shortest path μ_{opt} between s and t).
- (B) C_i : The current corridor.
- (C) Δ_i : A triangle having s_i as one of its vertices, and its two other vertices lie on a spoke of C_i . The shortest path μ_{opt} passes through s_i , and enters C_i through the base of Δ_i , and then exists the corridor through one of its “exit” spokes.

The task at hand is to trace the shortest path through C_i , in order to compute where the shortest path leaves the corridor.

► **Lemma 15.** *Tracing the shortest path μ_{opt} through a single corridor takes $O(n \log m + m \log m \log^4 \bar{n})$ expected time, using $O(m)$ space.*

Proof. We use the above notation. The algorithm glues together Δ_i to C_i to get a new polygon. Next, it clips the new polygon by extending the two edges of Δ_i from s_i . Let C'_i denote the resulting polygon, depicted in Figure 4.1 (ii). Let the three vertices of C'_i forming the two “exit” spokes be p_i, q_i, u_i . Next, the algorithm computes the shortest path from s_i to the three vertices p_i, q_i, u_i inside C'_i , and let π_i, π'_i, π''_i be these paths, respectively (this takes $O(|C'_i|) = O(m)$ time [12]). Using Lemma 14 the algorithm decides if $\pi_i \prec \mu_{\text{opt}} \prec \pi'_i$ or $\pi'_i \prec \mu_{\text{opt}} \prec \pi''_i$. We refer to a prefix path (that is part of the desired shortest path) followed by the two concave chains as a *funnel* – see Figure 4.1 (iii) and Figure 4.2 for an example.

Assume that $\pi_i \prec \mu_{\text{opt}} \prec \pi'_i$, and let F_i be the funnel created by these two shortest paths, where $p_i q_i$ is the base of the funnel. If the space bounded by the funnel is a triangle, then the algorithm sets its top vertex as s_{i+1} , the funnel triangle is Δ_{i+1} , and the algorithm computes the corridor on the other side of $p_i q_i$ using the algorithm of Lemma 13, set it as C_{i+1} , and continues the execution of the algorithm to the next iteration.

So the problem is when funnel chains are “complicated” concave polygons (with at most $O(m)$ vertices), see Figure 4.2. As long as the funnel F_i is not a triangle, pick a middle edge e on one side of the funnel, and extend it till it hits the edge $p_i p_{i+1}$, at a point x . This breaks F_i into two funnels, and using the algorithm of Lemma 14 on the edge e , decide which of these two funnels contains the shortest path μ_{opt} , and replace F_i by this funnel. Repeat this process till F_i becomes a triangle. Once this happens, the algorithm continues to the next iteration as described above. Clearly, this funnel “reduction” requires $O(\log m)$ calls to the algorithm of Lemma 14.

Note, that the algorithm “forgets” the portion of the funnel that is common to both paths as it moves from C_i to C_{i+1} . This polygonal path is a part of the shortest path μ_{opt}

computed by the algorithm, and it can be output at this stage, before moving to the next corridor C_{i+1} .

In the end of the iteration, this algorithm computes the next corridor C_{i+1} by calling the algorithm of Lemma 13. ◀

4.2 The algorithm

The overall algorithm works by first computing the corridor C_1 containing the source $s_1 = s$ using Lemma 13. The algorithm now iteratively applies Lemma 15 till arriving to the corridor containing t , where the remaining shortest path can be readily computed. Since every corridor gets visited only once by this walk, we get the following result.

► **Theorem 16.** *Given a simple polygon P with n vertices (stored in a read only memory), a start vertex s , a target vertex t , and a space parameter m , one can compute the length of the shortest path from s to t (and output it), using $O(m)$ additional space, in $O(n^2/m)$ expected time, if $m = O(n/\log^2 n)$. Otherwise, it is $O\left(\frac{n^2}{m} + n \log m \log^4 \bar{n}\right)$.*

Proof. The algorithm is described above, and let $\bar{n} = \lfloor (n-1)/m \rfloor + 1$. There are $O(\bar{n})$ corridors, and this bounds the number of iterations of the algorithm. As such, the overall expected running time is $O(\bar{n}(n \log m + m \log m \log^4 \bar{n})) = O\left(\frac{n^2}{m} \log m + n \log m \log^4 \bar{n}\right)$.

To get a better running time, observe that the extra log factor (on the first term), is rising out of the funnel reduction $O(\log m)$ queries inside each corridor, done in the algorithm of Lemma 15. If instead of reducing a funnel all the way to constant size, we reduce it to have say, at most $\lceil m/4 \rceil$ edges (triggered by the event that the funnel has at least $m/2$ edges), then at each invocation of Lemma 15, only a constant number of such queries would be performed. One has to adapt the algorithm such that instead of a triangle entering a new corridor, it is a funnel. The adaptation is straightforward, and we omit the easy details. The improved running time is $O\left(\frac{n^2}{m} + n \log m \log^4 \bar{n}\right)$. ◀

5 Conclusions

The most interesting open problem remaining from our work, is whether one can improve the running time for computing the shortest path in a polygon with $O(m)$ space to be faster than $O(n^2/m)$.

Acknowledgments. The author became aware of the low-space shortest path problem during Tetsuo Asano talk in the Workshop in honor of his 65th birthday during SoCG 2014. The author thanks him for the talk, and the subsequent discussions. The author also thanks Pankaj Agarwal, Chandra Chekuri, Jeff Erickson and Bernd Gärtner for useful discussions. The authors also thanks the anonymous referees for their detailed comments, and their patience with the numerous typos in the submitted version.

References

- 1 T. Asano, K. Buchin, M. Buchin, M. Korman, W. Mulzer, G. Rote, and A. Schulz. Memory-constrained algorithms for simple polygons. *Comput. Geom. Theory Appl.*, 46(8):959–969, 2013.
- 2 T. Asano, K. Buchin, M. Buchin, M. Korman, W. Mulzer, G. Rote, and A. Schulz. Reprint of: Memory-constrained algorithms for simple polygons. *Comput. Geom. Theory Appl.*, 47(3):469–479, 2014.

- 3 M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Santa Clara, CA, USA, 3rd edition, 2008.
- 4 Y. Brise and B. Gärtner. Clarkson’s algorithm for violator spaces. *Comput. Geom. Theory Appl.*, 44(2):70–81, 2011.
- 5 B. Chazelle, D. Liu, and A. Magen. Sublinear geometric algorithms. *SIAM J. Comput.*, 35(3):627–646, 2005.
- 6 B. Chazelle and J. Matoušek. On linear-time deterministic algorithms for optimization problems in fixed dimension. *J. Algorithms*, 21:579–597, 1996.
- 7 K. L. Clarkson. Las Vegas algorithms for linear and integer programming. *J. Assoc. Comput. Mach.*, 42:488–499, 1995.
- 8 K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- 9 S. J. Fortune. A sweepline algorithm for Voronoi diagrams. *Algorithmica*, 2:153–174, 1987.
- 10 B. Gärtner, J. Matoušek, L. Rüst, and P. Šavroň. Violator spaces: Structure and algorithms. In *Proc. 14th Annu. European Sympos. Algorithms (ESA)*, pages 387–398, 2006.
- 11 B. Gärtner, J. Matoušek, L. Rüst, and P. Šavroň. Violator spaces: Structure and algorithms. *Discrete Appl. Math.*, 156(11):2124–2141, 2008.
- 12 L. J. Guibas and J. Hershberger. Optimal shortest path queries in a simple polygon. *J. Comput. Syst. Sci.*, 39(2):126–152, October 1989.
- 13 S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Mathematical Surveys and Monographs*. Amer. Math. Soc., Boston, MA, USA, 2011.
- 14 S. Har-Peled. Quasi-polynomial time approximation scheme for sparse subsets of polygons. In *Proc. 30th Annu. Sympos. Comput. Geom. (SoCG)*, pages 120–129, 2014.
- 15 S. Har-Peled. Shortest path in a polygon using sublinear space. *CoRR*, abs/1412.0779, 2014.
- 16 P. Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *J. Assoc. Comput. Mach.*, 53(3):307–323, 2006.
- 17 D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.
- 18 N. Megiddo. Linear programming in linear time when the dimension is fixed. *J. Assoc. Comput. Mach.*, 31:114–127, 1984.
- 19 K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1994.
- 20 L. Y. Rüst. *The P-Matrix Linear Complementarity Problem – Generalizations and Specializations*. PhD thesis, ETH, 2007. Diss. ETH No. 17387.
- 21 P. Šavroň. *Abstract models of optimization problems*. PhD thesis, Charles University, 2007. <http://kam.mff.cuni.cz/~xofon/thesis/diplomka.pdf>.
- 22 R. Seidel. Small-dimensional linear programming and convex hulls made easy. *Discrete Comput. Geom.*, 6:423–434, 1991.
- 23 M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. Theoret. Aspects Comput. Sci.*, volume 577 of *Lect. Notes in Comp. Sci.*, pages 569–579, London, UK, 1992. Springer-Verlag.

Optimal Morphs of Convex Drawings*

Patrizio Angelini¹, Giordano Da Lozzo¹, Fabrizio Frati¹,
Anna Lubiw², Maurizio Patrignani¹, and Vincenzo Roselli¹

- 1 Department of Engineering, Roma Tre University, Italy
{angelini,dalozzo,frati,patrigna,roselli}@dia.uniroma3.it
- 2 Cheriton School of Computer Science, University of Waterloo, Canada
alubiw@uwaterloo.ca

Abstract

We give an algorithm to compute a morph between any two convex drawings of the same plane graph. The morph preserves the convexity of the drawing at any time instant and moves each vertex along a piecewise linear curve with linear complexity. The linear bound is asymptotically optimal in the worst case.

1998 ACM Subject Classification G.2.2. Graph Theory

Keywords and phrases Convex Drawings, Planar Graphs, Morphing, Geometric Representations

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.126

1 Introduction

Convex drawings of plane graphs are a classical topic of investigation in geometric graph theory. A characterization [25] of the plane graphs that admit convex drawings and a linear-time algorithm [10] to test whether a graph admits a convex drawing are known. Convex drawings in small area [5, 8, 11], orthogonal convex drawings [18, 19, 25], and convex drawings satisfying further geometric constraints [16, 17] have also been studied. It is intuitive, but far from trivial to prove, that the space of the convex drawings of any n -vertex plane graph G is connected; i.e., the points in \mathbb{R}^{2n} , each corresponding to the two-dimensional coordinates of a convex drawing of G , form a connected set. Expressed in yet another way, there exists a *convex morph* between any two convex drawings Γ_s and Γ_t of the same plane graph G , that is, a continuous deformation from Γ_s to Γ_t so that the intermediate drawing of G is convex at any instant of the deformation. The main result of this paper is the existence of a convex morph between any two convex drawings of the same plane graph such that each vertex moves along a piecewise linear curve with linear complexity during the deformation.

The existence of a convex morph between any two convex drawings of the same plane graph was first proved by Thomassen [24] more than 30 years ago. His result confirmed a conjecture of Grünbaum and Shepard [15] and improved upon a result of Cairns [9], stating that there exists a continuous deformation, called a *morph*, between any two straight-line planar drawings of the same plane graph such that any intermediate straight-line drawing is planar. More recently, motivated by applications in computer graphics, animation, and modeling, a number of algorithms for morphing graph drawings have been designed [12, 13, 14, 21, 22]. These algorithms aim to construct morphs that preserve the topology of the given drawings at any time, while guaranteeing that the trajectories of the vertices are “nice” curves.

* Work partially supported by MIUR project AMANDA “Algorithmics for MAssive and Networked DAta”, prot. 2012C4E3KT_001, and by NSERC of Canada. Because of space limitations some proofs are only sketched here; complete proofs will be found in the full version of the paper.



Straight-line segments are undoubtedly the most readable and appealing curves for the vertex trajectories. However, *linear morphs* – morphs in which the vertices move along straight lines – do not always exist [12]. A natural way to overcome this problem is to allow vertices to move along piecewise linear curves. Since trajectories of large complexity would have a dramatically detrimental impact on the readability of the morph, an important goal is to minimize the complexity of these curves. This problem is formalized as follows. Let Γ_s and Γ_t be two planar straight-line drawings of a plane graph G . Find a sequence $\Gamma_s = \Gamma_1, \dots, \Gamma_k = \Gamma_t$ of planar straight-line drawings of G such that, for $1 \leq i \leq k - 1$, the linear morph transforming Γ_i into Γ_{i+1} , called a *morphing step*, is planar and k is small.

The first polynomial upper bound for this problem was recently obtained by Alamdari *et al.* [1]. The authors proved that a morph between any two planar straight-line drawings of the same n -vertex connected plane graph exists with $O(n^4)$ morphing steps. The $O(n^4)$ bound was later improved to $O(n^2)$ [4] and then to a worst-case optimal $O(n)$ bound by Angelini *et al.* [3]. The algorithm of Angelini *et al.* [3] can be extended to work for disconnected graphs at the expense of an increase in the number of steps to $O(n^{1.5})$ [2].

In this paper we give an algorithm to construct a convex morph between any two convex drawings of the same n -vertex plane graph with $O(n)$ morphing steps. Our algorithm preserves the convexity of the drawing at any time instant and in fact preserves strict convexity, if the given drawings are strictly-convex. The linear bound is tight in the worst case, as can be shown by adapting the lower bound construction of Angelini *et al.* [3]. We remark that Thomassen's algorithm [24] constructs convex morphs with an exponential number of steps. To the best of our knowledge, no other algorithm is known to construct a convex morph between any two convex drawings of the same plane graph.

The outline of our algorithm is simple. Let Γ_s and Γ_t be two convex drawings of the same *convex graph* G , that is, a plane graph that admits a convex drawing. Determine a connected subgraph G' of G such that removing G' from G results in a smaller convex graph G'' . Then G' lies inside one face f of G'' . Morph Γ_s into a drawing Γ'_s of G and morph Γ_t into a drawing Γ'_t of G such that the cycle of G corresponding to f is delimited by a convex polygon in Γ'_s and in Γ'_t . These morphs consist of one morphing step each. Remove G' from Γ'_s and Γ'_t to obtain two convex drawings Γ''_s and Γ''_t of G'' . Finally, recursively compute a morph between Γ''_s and Γ''_t . Since f remains convex throughout the whole morph from Γ''_s to Γ''_t , a morph of G from Γ'_s to Γ'_t can be obtained from the morph of G'' from Γ''_s to Γ''_t by suitably drawing G' inside f at each intermediate step of such a morph. The final morph from Γ_s to Γ_t consists of the morph from Γ_s to Γ'_s followed by the morph from Γ'_s to Γ'_t , and then the reverse of the morph from Γ_t to Γ'_t . Our algorithm has two main ingredients.

The first ingredient is a structural decomposition of convex graphs that generalizes a well-known structural decomposition of triconnected planar graphs due to Barnette and Grünbaum [6]. The latter states that any subdivision of a triconnected planar graph contains a path whose removal results in a subdivision of a smaller triconnected planar graph. For convex graphs we can prove a similar theorem which states, roughly speaking, that any convex graph contains a path, or three paths incident to the same vertex, whose removal results in a smaller convex graph. Our approach is thus based on *removing* a subgraph from the input graph. This differs from the recent papers on morphing graph drawings [1, 3, 4], where the basic operation is to *contract* (i.e. move arbitrarily close) a vertex to a neighbor. One of the difficulties of the previous approach was to determine a trajectory for a contracted vertex inside the moving polygon of its neighbors. By removing a subgraph and forcing the newly formed face to be convex, we avoid this difficulty.

The second ingredient is a relationship between *unidirectional morphs* and level planar drawings of hierarchical graphs, which allows us to compute the above mentioned morphs

between Γ_s and Γ'_s and between Γ_t and Γ'_t with one morphing step. This relationship was first observed by Angelini *et al.* [3]. However, in order to use it in our setting, we need to prove that every strictly-convex graph admits a *strictly-convex* level planar drawing; this strengthens a result of Hong and Nagamochi [16] and might be of independent interest.

We leave open the question whether any two straight-line drawings of the same plane graph G can be morphed so that every intermediate drawing has polynomial *size* (e.g., the ratio between the length of any two edges is polynomial in the size of G during the entire morph). In order to solve this problem positively, our approach seems to be better than previous ones; intuitively, subgraph removals are more suitable than vertex contractions for a morphing algorithm that doesn't blow up the size of the intermediate drawings. Nevertheless, we haven't yet been able to prove that polynomial-size morphs always exist.

2 Definitions and Preliminaries

In this section we give some definitions and preliminaries.

Drawings and Embeddings. A *straight-line planar drawing* Γ of a graph maps vertices to points in the plane and edges to internally disjoint straight-line segments. Drawing Γ partitions the plane into topologically connected regions, called *faces*. The bounded faces are *internal* and the unbounded face is the *outer face*. A vertex (an edge) is *external* if it is incident to the outer face and *internal* otherwise. A vertex x is *convex*, *flat*, or *concave* in an incident face f in Γ , if the angle at x in f is smaller than, equal to, or larger than π radians, respectively. Drawing Γ is *convex* (*strictly-convex*) if for each vertex v and each face f vertex v is incident to, v is either convex or flat (is convex) in f , if f is internal, and v is either concave or flat (is concave) in f , if f is the outer face. A planar drawing determines a clockwise ordering of the edges incident to each vertex. Two planar drawings of a connected planar graph are *equivalent* if they determine the same clockwise orderings and have the same outer face. A *plane embedding* is an equivalence class of planar drawings. A graph with a plane embedding is a *plane graph*. A *convex* (*strictly-convex*) graph is a plane graph that admits a convex (resp. strictly-convex) drawing with the given plane embedding.

Subgraphs and Connectivity. A subgraph G' of a plane graph G is regarded as a plane graph whose plane embedding is obtained from G by removing all the vertices and edges not in G' . We denote by $G - e$ (by $G - S$) the plane graph obtained from G by removing an edge e of G (resp. a set S of vertices and their incident edges).

We denote by $\deg(G, v)$ the degree of a vertex v in a graph G . A graph G is *biconnected* (*triconnected*) if removing any vertex (resp. any two vertices) leaves G connected. A *separation pair* in a graph G is a pair of vertices whose removal disconnects G . A biconnected plane graph G is *internally triconnected* if introducing a new vertex in the outer face of G and connecting it to all the vertices incident to the outer face of G results in a triconnected graph. Thus, internally triconnected plane graphs form a super-class of triconnected plane graphs. A *split component* of a graph G with respect to a separation pair $\{u, v\}$ is either an edge (u, v) or a maximal subgraph G' of G that does not contain edge (u, v) , that contains vertices u and v , and such that $\{u, v\}$ is not a separation pair of G' ; we say that $\{u, v\}$ *determines* the split components with respect to $\{u, v\}$. For an internally triconnected plane graph G , every separation pair $\{u, v\}$ determines two or three split components; further, in the latter case, one of them is an edge (u, v) not incident to the outer face of G .

A *subdivision* G' of a graph G is a graph obtained from G by replacing each edge (u, v) with a path between u and v ; the internal vertices of this path are called *subdivision vertices*. Given a subgraph H of G , the subgraph H' of G' *corresponding to* H is obtained from H by replacing each edge (u, v) with a path with the same number of vertices as in G' .

Convex Graphs. Convex graphs have been thoroughly studied, both combinatorially and algorithmically. Most of the known results about convex graphs are stated in the following setting. The input consists of a plane graph G and a convex polygon P representing the cycle C delimiting the outer face of G . The problem asks whether G admits a convex drawing in which C is represented by P . The known characterizations for this setting imply characterizations and recognition algorithms for the class of convex graphs (with no constraint on the representation of the cycle delimiting the outer face). Quite surprisingly, the literature seems to lack explicit statements of the characterizations in this unconstrained setting. Here we present two theorems, whose proofs can be easily derived from known results [10, 16, 25].

► **Theorem 1.** *A plane graph is convex if and only if it is a subdivision of an internally triconnected plane graph.*

► **Theorem 2.** *A plane graph is strictly-convex if and only if it is a subdivision of an internally triconnected plane graph and every degree-2 vertex is external.*

Monotonicity. A *straight arc* \mathbf{xy} is a straight line segment directed from a point x to a point y ; \mathbf{xy} is *monotone* with respect to an oriented straight line \mathbf{d} if the projection of x on \mathbf{d} precedes the projection of y on \mathbf{d} according to the orientation of \mathbf{d} . A path (u_1, \dots, u_n) is *\mathbf{d} -monotone* if $\mathbf{u_i u_{i+1}}$ is monotone with respect to \mathbf{d} , for $i = 1, \dots, n - 1$; a polygon Q is *\mathbf{d} -monotone* if it contains two vertices s and t such that the two paths between s and t in Q are both \mathbf{d} -monotone. A path P (a polygon Q) is *monotone* if there exists an oriented straight line \mathbf{d} such that P (resp. Q) is \mathbf{d} -monotone. We have the following.

► **Lemma 3** (Angelini et al. [3]). *Let Q be a convex polygon and \mathbf{d} be an oriented straight line not orthogonal to any straight line through two vertices of Q . Then Q is \mathbf{d} -monotone.*

► **Lemma 4.** *Let Q_1 and Q_2 be strictly-convex polygons sharing an edge e and lying on opposite sides of the line through e . Let P_i be the path obtained from Q_i by removing edge e , for $i = 1, 2$. The polygon Q composed of P_1 and P_2 is monotone.*

Proof sketch. Q is monotone with respect to a line \mathbf{l} orthogonal to e – unless Q contains edges parallel to e , in which case a slight perturbation of \mathbf{l} suffices. ◀

Morphing. A *linear morph* $\langle \Gamma_1, \Gamma_2 \rangle$ between two straight-line planar drawings Γ_1 and Γ_2 of a plane graph G moves each vertex at constant speed along a straight line from its position in Γ_1 to its position in Γ_2 . A linear morph is *planar* if no crossing or overlap occurs between any two edges or vertices during the transformation. A linear morph is *convex* (*strictly-convex*) if it is planar and each face is delimited by a convex (resp. strictly-convex) polygon at any time instant of the morph. A convex linear morph is called a *morphing step*. A *unidirectional* linear morph [7] is a linear morph in which the straight-line trajectories of the vertices are parallel. A *convex morph* (a *strictly-convex morph*) $\langle \Gamma_s, \dots, \Gamma_t \rangle$ between two convex drawings Γ_s and Γ_t of a plane graph G is a finite sequence of convex (resp. strictly-convex) linear morphs that transforms Γ_s into Γ_t . A *unidirectional* (*strictly-*) *convex morph* is such that each of its morphing steps is unidirectional.

3 Decompositions of Convex Graphs

Our morphing algorithm relies on a lemma stating that, roughly speaking, any convex graph has a “simple” subgraph whose removal results in a smaller convex graph. A similar result is known for a restricted graph class, namely the subdivisions of triconnected planar graphs.

On the way to proving that every triconnected planar graph is the skeleton of a convex polytope in \mathbb{R}^3 , Barnette and Grünbaum [6] proved that every subdivision of a triconnected planar graph G can be decomposed as follows (see also [20]). Starting from G , repeatedly remove a path whose internal vertices have degree two in the current graph, until a subdivision of K_4 is obtained. Barnette and Grünbaum proved that there is such a decomposition in which every intermediate graph is a subdivision of a simple triconnected plane graph.

We now present a lemma that generalizes Barnette and Grünbaum’s decomposition technique so that it applies to convex (not necessarily triconnected) graphs.

► **Lemma 5.** *Let G be a convex graph. There exists a sequence G_1, \dots, G_ℓ of graphs such that: (i) $G_1 = G$; (ii) G_ℓ is the simple cycle C delimiting the outer face of G ; (iii) for each $1 \leq i \leq \ell$, graph G_i is a subgraph of G and is a subdivision of a simple internally triconnected plane graph H_i ; and (iv) for each $1 \leq i < \ell$, graph G_{i+1} is obtained either:*

- *by deleting the edges and the internal vertices of a path (u_1, u_2, \dots, u_k) with $k \geq 2$ from G_i , where u_2, \dots, u_{k-1} are degree-2 internal vertices of G_i ; or*
- *by deleting a degree-3 internal vertex u of G_i as well as the edges and the internal vertices of three paths P_1, P_2 , and P_3 connecting u with three vertices of the cycle C delimiting the outer face of G , where P_1, P_2 , and P_3 are vertex-disjoint except at u and the internal vertices of P_1, P_2 , and P_3 are degree-2 internal vertices of G_i .*

Proof. Set $G_1 = G$. Suppose that a sequence G_1, \dots, G_i has been determined. If $G_i = G_\ell$ is the cycle delimiting the outer face of G , then we are done. Otherwise, we distinguish two cases, based on whether G_i is a subdivision of a triconnected plane graph or not.

Suppose first that G_i is a subdivision of a triconnected plane graph H_i . We construct graphs G_i, \dots, G_ℓ one by one, in reverse order. Throughout the construction, we maintain the following invariant for every $\ell \geq j > i$. Suppose that H_j contains an internal edge (u, v) that is also an edge of H_i . Then there exists no path in H_i that connects u and v , that is different from edge (u, v) , and all of whose internal vertices are not in H_j .

Let G_ℓ be the cycle C delimiting the outer face of G_i . Next, we determine $G_{\ell-1}$ (see Fig. 1(a)). Let C_i be the cycle delimiting the outer face of H_i . Since H_i is triconnected and has at least four vertices, there exist three paths that connect an internal vertex v of H_i with vertices of C_i , that share no vertices other than v , and whose internal vertices are not in C_i (see Theorem 5.1 in [23]). Among all the triples of paths with these properties, choose a triple (P_x, P_y, P_z) involving the largest number of vertices of H_i . Paths P_x, P_y , and P_z and cycle C_i form a graph $G_{\ell-1}^H$ that is a subdivision of K_4 . The subgraph $G_{\ell-1}$ of G_i corresponding to $G_{\ell-1}^H$ is hence a subdivision of K_4 in which v is the only degree-3 internal vertex. The invariant is satisfied since P_x, P_y , and P_z involve the largest number of vertices of H_i . Further, G_ℓ is obtained from $G_{\ell-1}$ by deleting a degree-3 internal vertex v of $G_{\ell-1}$ as well as the edges and the internal vertices of P_x, P_y , and P_z , as required by the lemma.

Next, assume that a sequence G_ℓ, \dots, G_j has been determined, for some $j \leq \ell - 1$. If $G_j = G_i$, then we are done. Otherwise, G_{j-1} is obtained by adding a path P to G_j . The choice of P distinguishes two cases (as in the proof of Theorem 2 in [6]).

In Case (A), a vertex z exists such that $\deg(G_j, z) = 2$ and $\deg(G_i, z) \geq 3$. Then, consider the unique path P_{xy} in G_j that contains z as an internal vertex, whose internal vertices have degree two in G_j , and whose end-points x and y have degree at least three in G_j . Note

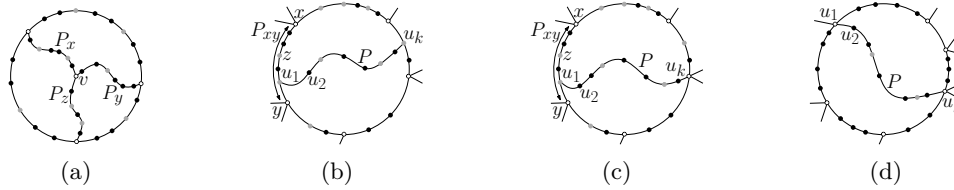


Figure 1 Illustration for the proof of Lemma 5 if G_i is a subdivision of a triconnected plane graph H_i . White vertices belong to $H_j, G_j, H_i,$ and G_i ; grey vertices belong to $G_j, H_i,$ and G_i , and not to H_j ; black vertices belong to G_j and G_i , and not to H_j and H_i . (a) Graph $G_{\ell-1}$. (b)–(d) Graph G_j and path P , together forming graph G_{j-1} ; (b) and (c) illustrate Case (A) with u_k having degree two and greater than two in G_j , respectively, while (d) depicts Case (B).

that (x, y) is an edge of H_j . Since $\{x, y\}$ is not a separation pair in H_i , there exists a path $P = (u_1, u_2, \dots, u_k)$ in G_i such that u_1 is an internal vertex of P_{xy} , vertex u_h does not belong to G_j , for every $2 \leq h \leq k - 1$, and u_k is a vertex of G_j not in P_{xy} . Choose the path with these properties involving the largest number of vertices of H_i . Observe that u_k might have degree two (as in Fig. 1(b)) or greater than two (as in Fig. 1(c)) in G_j .

In Case (B), there exists no vertex z such that $\deg(G_j, z) = 2$ and $\deg(G_i, z) \geq 3$ (see Fig. 1(d)). Since G_j is different from G_i , there exists a path $P = (u_1, u_2, \dots, u_k)$ in G_i such that u_1 and u_k belong to H_j , and u_2, \dots, u_{k-1} do not belong to G_j . Also, a path P satisfying these properties exists such that u_1 is an internal vertex of H_i (otherwise H_i would contain a separation pair composed of two external vertices). Choose a path P involving the largest number of vertices of H_i , subject to the constraint that u_1 is an internal vertex of H_i .

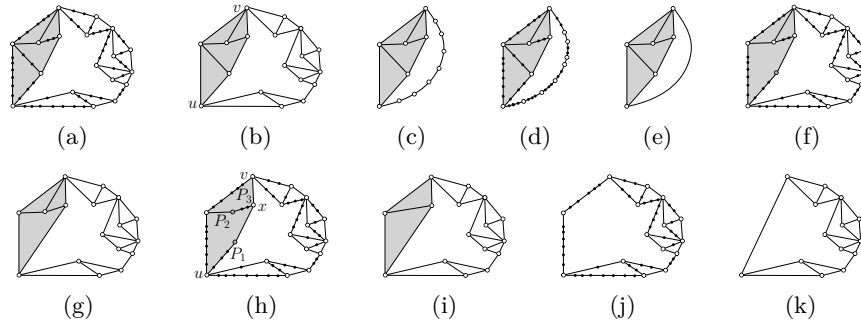
In both cases, path P has to be embedded inside a face f of G_j , according to the plane embedding of G_i . Since G_j contains the cycle delimiting the outer face of G_i , we have that f is an internal face of G_j . Graph G_{j-1} is obtained by inserting P in f . Since P and G_j are subgraphs of G_i , graph G_{j-1} is a subgraph of G_i . Also, it satisfies the invariant since P is chosen as a path involving the largest number of vertices of H_i . It remains to prove that G_{j-1} is a subdivision of a simple triconnected plane graph H_{j-1} . Let H_{j-1} be the graph obtained from G_{j-1} by replacing each maximal path whose internal vertices have degree two with a single edge. Thus, G_{j-1} is a subdivision of H_{j-1} .

► **Claim 1.** *Graph H_{j-1} is plane, simple, and triconnected.*

Proof sketch. First, H_{j-1} is a plane graph since G_{j-1} is a plane graph. Second, in Case (A) H_{j-1} is simple because H_j is simple and u_1 does not belong to H_j ; further, it can be proved that H_{j-1} contains no separation pair (hence it is triconnected) because H_j contains three internally disjoint paths between any pair of vertices and because each of u_1 and u_k contains three internally disjoint paths to vertices of H_j . Third, in Case (B) H_{j-1} is simple because of the invariant and it is triconnected because H_j is triconnected. ◀

We now turn to the case in which G_i is not a subdivision of a triconnected plane graph. In this case G_i is a subdivision of a simple internally triconnected plane graph H_i with minimum degree three and containing some separation pairs. Recall that H_i has either two or three split components with respect to any separation pair $\{u, v\}$.

Suppose that a separation pair $\{u, v\}$ exists in H_i determining three split components. Since H_i is internally triconnected, one of these split components is an internal edge (u, v) of H_i corresponding to a path $P = (u = u_1, \dots, u_k = v)$ in G_i , where u_2, \dots, u_{k-1} are degree-2 internal vertices of G_i . Let $G_{i+1} = G_i - \{u_2, \dots, u_{k-1}\}$ and let $H_{i+1} = H_i - (u, v)$. Note that



■ **Figure 2** Illustration for the proof of Lemma 5 if G_i is not a subdivision of a triconnected plane graph H_i . The faces of D_1, \dots, D_m not incident to Q are colored gray in G_i, \dots, G_{i+m-1} . The faces of M_1, \dots, M_m not incident to (u, v) are colored gray in H_i, \dots, H_{i+m-1} . (a) Graph G_i . (b) Graph H_i and separation pair $\{u, v\}$. (c) Graph L . (d) Graph $D = D_1$. (e) Graph $M = M_1$. (f) Graph G_{i+1} . (g) Graph H_{i+1} . (h) Graph G_{i+2} . (i) Graph H_{i+2} . (j) Graph G_{i+3} . (k) Graph H_{i+3} .

G_{i+1} is a subdivision of H_{i+1} . Then H_{i+1} is an internally triconnected simple plane graph, given that H_i is an internally triconnected simple plane graph with three split components with respect to $\{u, v\}$.

Suppose next that every separation pair of H_i determines two split components, as in Fig. 2(a). Let $\{u, v\}$ be a separation pair of H_i determining two split components A and B such that A does not contain any separation pair of H_i different from $\{u, v\}$, as in Fig. 2(b), (e.g., let $\{u, v\}$ be a separation pair such that the number of vertices in A is minimum among all separation pairs). Let L be the subgraph of H_i composed of A and of the path Q between u and v that delimits the outer face of H_i and that belongs to B ; see Fig. 2(c). Let D be the subgraph of G_i corresponding to L ; see Fig. 2(d). The graph M obtained from L by replacing Q with an edge (u, v) , shown in Fig. 2(e), is triconnected, given that the vertex set of A does not contain any separation pair of H_i different from $\{u, v\}$. Thus, D is a subdivision of a simple triconnected plane graph M .

By means of the same algorithm described in the case in which G_i is a subdivision of a triconnected plane graph, we determine a sequence D_1, \dots, D_m of subdivisions of triconnected plane graphs M_1, \dots, M_m , where $D_1 = D$, $M_1 = M$, and $M_m = K_3$. Further, we define a sequence $H_{i+1}, \dots, H_{i+m-1}$ of graphs where, for each $2 \leq j \leq m-1$, graph H_{i+j-1} is obtained from H_i by replacing M with M_j (see Figs. 2(b), 2(g), and 2(i)), and where H_{i+m-1} is obtained from H_i by replacing M with an edge (u, v) (see Fig. 2(k)). Analogously, we define a sequence $G_{i+1}, \dots, G_{i+m-1}$ of graphs where, for each $2 \leq j \leq m$, graph G_{i+j-1} is obtained from G_i by replacing D with D_j (see Figs. 2(a), 2(f), 2(h), and 2(j)). Then, for each $2 \leq j \leq m$, graph G_{i+j-1} is a subdivision of H_{i+j-1} . Further, for each $1 \leq j \leq m-2$, graph G_{i+j} is obtained from G_{i+j-1} by deleting the edges and the internal vertices of a path (u_1, \dots, u_k) with $k \geq 2$, where u_2, \dots, u_{k-1} are degree-2 internal vertices of G_{i+j-1} . Moreover, graph G_{i+m-1} is obtained by deleting from G_{i+m-2} a degree-3 internal vertex x as well as the edges and the internal vertices of three paths P_1, P_2 , and P_3 , as required by the lemma. Finally, since M_2, \dots, M_m are simple triconnected plane graphs, $H_{i+1}, \dots, H_{i+m-1}$ are simple internally triconnected plane graphs.

Note that H_{i+m-1} is obtained from H_i by replacing A with edge (u, v) , hence $\{u, v\}$ is not a separation pair in H_{i+m-1} . Thus, the repetition of the described transformations over different separation pairs $\{u, v\}$ eventually leads to a graph G_x that is the subdivision of a simple triconnected plane graph H_x ; then a sequence G_x, \dots, G_ℓ of subdivisions of triconnected plane graphs such that G_ℓ is a subdivision of K_3 is determined as above. ◀

4 Convex Drawings of Hierarchical Convex Graphs

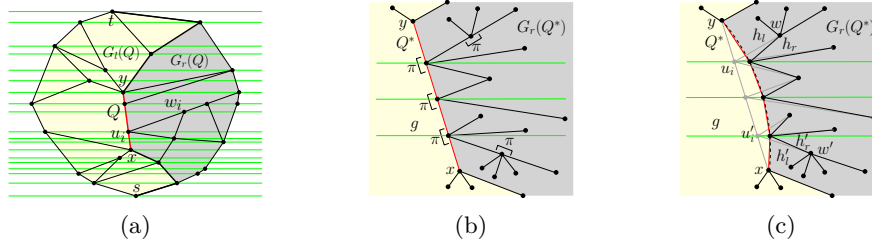
A *hierarchical graph* is a tuple $(G, \mathbf{d}, L, \gamma)$ where G is a graph, \mathbf{d} is an oriented straight line in the plane, L is a set of parallel lines orthogonal to \mathbf{d} , and γ is a function that maps each vertex of G to a line in L so that adjacent vertices are mapped to distinct lines. The lines in L are ordered as they are encountered when traversing \mathbf{d} according to its orientation (we write $l_1 < l_2$ if a line l_1 precedes a line l_2 in L). Furthermore, each line $l_i \in L$ is oriented so that \mathbf{d} cuts l_i from the right to the left of l_i ; a point a *precedes* a point b on l_i if a is encountered before b when traversing l_i according to its orientation. For the sake of readability, we will often write G instead of $(G, \mathbf{d}, L, \gamma)$ to denote a hierarchical graph. A *level drawing* of a hierarchical graph G maps each vertex v to a point on the line $\gamma(v)$ and each edge (u, v) of G with $\gamma(u) < \gamma(v)$ to an arc uv monotone with respect to \mathbf{d} . A hierarchical graph G with a prescribed plane embedding is a *hierarchical plane graph* if there is a level planar drawing Γ of G that respects the prescribed plane embedding. A path (u_1, \dots, u_k) in G is *monotone* if $\gamma(u_i) < \gamma(u_{i+1})$, for $1 \leq i \leq k - 1$. An *st-face* in a hierarchical plane graph G is a face delimited by two monotone paths connecting two vertices s and t , where s is the *source* and t is the *sink* of the face. Furthermore, G is a *hierarchical-st plane graph* if every face of G is an st-face; note that a face f of G is an st-face if and only if the polygon delimiting f in a straight-line level planar drawing of G is \mathbf{d} -monotone.

In this section we give an algorithm to construct strictly-convex level planar drawings of *hierarchical-st strictly-convex graphs*, that are hierarchical-st plane graphs $(G, \mathbf{d}, L, \gamma)$ such that G is a strictly-convex graph. We have the following.

► **Theorem 6.** *Every hierarchical-st strictly-convex graph admits a drawing which is simultaneously strictly-convex and level planar.*

Proof. Let $(G, \mathbf{d}, L, \gamma)$ be a hierarchical-st strictly-convex graph, in the following simply denoted by G , and let C be the cycle delimiting the outer face f of G . Construct a strictly-convex level planar drawing P_C of C in which the clockwise order of the vertices along P_C is the same as prescribed in G . Hong and Nagamochi [16] showed an algorithm to construct a (non-strictly) convex level planar drawing Γ of G in which C is represented by P_C . We show how to modify Γ into a strictly-convex level planar drawing of G .

We give some definitions. Let s and t be the vertices of G such that $\gamma(s) < \gamma(u) < \gamma(t)$, for every vertex $u \neq s, t$ of G . Given a vertex v of G , the *leftmost (rightmost) top neighbor* of v is the neighbor x of v with $\gamma(x) > \gamma(v)$ such that for the neighbor y of v counter-clockwise (clockwise) following x we have that either $\gamma(y) < \gamma(v)$, or $\gamma(y) > \gamma(v)$ and both x and y are incident to f (this only happens when $v = s$). The *leftmost* and the *rightmost bottom neighbor* of v are defined analogously. Also, the *leftmost (rightmost) top path* of v is the monotone path P from v to t obtained by initializing $P = (v)$ and by repeatedly adding the leftmost (resp. rightmost) top neighbor of the last vertex. The *leftmost* and *rightmost bottom path* of v are defined analogously. Let v be a vertex of G that is flat in a face g of Γ ; v is an internal vertex of G , since P_C is strictly-convex. Let x and y be the neighbors of v in g ; then either $\gamma(x) < \gamma(v) < \gamma(y)$ or $\gamma(y) < \gamma(v) < \gamma(x)$. Assume the former. If g lies to the left of path (x, v, y) when traversing it from x to y , then we say that v is a *left-flat vertex* in Γ , otherwise v is a *right-flat vertex*. By Theorem 2 and since v is an internal vertex of G , we have $\deg(G, v) \geq 3$, hence v cannot be both a left-flat and a right-flat vertex in Γ . A *left-flat (right-flat) path* in Γ is a maximal path whose internal vertices are all left-flat (resp. right-flat) vertices and are all flat in the same face (see Fig. 3(a)). Let $Q = (x, \dots, y)$ be a left-flat path in Γ ; the *elongation* E_Q of Q is the monotone path between s and t obtained by concatenating the rightmost bottom path of x , Q , and the rightmost top path of y . Let



■ **Figure 3** (a) A left-flat path Q (red thick line), its elongation $E(Q)$ (red and black thick lines), graphs $G_r(Q)$ (gray) and $G_l(Q)$ (yellow). (b) Drawing Γ . (c) Drawing Γ' .

$G_l(Q)$ ($G_r(Q)$) be the subgraph of G whose outer face is delimited by the cycle composed of E_Q and of the leftmost (resp. rightmost) top path of s . For a right-flat path Q in Γ , the elongation E_Q of Q , and graphs $G_l(Q)$ and $G_r(Q)$ are defined analogously.

In order to modify Γ into a strictly-convex level planar drawing of G , we proceed by induction on the number $a(\Gamma)$ of flat angles in Γ . If $a(\Gamma) = 0$, then Γ is strictly-convex and there is nothing to be done. If $a(\Gamma) \geq 1$, then there exists a path Q that is either a left-flat path or a right-flat path in Γ . Assume the former, the other case is symmetric. Also, assume w.l.o.g. up to a rotation of the axes, that the lines in L are horizontal.

Ideally, we would like to move the internal vertices of Q to the right, so that the polygon delimiting the face on which the internal vertices of Q are flat becomes strictly-convex. There is one obstacle to such a modification, though: An internal vertex of Q might be the first or the last vertex of a left-flat path Q' ; thus, moving that vertex to the right would cause the polygon delimiting the face on which the internal vertices of Q' are flat to become concave (in Fig. 3(a) moving u_i to the right causes an angle incident to w_i to become concave). We now argue that there is a left-flat path Q^* such that $G_r(Q^*)$ contains no internal left-flat path; then we modify Γ by moving the internal vertices of Q^* to the right.

Let $Q^* = (x, \dots, y)$ be a left-flat path such that the number of internal vertices of $G_r(Q^*)$ is minimum. Suppose, for a contradiction, that $G_r(Q^*)$ contains an internal left-flat path Q' . Then $G_r(Q')$ has less internal vertices than $G_r(Q^*)$, since $G_r(Q')$ is a subgraph of $G_r(Q^*)$ and the internal vertices of Q' are internal vertices of $G_r(Q^*)$ and external vertices of $G_r(Q')$. This contradiction proves that $G_r(Q^*)$ does not contain any internal left-flat path.

We construct a convex drawing Γ' of G with $a(\Gamma') < a(\Gamma)$. Initialize $\Gamma' = \Gamma$ and remove the internal vertices of Q^* . Let $\epsilon > 0$ be to be determined later. Consider segment \overline{xy} , its mid-point z , and a point p in the half-plane to the right of \overline{xy} such that segment \overline{zp} is orthogonal to \overline{xy} and has length ϵ . Let a be the arc of circumference between x and y passing through p . Place each internal vertex v of Q^* at the intersection point of $\gamma(v)$ with a , which exists since Q^* is monotone. Denote by Γ' the resulting drawing. We have the following.

► **Claim 2.** *The following statements hold, provided that ϵ is sufficiently small: (i) Γ' is convex; (ii) every vertex that is flat in an incident face in Γ' is flat in the same face in Γ ; and (iii) every internal vertex of Q^* is convex in every incident face in Γ' .*

Proof sketch. Moving the internal vertices of Q^* from \overline{xy} to a results in these vertices being convex in the unique face g of $G_l(Q^*)$ they are all incident to in Γ' . Further, the difference between the size of any angle in Γ' and the size of the corresponding angle in Γ tends to 0 as $\epsilon \rightarrow 0$; in particular, angles that are flat in Γ either have the same or smaller size in Γ' (see w in Figs. 3(b)–(c)), given that $G_r(Q^*)$ does not contain any internal left-flat path. ◀

Claim 2 implies that Γ' is convex and that $a(\Gamma') < a(\Gamma)$. The theorem follows. ◀

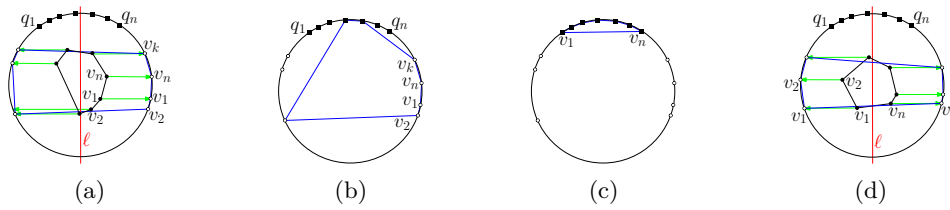


Figure 4 (a) Drawings Γ_s (black circles and black lines) and Γ'_s (white circles and blue lines), together with points q_1, \dots, q_n . (b) Morph $\langle \Gamma'_s, \dots, \Gamma \rangle$ after two steps. (c) Drawing Γ . (d) Drawings Γ_t (black circles and black lines) and Γ'_t (white circles and blue lines), together with points q_1, \dots, q_n .

5 A Morphing Algorithm

In this section we give algorithms to morph convex drawings of plane graphs. We start with a lemma about unidirectional linear morphs. Two level planar drawings Γ_1 and Γ_2 of a hierarchical plane graph $(G, \mathbf{d}, L, \gamma)$ are *left-to-right equivalent* if, for any line $l_i \in L$, for any vertex or edge x of G , and for any vertex or edge y of G , we have that x precedes (follows) y on l_i in Γ_1 if and only if x precedes (resp. follows) y on l_i in Γ_2 . We have the following.

► **Lemma 7.** *The linear morph $\langle \Gamma_1, \Gamma_2 \rangle$ between two left-to-right equivalent strictly-convex level planar drawings Γ_1 and Γ_2 of a hierarchical-st strictly-convex graph $(G, \mathbf{d}, L, \gamma)$ is strictly-convex and unidirectional.*

Proof sketch. Morph $\langle \Gamma_1, \Gamma_2 \rangle$ is planar and unidirectional [3]. Also, it is strictly-convex since an angle \widehat{vuz} that is convex in Γ_1 and in Γ_2 stays convex during $\langle \Gamma_1, \Gamma_2 \rangle$; this descends from the planarity of $\langle \Gamma_1, \Gamma_2 \rangle$ if $\gamma(u) < \gamma(v), \gamma(z)$ or $\gamma(v), \gamma(z) < \gamma(u)$ and from the fact that u, v , and z are never aligned during $\langle \Gamma_1, \Gamma_2 \rangle$ if $\gamma(z) < \gamma(u) < \gamma(v)$ (see [7]). ◀

We now describe an algorithm to construct a strictly-convex morph between any two strictly-convex drawings Γ_s and Γ_t of a plane graph G with n vertices and m internal faces. The algorithm works by induction on m and consists of at most $2n + 2m$ morphing steps.

In the base case we have $m = 1$, hence G is a cycle. We have the following.

► **Claim 3.** *There exists a strictly-convex unidirectional morph with at most $2n + 2$ steps between any two strictly-convex drawings Γ_s and Γ_t of cycle G .*

Proof Sketch. Let v_1, \dots, v_n be the vertices of G as they appear clockwise around G . Let ℓ be a straight line not orthogonal to any line through two vertices of G in Γ_s and in Γ_t . Draw a circumference \mathcal{C} enclosing both Γ_s and Γ_t . Morph Γ_s (Γ_t) into a drawing Γ'_s (Γ'_t) such that all the vertices of G are on \mathcal{C} (see Fig. 4(a) and Fig. 4(d)) with a single strictly-convex morphing step which is unidirectional in the direction orthogonal to ℓ (each vertex moves in the direction that does not make it collide with the initial drawing of G).

Consider n points q_1, \dots, q_n in this clockwise order on \mathcal{C} both in Γ'_s and in Γ'_t such that the arc of \mathcal{C} between q_1 and q_n containing q_2 does not contain any vertex of G . Morph Γ'_s (Γ'_t) into a drawing Γ of G in which v_i is placed at q_i , for $1 \leq i \leq n$, as follows (see Figs. 4(a)–(c)). Let v_k be the first vertex of G encountered when clockwise traversing \mathcal{C} from q_n . For $j = k - 1, \dots, 1, k, \dots, n$, move v_j to p_j . These morphs consist of n unidirectional strictly-convex morphing steps each. Hence, $\langle \Gamma_s, \Gamma'_s, \dots, \Gamma, \dots, \Gamma'_t, \Gamma_t \rangle$ is a unidirectional strictly-convex morph between Γ_s and Γ_t with $2n + 2$ morphing steps. ◀

In the inductive case we have $m > 1$. Then we apply Lemma 5 to G in order to obtain a graph G' with $m' < m$ internal faces. We proceed as follows.

Assume first that, according to Lemma 5, a degree-3 internal vertex u of G as well as the edges and the internal vertices of paths P_1 , P_2 , and P_3 can be removed from G resulting in a convex graph G' , where: (i) P_1 , P_2 , and P_3 respectively connect u with vertices u_1 , u_2 , and u_3 of the cycle C delimiting the outer face f of G ; (ii) P_1 , P_2 , and P_3 are vertex-disjoint except at u ; and (iii) the internal vertices of P_1 , P_2 , and P_3 are degree-2 internal vertices of G . Graph G has no degree-2 internal vertices, since it is strictly-convex (see Theorem 2), hence P_1 , P_2 , and P_3 are edges (u, u_1) , (u, u_2) , and (u, u_3) , respectively.

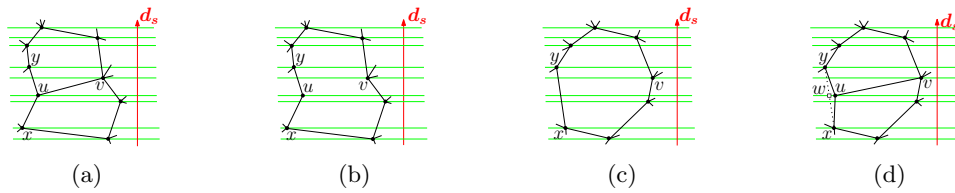
Vertex u lies in the interior of triangle $\Delta(u_1, u_2, u_3)$ both in Γ_s and in Γ_t , since $\deg(G, u) = 3$ and the angles incident to u are smaller than π both in Γ_s and in Γ_t . Hence, the position of u is a convex combination of the positions of u_1 , u_2 , and u_3 both in Γ_s and in Γ_t (the coefficients of such convex combinations might be different in Γ_s and in Γ_t). Further, no vertex other than u and no edge other than those incident to u lie in the interior of triangle $\Delta(u_1, u_2, u_3)$ in Γ_s and Γ_t , since these drawings are strictly-convex. With a single unidirectional linear morph, move u in Γ_s to the point that is a convex combination of the positions of u_1 , u_2 , and u_3 with the same coefficients as in Γ_t . This morph is strictly-convex since u stays inside $\Delta(u_1, u_2, u_3)$ at any time instant. Let Γ'_s be the resulting drawing of G .

Let Q_1 , Q_2 , and Q_3 be the polygons delimiting the faces of G incident to u in Γ_s . Let Λ'_s be the drawing of G' obtained from Γ'_s by removing u and its incident edges. We claim that Λ'_s is strictly-convex. Indeed, every internal face of G' different from the face f_u that used to contain u is also a face in Γ'_s , hence it is delimited by a strictly-convex polygon. Further, every internal angle of the polygon delimiting f_u is either an internal angle of Q_1 , Q_2 , or Q_3 , hence it is smaller than π , since Γ'_s is strictly-convex, or is incident to u_1 , u_2 , or u_3 ; however, these vertices are concave in f , hence they are convex in $f_u \neq f$. Analogously, the drawing Λ'_t of G' obtained from Γ_t by removing u and its incident edges is strictly-convex.

Inductively construct a unidirectional convex morph $\langle \Lambda'_s = \Lambda_0, \dots, \Lambda_\ell = \Lambda'_t \rangle$ with $\ell \leq 2(n-1) + 2(m-2)$ morphing steps. For each $1 \leq j \leq \ell - 1$, draw u in Λ_j at a point that is the convex combination of the positions of u_1 , u_2 , and u_3 with the same coefficients as in Γ'_s and in Γ_t ; denote by Γ_j the resulting drawing of G . Morph $\langle \Gamma'_s = \Gamma_0, \dots, \Gamma_\ell = \Gamma_t \rangle$ is strictly-convex and unidirectional. Namely, in every morphing step $\langle \Gamma_j, \Gamma_{j+1} \rangle$, vertex u moves between two points that are convex combinations of the positions of u_1 , u_2 , and u_3 with the same coefficients, hence it moves parallel to each of u_1 , u_2 , and u_3 (from which $\langle \Gamma_0, \dots, \Gamma_\ell \rangle$ is unidirectional) and it stays inside $\Delta(u_1, u_2, u_3)$ at any time instant of $\langle \Gamma_j, \Gamma_{j+1} \rangle$ (from which $\langle \Gamma_0, \dots, \Gamma_\ell \rangle$ is strictly-convex). Thus, $\langle \Gamma_s, \Gamma'_s = \Gamma_0, \dots, \Gamma_\ell = \Gamma_t \rangle$ is a unidirectional strictly-convex morph between Γ_s and Γ_t with $\ell + 1 \leq 2n + 2m - 5$ morphing steps.

Assume next that, according to Lemma 5, the edges and the internal vertices of a path P , whose internal vertices are degree-2 internal vertices of G , can be deleted from G so that the resulting graph G' is convex. Graph G has no degree-2 internal vertices, since it is strictly-convex (see Theorem 2), hence P is an edge (u, v) . Removing (u, v) from Γ_s (from Γ_t) results in a drawing Λ_s (resp. Λ_t) of G' which is not, in general, convex, since vertices u and v might be concave in the face f_{uv} of G' that used to contain (u, v) , as in Fig. 5. By Lemma 4, there exists an oriented straight line \mathbf{d}_s such that the polygon Q_{uv} representing the cycle C_{uv} delimiting f_{uv} is \mathbf{d}_s -monotone. By slightly perturbing the slope of \mathbf{d}_s , we can assume that it is not orthogonal to any line through two vertices of G' . Let L'_s be the set of parallel and distinct lines through vertices of G' and orthogonal to \mathbf{d}_s . Let γ'_s be the function that maps each vertex of G' to the line in L'_s through it. We have the following.

► **Lemma 8.** $(G', \mathbf{d}_s, L'_s, \gamma'_s)$ is a hierarchical-st convex graph.



■ **Figure 5** Drawings (a) Γ_s , (b) Λ_s , (c) Λ'_s , and (d) Γ'_s .

Analogously, there exists an oriented straight line \mathbf{d}_t that leads to define a hierarchical-st convex graph $(G', \mathbf{d}_t, L'_t, \gamma'_t)$ for which Λ_t is a straight-line level planar drawing.

We now distinguish three cases, based on whether $\deg(G', u), \deg(G', v) > 2$ (**Case 1**), $\deg(G', u) = 2$ and $\deg(G', v) > 2$ (**Case 2**), or $\deg(G', u) = \deg(G', v) = 2$ (**Case 3**). The case in which $\deg(G', u) > 2$ and $\deg(G', v) = 2$ is symmetric to Case 2.

In Case 1 graph G' is strictly-convex, since it is convex and all its internal vertices have degree greater than two. By Theorem 6, $(G', \mathbf{d}_s, L'_s, \gamma'_s)$ and $(G', \mathbf{d}_t, L'_t, \gamma'_t)$ admit strictly-convex level planar drawings Λ'_s and Λ'_t , respectively. Let Γ'_s (Γ'_t) be the strictly-convex level planar drawing of $(G, \mathbf{d}_s, L'_s, \gamma'_s)$ (resp. of $(G, \mathbf{d}_t, L'_t, \gamma'_t)$) obtained by inserting edge (u, v) as a straight-line segment in Λ'_s (resp. Λ'_t). Drawings Γ_s and Γ'_s (Γ_t and Γ'_t) are left-to-right equivalent. This is argued as follows. First, since G is a plane graph, its outer face is delimited by the same cycle C in both Γ_s and Γ'_s ; further, the clockwise order of the vertices along C is the same in Γ_s and in Γ'_s (recall that Theorem 6 allows us to arbitrarily prescribe the strictly-convex polygon representing C). Consider any two vertices or edges x and y both intersecting a line ℓ in L'_s ; assume this line to be oriented in any way. Suppose, for a contradiction, that x precedes y on ℓ in Γ_s and follows y on ℓ in Γ'_s . Since Γ_s and Γ'_s are strictly-convex, there exists a \mathbf{d}_s -monotone path P_x (P_y) containing x (resp. y) and connecting two vertices of C . Then P_x and P_y properly cross, contradicting the planarity of Γ_s or of Γ'_s , or they share a vertex which has a different clockwise order of its incident edges in the two drawings, contradicting the fact that Γ_s and Γ'_s are drawings of the same plane graph. By Lemma 7, linear morphs $\langle \Gamma_s, \Gamma'_s \rangle$ and $\langle \Gamma_t, \Gamma'_t \rangle$ are strictly-convex and unidirectional.

Inductively construct a unidirectional strictly-convex morph $\langle \Lambda'_s = \Lambda_0, \Lambda_1, \dots, \Lambda_\ell = \Lambda'_t \rangle$ with $\ell \leq 2n + 2(m - 1)$ morphing steps between Λ'_s and Λ'_t . For each $0 \leq j \leq \ell$, draw edge (u, v) in Λ_j as a straight-line segment \overline{uv} ; let Γ_j be the resulting drawing of G . We have that morph $\langle \Gamma'_s = \Gamma_0, \Gamma_1, \dots, \Gamma_\ell = \Gamma'_t \rangle$ is strictly-convex and unidirectional given that $\langle \Lambda_0, \Lambda_1, \dots, \Lambda_\ell \rangle$ is strictly-convex and unidirectional and given that, at any time instant of $\langle \Lambda_0, \Lambda_1, \dots, \Lambda_\ell \rangle$, segment \overline{uv} splits the strictly-convex polygon delimiting f_{uv} into two strictly-convex polygons. Thus, $\langle \Gamma_s, \Gamma'_s = \Gamma_0, \Gamma_1, \dots, \Gamma_\ell = \Gamma'_t, \Gamma_t \rangle$ is a unidirectional strictly-convex morph between Γ_s and Γ_t with $\ell + 2 \leq 2n + 2m$ morphing steps.

In Case 2 let G'' be the graph obtained from G' by replacing path (x, u, y) with edge (x, y) , where x and y are the only neighbors of u in G' . Graph G'' is strictly-convex, since G' is convex and is a subdivision of G'' , and since all the internal vertices of G'' have degree greater than two. Moreover, since $(G', \mathbf{d}_s, L'_s, \gamma'_s)$ and $(G', \mathbf{d}_t, L'_t, \gamma'_t)$ are hierarchical-st convex graphs, it follows that $(G'', \mathbf{d}_s, L''_s, \gamma''_s)$ and $(G'', \mathbf{d}_t, L''_t, \gamma''_t)$ are hierarchical-st strictly-convex graphs, where $L''_s = L'_s \setminus \{\gamma'_s(u)\}$, $L''_t = L'_t \setminus \{\gamma'_t(u)\}$, $\gamma''_s(z) = \gamma'_s(z)$ for each vertex z in G'' , and $\gamma''_t(z) = \gamma'_t(z)$ for each vertex z in G'' . By Theorem 6, $(G'', \mathbf{d}_s, L''_s, \gamma''_s)$ and $(G'', \mathbf{d}_t, L''_t, \gamma''_t)$ admit strictly-convex level planar drawings Λ''_s and Λ''_t , respectively.

We modify Λ''_s into a drawing Γ'_s of $(G, \mathbf{d}_s, L'_s, \gamma'_s)$, as in Fig. 5. Assume w.l.o.g. that

$\gamma'_s(x) < \gamma'_s(u) < \gamma'_s(y)$. Let w be the intersection point of $\gamma'_s(u)$ and \overline{xy} in Λ''_s (where line $\gamma'_s(u)$ is the same as in Λ_s). Let C''_{uv} be the facial cycle of G'' such that the facial cycle C_{uv} of G' is a subdivision of C''_{uv} . Insert u in the interior of C''_{uv} , on $\gamma'_s(u)$, at distance $\epsilon > 0$ from w . Remove edge (x, y) from Λ''_s and insert edges (u, v) , (u, x) , and (u, y) as straight-line segments. Denote by Γ'_s the resulting drawing. We have the following.

► **Claim 4.** Γ'_s is a strictly-convex level planar drawing of $(G, \mathbf{d}_s, L'_s, \gamma'_s)$, provided that $\epsilon > 0$ is sufficiently small.

A strictly-convex level planar drawing Γ'_t of $(G, \mathbf{d}_t, L'_t, \gamma'_t)$ can be constructed analogously from Λ''_t . Drawings Γ_s and Γ'_s (Γ_t and Γ'_t) are left-to-right equivalent, which can be proved as in Case 1. By Lemma 7, morphs $\langle \Gamma_s, \Gamma'_s \rangle$ and $\langle \Gamma_t, \Gamma'_t \rangle$ are strictly-convex and unidirectional.

Inductively construct a unidirectional strictly-convex morph $\langle \Lambda''_s = \Lambda_0, \Lambda_1, \dots, \Lambda_\ell = \Lambda''_t \rangle$ with $\ell \leq 2(n-1) + 2(m-1)$ morphing steps between Λ''_s and Λ''_t . Let $0 < \xi < 1$ be sufficiently small so that the following holds true: For every $0 \leq j \leq \ell$, insert u in Λ_j at a point which is a convex combination of the positions of x , y , and v with coefficients $(\frac{1-\xi}{2}, \frac{1-\xi}{2}, \xi)$, remove edge (x, y) , and insert edges (u, x) , (u, y) , and (u, v) as straight-line segments; then the resulting drawing Γ_j of G is strictly-convex. Such a $\xi > 0$ exists. Namely, placing v as a convex combination of the positions of x , y , and v results in angles incident to u and v that are all convex. Moreover, as $\xi \rightarrow 0$, the point at which u is placed approaches segment \overline{xy} , hence the size of any angle incident to x or y approaches the size of an angle incident to x or y in Λ_j , and the latter is strictly less than π radians.

With a single unidirectional strictly-convex linear morph, move u in Γ'_s to the point that is a convex combination of the positions of x , y , and v with coefficients $(\frac{1-\xi}{2}, \frac{1-\xi}{2}, \xi)$; denote by Γ''_s the drawing of G obtained from this morph. Analogously, let $\langle \Gamma'_t, \Gamma''_t \rangle$ be a unidirectional strictly-convex linear morph, where the point at which u is placed in Γ''_t is a convex combination of the positions of x , y , and v with coefficients $(\frac{1-\xi}{2}, \frac{1-\xi}{2}, \xi)$.

For each $0 \leq j \leq \ell - 1$, Γ_j and Γ_{j+1} are left-to-right equivalent strictly-convex level planar drawings of the hierarchical-st strictly-convex graph $(G, \mathbf{d}_j, L_j, \gamma_j)$, where \mathbf{d}_j is an oriented straight line orthogonal to the direction of morph $\langle \Lambda_j, \Lambda_{j+1} \rangle$, L_j is the set of lines through vertices of G orthogonal to \mathbf{d}_j , and γ_j maps each vertex of G to the line in L_j through it. In particular, Γ_j and Γ_{j+1} are strictly-convex drawings of G since Λ_j and Λ_{j+1} are strictly-convex drawings of G'' and by the choice of ξ ; further, every face of G is an st-face in Γ_j and Γ_{j+1} by Lemmata 3 and 4; moreover, u moves parallel to the other vertices since $\langle \Lambda_j, \Lambda_{j+1} \rangle$ is unidirectional and since the points at which u is placed in Γ_j and Γ_{j+1} are convex combinations of the positions of x , y , and v with the same coefficients. By Lemma 7, $\langle \Gamma_j, \Gamma_{j+1} \rangle$ is strictly-convex and unidirectional. Hence, $\langle \Gamma_s, \Gamma'_s, \Gamma''_s = \Gamma_0, \Gamma_1, \dots, \Gamma_\ell = \Gamma''_t, \Gamma'_t, \Gamma_t \rangle$ is a unidirectional strictly-convex morph between Γ_s and Γ_t with $\ell + 4 \leq 2n + 2m$ morphing steps.

Case 3 is very similar to Case 2, hence we only sketch the algorithm here. Let G'' be the graph obtained from G' by replacing paths (x_u, u, y_u) and (x_v, v, y_v) with edges (x_u, y_u) and (x_v, y_v) , respectively, where x_u and y_u (x_v and y_v) are the only neighbors of u (resp. v) in G' ; $(G'', \mathbf{d}_s, L''_s, \gamma''_s)$ and $(G'', \mathbf{d}_t, L''_t, \gamma''_t)$ are hierarchical-st strictly-convex graphs, where $L''_s = L'_s \setminus \{\gamma'_s(u), \gamma'_s(v)\}$, $L''_t = L'_t \setminus \{\gamma'_t(u), \gamma'_t(v)\}$, $\gamma''_s(z) = \gamma'_s(z)$ for each vertex z in G'' , and $\gamma''_t(z) = \gamma'_t(z)$ for each vertex z in G'' . By Theorem 6, $(G'', \mathbf{d}_s, L''_s, \gamma''_s)$ and $(G'', \mathbf{d}_t, L''_t, \gamma''_t)$ admit strictly-convex level planar drawings Λ''_s and Λ''_t , respectively. We modify Λ''_s into a strictly-convex level planar drawing Γ'_s of $(G, \mathbf{d}_s, L'_s, \gamma'_s)$ by inserting u (v) on $\gamma'_s(u)$ (resp. $\gamma'_s(v)$) at distance $\epsilon > 0$ from the intersection point of $\gamma'_s(u)$ with segment $\overline{x_u y_u}$ (of $\gamma'_s(v)$)

with segment $\overline{x_v y_v}$ in the interior of the facial cycle of G'' such that the facial cycle C_{uv} of G' is a subdivision of C''_{uv} . Analogously, we modify Λ''_t into a strictly-convex level planar drawing Γ'_t of $(G, \mathbf{d}_t, L'_t, \gamma'_t)$. Drawings Γ_s and Γ'_s (Γ_t and Γ'_t) are left-to-right equivalent.

Inductively construct a unidirectional strictly-convex morph $\langle \Lambda''_s = \Lambda_0, \dots, \Lambda_\ell = \Lambda''_t \rangle$ with $\ell \leq 2(n - 2) + 2(m - 1)$ morphing steps. Let $\xi > 0$ be sufficiently small so that for every $0 \leq j \leq \ell$, inserting u (v) in Λ_j at a convex combination of the positions of x_u, y_u, x_v , and y_v with coefficients $(\frac{1-\xi}{2}, \frac{1-\xi}{2}, \frac{\xi}{2}, \frac{\xi}{2})$ (resp. $(\frac{\xi}{2}, \frac{\xi}{2}, \frac{1-\xi}{2}, \frac{1-\xi}{2})$), removing edges (x_u, y_u) and (x_v, y_v) , and inserting edges (x_u, u) , (y_u, u) , (x_v, v) , (y_v, v) , and (u, v) results in a strictly-convex drawing Γ_j of G . With a unidirectional strictly-convex linear morph $\langle \Gamma'_s, \Gamma''_s \rangle$, move u in Γ'_s to the point that is a convex combination of the positions of x_u, y_u, x_v , and y_v with coefficients $(\frac{1-\xi}{2}, \frac{1-\xi}{2}, \frac{\xi}{2}, \frac{\xi}{2})$. With a unidirectional strictly-convex linear morph $\langle \Gamma''_s, \Gamma'''_s \rangle$, move v in Γ''_s to the point that is a convex combination of the positions of x_u, y_u, x_v , and y_v with coefficients $(\frac{\xi}{2}, \frac{\xi}{2}, \frac{1-\xi}{2}, \frac{1-\xi}{2})$. Define morph $\langle \Gamma'_t, \Gamma''_t, \Gamma'''_t \rangle$ analogously. For each $0 \leq j \leq \ell - 1$, Γ_j and Γ_{j+1} are left-to-right equivalent strictly-convex level planar drawings of the hierarchical-st strictly-convex graph $(G, \mathbf{d}_j, L_j, \gamma_j)$, where \mathbf{d}_j is an oriented line orthogonal to the direction of morph $\langle \Lambda_j, \Lambda_{j+1} \rangle$, L_j is the set of lines through vertices of G and orthogonal to \mathbf{d}_j , and γ_j maps each vertex of G to the line in L_j through it. By Lemma 7, $\langle \Gamma_s, \Gamma'_s, \Gamma''_s, \Gamma'''_s = \Gamma_0, \dots, \Gamma_\ell = \Gamma''_t, \Gamma'_t, \Gamma'''_t, \Gamma_t \rangle$ is a unidirectional strictly-convex morph between Γ_s and Γ_t with $\ell + 6 \leq 2n + 2m$ morphing steps. We get the following.

► **Theorem 9.** *There exists an algorithm to construct a strictly-convex unidirectional morph with $O(n)$ morphing steps between any two strictly-convex drawings of the same n -vertex plane graph.*

A simple enhancement of the above described algorithm allows us to extend our results to (non-strictly) convex drawings of convex graphs. We have the following.

► **Theorem 10.** *There exists an algorithm to construct a convex unidirectional morph with $O(n)$ morphing steps between any two convex drawings of the same n -vertex plane graph.*

Proof sketch. First, with $O(n)$ unidirectional convex morphing steps we morph Γ_s (Γ_t) into a convex drawing Γ'_s (resp. Γ'_t) such that the polygon delimiting the outer face of G is strictly-convex. This is done by moving, during each morphing step, all the internal vertices of a maximal path incident to the outer face whose internal vertices have degree two.

Second, we consider each maximal path $P = (u_1, \dots, u_k)$ such that u_2, \dots, u_{k-1} are degree-2 internal vertices of G ; with a single linear morph in the direction of $\overline{u_1 u_k}$, we move each of u_2, \dots, u_{k-1} in Γ'_s to the point which is a convex combination of the positions of u_1 and u_k with the same coefficients as in Γ'_t . Over all such paths P this amounts to $O(n)$ unidirectional convex morphing steps; denote by Γ''_s the resulting drawing of G .

Third, we replace each maximal path $P = (u_1, \dots, u_k)$ such that u_2, \dots, u_{k-1} are degree-2 internal vertices of G with an edge (u_1, u_k) in G , Γ''_s , and Γ'_t ; we obtain a strictly-convex graph G' , and two strictly-convex drawings Λ''_s and Λ'_t of G' . We compute a strictly-convex unidirectional morph $\langle \Lambda''_s = \Lambda_0, \dots, \Lambda_\ell = \Lambda'_t \rangle$ with $\ell \in O(n)$ morphing steps as in Theorem 9. For each $0 \leq j \leq \ell$, we reinsert the internal vertices of each path $P = (u_1, \dots, u_k)$ in Λ_j at the points that are the convex combinations of the positions of u_1 and u_k in Λ_j with the same coefficients as in Γ''_s and in Γ'_t . Each morphing step $\langle \Gamma_j, \Gamma_{j+1} \rangle$ is convex and unidirectional, since u_i moves between two points that are convex combinations of the positions of u_1 and u_k with the same coefficients. Hence, $\langle \Gamma_s, \dots, \Gamma'_s, \dots, \Gamma''_s = \Gamma_0, \Gamma_1, \dots, \Gamma_\ell = \Gamma'_t, \dots, \Gamma_t \rangle$ is a unidirectional convex morph between Γ_s and Γ_t with $O(n)$ morphing steps. ◀

References

- 1 S. Alamdari, P. Angelini, T. M. Chan, G. Di Battista, F. Frati, A. Lubiw, M. Patrignani, V. Roselli, S. Singla, and B. T. Wilkinson. Morphing planar graph drawings with a polynomial number of steps. In *SODA*, pages 1656–1667, 2013.
- 2 G. Aloupis, L. Barba, P. Carmi, V. Dujmovic, F. Frati, and P. Morin. Compatible connectivity-augmentation of planar disconnected graphs. In *SODA*, pages 1602–1615, 2015.
- 3 P. Angelini, G. Da Lozzo, G. Di Battista, F. Frati, M. Patrignani, and V. Roselli. Morphing planar graph drawings optimally. In *ICALP*, volume 8572 of *LNCS*, pages 126–137, 2014.
- 4 P. Angelini, F. Frati, M. Patrignani, and V. Roselli. Morphing planar graph drawings efficiently. In *GD*, volume 8242 of *LNCS*, pages 49–60, 2013.
- 5 I. Bárány and G. Rote. Strictly convex drawings of planar graphs. *Documenta Mathematica*, 11:369–391, 2006.
- 6 D. Barnette and B. Grünbaum. On Steinitz’s theorem concerning convex 3-polytopes and on some properties of planar graphs. In *Many Facets of Graph Theory*, volume 110 of *Lecture Notes in Mathematics*, pages 27–40. Springer, 1969.
- 7 F. Barrera-Cruz, P. Haxell, and A. Lubiw. Morphing planar graph drawings with unidirectional moves. Mexican Conference on Discr. Math. and Comput. Geom., 2013.
- 8 N. Bonichon, S. Felsner, and M. Mosbah. Convex drawings of 3-connected plane graphs. *Algorithmica*, 47(4):399–420, 2007.
- 9 S. Cairns. Deformations of plane rectilinear complexes. *Am. Math. Mon.*, 51:247–252, 1944.
- 10 N. Chiba, T. Yamanouchi, and T. Nishizeki. Linear algorithms for convex drawings of planar graphs. In *Progress in Graph Theory*, pages 153–173. Academic Press, New York, NY, 1984.
- 11 M. Chrobak and G. Kant. Convex grid drawings of 3-connected planar graphs. *Int. J. Comput. Geometry Appl.*, 7(3):211–223, 1997.
- 12 C. Erten, S. G. Kobourov, and C. Pitta. Intersection-free morphing of planar graphs. In *GD*, volume 2912 of *LNCS*, pages 320–331, 2004.
- 13 C. Friedrich and P. Eades. Graph drawing in motion. *J. Graph Alg. Ap.*, 6:353–370, 2002.
- 14 C. Gotsman and V. Surazhsky. Guaranteed intersection-free polygon morphing. *Computers & Graphics*, 25(1):67–75, 2001.
- 15 B. Grünbaum and G.C. Shephard. *The geometry of planar graphs*. Camb. Univ. Pr., 1981.
- 16 S. H. Hong and H. Nagamochi. Convex drawings of hierarchical planar graphs and clustered planar graphs. *J. Discrete Algorithms*, 8(3):282–295, 2010.
- 17 S. H. Hong and H. Nagamochi. A linear-time algorithm for symmetric convex drawings of internally triconnected plane graphs. *Algorithmica*, 58(2):433–460, 2010.
- 18 M. S. Rahman, S. I. Nakano, and T. Nishizeki. Rectangular grid drawings of plane graphs. *Comput. Geom.*, 10(3):203–220, 1998.
- 19 M. S. Rahman, T. Nishizeki, and S. Ghosh. Rectangular drawings of planar graphs. *J. of Algorithms*, 50:62–78, 2004.
- 20 J. M. Schmidt. Contractions, removals, and certifying 3-connectivity in linear time. *SIAM J. Comput.*, 42(2):494–535, 2013.
- 21 V. Surazhsky and C. Gotsman. Controllable morphing of compatible planar triangulations. *ACM Trans. Graph*, 20(4):203–231, 2001.
- 22 V. Surazhsky and C. Gotsman. Intrinsic morphing of compatible triangulations. *Internat. J. of Shape Model.*, 9:191–201, 2003.
- 23 C. Thomassen. Planarity and duality of finite and infinite graphs. *J. Comb. Theory, Ser. B*, 29(2):244–271, 1980.
- 24 C. Thomassen. Deformations of plane graphs. *J. Comb. Th. Ser. B*, 34(3):244–257, 1983.
- 25 C. Thomassen. Plane representations of graphs. In J. A. Bondy and U. S. R. Murty, editors, *Progress in Graph Theory*, pages 43–69. Academic Press, New York, NY, 1984.

1-String B_2 -VPG Representation of Planar Graphs*

Therese Biedl and Martin Derka

David R. Cheriton School of Computer Science, University of Waterloo
200 University Ave W, Waterloo, ON N2L 3G1, Canada
{biedl, mderka}@uwaterloo.ca

Abstract

In this paper, we prove that every planar graph has a 1-string B_2 -VPG representation – a string representation using paths in a rectangular grid that contain at most two bends. Furthermore, two paths representing vertices u, v intersect precisely once whenever there is an edge between u and v .

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Graph drawing, string graphs, VPG graphs, planar graphs

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.141

1 Preliminaries

One way of representing graphs is to assign to every vertex a curve so that two curves cross if and only if there is an edge between the respective vertices. Here, two curves \mathbf{u}, \mathbf{v} *cross* means that they share a point s internal to both of them and the boundary of a sufficiently small closed disk around s is crossed by $\mathbf{u}, \mathbf{v}, \mathbf{u}, \mathbf{v}$ (in this order). The representation of graphs using crossing curves is referred to as a *string representation*, and graphs that can be represented in this way are called *string graphs*.

In 1976, Ehrlich, Even and Tarjan showed that every planar graph has a string representation [9]. It is only natural to ask if this result holds if one is restricted to using only some “nice” types of curves. In 1984, Scheinerman conjectured that all planar graphs can be represented as intersection graphs of line segments [12]. This was proved first for bipartite graphs [8, 11] with the strengthening that every segment is vertical or horizontal. The result was extended to triangle-free graphs, which can be represented by line segments with at most three distinct slopes [7].

Since Scheinerman’s conjecture seemed difficult to prove for all planar graphs, interest arose in possible relaxations. Note that any two line segments can generally intersect at most once. Define 1-STRING to be the class of graphs that are intersection graphs of curves (of arbitrary shape) that intersect at most once. We also say that graphs in this class have a *1-string representation*. The original construction of string representations for planar graphs given in [9] requires curves to cross multiple times. In 2007, Chalopin, Gonçalves and Ochem showed that every planar graph is in 1-STRING [4, 5]. With respect to Scheinerman’s conjecture, while the argument of [4, 5] shows that the prescribed number of intersections can be achieved, it provides no idea on the complexity of curves that is required.

Another way of restricting curves in string representations is to require them to be *orthogonal*, i.e., to be paths in a grid. Call a graph a *VPG-graph* (as in “Vertex-intersection

* Research supported by NSERC. The second author was supported by the Vanier CGS.



graph of Paths in a Grid”) if it has a string representation with orthogonal curves. It is easy to see that all planar graphs are VPG-graphs (e.g. by generalizing the construction of Ehrlich, Even and Tarjan). For bipartite planar graphs, curves can even be required to have no bends [8, 11]. For arbitrary planar graphs, bends are required in orthogonal curves. Recently, Chaplick and Ueckerdt showed that 2 bends per curve always suffice [6]. Let B_2 -VPG be the graphs that have a string representation where curves are orthogonal and have at most 2 bends; the result in [6] then states that planar graphs are in B_2 -VPG. Unfortunately, in Chaplick and Ueckerdt’s construction, curves may cross each other repeatedly, and so it does not prove that planar graphs are in 1-STRING.

The conjecture of Scheinerman remained open until 2009 when it was proved true by Chalopin and Gonçalves [3].

Our Results: In this paper, we show that every planar graph has a string representation that simultaneously satisfies the requirements for 1-STRING (any two curves cross at most once) and the requirements for B_2 -VPG (any curve is orthogonal and has at most two bends). Our result hence re-proves, in one construction, the results by Chalopin et al. [4, 5] and the result by Chaplick and Ueckerdt [6].

► **Theorem 1.** *Every planar graph has a 1-string B_2 -VPG representation.*

Our approach is inspired by the construction of 1-string representations from 2007 [4, 5]. The authors proved the result in two steps. First, they showed that triangulations without separating triangles admit 1-string representations. By induction on the number of separating triangles, they then showed that a 1-string representation exists for any planar triangulation, and consequently for any planar graph.

In order to show that triangulations without separating triangles have 1-string representations, Chalopin et al. [5] used a method inspired by Whitney’s proof that 4-connected planar graphs are Hamiltonian [13]. Asano, Saito and Kikuchi later improved Whitney’s technique and simplified his proof [1]. Our paper uses the same approach as [5], but borrows ideas from [1] and develops them further to reduce the number of cases.

2 Definitions and Basic Results

Let us begin with a formal definition of a 1-string B_2 -VPG representation.

► **Definition 2** (1-string B_2 -VPG representation). A graph G has a 1-string B_2 -VPG representation if every vertex v of G can be represented by a curve \mathbf{v} such that:

1. Curve \mathbf{v} is *orthogonal*, i.e., it consists of horizontal and vertical segments.
2. Curve \mathbf{v} has at most two bends.
3. Curves \mathbf{u} and \mathbf{v} intersect at most once, and \mathbf{u} intersects \mathbf{v} if and only if (u, v) is an edge of G .

We always use \mathbf{v} to denote the curve of vertex v , and write \mathbf{v}^R if the representation R is not clear from the context. We also often omit “1-string B_2 -VPG” since we do not consider any other representations.

Our technique for constructing representations of a graph uses an intermediate step referred to as a “partial 1-string B_2 -VPG representation of a W -triangulation that satisfies the chord condition with respect to three chosen corners.” We define these terms, and related graph terms, first.

A *planar graph* is a graph that can be embedded in the plane, i.e., it can be drawn so that no edges intersect except at common endpoints. All graphs in this paper are planar. We assume throughout the paper that one combinatorial embedding of the graph has been fixed by specifying the clockwise (CW) cyclic order of incident edges around each vertex. Subgraphs inherit this embedding, i.e., they use the induced clockwise orders. A *facial region* is a connected region of $\mathbb{R}_2 - \Gamma$ where Γ is a planar drawing of G that conforms with the combinatorial embedding. The circuit bounding this region can be read from the combinatorial embedding of G and is referred to as a *face*. The *outer-face* is the one that corresponds to the unbounded region; all others are called *interior faces*. The outer-face cannot be read from the embedding; we assume throughout this paper that the outer-face of G has been specified. Subgraphs inherit the outer-face by using as outer-face the one whose facial region contains the facial region of the outer-face of G . An edge of G is called *interior* if it does not belong to the outer-face.

A *triangulated disk* is a planar graph G for which the outer-face is a simple cycle and every interior face is a triangle. A *separating triangle* is a cycle C of length 3 such that G has vertices both inside and outside the region bounded by C (with respect to the fixed embedding and outer-face of G). Following the notation of [5], a *W-triangulation* is a triangulated disk that does not contain a separating triangle. A *chord* of a triangulated disk is an interior edge for which both endpoints are on the outer-face.

For two vertices X, Y on the outer-face of a connected planar graph, define P_{XY} to be the counter-clockwise (CCW) path on the outer-face from X to Y (X and Y inclusive). We often study triangulated disks with three specified distinct vertices A, B, C called the *corners*. A, B, C must appear on the outer-face in CCW order. We denote $P_{AB} = (a_1, a_2, \dots, a_r)$, $P_{BC} = (b_1, b_2, \dots, b_s)$ and $P_{CA} = (c_1, c_2, \dots, c_t)$, where $c_t = a_1 = A$, $a_r = b_1 = B$ and $b_s = c_1 = C$.

► **Definition 3** (Chord condition). A W -triangulation G satisfies the *chord condition* with respect to the corners A, B, C if G has no chord within P_{AB}, P_{BC} or P_{CA} , i.e., no interior edge of G has both ends on P_{AB} , or both ends on P_{BC} , or both ends on P_{CA} .¹

► **Definition 4** (Partial 1-string B_2 -VPG representation). Let G be a connected planar graph and $E' \subseteq E(G)$ be a set of edges. An (E') -1-string B_2 -VPG representation of G is a 1-string B_2 -VPG representation of the subgraph $(V(G), E')$, i.e., curves \mathbf{u}, \mathbf{v} cross if and only if (u, v) is an edge in E' . If E' consists of all interior edges of G as well as some set of edges F on the outer-face, then we write $(int \cup F)$ representation instead.

In our constructions, we use $(int \cup F)$ representations with $F = \emptyset$ or $F = e$, where e is an outer-face edge incident to corner C of a W -triangulation. Edge e is called the *special edge*, and we sometimes write $(int \cup e)$ representation, rather than $(int \cup \{e\})$ representation.

2.1 2-Sided, 3-Sided and Reverse 3-Sided Layouts

To create representations where vertex-curves have few bends, we need to impose geometric restrictions on representations of subgraphs. Unfortunately, no one type of layout seems sufficient for all cases, and we will hence have three different layout types whose existence we will prove in parallel.

¹ For readers familiar with [5] or [1]: A W -triangulation that satisfies the chord condition with respect to corners A, B, C is called a *W-triangulation with 3-boundary* P_{AB}, P_{BC}, P_{CA} in [5], and the chord condition is the same as *Condition (W2b)* in [1].

► **Definition 5** (2-sided layout). Let G be a connected planar graph and A, B be two distinct outer-face vertices. An $(int \cup F)$ B_2 -VPG representation of G has a *2-sided layout* (with respect to corners A, B) if:

1. There exists a rectangle Θ that contains all intersections of curves and such that the top of Θ is intersected, from right to left in order, by the curves of the vertices of P_{AB} , and the bottom of Θ is intersected, from left to right in order, by the curves of the vertices of P_{BA} .
2. The curve \mathbf{v} of an outer-face vertex v has at most one bend. (By 1., this implies that \mathbf{A} and \mathbf{B} have no bends.)

► **Definition 6** (3-sided layout). Let G be a connected planar graph and A, B, C be three distinct vertices in CCW order on the outer-face of G . Let F be a set of exactly one outer-face edge incident to C . An $(int \cup F)$ B_2 -VPG representation of G has a *3-sided layout* (with respect to corners A, B, C) if:

1. There exists a rectangle Θ containing all intersections of curves so that
 - (i) the top of Θ is intersected, from right to left in order, by the curves of the vertices on P_{AB} ;
 - (ii) the left side of Θ is intersected, from top to bottom in order, by the curves of the vertices on $P_{Bb_{s-1}}$, possibly followed by \mathbf{C} ; ²
 - (iii) the bottom of Θ is intersected, from right to left in order, by the curves of vertices on P_{c_2A} in reversed order, possibly followed by \mathbf{C} ; ²
 - (iv) curve $\mathbf{b}_s = \mathbf{C} = \mathbf{c}_1$ intersects the boundary of Θ exactly once; it is the bottommost curve to intersect the left side of Θ if the special edge in F is (C, c_2) , and \mathbf{C} is the leftmost curve to intersect the bottom of Θ if the special edge in F is (C, b_{s-1}) .
5. The curve \mathbf{v} of an outer-face vertex v has at most one bend. (By 1., this implies that \mathbf{B} has precisely one bend.)
6. \mathbf{A} and \mathbf{C} have no bends.

We sometimes refer to the rectangle Θ for both 2- and 3-sided representation as a *bounding box*. Figure 1 (which will serve as base case later) shows such layouts for a triangle and varying choices of F . We also need the concept of a *reverse 3-sided layout*, which is similar to the 3-sided layout except that B is straight and A has a bend. Formally, it satisfies conditions 1(ii-iv) and (2). 1(i) is replaced by “the right side of Θ is intersected, from bottom to top in order, by the curves of the vertices on P_{AB} ” and (3) is replaced by “ \mathbf{B} and \mathbf{C} have no bends.”

2.2 Private Regions

Our proof starts with constructing representation for triangulations without separating triangles. The construction is then extended to all triangulations by merging representations of subgraphs obtained by splitting at separating triangles. To permit the merge, we apply the technique used in [5] (and re-discovered in [10]): With every triangular face, create a region that intersects the curves of vertices of the face in a predefined way and does not intersect anything else, specifically any other private regions. Following the notation of [10], we call this a *private region* (but we use a different shape).

² Recall (b_{s-1}, C) and (C, c_2) are the two incident edges of C on the outer-face.

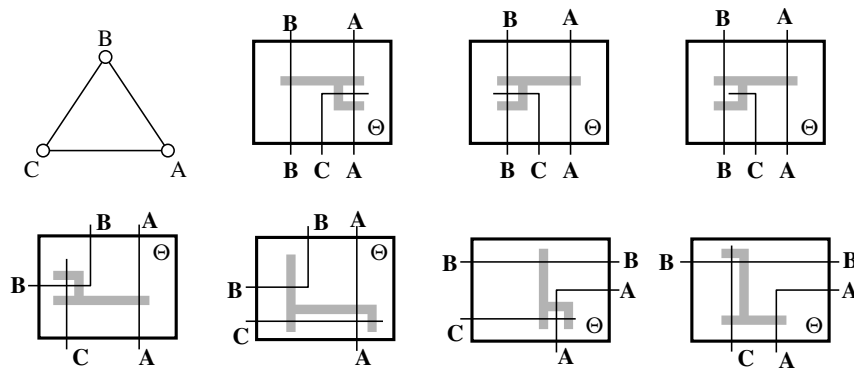


Figure 1 $(int \cup F)$ representations of a triangle: (Top) 2-sided representations for $F \in \{(A,C), \{(B,C)\}, \emptyset\}$. (Bottom) 3-sided and reverse 3-sided representations for $F \in \{(A,C), \{(B,C)\}\}$. Private regions are shaded in grey.

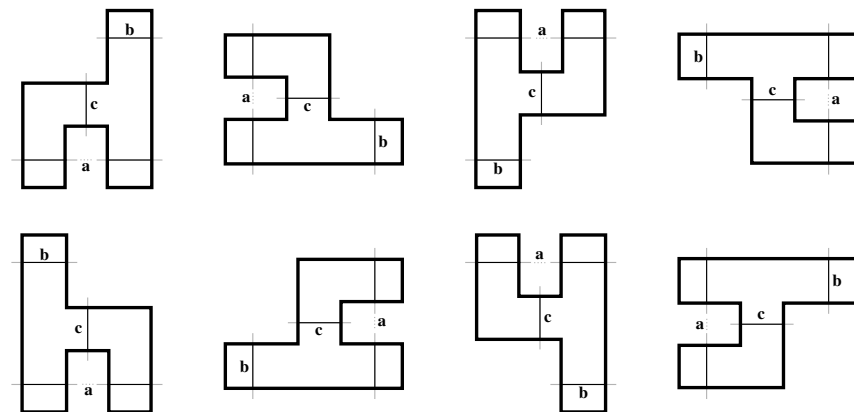


Figure 2 The private region of a triangle a, b, c with possible rotations and flips.

► **Definition 7** (Chair-shape). A *chair-shaped area* is a region bounded by a 10-sided orthogonal polygon with CW (clockwise) or CCW (counter-clockwise) sequence of interior angles $90^\circ, 90^\circ, 270^\circ, 270^\circ, 90^\circ, 90^\circ, 90^\circ, 90^\circ, 270^\circ, 90^\circ$. See also Figure 2.

► **Definition 8** (Private region). Let G be a planar graph with a partial 1-string B_2 -VPG representation R and let f be a facial triangle in G . A *private region* of f is a chair-shaped area Φ inside R such that:

1. Φ is intersected by no curves except for the ones representing vertices on f .
2. All the intersections of R are located outside of Φ .
3. For a suitable labeling of vertices of f as $\{a, b, c\}$, Φ is intersected by two segments of a and one segment of b and c . The intersections between these segments and Φ occur at the edges of Φ as depicted in Figure 2.

3 Constructions for W-Triangulations

Our key tool for proving Theorem 1 is the following lemma:

► **Lemma 9.** Let G be a W -triangulation that satisfies the chord condition with respect to corners A, B, C . For any $e \in \{(C, b_{s-1}), (C, c_2)\}$, G has an $(int \cup e)$ 1-string B_2 -VPG

representation with 3-sided layout and an $(int \cup e)$ 1-string B_2 -VPG representation with reverse 3-sided layout. Both representations have a chair-shaped private region for every interior face.

The proof of Lemma 9 will use induction on the number of vertices. To combine the representations of subgraphs, we sometimes need them to have a 2-sided layout, and hence prove the following result:

► **Lemma 10.** *Let G be a W -triangulation that satisfies the chord condition with respect to corners A, B, C . Then G has an $(int \cup F)$ 1-string B_2 -VPG representation with 2-sided layout with respect to A, B and for any set F of at most one outer-face edge incident to C . Furthermore, this representation has a chair-shaped private region for every interior face of G .*

Notice that for Lemma 9 the special edge *must* exist (this is needed in Case 1 to find private regions), while for Lemma 10, F is allowed to be empty.

We will prove both lemmas simultaneously by induction on the number of vertices. First, let us make an observation that will greatly help to reduce the number of cases. Define G^{rev} to be the graph obtained from graph G by reversing the combinatorial embedding, but keeping the same outer-face. This effectively switches corners A and B , and replaces special edge (C, c_2) by (C, b_{s-1}) and vice versa. If G satisfies the chord condition with respect to corners (A, B, C) , then G^{rev} satisfies the chord condition with respect to corners (B, A, C) . (With this new order, the corners are CCW on the outer-face of G^{rev} , as required.)

Presume we have a 2-sided/3-sided/reverse 3-sided representation of G^{rev} . Then we can obtain a 2-sided representation of G by flipping the 2-sided one of G^{rev} horizontally, i.e., along the y -axis. We can obtain a 3-sided/reverse 3-sided representation of G by flipping the reverse 3-sided/3-sided representation of G^{rev} diagonally (i.e., along the line defined by $(x = y)$). Hence for all the following cases, we may (after possibly applying the above flipping operation) either make a restriction on which edge the special edge is, or we only need to give the construction for 3-sided, but not for reverse 3-sided layout.

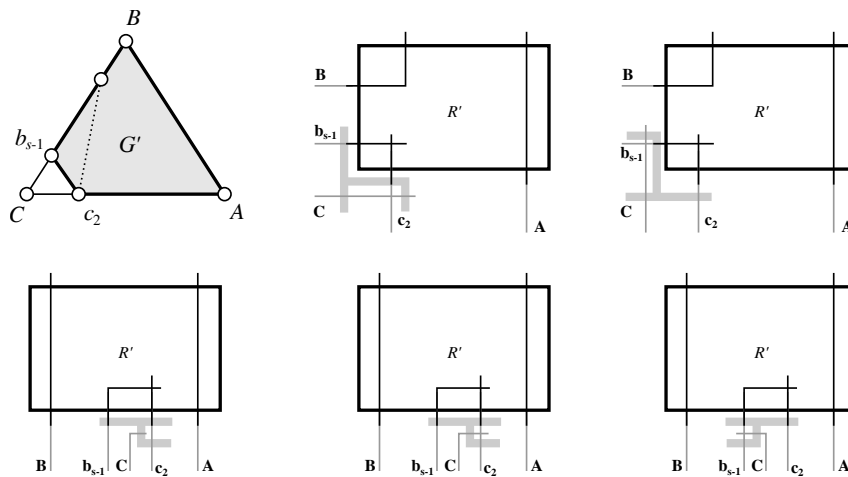
Now we begin the induction. In the base case, $n = 3$, so G is a triangle, and the three corners A, B, C must be the three vertices of this triangle. The desired $(int \cup F)$ representations for all possible choices of F are depicted in Figure 1. The induction step for $n \geq 4$ is divided into three cases which we describe in separate subsections.

3.1 C has degree 2

Since G is a triangulated disk with $n \geq 4$, (b_{s-1}, c_2) is an edge. Define $G' := G - C$ and $F' := \{(b_{s-1}, c_2)\}$. We claim that G' satisfies the chord condition for corners $A' := A, B' := B$ and a suitable choice of $C' \in \{b_{s-1}, c_2\}$, and argue this as follows. If c_2 is not incident to a chord that ends on P_{BC} , then set $C' := c_2$; clearly the chord condition holds for G' . If c_2 is incident to such a chord, then b_{s-1} cannot be incident to a chord by planarity and the chord condition for G . So, in this case with the choice $C' := b_{s-1}$ the chord condition holds for G' . Thus in either case, we can apply induction to G' .

To create a 2-sided representation of G , we use a 2-sided $(int \cup F')$ representation R' of G' . We introduce a new vertical curve \mathbf{C} placed between \mathbf{b}_{s-1} and \mathbf{c}_2 below R' . Add a bend at the upper end of \mathbf{C} and extend it leftwards or rightwards. If the special edge e exists, then extend \mathbf{C} until it hits the curve of the other endpoint of e ; else extend it only far enough to allow for the creation of the private region.

To create a 3-sided representation of G , we use a 3-sided $(int \cup F')$ representation R' of G' . Note that regardless of which vertex is C' , we have \mathbf{b}_{s-1} as bottommost curve on the left and



■ **Figure 3** Case 1: C has degree 2. (Top) 3-sided representation. (Bottom) 2-sided representation.

c_2 as leftmost curve on the bottom. Introduce a new horizontal segment representing C which intersects c_2 if $F = \{(C, c_2)\}$, or a vertical segment which intersects b_{s-1} if $F = \{(C, b_{s-1})\}$.

In both constructions, after suitable lengthening, the curves intersect the bounding box in the required order. One can find the chair-shaped private region for the only new face $\{C, c_2, b_{s-1}\}$ as shown in Figure 3. Observe that no bends were added to the curves of R' and that C has the required number of bends in both representations.

Since we have given the constructions for both possible special edges, we can obtain the reverse 3-sided representation by diagonally flipping a 3-sided representation of G^{rev} .

3.2 G has a chord incident to C

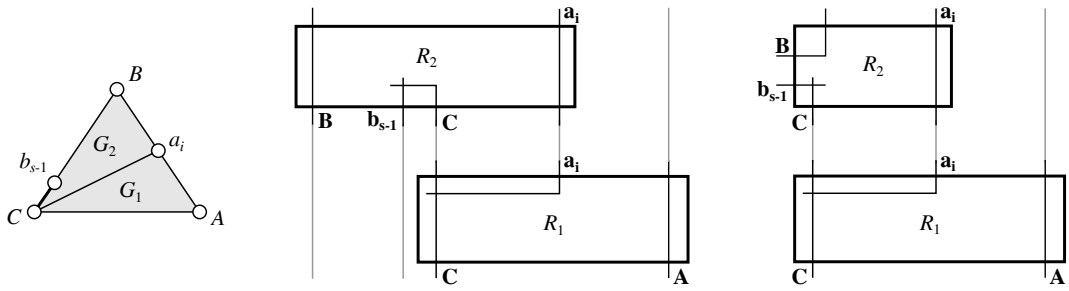
By the chord condition, this chord has the form (C, a_i) for some $1 < i < r$. Select the chord that minimizes i . The graph G can be split along the chord (C, a_i) into two graphs G_1 and G_2 . Both G_1 and G_2 are bounded by simple cycles, hence they are triangulated disks. No edges were added, so neither G_1 nor G_2 contains a separating triangle. So, both of them are W -triangulations.

We select (C, A, a_i) as corners for G_1 and (a_i, B, C) as corners for G_2 and can easily verify that G_1 and G_2 satisfy the chord condition with respect to those corners:

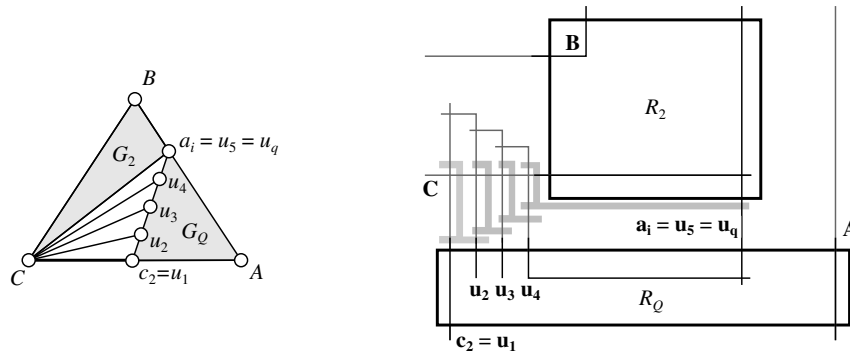
- G_1 has no chords on P_{Aa_i} or P_{CA} as they would violate the chord condition in G . There is no chord on P_{a_iC} as it is a single edge.
- G_2 has no chords on P_{a_iB} or P_{BC} as they would violate the chord condition in G . There is no chord on P_{a_iC} as it is a single edge.

So we can apply induction to both G_1 and G_2 , obtain representations R_1 and R_2 for them, and combine them suitably. In the 3-sided case, we will do so for all possible choices of special edge, and hence need not give the constructions for reverse 3-sided layout as explained earlier.

Case 2(a): $F = \emptyset$ or $F = \{(C, b_{s-1})\}$. Inductively construct a 2-sided $(\text{int} \cup (C, a_i))$ representation R_1 of G_1 . Inductively, construct an $(\text{int} \cup F)$ representation R_2 of G_2 , which should be 2-sided if we want the result to be 2-sided and 3-sided if we want the result to be 3-sided. Note that either way C^{R_2} and $a_1^{R_2}$ on the bottom side of R_2 with C^{R_2} to the left of $a_1^{R_2}$.



■ **Figure 4** Case 2(a): Constructing an $(int \cup (C, b_{s-1}))$ representation when C is incident to a chord, in 2-sided (middle) and 3-sided (right) layout.



■ **Figure 5** Case 2(b)1: C is incident to a chord, $F = (C, c_2)$, and $c_2 \neq A$.

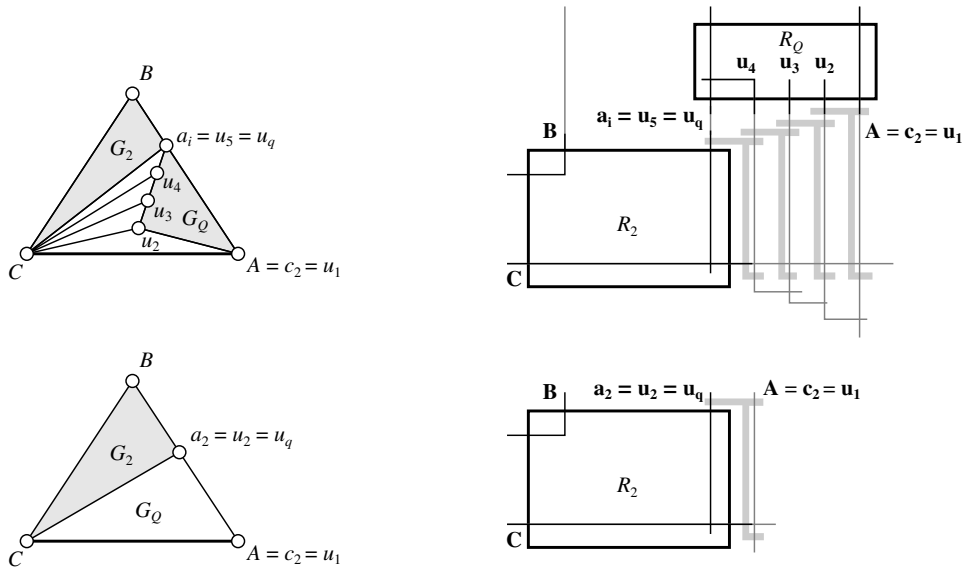
Rotate R_1 by 180° , and translate it so that it is below R_2 with $\mathbf{a}_1^{R_1}$ in the same column as $\mathbf{a}_1^{R_2}$. Stretch R_1 horizontally as needed until \mathbf{C}^{R_1} is in the same column as \mathbf{C}^{R_2} . Then \mathbf{a}_1^R and \mathbf{C}^R for $R \in \{R_1, R_2\}$ can each be unified without adding bends by adding vertical segments. The curves of outer-face vertices of G then cross (after suitable lengthening) the bounding box in the required order. See also Figure 4.

Every interior face f of G is contained in G_1 or G_2 and hence has a private region in R_1 or R_2 . As our construction does not make any changes inside the bounding boxes of R_1 and R_2 , the private region of f is contained in R as well.

Case 2(b): $F = \{(C, c_2)\}$. For the 2-sided construction, we apply the reversal trick: Construct a 2-sided representation of G^{rev} with suitable selection of corners (here Case 2(a) then applies) and flip it horizontally.

For the 3-sided construction, we need a different approach, which is quite similar to Case 1 in [1, Proof of Lemma 2]. Let $G_Q = G_1 - C$, and observe that it is bounded by P_{c_2A} , P_{A, a_i} , and the path formed by the neighbours $c_2 = u_1, u_2, \dots, u_q = a_i$ of C in CCW order. We must have $q \geq 2$, but possibly G_1 is a triangle $\{C, A, a_i\}$ and G_Q then degenerates into an edge. If G_Q contains at least three vertices, then none of u_2, \dots, u_{q-1} belongs to P_{AB} since chord (C, a_i) was chosen closest to A , and so G_Q is a W -triangulation.

We divide the proof into two subcases, depending on whether $A \neq c_2$ or $A = c_2$. As the constructions are sufficiently simple, we refer the reader to Figures 5 and 6 here. A detailed description is in [2, Case 2(b)].



■ **Figure 6** Case 2(b)2: Construction when C is incident to a chord, $c_2 = A$, and (A, a_i, C) is not a face (top), and when (A, a_i, C) is a face (bottom). We only show the 3-sided constructions.

3.3 G has no chords incident with C and $\text{deg}(C) \geq 3$

We will give explicit constructions for 2-sided, 3-sided and reverse 3-sided layout, and may hence (after applying the reversal trick) assume that the special edge, if it exists, is (C, c_2) .

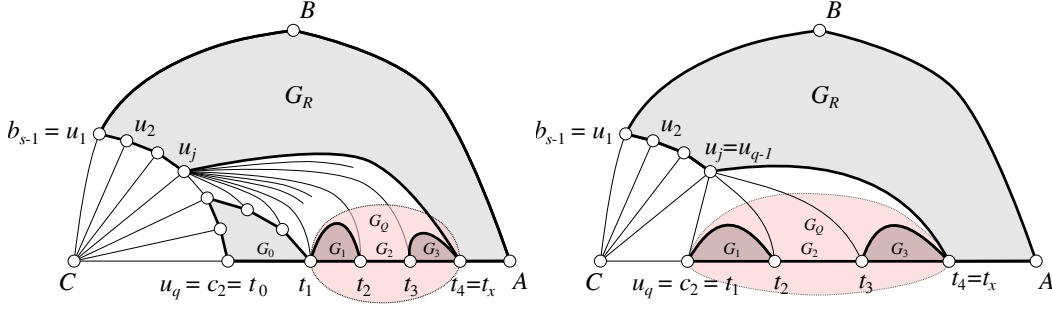
Let u_1, \dots, u_q be the neighbours of vertex C in clockwise order, starting with b_{s-1} and ending with c_2 . We know that $q = \text{deg}(C) \geq 3$ and that u_2, \dots, u_{q-1} are not on the outer-face, since C is not incident to a chord. Let u_j be a neighbour of C that has at least one neighbour other than C on P_{CA} , and among all those, choose j to be minimal. Such a j exists because G is triangulated and therefore u_{q-1} is adjacent to both C and u_q . We distinguish two sub-cases.

Case 3(a): $j \neq 1$. Denote the neighbours of u_j on P_{c_2A} by t_1, \dots, t_x in the order in which they appear on P_{c_2A} . Separate G into subgraphs as follows (see also Figure 7):

- The *right* graph G_R is bounded by $(A, P_{AB}, B, P_{B u_1}, u_1, u_2, \dots, u_j, t_x, P_{t_x A}, A)$.
- Let G_B be the graph bounded by $(u_j, t_1, P_{t_1 t_x}, t_x, u_j)$. We are chiefly interested in its subgraph $G_Q := G_B - u_j$.
- Let G_L be the graph bounded by $(C, P_{C t_1}, t_1, u_j, C)$. We are chiefly interested in its subgraph $G_0 := G_L - \{u_j, C\}$.

The idea is to obtain representations of these subgraphs and then to combine them suitably. We first explain how to obtain the representation R_R used for G_R . Clearly G_R is a W -triangulation, since u_2, \dots, u_j are interior vertices of G , and hence the outer-face of G_R is a simple cycle. Set $A_R := A$ and $B_R := B$. If $B \neq u_1$ then set $C_R := u_1$ and observe that G_R satisfies the chord condition with respect to these corners:

- G_R does not have any chords with both ends on $P_{A_R B_R} = P_{AB}$, $P_{B_R u_1} \subseteq P_{BC}$, or $P_{t_x A_R} \subseteq P_{CA}$ since G satisfies the chord condition.
- If there were any chords between a vertex in u_1, \dots, u_j and a vertex on $P_{C_R A_R}$, then by $C_R = u_1$ the chord would either connect two neighbours of C (hence give a separating triangle of G), or connect some u_i for $i < j$ to P_{CA} (contradicting the minimality of j),



■ **Figure 7** Case 3: Splitting the graph when $\deg(C) \geq 3$ and no chord is incident to C . (Left) $j < q - 1$; G_0 is non-trivial. (Right) $j = q - 1$; $G_0 = \{c_2\}$.

or connect u_j to some other vertex on $P_{t_x A}$ (contradicting that t_x is the last neighbour of u_j on P_{CA}). Hence no such chord can exist either.

If $B = u_1$, then set $C_R := u_2$ (which exists by $q \geq 3$) and similarly verify that it satisfies the chord condition as $P_{B_R C_R}$ is the edge (B, u_2) . Since $C_R \in \{u_1, u_2\}$ in both cases, we can apply induction on G_R and obtain an $(\text{int} \cup \{u_1, u_2\})$ representation R_R . We use as layout for R_R the type that we want for G , i.e., use a 2-sided/3-sided/reverse 3-sided layout if we want G to have a 2-sided/3-sided/reverse 3-sided representation.

Next consider the graph G_0 , which is bounded by $u_{j+1}, \dots, u_q, P_{c_2 t_1}$ and the neighbours of u_j in CCW order between t_1 and u_{j+1} . We distinguish two cases:

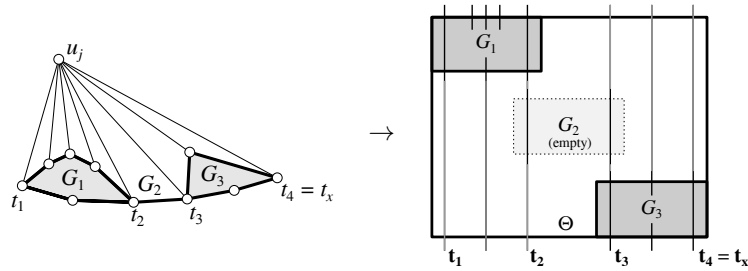
- (1) $j = q - 1$, and hence $t_1 = u_q = c_2$ and G_0 consists of only c_2 . In this case, the representation of R_0 consists of a single vertical line segment \mathbf{c}_2 .
- (2) $j < q - 1$, so G_0 contains at least three vertices u_{q-1}, u_q and t_1 . Then G_0 is a W-triangulation since C is not incident to a chord and by the choice of t_1 . Also, it satisfies the chord condition with respect to corners $A_0 := c_2, B_0 := t_1$ and $C_0 := u_{j+1}$ since the three paths on its outer-face are sub-paths of P_{CA} or contained in the neighbourhood of C or u_j . In this case, construct a 2-sided $(\text{int} \cup \{u_{j+1}, u_{j+2}\})$ representation R_0 of G_0 with respect to these corners inductively.

Finally, we create a representation R_Q of $G_Q = G_B - u_j$. If G_Q is a single vertex or a single edge, then simply use vertical segments for the curves of its vertices. Otherwise, we can show:

► **Claim 11.** G_Q has a 2-sided $(\text{int} \cup \emptyset)$ 1-string B_2 -VPG representation with respect to corners t_1 and t_x .

Proof. G_Q is not necessarily 2-connected, so we cannot apply induction directly. Instead we break it into $x - 1$ graphs G_1, \dots, G_{x-1} , where for $i = 1, \dots, x - 1$ graph G_i is bounded by $P_{t_i t_{i+1}}$ as well as the neighbours of u_j between t_i and t_{i+1} in CCW order. Note that G_i is either a single edge, or it is bounded by a simple cycle since u_j has no neighbours on P_{CA} between t_i and t_{i+1} . In the latter case, use $B_i := t_i, A_i := t_{i+1}$, and C_i an arbitrary third vertex on $P_{t_i t_{i+1}} \subseteq P_{CA}$, which exists since the outer-face of G_i is a simple cycle and (t_i, t_{i+1}, u_j) is not a separating triangle. Observe that G_i satisfies the chord condition since all paths on the outer-face of G_i are either part of P_{CA} or in the neighbourhood of u_j . Hence by induction there exists a 2-sided $(\text{int} \cup \emptyset)$ representation R_i of G_i . If G_i is a single edge (t_i, t_{i+1}) , then let R_i consists of two vertical segments \mathbf{t}_i and \mathbf{t}_{i+1} .

Since each representation R_i has at its leftmost end a vertical segment \mathbf{t}_i and at its rightmost end a vertical segment \mathbf{t}_{i+1} , we can combine all these representations by aligning



■ **Figure 8** Left: Graph G_B . The boundary of G_Q is shown bold. Right: Merging 2-sided ($int \cup \emptyset$) representations of $G_i, 1 \leq i \leq 3$, into a 2-sided ($int \cup \emptyset$) representation of G_Q .

$t_i^{R_i}$ and $t_i^{R_{i+1}}$ horizontally and filling in the missing segment. See also Figure 8. One easily verifies that the result is a 2-sided ($int \cup \emptyset$) representation of G_Q . ◀

We now explain how to combine these three representations R_R, R_Q and R_0 ; see also Figure 9. Translate R_Q so that it is below R_R with $t_x^{R_R}$ and $t_x^{R_Q}$ in the same column; then connect these two curves with a vertical segment. Rotate R_0 by 180° and translate it so that it is below R_R and to the left and above R_Q , and $t_1^{R_0}$ and $t_1^{R_Q}$ are in the same column; then connect these two curves with a vertical segment. Notice that the vertical segments of $u_2^{R_R}, \dots, u_j^{R_R}$ are at the bottom left of R_R . Horizontally stretch R_0 and/or R_R so that $u_2^{R_R}, \dots, u_j^{R_R}$ are to the left of the vertical segment of $u_{j+1}^{R_0}$, but to the right (if $j < q - 1$) of the vertical segment of $u_{j+2}^{R_0}$. There are such segments by $j > 1$.

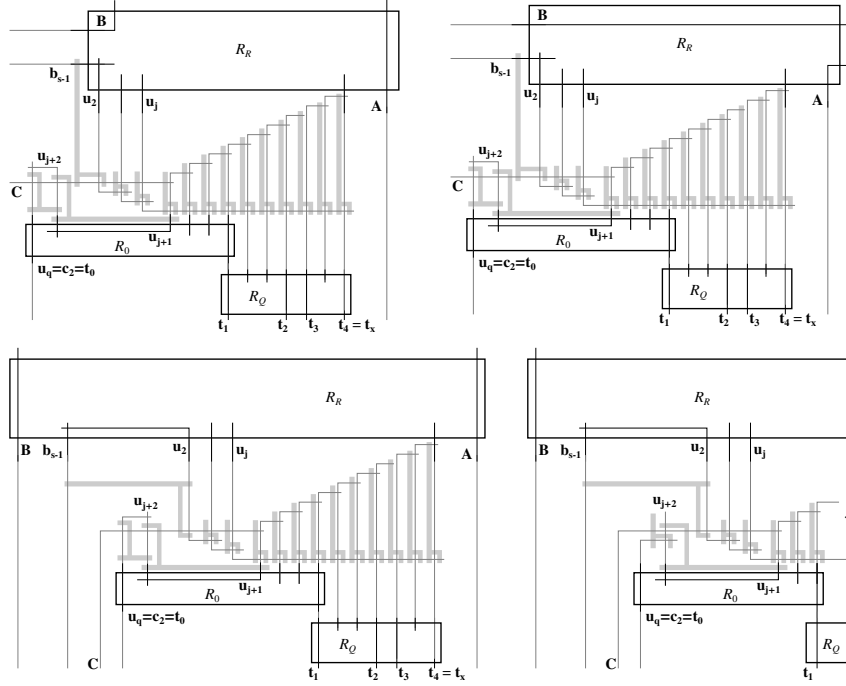
Introduce a new horizontal segment C and place it so that it intersects curves $u_q, \dots, u_{j+2}, u_2, \dots, u_j, u_{j+1}$ (after lengthening them, if needed). For a 2-sided layout also attach a vertical segment to C . If $j < q - 1$ then top-tangle u_q, \dots, u_{j+2} leftwards (see [2, Section 2.2] for a precise definition of this operation). Bottom-tangle u_2, \dots, u_j rightwards. The construction hence creates intersections for all edges in the path u_1, \dots, u_q , except for (u_{j+2}, u_{j+1}) (which was represented in R_0) and (u_2, u_1) (which was represented in R_R).

Bend and stretch $u_j^{R_R}$ rightwards so that it crosses the curves of all its neighbours in $G_0 \cup G_Q$. Finally, consider the path between the neighbours of u_j CCW from u_{j+1} to t_x . Create intersections for any edge on this path that is interior in G by top-tangling their curves rightwards.

One verifies that the curves intersect the bounding boxes as desired. The constructed representations contain private regions for all interior faces of G_R, G_Q and G_0 by induction. The remaining faces are of the form $(C, u_i, u_{i+1}), 1 \leq i < q$, and (u_j, w_k, w_{k+1}) where w_k and w_{k+1} are two consecutive neighbours of u_j on the outer-face of G_0 or G_Q . Private regions for those faces are shown in Figure 9.

Case 3(b): $j = 1$, i.e., there exists a chord (b_{s-1}, c_i) . In this case we cannot use the above construction directly since b_{s-1} ends on the left (in the 3-sided construction) while we need u_j to end at the bottom and not to be on the outer-face. However, if we use a different vertex as u_j (and argue carefully that the chord condition then holds), then the same construction works.

Recall that u_1, \dots, u_q are the neighbours of corner C in CW order starting with b_{s-1} and ending with c_2 . We know that $q \geq 3$ and u_2, \dots, u_{q-1} are not on the outer-face. Now define j' as follows: Let $u_{j'}, j' > 1$ be a neighbour of C that has at least one neighbour on P_{CA} other than C , and choose $u_{j'}$ so that j' is minimal while satisfying $j' > 1$. Such a j' exists since u_{q-1} has another neighbour on P_{CA} , and by $q \geq 3$ we have $q - 1 > 1$. Now,



■ **Figure 9** Case 3: Combining subgraphs when $\deg(C) \geq 3$, there is no chord incident with C , and $F \subseteq \{(C, c_2)\}$. (Top left) 3-sided and (top right) reverse 3-sided construction. (Bottom) 2-sided construction for the case $F = \{(C, c_2)\}$ and $F = \emptyset$. The construction matches the graph depicted in Figure 7 left.

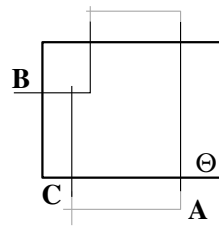
separate G as in the previous case, except use j' in place of j . Thus, define t_1, \dots, t_x to be the neighbours of $u_{j'}$ on P_{C_2A} , in order, and separate G into three graphs as follows:

- The *right* graph G_R is bounded by $(A, P_{AB}, B, P_{B u_1}, u_1, u_2, \dots, u_{j'}, t_x, P_{t_x A}, A)$.
- Let G_B be the graph bounded by $(u_{j'}, t_1, P_{t_1 t_x}, t_x, u_{j'})$. Define $G_Q := G_B - u_{j'}$.
- Let G_L be the graph bounded by $(C, P_{C t_1}, t_1, u_{j'}, C)$. Define $G_0 := G_L - \{u_{j'}, C\}$.

Observe that the boundaries of all the graphs are simple cycles, and thus they are W -triangulations. Select $(A_R := A, B_R := B, C_R := u_2)$ to be the corners of G_R and argue the chord condition as follows:

- G_R does not have any chords on $P_{C_R A_R}$ as such chords would either contradict minimality of j' , or violate the chord condition in G .
- G_R does not have any chords of $P_{A_R B_R} = P_{AB}$.
- G_R does not have any chords on $P_{B b_{s-1}}$ as it is a sub-path of P_{BC} and they would violate the chord condition in G . It also does not have any chords in the form $(C_R = u_2, b_\ell), 1 \leq \ell < s - 1$ as they would have to intersect the chord (b_{s-1}, c_i) , violating the planarity of G . Hence, G_R does not have any chords on $P_{C_R A_R}$.
- Notice in particular that the chord (u_1, c_i) of G_R is *not* a violation of the chord condition since we chose u_2 as a corner.

Hence, we can obtain a representation R_R of G_R with 2-sided, 3-sided and reverse 3-sided layout and special edge $(u_1 = b_{s-1}, u_2)$. For graphs G_Q and G_0 the corners are chosen, the chord condition is verified, and the representations are obtained exactly as in Case 3a. Since the special edge of G_R is (u_1, u_2) as before, curves \mathbf{u}_1 and \mathbf{u}_2 are situated precisely as in Case 3a, and we merge representations and find private regions as before.



■ **Figure 10** Completing a 3-sided ($int \cup (B, C)$) representation by adding intersections for (A, B) and (A, C) .

This ends the description of the construction in all cases, and hence proves Lemma 9 and Lemma 10.

4 From 4-Connected Triangulations to All Planar Graphs

In this section, we prove Theorem 1. Observe that Lemma 9 essentially proves it for 4-connected triangulations. As in [5] we extend it to all triangulations by induction on the number of separating triangles.

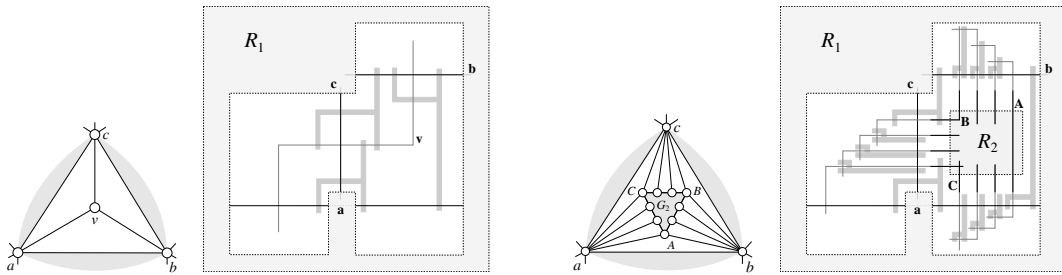
► **Theorem 12.** *Let G be a triangulation with outer-face (A, B, C) . G has a 1-string B_2 -VPG representation with a chair-shaped private region for every interior face f of G .*

Proof. Our approach is exactly the same as in [5], except that we must be careful not to add too many bends when merging subgraphs at separating triangles, and hence must use 3-sided layouts. Formally, we proceed by induction on the number of separating triangles. In the base case, G has no separating triangle, i.e., it is 4-connected. As the outer-face is a triangle, G clearly satisfies the chord condition. Thus, by Lemma 9, it has a 3-sided ($int \cup (B, C)$) representation R with private region for every face. R has an intersection for every edge except for (A, B) and (A, C) . These intersection can be created by tangling **B**, **A** and **C**, **A** suitably (see Figure 10). Recall that **A** initially did not have any bends, so it has 2 bends in the constructed representation of G . The existence of private regions is guaranteed by Lemma 9.

Now assume for induction that G has $k + 1$ separating triangles. Let $\Delta = (a, b, c)$ be an inclusion-wise minimal separating triangle of G . It was shown in [5] that the subgraph G_2 induced by the vertices inside Δ is either an isolated vertex, or a W-triangulation (A, B, C) such that the vertices on P_{AB} are adjacent to b , the vertices on P_{BC} are adjacent to c , and the vertices on P_{CA} are adjacent to a . Furthermore, G_2 satisfies the chord condition. Also, graph $G_1 = G - G_2$ is a W-triangulation that satisfies the chord condition and has k separating triangles. By induction, G_1 has a representation R_1 with a chair-shaped private region for every interior face f . Let Φ be the region for face Δ . Permute a, b, c , if needed, so that the naming corresponds to the one needed for the private region.

Case 1: G_2 is a single vertex v . Represent v by inserting into Φ an orthogonal curve \mathbf{v} with 2 bends that intersects **a**, **b** and **c**. The construction, together with private regions for the newly created faces (a, b, v) , (a, c, v) and (b, c, v) , is shown in Figure 11 (left).

Case 2: G_2 is a W-triangulation. Recall that G_2 satisfies the chord condition with respect to corners (A, B, C) . Apply Lemma 9 to construct a 3-sided ($int \cup (C, b_{s-1})$) representation R_2 of G_2 . Let us assume that (after possible rotation) Φ has the orientation shown in



■ **Figure 11** Separating triangle with one vertex and the construction (left), and separating triangle enclosing a W -triangulation and the construction (right).

Figure 11 (right); if it had the symmetric orientation then we would do a similar construction using a reverse 3-sided representation of G_2 . Place R_2 inside Φ as shown in Figure 11 (right). Stretch the curves representing vertices on P_{CA} , P_{AB} and $P_{Bb_{s-1}}$ downwards, upwards and leftwards respectively so that they intersect a , b and c . Top-tangle leftwards the curves $\mathbf{A} = \mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_r = \mathbf{B}$. Left-tangle downwards the curves $\mathbf{B} = \mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_{s-1}$ and bend and stretch \mathbf{C} downwards so that it intersects a . Bottom-tangle leftwards the curves $\mathbf{C} = \mathbf{c}_1, \dots, \mathbf{c}_t = \mathbf{A}$. It is easy to verify that the construction creates intersection for all the edges between vertices of Δ and the outer-face of G_2 . The tangling operation then creates intersections for all the outer-face edges of G_2 except edge (C, b_{s-1}) , which is already represented in R_2 .

Every curve that receives a new bend represents a vertex on the outer-face of G_2 , which means that it initially had at most 1 bend. Curve \mathbf{A} is the only curve that receives 2 new bends, but this is allowed as \mathbf{A} does not have any bends in R_2 . Hence, the number of bends for every curve does not exceed 2.

Private regions for faces formed by vertices a, b, c and vertices on the outer-face of G_2 can be found as shown in Figure 11 right. ◀

With Theorem 12 in hand, we can show our main result: every planar graph has a 1-string B_2 -VPG representation.

Proof of Theorem 1. If G is a planar triangulated graph, the claim holds by Theorem 12. So, assume that G is a planar graph. Then *stellate* the graph, i.e., insert a vertex into each non-triangulated face and connect it to all vertices on that face. It is well known that after at most 3 repetitions, the construction produces a 3-connected triangulated graph G' such that G is an induced subgraph of G' . Apply Theorem 12 to construct a 1-string B_2 -VPG representation R' of G' . By removing curves representing vertices that are not in G , we obtain a 1-string B_2 -VPG representation of G . ◀

5 Conclusions and Outlook

We showed that every planar graph has a 1-string B_2 -VPG representation, i.e., a representation as an intersection graph of strings where strings cross at most once and each string is orthogonal with at most two bends. One advantage of this is that the coordinates to describe such a representation are small, since orthogonal drawings can be deformed easily such that all bends are at integer coordinates. Every vertex curve has at most two bends and hence at most 3 segments, so the representation can be made to have coordinates in an $O(n) \times O(n)$ -grid with perimeter at most $3n$. Note that none of the previous results provided an intuition of the required size of the grid.

Following the steps of our proof, it is not hard to see that our representation can be found in linear time, since the only non-local operation is to test whether a vertex has a neighbour on the outer-face. This can be tested by marking such neighbours whenever they become part of the outer-face. Since no vertex ever is removed from the outer-face, this takes overall linear time.

The representation constructed in this paper uses curves of 8 possible shapes for planar graphs. One can in fact verify that the 2-sided layout (which only uses 2-sided layouts in its recursions) uses only 4 possible shapes: C, Z and their horizontal mirror images. Hence for triangulations without separating triangles (and, after stellating, all 4-connected planar graphs) 4 shapes suffice. A natural question is if one can restrict the number of shapes required to represent all planar graphs.

Bringing this effort further, is it possible to restrict the curves even more? Felsner et al. [10] asked the question whether every planar graph is the intersection graph of only two shapes, namely $\{L, \Gamma\}$. As they point out, a positive result would provide a different proof of Scheinerman's conjecture. Somewhat inbetween: is every planar graph the intersection graph of xy -monotone orthogonal curves, preferably in the 1-string model and with few bends?

References

- 1 Takao Asano, Shunji Kikuchi, and Nobuji Saito. A linear algorithm for finding Hamiltonian cycles in 4-connected maximal planar graphs. *Discr. Applied Mathematics*, 7(1):1–15, 1984.
- 2 Therese C. Biedl and Martin Derka. 1-string B_2 -VPG representation of planar graphs. *CoRR*, abs/1411.7277, 2014.
- 3 Jérémie Chalopin and Daniel Gonçalves. Every planar graph is the intersection graph of segments in the plane: extended abstract. In *ACM Symposium on Theory of Computing, STOC 2009*, pages 631–638. ACM, 2009.
- 4 Jérémie Chalopin, Daniel Gonçalves, and Pascal Ochem. Planar graphs are in 1-string. In *ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, pages 609–617. SIAM, 2007.
- 5 Jérémie Chalopin, Daniel Gonçalves, and Pascal Ochem. Planar graphs have 1-string representations. *Discrete & Computational Geometry*, 43(3):626–647, 2010.
- 6 Steven Chaplick and Torsten Ueckerdt. Planar graphs as VPG-graphs. *J. Graph Algorithms Appl.*, 17(4):475–494, 2013.
- 7 Natalia de Castro, Francisco Javier Cobos, Juan Carlos Dana, Alberto Márquez, and Marc Noy. Triangle-free planar graphs and segment intersection graphs. *J. Graph Algorithms Appl.*, 6(1):7–26, 2002.
- 8 Hubert de Fraysseix, Patrice Ossona de Mendez, and János Pach. Representation of planar graphs by segments. *Intuitive Geometry*, 63:109–117, 1991.
- 9 Gideon Ehrlich, Shimon Even, and Robert Endre Tarjan. Intersection graphs of curves in the plane. *J. Comb. Theory, Ser. B*, 21(1):8–20, 1976.
- 10 Stefan Felsner, Kolja B. Knauer, George B. Mertzios, and Torsten Ueckerdt. Intersection graphs of L -shapes and segments in the plane. In *Mathematical Foundations of Computer Science (MFCS'14), Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 299–310. Springer, 2014.
- 11 Irith Ben-Arroyo Hartman, Ilan Newman, and Ran Ziv. On grid intersection graphs. *Discrete Mathematics*, 87(1):41–52, 1991.
- 12 Edward R. Scheinerman. *Intersection Classes and Multiple Intersection Parameters of Graphs*. PhD thesis, Princeton University, 1984.
- 13 Hassler Whitney. A theorem on graphs. *The Annals of Mathematics*, 32(2):387–390, 1931.

Spanners and Reachability Oracles for Directed Transmission Graphs*

Haim Kaplan¹, Wolfgang Mulzer², Liam Roditty³, and Paul Seiferth²

1 School of Computer Science, Tel Aviv University, Israel
haimk@post.tau.ac.il

2 Institut für Informatik, Freie Universität Berlin, Germany
{mulzer,pseiferth}@inf.fu-berlin.de

3 Department of Computer Science, Bar Ilan University, Israel
liamr@macs.biu.ac.il

Abstract

Let $P \subset \mathbb{R}^d$ be a set of n points, each with an associated radius $r_p > 0$. The *transmission graph* G for P has vertex set P and an edge from p to q if and only if q lies in the ball with radius r_p around p . Let $t > 1$. A t -*spanner* H for G is a sparse subgraph of G such that for any two vertices p, q connected by a path of length ℓ in G , there is a p - q -path of length at most $t\ell$ in H . We show how to compute a t -spanner for G if $d = 2$. The running time is $O(n(\log n + \log \Psi))$, where Ψ is the ratio of the largest and smallest radius of two points in P . We extend this construction to be independent of Ψ at the expense of a polylogarithmic overhead in the running time. As a first application, we prove a property of the t -spanner that allows us to find a BFS tree in G for any given start vertex $s \in P$ in the same time.

After that, we deal with *reachability oracles* for G . These are data structures that answer *reachability queries*: given two vertices, is there a directed path between them? The quality of an oracle is measured by the space $S(n)$, the query time $Q(n)$, and the preprocessing time. For $d = 1$, we show how to compute an oracle with $Q(n) = O(1)$ and $S(n) = O(n)$ in time $O(n \log n)$. For $d = 2$, the radius ratio Ψ again turns out to be an important measure for the complexity of the problem. We present three different data structures whose quality depends on Ψ : (i) if $\Psi < \sqrt{3}$, we achieve $Q(n) = O(1)$ with $S(n) = O(n)$ and preprocessing time $O(n \log n)$; (ii) if $\Psi \geq \sqrt{3}$, we get $Q(n) = O(\Psi^3 \sqrt{n})$ and $S(n) = O(\Psi^5 n^{3/2})$; and (iii) if Ψ is polynomially bounded in n , we use probabilistic methods to obtain an oracle with $Q(n) = O(n^{2/3} \log n)$ and $S(n) = O(n^{5/3} \log n)$ that answers queries correctly with high probability. We employ our t -spanner to achieve a fast preprocessing time of $O(\Psi^5 n^{3/2})$ and $O(n^{5/3} \log^2 n)$ in case (ii) and (iii), respectively.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems – Geometrical Problems and Computations

Keywords and phrases Transmission Graphs, Reachability Oracles, Spanner, Intersection Graph

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.156

1 Introduction

A common model for wireless sensor networks is the *unit-disk graph*: each sensor p is modeled by a unit disk centered at p , and there is an edge between two sensors iff their disks intersect [7]. Intersection graphs of disks with arbitrary radii have also been used to model

* This work is supported by GIF project 1161 & DFG project MU/3501/1.



sensors with different transmission radii [2, Chapter 4]. Intersection graphs of disks are undirected, however. For some networks we may want a directed model. In such networks, a sensor p that can transmit information to a sensor q may not be able to receive information from q . This motivated various researchers to consider what we call here *transmission graphs* [16, 15]. A transmission graph G is defined for a set of points $P \subset \mathbb{R}^2$ where each point $p \in P$ has a (transmission) radius r_p associated with it. Each vertex of G corresponds to a point of P , and there is a directed edge from p to q iff q lies in the disk $D(p)$ of radius r_p around p . We can weight each edge pq of G by its Euclidean length $|pq|$ and treat G as a weighted graph. We study (approximate) shortest path and reachability problems for transmission graphs.

Even though transmission graphs have a linear size representation, they may be very dense, even with $\Theta(n^2)$ edges (similar to many other geometric intersection graphs). Thus, if one applies a standard graph algorithm, like breadth first search (BFS), to a dense transmission graph, it runs slowly, since it requires an explicit representation of all the edges in the graph. Thus, given an transmission graph G implicitly as points with radii, it is desirable to construct a sparse approximation of G that preserves connectivity and proximity properties. For any $t > 1$, a subgraph H of G is a t -spanner for G if the distance between any pair of vertices p and q in H is at most t times the distance between p and q in G , i.e., $d_H(p, q) \leq t \cdot d_G(p, q)$ for any pair p, q (see [14] for an overview of spanners for geometric graphs). Fürer and Kasiviswanathan show how to compute a t -spanner for unit- and general disk graphs using a variant of the Yao graph [9, 17]. Peleg and Roditty [15] give a construction for t -spanners in transmission graphs in any metric space with bounded doubling dimension. However, except for the unit-disk case, the running times of these algorithms depend on the number of edges in the intersection graph. We avoid this dependency and give an almost linear time algorithm that constructs a t -spanner of a transmission graph for the Euclidean metric in the plane. Our construction is based on the *Yao graph* [17]. The basic Yao graph is a spanner for the complete graph defined by n points in the plane (with Euclidean distances). To determine the points adjacent to a particular point q , we divide the plane by equally spaced rays emanating from q and connect q to the closest point in each wedge (the number of wedges increases as t gets smaller). Transmission graphs, being directed, pose a severe computational difficulty as we want to consider, in each wedge, only the points p with $q \in D(p)$ and pick the closest to q only among those. Our spanner construction generalizes the Yao graph in this manner. We further need to relax this construction in a subtle way, without hurting the approximation too much, in order to construct the spanner efficiently. Even with a good approximation in terms of a t -spanner at hand, we sometimes wish to obtain exact solutions for certain problems on disk graphs. Working in this direction, Cabello and Jeřčič gave an $O(n \log n)$ time algorithm for computing a BFS tree in a unit-disk graph, rooted at any given vertex [3]. For this, they exploited the special structure of the Delaunay triangulation of the disk centers. We show that our spanner admits similar properties for transmission graphs. As a first application of our spanner, we get an efficient algorithm to compute a BFS tree in a transmission graph.

A classical data structure problem for a directed graph G is to construct a space efficient *reachability oracle* that can answer *reachability queries* quickly. In a reachability query we get two vertices p and q and we would like to determine if there is a directed path from p to q . The quality of a reachability oracle for a graph G with n vertices is measured by three parameters: the query time $Q(n)$, the space requirement $S(n)$, and the preprocessing time. In the planar case, efficient reachability oracles exist and a recent result by Holm, Rotenberg and Thorup achieves optimal parameters [11]. However, for general directed graphs, there are no nontrivial results, and special cases, such as transmission graphs, are of great interest.

We give efficient constructions of reachability oracles for transmission graphs by exploiting their geometry. For points in 1D, we give an $O(n)$ space oracle with query time $O(1)$. In 2D it turns out that the ratio Ψ of the largest and smallest radius of points in P is an important complexity measure for transmission graphs. We give three oracles for different ranges of Ψ .

Our Contribution and Organization of the Paper. In Section 2, we show how to compute, for every fixed $t > 1$, a t -spanner H of G . Our construction is quite generic and can be adapted to several situations. In the simplest case, if the *spread* Φ (i.e., the ratio between the largest and the smallest distance in P) is bounded, we can obtain a t -spanner in time $O(n(\log n + \log \Phi))$ (Section 2.1). With a little more work, we can weaken the assumption to a bounded *radius ratio* Ψ (the ratio between the largest and smallest radius in P), giving a running time of $O(n(\log n + \log \Psi))$ (Section 2.2). Using even more advanced data structures, we can compute a t -spanner in expected time $O(n \log^6 n)$, without any dependence on Φ or Ψ (Section 2.3). Our spanners have several applications. For one, we adapt a result by Cabello and Jeřić [3] to show that once a spanner is at hand, we can compute the BFS-tree of any given vertex $p \in P$ with additional time $O(n \log n)$ (Section 2.4). Furthermore, we use t -spanners to obtain efficient preprocessing algorithms for reachability oracles.

The remaining paper is dedicated to these reachability oracles. We will see that in 1D transmission graphs admit a rich structure that can be exploited to construct a simple linear space reachability oracle with constant query time and $O(n \log n)$ preprocessing time. This construction is described in Section 3. Unfortunately, in 2D most of their structure vanishes. However, if the radius ratio Ψ is less than $\sqrt{3}$, we show how to make the transmission graph planar in $O(n \log n)$ time, while preserving the reachability structure and keeping the number of vertices linear in n . Now we can construct a reachability oracle for the resulting planar graph. A recent construction of Holm, Rotenberg and Thorup [11] gives a distance oracle for planar graphs in linear time that takes linear space and can answer a query in $O(1)$ time. This construction is in Section 4.1. When $\Psi \geq \sqrt{3}$ we do not know how to planarize G . Fortunately, we can use a separation theorem by Alber and Fiala that allows us to find a small and balanced separator with respect to the area of the union of the disks [1]. This allows us to build a reachability oracle with query time $O(\Psi^3 \sqrt{n})$ and space and preprocessing time $O(\Psi^5 n^{3/2})$. See Section 4.2. When Ψ is even larger but still polynomially bounded in n , we use random sampling combined with a quad tree of logarithmic depth to obtain a reachability oracle with query time $O(n^{2/3} \log n)$, space $O(n^{5/3} \log n)$, and preprocessing time $O(n^{5/3} \log^2 n)$. Refer to Section 4.3.

Many of our constructions rely on planar grids. For $i = 0, 1, \dots$, we define \mathcal{Q}_i to be the *grid at level i* . It consists of axis-parallel squares with diameter 2^i that partition the plane in grid-like fashion (the *cells*). \mathcal{Q}_i is aligned so that the origin is a vertex of the grid. The *distance* between two grid cells is the smallest distance of any two points contained in them. Furthermore, we assume that the input is scaled so that the distance of the closest pair in P is 1. We assume that in our model of computation we can find for any given point the grid cell that contains it in $O(1)$ time. For space reasons, all proofs in this extended abstract are omitted. We refer the interested and ambitious reader to the full version.

2 Spanners and BFS Trees

2.1 Efficient Spanner Construction

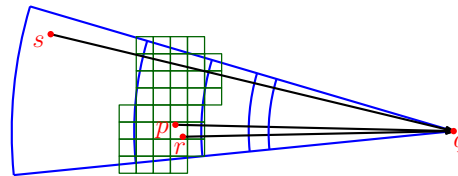
Let $P \subset \mathbb{R}^2$ be a point set with radii, and let $\Phi = \max_{p,q \in P} |pq| / \min_{p \neq q \in P} |pq|$ be the *spread* of P . First, we give a spanner construction for the transmission graph G of P that depends on the spread of P . In Section 2.2, we will weaken this to a dependence on the radius ratio.

► **Theorem 2.1.** *Let G be the transmission graph for a two-dimensional n -point set P with spread Φ . For any $t > 1$, we can compute a t -spanner for G in time $O(n(\log n + \log \Phi))$.*

Our construction creates a subgraph H of G that is similar to the Yao graph [17], but modified to take the disks into account. Ideally, our spanner should look as follows: we pick a suitable integer k , and we take a set \mathcal{C} of k cones with opening angle $2\pi/k$ that partition the plane and that have the origin as apex. For each vertex $q \in P$, we attach the cones in \mathcal{C} to q , and in each translated cone we pick the closest vertex $p \in P$ with $q \in D(p)$. We add the edge pq to H . The resulting graph has $O(kn)$ edges, and using standard techniques, one can show that it is a t -spanner for large enough k . This construction seems to be folklore [5, 15].

However, the standard algorithms for computing the Yao graph do not seem to adapt easily for our setting without affecting the running time. Thus, we need a more sophisticated construction that gives a graph with similar properties. The idea is to partition each cone C_q attached to q into “intervals” obtained by intersecting C_q with annuli centered at q whose inner and outer radius grows exponentially; see Figure 1. Each of these intervals is discretized by covering it with $O(1)$ grid cells whose diameter is relatively small compared to the distance between the interval and q . This enforces two properties that help us to find an approximately shortest incoming edge for q in C_q : we only need to consider edges from the interval that is closest to q since these edges will be shorter than any edge from any later interval; and if there are multiple edges from the same grid cell to q , it suffices to pick only one of them since their endpoints are close together.

We define a decomposition of P that represents the discretized intervals by a neighborhood relation between grid cells. Given this decomposition, there is a simple (rather inefficient) rule how to pick incoming edges for each $q \in P$ such that the resulting graph H is a spanner. We first give the definition of the decomposition and prove that H is a t -spanner if we pick the parameters appropriately.



■ **Figure 1** A cone C_q covered by discretized intervals. We only need one of the edges \vec{pq} , $\vec{r\hat{q}}$ for H .

Then we show how compute the decomposition using a quadtree T . Finally, we use the structure of T to find the edges within the desired time bound. Let $c > 2$ be a large constant. For a grid cell σ , let m_σ be the point in $P \cap \sigma$ with the largest radius.

► **Definition 2.2.** Let G be the transmission graph of a point set $P \subset \mathbb{R}^2$. A c -separated annulus decomposition for G consists of a finite set $\mathcal{Q} \subset \bigcup_{i=0}^\infty \mathcal{Q}_i$, a symmetric neighborhood relation $N \subseteq \mathcal{Q} \times \mathcal{Q}$, and assigned sets $R_\sigma \subseteq P \cap \sigma$ for each $\sigma \in \mathcal{Q}$ so that (i) for all $(\sigma, \sigma') \in N$, $\text{diam}(\sigma) = \text{diam}(\sigma')$ and $d(\sigma, \sigma') \in [(c - 2)\text{diam}(\sigma), 2c\text{diam}(\sigma)]$; and (ii) for every edge \vec{pq} of G , there is a $(\sigma, \sigma') \in N$ with $p \in \sigma$, $q \in \sigma'$, and with either $p \in R_\sigma$ or $q \in D(m_\sigma)$.

For $\sigma \in \mathcal{Q}$, we define $N(\sigma) = \{\sigma' \mid (\sigma, \sigma') \in N\}$. Definition 2.2(i) implies $|N(\sigma)| = O(1)$.

Getting a Spanner. Let $t > 1$ be the desired stretch. Depending on t , we pick suitable constants c (separation parameter) and k (number of cones). Let \mathcal{Q} be a c -separated annulus decomposition for G . To obtain a t -spanner $H \subseteq G$, we pick the incoming edges for each point $q \in P$ and each cone $C \in \mathcal{C}$ as in Alg. 1. For $\sigma \in \mathcal{Q}$ let C_σ be the translated copy of C that has the center of σ as apex and let C_σ^2 be the cone obtained by doubling the opening angle of C_σ . Instead of C_q we use the cones C_σ^2 with $q \in \sigma$ to find incoming edges for q . This gives the generality needed for later extensions of this algorithm.

```

1  $\mathcal{Q}_q \leftarrow$  cells of  $\mathcal{Q}$  that contain  $q$ 
2 Sort  $\mathcal{Q}_q$  by the diameter of the cells in increasing order; give  $q$  the status active
3 while  $q$  is active do
4    $\sigma \leftarrow$  next largest cell in  $\mathcal{Q}_q$ 
5   foreach cell  $\sigma' \in N(\sigma)$  that is contained in  $C_\sigma^2$  do
6     if there is a  $r \in R_{\sigma'} \cup \{m_{\sigma'}\}$  with  $q \in D(r)$  then
7       take an arbitrary such  $r$ , add the edge  $rq$  to  $H$ , and set  $q$  to be inactive.

```

Algorithm 1: Selecting the incoming edges for q and the cone C_q .

For each cone $C \in \mathcal{C}$ and each $q \in P$ there is only one $\sigma \in \mathcal{Q}_q$ that produces incoming edges for q : after σ is processed, q is inactive. Since we have k cones and since $|N(\sigma)| = O(1)$, q has $O(k)$ incoming edges, for a total of $O(n)$ edges in H . To show that H is a t -spanner, we use induction on the rank of the edge lengths. The proof is done in a similar manner as for standard Yao graphs, but with a few additional twists.

► **Lemma 2.3.** *For any $t > 1$, there are constants c and k such that H is a t -spanner for G .*

Finding the Decomposition. Let $c > 3$ be the separation parameter. We scale P s.t. the smallest distance in P is c . A *quadtree* for P is a rooted tree T in which each internal node has degree four. Each node v of T has an associated cell σ_v from a grid \mathcal{Q}_i , $i \geq 0$, and we say that v has *level* i . If v is an internal node, the cells of its four children partition σ_v into four congruent squares, each with half the diameter of σ_v . We describe how to compute a quadtree T for P s.t. the cells of T form the set \mathcal{Q} for the c -separated annulus decomposition.

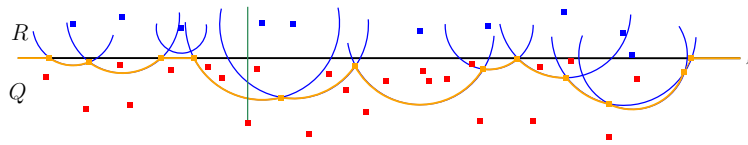
We construct T level-wise. Let L be the smallest integer such that there is a cell $\sigma \in \mathcal{Q}_L$ that (possibly after shifting) contains P . Since c is constant and since P has spread Φ , the scaled point set has diameter $c\Phi$, and we can take $L = O(\log \Phi)$. We create the root node v and set $\sigma_v = \sigma$. This will be level L . To construct level $i - 1$, given level i , we do the following for each level i node v whose cell σ_v is non-empty: take the four cells of \mathcal{Q}_{i-1} that partition σ_v and create four nodes w_1, \dots, w_4 . To each of the four nodes w_1, \dots, w_4 we assign one of the four cells, and we make w_1, \dots, w_4 children of v . This process stops at level 0. The scaling of P ensures that a cell of a level 0 node contains at most one point of P .

We now set $\mathcal{Q} = \{\sigma_v \mid v \in T\}$. We let $(\sigma_v, \sigma_w) \in N$ if v and w have the same level and if $d(\sigma_v, \sigma_w) \in [(c-2) \text{diam}(\sigma_v), 2c \text{diam}(\sigma_v)]$. As R_{σ_v} we take all points in $\sigma_v \cap P$ whose radius is between $(c-2) \text{diam}(\sigma_v)$ and $2(c+1) \text{diam}(\sigma_v)$. To see that this satisfies Definition 2.2(ii), consider an edge pq of G with $q \in \sigma_v$ and $p \in \sigma' \in N(\sigma_v)$. Since $D(p)$ must intersect σ_v , we have $r_p \geq (c-2) \text{diam}(\sigma_v)$. Thus, we have either $p \in R_{\sigma'}$ or $r_p > 2(c+1) \text{diam}(\sigma')$. In the second case for any $r \in \sigma'$ with radius $r_r \geq r_p$ the disk $D(r)$ fully contains σ_v . In particular this holds for $r = m_{\sigma'}$. Since Def. 2.2(i) is satisfied by construction, we get the next lemma.

► **Lemma 2.4.** *The set \mathcal{Q} with N and the assignment R_σ described above is a c -separated annulus decomposition for G .*

Finding the Edges. To find the edges for the spanner H more quickly, we use the cells of \mathcal{Q} to group the points and find incoming edges for multiple points at once. We process the cells of \mathcal{Q} by increasing diameter, following the structure of the quadtree T .

Fix one cone $C \in \mathcal{C}$ of the k cones we want to process. For $\sigma \in \mathcal{Q}$, let C_σ^2 be the cone with opening angle $4\pi/k$ whose apex is the center of σ obtained by translating C and doubling its opening angle. We give all points in P the status *active*. We process T in level-order,



■ **Figure 2** The lower envelope (orange), the points Q (red) and R (blue), and the sweepline (green).

starting with level 0. For each $v \in T$, we select incoming edges for the active points Q in $\sigma_v \cap P$ as in Algorithm 2. First we sort Q by x and y -direction in linear time, using the sorted lists of v 's children (preprocessing). Let $\sigma' \in N(\sigma_v)$ be a neighbor of σ_v . The sorting enables us to efficiently find incoming edges for points in Q from points in $R = R_{\sigma'} \cup \{m_{\sigma'}\}$ (edge selection): Q and R are separated by a line ℓ that is parallel to either the x or the y axis, namely one of the supporting lines of the boundary of σ_v . We can compute the lower envelope E of the disks in R and sweep over Q in ℓ direction, see Fig. 2. This takes time linear in $|Q|$ since Q is sorted in ℓ direction. To check whether the current point $q \in Q$ is contained in a disk of R , we only need to test the disk of the arc of E intersected by the sweepline through q orthogonal to ℓ . We summarize the above discussion in Lemma 2.5.

► **Lemma 2.5.** *Let Q, R , and ℓ be as above with $|Q| = n$ and $|R| = m$. Suppose that Q is sorted along ℓ and that ℓ separates Q and R . We can compute in time $O(m \log m + n)$ for each $q \in Q$ one disk from R that contains it, provided that such a disk exists.*

The edges selected by Algorithm 2 have the same properties as the edges selected by Algorithm 1. Thus, by Lemma 2.3, the resulting graph is a t -spanner.

```

1 for  $i = 0, \dots, L$  do
2   foreach  $v \in T$  of level  $i$  do
3      $Q \leftarrow$  active points in  $\sigma_v \cap P$ 
4     // preprocessing
5     Sort  $Q$  in  $x$  and  $y$ -direction by merging the sorted lists of the children of  $v$ 
6     foreach  $\sigma' \in N(\sigma_v)$  contained in  $C_{\sigma_v}^2$  do
7        $R \leftarrow R_{\sigma'} \cup \{m_{\sigma'}\}$ 
8       // edge selection
9       For each  $q \in Q$  find a  $r \in R$  with  $q \in D(r)$ , if it exists; add the edge  $\overrightarrow{rq}$  to  $H$ 
10      Set all  $q \in Q$  for which at least one incoming edge was found to inactive

```

Algorithm 2: Selecting the edges for H for a fixed cone C .

Running Time. By Lemma 2.5, we can argue that the running time of Algorithm 2 is dominated by the edge selection step. Since T has depth $O(\log \Phi)$, each $p \in P$ takes part in $O(\log \Phi)$ edge selections as a point in Q for incoming edges, taking $O(1)$ time for that point (by Lemma 2.5). Furthermore, each point is in $O(1)$ different sets R_σ and thus takes part in $O(1)$ edges selections as a disk-center in R , taking $O(\log |R|) = O(\log n)$ time for that point. Thus, we have a total running time of $O(n(\log \Phi + \log n))$, as stated in the next lemma.

► **Lemma 2.6.** *The construction of the spanner H of G takes $O(n(\log \Phi + \log n))$ time.*

Theorem 2.1 follows by combining Lemmas 2.3 and 2.6.

2.2 From Bounded Spread to Bounded Radius Ratio

Let $P \subset \mathbb{R}^2$ be a point set with radii and let Ψ be the radius ratio of P . We extend the spanner construction from Section 2.1 to be depended on the radius ration Ψ of P .

► **Theorem 2.7.** *Let G be the transmission graph for a n -point set $P \subset \mathbb{R}^2$ with radius ratio Ψ . For any $t > 1$, we can compute a t -spanner for G in $O(n(\log n + \log \Psi))$ time.*

The main observation is that the spread is irrelevant in our setting: points that are close together form a clique in G and can be handled through classic spanners, and points that are far away from each other form distinct components and can be dealt with independently.

Given t , we pick large enough constants k and c . Then, we scale the input such that the smallest radius is c . Let $M = O(\Psi)$ be the largest radius. First, we partition P into sets that are far away from each other and can be handled separately.

► **Lemma 2.8.** *In $O(n \log n)$ time, we can partition P into sets P_1, \dots, P_ℓ so that each P_i has diameter $O(n\Psi)$ and so that for any $i \neq j$, no point in P_i can reach a point of P_j in G .*

By Lemma 2.8, we may assume that our input point set has diameter $O(n\Psi)$. As in Section 2.1, we can compute a quadtree T for P with L levels and $L = O(\log(n\Psi))$: take a large enough grid cell that contains P and recursively subdivide each non-empty cell into four cells of half the diameter. We stop when the diameter of the cells is 1. Unlike in Section 2.1, the set of the cells of all nodes of T does not yield a c -separated annulus decomposition for G . In particular, Definition 2.2(ii) is not true anymore. Therefore, there can be edges in G that do not go between neighboring cells. These are the *short edges*.

First, we handle *very short edges*: let $v \in T$ be a level 0 node and let $\sigma_v \in \mathcal{Q}_0$ be the cell of v . Let $Q \subseteq P$ be all points that lie in cells of \mathcal{Q}_0 with distance at most $c/2 - 3$ from σ . Since any pair of points in Q has distance at most c , the set Q forms a clique in G . We compute a (classic) t -spanner for Q in $O(|Q| \log |Q|)$ time [14]. Since any $p \in P$ participates in $O(c^2)$ such spanners, we generate $O(n)$ edges in total and require $O(n \log n)$ time.

Second, we handle *not quite so short edges*: for each $q \in P$, let v be the level 0 node of T whose cell σ_v contains q . For any cell $\sigma \in \mathcal{Q}_0$ with $d(\sigma_v, \sigma) \in (c/2 - 3, c - 2)$, we take an arbitrary point $r \in \sigma \cap P$ and add the edge $\vec{r}\vec{q}$ to our spanner. All these edges have length at most c and are therefore valid edges in G . This takes $O(n)$ time and creates $O(n)$ edges.

Finally, we handle the remaining edges: for this, we mark all points of P as active, and we run Algorithm 2 from Section 2.1 starting from level 0 of T . Call the resulting graph H .

As in Lemma 2.3, induction on the rank of the edges lengths shows that H is a t -spanner.

► **Lemma 2.9.** *The graph H is a t -spanner for G if c and k are large enough constants.*

Using Lemma 2.9, Theorem 2.7 follows in the same way as in Section 2.1. The running time analysis goes exactly as in Lemma 2.6, but the quadtree now has $O(\log n + \log \Psi)$ levels.

2.3 Spanners for Unbounded Spread and Radius Ratio

We show how eliminate the bounded radius ratio assumption at the expense of using a more involved data structure and of losing a polylog factor in the running time. Let $P \subset \mathbb{R}^2$ and the desired stretch factor $t > 1$ be given. Assume that the closest pair in P has distance 1.

First we compute a *compressed quadtree* T for P . It is a rooted tree in which each internal node has degree 1 or 4. Each node v has an associated cell σ_v from a grid \mathcal{Q}_i . To keep the notation simple, we write $\text{diam}(v)$ for $\text{diam}(\sigma_v) = 2^i$, and for two nodes v, w , we write $d(v, w)$ for $d(\sigma_v, \sigma_w)$. If v has degree 4, then the associated cells of its children partition

σ_v into 4 congruent squares of half the diameter, and at least two of them are non-empty. If v has degree 1, then the associated cell of the only child w of v has diameter at most $\text{diam}(v)/4$. Furthermore, there are no points from P in $\sigma_v \setminus \sigma_w$. Each internal node of T contains at least 2 points from P in its cell and each leaf at most 1 point. A compressed quadtree for P with $O(n)$ nodes can be computed in $O(n \log n)$ time [10].

Our goal is to use the algorithm from Section 2.1 on the compressed quadtree T . There are two problems with this: since the depth of T can be linear, we cannot consider all points for incoming edges in each level, as in Algorithm 2. Instead we use Chan's dynamic nearest neighbor data structure to quickly identify the relevant points. It has the following properties.

► **Theorem 2.10** (Chan [6]). *There exists a dynamic data structure that maintains a planar point set S such that (i) we can insert a point into S in expected, amortized time $O(\log^3 n)$; (ii) we can delete a point from S in expected, amortized time $O(\log^6 n)$; and (iii) given a query point q , we can find the nearest neighbor for q in S in worst-case time $O(\log^2 n)$.*

Furthermore, the cells of T do not form a c -separated annulus decomposition anymore. The notion of neighborhood needs to be adapted to accommodate internal nodes of degree 1 and to ensure that Definition 2.2(ii) holds. We fix this by inserting $O(n)$ additional nodes into T that have the desired properties. To find these nodes, we use the well-separated pair decomposition algorithm of Callahan and Kosaraju [4]. Let a large enough constant c be given. As in Section 2.1, we define the neighborhood relation N as the pairs (σ_v, σ_w) whose nodes v and w have the same level in T and that satisfy $d(\sigma_v, \sigma_w) \in [(c-2) \text{diam}(\sigma_v), 2c \text{diam}(\sigma_v))$. The set R_{σ_v} are all points in $\sigma_v \cap P$ whose radius is between $(c-2) \text{diam}(\sigma_v)$ and $2(c+1) \text{diam}(\sigma_v)$.

► **Lemma 2.11.** *For any $c > 0$ we can in $O(n \log n)$ time insert $O(n)$ nodes into the compressed quadtree T s.t. $\mathcal{Q} = \{\sigma_v \mid v \in T\}$ with N and the assignment R_{σ} is a c -separated annulus decomposition for G . In the same time we can compute N and R_{σ_v} for each $\sigma_v \in \mathcal{Q}$.*

Finding the Edges. To find the edges for the spanner $H \subseteq G$, we choose constants k and c depending on t . The algorithm proceeds as follows: we compute a compressed quadtree T for P . To obtain a c -separated annulus decomposition \mathcal{Q} , N , R_{σ_v} for G , we augment T with $O(n)$ nodes as in Lemma 2.11. We create the dynamic nearest neighbor (NN) data structure from Theorem 2.10 for each leaf node v of T whose cell σ_v is non-empty. We sort all cells of nodes of T by increasing diameter. A point is called *active* if it is in the NN data structure of some v , thus initially all points of P are active. Fix a cone C . For $\sigma \in \mathcal{Q}$ let C_{σ}^2 be the cone whose apex is the center of σ and such that C_{σ}^2 is obtained from C by translating and doubling the opening angle to $4\pi/k$. To select the spanner edges for C , we consider the nodes of T in increasing order and perform two steps for each node v , similar to Algorithm 2 of Section 2.1: let w be the child of v that has the most active points in its NN structure. To get the NN data structure for v , we insert all active points of the remaining children of v into the NN data structure of w (preprocessing). Since w has the most points, overall each point is inserted $O(\log n)$ times in some NN structure. Then we do the edge selection for all $\sigma' \in N(\sigma_v)$ contained in $C_{\sigma_v}^2$ using the NN structure of v ; see Algorithm 3. We take each point $r \in R_{\sigma'} \cup \{m_{\sigma'}\}$ and repeatedly query the NN structure of v . Let q be the result. If rq constitutes an edge in G , we call the query *successful*, add rq to H , delete q , and do another query with r . Otherwise, we proceed with the next point of R . Each such query causes $O(1)$ additional insertion/deletions to a NN structure. If it was successful, we charge these costs to the created edge. Otherwise, we charge the costs to this point r . Since each point $p \in P$ is in $O(1)$ sets R_{σ} , it can only be responsible for $O(1)$ unsuccessful queries. Thus, since H has n vertices and $O(n)$ edges, we can prove the next lemma.

► **Lemma 2.12.** *The algorithm has total expected running time $O(n \log^6 n)$.*

The edges selected by this procedure have the same properties as the edges selected by Algorithm 1. Thus, by Lemma 2.3 we obtain a t -spanner H , which establishes Theorem 2.10.

```

// preprocessing
1 Let  $w$  be the child of  $v$  whose NN structure contains the most points
2 Insert all points of each child  $w' \neq w$  of  $v$  into the NN structure of  $w$ 
3 foreach  $\sigma' \in N(\sigma_v)$  contained in  $C_{\sigma_v}^2$  do
4   foreach  $r \in R = R_{\sigma'} \cup \{m_{\sigma'}\}$  do
5     // edge selection
6      $q \leftarrow \text{NN}(v, r)$  // query NN structure of  $v$  with  $r$ 
7     while  $q \in D(r)$  and  $q \neq \emptyset$  do
8       | add the edge  $rq$  to  $H$ ; delete  $q$  from  $\text{NN}(v)$ ;  $q \leftarrow \text{NN}(v, r)$ 
9     reinsert all deleted points into  $\text{NN}(v)$ 
10  delete all  $q$  from  $\text{NN}(v)$  for which at least one edge  $rq$  was found

```

Algorithm 3: Selecting incoming edges for the points of a node v of T and a cone C .

2.4 From Spanners to BFS Trees

We show how to compute BFS trees for a transmission graph G . Let the desired root $s \in P$ be given. We apply a technique that Cabello and Jejêcê used for unit-disk graphs [3]. Denote by $d_h(s, p)$ the BFS distance from s to p in G . For $i \in \mathbb{N}_0$ let $W_i \subseteq P$ be the vertices $p \in P$ with $d_h(s, p) = i$. Cabello and Jejêcê used the Delaunay triangulation (DT) to efficiently identify W_{i+1} , given W_0, \dots, W_i . Our t -spanner provides similar properties for transmission graphs as the DT does for unit-disk graphs.

► **Lemma 2.13.** *Let H be the t -spanner for G from Theorem 2.7 for t small enough, and let $v \in W_{i+1}$, for some $i \geq 0$. Then there is a vertex $u \in W_i$ and a path $u = q_l, \dots, q_1 = v$ in H with $d_h(s, q_j) = i + 1$ for $j = l - 1, \dots, 1$.*

```

1  $W_0 \leftarrow \{s\}$ ;  $d[s] = 0$ ;  $\pi[s] = s$ ;  $i = 0$ ; and, for  $p \in P \setminus \{s\}$ ,  $d[p] = \infty$  and  $\pi[p] = \text{NIL}$ 
2 while  $W_i \neq \emptyset$  do
3   compute power diagram with point location structure  $\text{PD}_i$  of  $W_i$ 
4   queue  $Q \leftarrow W_i$ ;  $W_{i+1} \leftarrow \emptyset$ 
5   while  $Q \neq \emptyset$  do
6      $p \leftarrow \text{dequeue}(Q)$ 
7     foreach edge  $pq$  of  $H$  do
8       |  $u \leftarrow \text{PD}_i(q)$  // query  $\text{PD}_i$  with  $q$ 
9       | if  $q \in D(u)$  and  $d[q] = \infty$  then
10      | | enqueue( $Q, q$ );  $d[q] = i + 1$ ;  $\pi[q] = u$ ; add  $q$  to  $W_{i+1}$ 
11   $i \leftarrow i + 1$ 

```

Algorithm 4: Compute the BFS tree for G with root s using H .

The BFS tree for s is computed iteratively; see Alg. 4 for pseudocode. Initially, we set $W_0 = \{s\}$. Now assume we computed everything up to W_i . By Lemma 2.13, all vertices in

W_{i+1} can be reached from W_i in the subgraph of H induced by $W_i \cup W_{i+1}$. Thus, we can compute W_{i+1} as follows: for each $u \in W_i$, start a BFS search in H from u . Every time we encounter a new vertex q , we check if it lies in a disk of W_i . If so, we add q to W_{i+1} and add the new neighbors of q to the queue. Otherwise, we discard q for now. To test whether q lies in a disk of W_i , we use the *power diagram*. This weighted version of the Voronoi Diagram represents the union of the W_i -disks as a planar subdivision. It takes $O(|W_i| \log |W_i|)$ time to compute, and augmented with a point location structure it supports the following queries in time $O(\log |W_i|)$: given a point q , find a disk in W_i that contains it, if it exists [12, 13].

Each edge pq of H is considered at most twice by Alg. 4, and each time we query a power diagram with q (in $O(\log n)$ time). Since H is sparse, the total time is $O(n \log n)$.

3 Reachability Oracles for 1-dimensional Graphs

In this section we prove the following theorem.

► **Theorem 3.1.** *Let G be the transmission graph of an n -point set $P \subset \mathbb{R}$. We can construct in $O(n \log n)$ time a reachability oracle for G with $S(n) = O(n)$ and $Q(n) = O(1)$.*

Let \mathcal{C} be the set of strongly connected components (SCCs) of G and let $C \in \mathcal{C}$. We say that C can *reach* a point $p \in P$ if there is a path in G from a point in C to p . We say that C can reach an SCC $D \in \mathcal{C}$ if C can reach a point in D . By strong connectivity, this means that all points in C can reach all points in D . Next, we define several points related to C : the *leftmost point* of C , $l(C)$, is the point in C with the smallest x -coordinate; the *left reachpoint* of C , $lr(C)$, is the leftmost point in \mathbb{R} that lies in a ball around a point in P reachable from C ; and the *direct left reachpoint* of C , $dl(C)$, is the leftmost point in \mathbb{R} that lies in a ball around a point in C , i.e., $dl(C) = \min_{p \in C} (p - r_p)$. The right versions $r(C)$, $rr(C)$, and $dr(C)$ are defined analogously. The *interval* of C , I_C , is defined as $I_C = [l(C), r(C)]$.

► **Lemma 3.2.** *Let $C \in \mathcal{C}$ be an SCC, and let $p \in C$ a point in C . For any $q \in P$, there is a path in G from p to q if and only if $q \in [lr(C), rr(C)]$.*

Lemma 3.2 suggests the following reachability oracle with $O(n)$ space and $O(1)$ query time: for each $C \in \mathcal{C}$, store the reachpoints $lr(C)$ and $rr(C)$; and for each point $p \in P$, store the SCC of G that contains it. Given two query points p, q , we look up the SCC C for p , and we return YES iff $q \in [lr(C), rr(C)]$. It remains to describe an efficient preprocessing algorithm. To find the reachpoints quickly, we investigate the structure of the SCCs in G .

► **Lemma 3.3.** *The intervals $\{I_C \mid C \in \mathcal{C}\}$ for the SCCs form a laminar set family, i.e., for any $C, D \in \mathcal{C}$, we have either $I_C \cap I_D = \emptyset$, $I_C \subseteq I_D$, or $I_D \subseteq I_C$.*

By Lemma 3.3, we can obtain a forest with vertex set \mathcal{C} by considering the set containment relation on the intervals $\{I_C \mid C \in \mathcal{C}\}$. If necessary, we add a common root node to get a tree T . The next lemma characterizes the left and right reachpoints.

► **Lemma 3.4.** *Let $C \in \mathcal{C}$. The left reachpoint $lr(C)$ of C is either $dl(C)$ or $dl(D)$, where D is a sibling of C in T . The situation for the right reachpoints is analogous.*

Reachability Between Siblings. By Lemma 3.4, for an SCC $C \in \mathcal{C}$, it suffices to search for $lr(C)$ and $rr(C)$ among the siblings of C in T . Let C_1, \dots, C_k be the children of a node in T , sorted from left to right according to their intervals. To compute the left reachpoints of C_1, \dots, C_k , we set $lr(C_1) = dl(C_1)$ and we push C_1 onto an empty stack S . Then we go

through C_2, \dots, C_k , from left to right. For the current child C_i , we initialize the tentative left reachpoint $\text{lr}(C_i) = \text{dl}(C_i)$. While the current tentative reachpoint lies to the left of the right interval endpoint for the top of the stack, we pop the stack and we update the tentative reachpoint of C_i to the left reachpoint of the popped component, if it lies further to the left. Then we push C_i onto the stack and proceed to the next child; see Algorithm 5.

```

1  $\text{lr}(C_1) \leftarrow \text{dr}(C_1)$ ; push  $C_1$  onto an empty stack  $S$ 
2 for  $i \leftarrow 2$  to  $k$  do
3    $\text{lr}(C_i) \leftarrow \text{dr}(C_i)$ 
4   while  $S \neq \emptyset$  and  $\text{lr}(C_i) \leq \text{r}(\text{top}(S))$  do
5      $D \leftarrow \text{pop}(S)$ ;  $\text{lr}(C_i) \leftarrow \min\{\text{lr}(C_i), \text{lr}(D)\}$ 
6   push  $C_i$  onto  $S$ 

```

Algorithm 5: Computing left reachpoints.

The right reachpoints are computed analogously. Since each SCC is pushed/popped at most once onto/from S , and sorting the SSCs needs $O(n \log n)$ time, we get the following lemma.

► **Lemma 3.5.** *We can compute the reachability for all nodes in T in $O(n \log n)$ time.*

It remains to find the SCCs without explicitly constructing G . To do so, we can use the Kosaraju-Sharir algorithm [8] together with geometric data structures that allow us to efficiently find unvisited edges. See the full version for details. This establishes Theorem 3.1.

4 Reachability Oracles for 2-dimensional Graphs

4.1 Ψ is less than $\sqrt{3}$

Suppose that $\Psi \in [1, \sqrt{3})$. We show that we can *planarize* G by first removing unnecessary edges and then resolving edge crossings by adding $O(n)$ additional vertices. This will not change the reachability between the original vertices. The existence of efficient reachability oracles then follows from known results for planar graphs. We prove the following lemma.

► **Lemma 4.1.** *Let G be the transmission graph for a planar n -point set P , such that $\Psi < \sqrt{3}$. In time $O(n \log n)$, we can find a plane graph $H = (V, E)$ s.t. (i) $|V| = O(n)$ and $|E| = O(n)$; (ii) $P \subseteq V$; (iii) for any $p, q \in P$, p can reach q in G iff p can reach q in H .*

Given Lemma 4.1, we can obtain the following result by constructing the distance oracle from Holm, Rotenberg and Thorup for H [11]. It has $O(1)$ query time and needs $O(n)$ space.

► **Theorem 4.2.** *Let G be the transmission graph for a two-dimensional set P of n points and let Ψ be the ratio between the largest and smallest radius in P . If $\Psi < \sqrt{3}$, we can construct in $O(n \log n)$ time a reachability oracle for G with $S(n) = O(n)$ and $Q(n) = O(1)$.*

We prove Lemma 4.1 in three steps. First, we show how to reduce the number of edges in G to $O(n)$ without changing the reachability. Then we show how to remove the crossings from G . Finally, we argue that we can combine these two operations to get the desired result.

Pruning the Graph. We construct a sparse subgraph $H \subseteq G$ with the same reachability as G but with $O(n)$ edge crossings. Consider the grid \mathcal{Q}_0 whose cells have side length $1/\sqrt{2}$. Let $\sigma \in \mathcal{Q}_0$ be a grid cell. We say that an edge of G *lies in* σ if both endpoints are contained

in σ . The *neighborhood* $N(\sigma)$ of σ consists of the 7×7 block of cells in \mathcal{Q}_0 with σ at the center. Two grid cells are *neighboring* if they lie in each other's neighborhood. For any edge in G , its two endpoints must lie in neighboring grid cells. We assign each point in P to the cell of \mathcal{Q}_0 that contains it. The subgraph H has P as vertex set, and we pick the edges as follows: for each non-empty cell σ , let $P_\sigma \subseteq P$ be the points in σ . We compute the Euclidean minimum spanning tree (EMST) T_σ of P_σ , and for each edge pq of T_σ , we add the directed edges pq and qp to H . Then, for cell $\sigma' \in N(\sigma)$, we check if there are any edges from σ to σ' in G . If so, we add an arbitrary such edge to H . The following lemma states properties of H .

► **Lemma 4.3.** *The graph H **a)** has the same reachability as G ; **b)** has $O(n)$ edges; **c)** can be constructed in $O(n \log n)$ time; and **d)** has $O(n)$ edge crossings if it is drawn in the plane with vertex set P .*

Removing the Crossings. Suppose an edge pq of G and an edge uv of G cross at a point x . To eliminate the crossing, we add x as a new site to the graph, and we replace pq and uv by the four new edges px, xq, ux and xv . Furthermore, if qp is an edge of G , we replace it by the two edges qx, xp , and if vu is an edge of G , we replace it by the two edges vx, xv . We say that this *resolves* the crossing between p, q, u and v . Let \tilde{G} be the graph obtained by iteratively resolving all crossings in G . We can show that the reachability on the vertices of G stays the same in \tilde{G} . Intuitively speaking, the $\Psi < 3$ restriction forces the vertices to be close together. This guarantees the existence of additional edges between p, q, u, v in G and these edges are always sufficient to cover all new paths introduced by resolving the crossing.

► **Lemma 4.4.** *For any two sites $p, q \in P$, if p can reach q in \tilde{G} then p can reach q in G .*

Putting it together. To prove Lemma 4.1, we first construct the sparse subgraph H as in Lemma 4.3 in time $O(n \log n)$. Then we iteratively resolve all crossings in H to obtain \tilde{H} . Since H has $O(n)$ crossings that can be found in the same time, this takes $O(n)$ time.

Let $p, q \in P$. We must argue that p can reach q in G if and only if p can reach q in \tilde{H} . Let \tilde{G} be the graph obtained by resolving the crossings in G , as in Lemma 4.4. We know that the reachability between p and q is the same in G, H , and \tilde{G} . Furthermore, if p can reach q in H , then also in \tilde{H} , and if p can reach q in \tilde{H} , then also in \tilde{G} , because (a subdivision of) every edge of \tilde{H} is present in \tilde{G} . Thus, \tilde{H} and G have the same reachability properties.

4.2 Ψ is constant

Our goal is to prove the following theorem:

► **Theorem 4.5.** *Let G be the transmission graph for an n -point set $P \subset \mathbb{R}^2$ and let Ψ be the ratio between the largest and smallest radius of the points in P . We can construct a reachability oracle for G with $S(n) = O(\Psi^5 n^{3/2})$ and $Q(n) = O(\Psi^3 \sqrt{n})$ in time $O(\Psi^5 n^{3/2})$.*

Let \mathcal{D} be the disks induced by P . Let $\mu(\mathcal{D})$ be the area occupied by $\bigcup \mathcal{D} := \bigcup_{D \in \mathcal{D}} D$. Alber and Fiala show how to compute a separator for disks with respect to $\mu(\cdot)$ [1].

► **Theorem 4.6** (Theorem 4.12 in [1]). *There exist positive constants $\alpha < 1$ and β such that the following holds: let \mathcal{D} be a set of n disks and Ψ the ratio of the largest and the smallest radius in \mathcal{D} . Then we can find in time $O(\Psi^2 n)$ a partition $\mathcal{A} \cup \mathcal{B} \cup \mathcal{S}$ of \mathcal{D} satisfying (i) $\mathcal{A} \cap \mathcal{B} = \emptyset$, (ii) $\mu(\mathcal{S}) \leq \Psi^2 \beta \sqrt{\mu(\mathcal{D})}$ and (iii) $\mu(\mathcal{A}), \mu(\mathcal{B}) \leq \alpha \mu(\mathcal{D})$.*

To obtain the data structure, consider the grid $\mathcal{Q} = \mathcal{Q}_0$ whose cells have diameter 1. All vertices in one cell form a clique in G , so it suffices to determine the reachability for one of them. For each non-empty cell $\sigma \in \mathcal{Q}$ we pick an arbitrary vertex as the *representative* of σ . Let $P_{\mathcal{D}}$ be the set of all representatives for \mathcal{D} . We recursively create a separator tree T that contains all needed reachability information: compute \mathcal{A}, \mathcal{B} , and \mathcal{S} according to Theorem 4.6. We create a node v of the separator tree. Let Q_v be all cells in \mathcal{Q} that intersect $\bigcup \mathcal{S}$, and let P_v be their representatives and \mathcal{D}_v all disks with centers in Q_v . For each $s \in P_v$, we store all representatives of $P_{\mathcal{D}}$ that s can reach and all the representatives that can be reached by s in the transmission graph induced by \mathcal{D} (this graph is a subgraph of G). We recursively compute separator trees for $\mathcal{A} \setminus \mathcal{D}_v$ and $\mathcal{B} \setminus \mathcal{D}_v$, and we connect them to v .

For the space requirement, we can show that $|P_{\mathcal{D}}| = O(\mu(\mathcal{D}))$ for any set of disks \mathcal{D} .

► **Lemma 4.7.** *Let \mathcal{D} be a set of n disks with radius at least 1. Then the number of cells in \mathcal{Q}_0 that intersect $\bigcup \mathcal{D}$ is $O(\mu(\mathcal{D}))$.*

Then, the space requirement $S(\mu(\mathcal{D}))$ for a set of disks \mathcal{D} with respect to $\mu(\cdot)$ is

$$S(\mu(\mathcal{D})) = S((1 - \alpha)\mu(\mathcal{D})) + S(\alpha\mu(\mathcal{D})) + O(\Psi^2\mu(\mathcal{D})^{3/2}), \quad (1)$$

where the last term accounts for storing reachability between the $O(\Psi^2\sqrt{\mu(\mathcal{D})})$ vertices of $P_{\mathcal{S}}$ and the $O(\mu(\mathcal{D}))$ vertices of $P_{\mathcal{A}} \cup P_{\mathcal{B}}$. For $\mu(\mathcal{D}) = O(1)$, we have $S(\mu(\mathcal{D})) = O(1)$, and Eqn. 1 solves to $S(\mu(\mathcal{D})) = O(\Psi^2\mu(\mathcal{D})^{3/2})$. Since $\mu(\mathcal{D}) = O(n\Psi^2)$, the total space is $O(\Psi^5n^{3/2})$.

Performing a Query. Let $p, q \in P$ be given. We may assume that p and q are representative for their cells. If $p = q$, we say YES. If $p \neq q$, we let v_p and v_q be the nodes in T with $p \in P_{v_p}$ and $q \in P_{v_q}$, respectively. Let u be least common ancestor of v_p and v_q . It can be found in $O(\log n)$ time by walking up the tree. Let L be the path from u to the root of T . We check for each $s \in \bigcup_{v \in L} P_v$ whether p can reach s and s can reach q . If so, we say YES. If there is no such s , we say NO. Since $|P_v|$ decreases geometrically along L , the running time is dominated by the root, and it is $O(\Psi^2\mu(\mathcal{D})^{1/2})$. Bounding $\mu(\mathcal{D})$ by $O(\Psi^2n)$, the total query time is $O(\Psi^3\sqrt{n})$. We now argue correctness. First, note that we will say YES only if there is a path from p to q . Now suppose there is a path π in G from p to q , where $p \neq q$ and p, q are representatives. Let v_p, v_q be the nodes in T for p and q , let u be their least common ancestor, and L be the path from u to the root. By construction, $\bigcup_{v \in L} \mathcal{D}_v$ must contain a disk for a point r in π . We pick r such that the corresponding node v is closest to the root. Let s be the representative for the cell containing r . Then there is an edge from r to s and from s to r , so p can reach s and s can reach q in the transmission graph of v . Thus, when walking along L , the algorithm will discover s and the connection between p and q .

Preprocessing Time. We compute for each node v in T a spanner H_v for the corresponding transmission graph, as in Theorem 2.7. Since we are only interested in the reachability H_v , we can choose $t > 1$ to be some small constant. Since T has $O(\log n)$ levels, the total running time for this step is $O(n \log n (\log n + \log \Psi))$. Then we go through all the nodes $v \in T$. For each $s \in P_v$, we compute a BFS tree in H_v with root s . Next, we reverse all edges in H_v and we again compute BFS-trees for all $s \in P_v$ in the transposed graph. This gives the necessary information we want to store for s . Since the amount of work is proportional to the total size of the BFS-trees, we get a total running time of $O(\Psi^5n^{3/2})$. Theorem 4.5 now follows.

4.3 Ψ is polynomially bounded

Now we assume that Ψ is bounded by some polynomial in n . Then we can show the following.

► **Theorem 4.8.** *Let G be the transmission graph for a two-dimensional set P of n points and let Ψ be the ratio between the largest and smallest radii of the points in P . If $\Psi = O(\text{poly}(n))$, we can construct a reachability oracle for G in $O(n^{5/3} \log^2 n)$ time with $S(n) = O(n^{5/3} \log n)$ and $Q(n) = O(n^{2/3} \log n)$. All queries are answered correctly with high probability.*

We scale everything such that the smallest radius in P is 1. Our approach is as follows: let $p, q \in P$. If there is a p - q -path with “many” vertices, we detect this by taking a large enough random sample $S \subseteq P$ and by storing the reachability information for every vertex in S . If there is a path from p to q with “few” vertices, then p must be “close” to q , where “closeness” is defined relative to the largest radius along the path. The radii from P can lie in $O(\log \Psi)$ different scales, and for each scale we store few local information to find such a “short” path.

First we consider long paths. Let $0 < \alpha < 1$ be some constant to be determined later. First, we show that a random sample can be used to detect paths with many vertices.

► **Lemma 4.9.** *We can sample a set $S \subset P$ of size $O(n^\alpha \log n)$ s.t. the following holds w.h.p.: for any $p, q \in P$, if there is a path π from p to q in G of length at least $n^{1-\alpha}$, then $\pi \cap S \neq \emptyset$.*

We find such a sample S , and for each $s \in S$, we store two Boolean arrays that indicate for each $p \in P$ whether p can reach s and whether s can reach p . This needs space $O(n^{1+\alpha} \log n)$.

Now we treat short paths. Let $L = \lceil \log \Psi \rceil$. We consider L grids $\mathcal{Q}_0, \dots, \mathcal{Q}_L$, s.t. the cells in \mathcal{Q}_i have diameter 2^i . For each $\sigma \in \mathcal{Q}_i$, let $Q_\sigma \subseteq P$ be the vertices $p \in P \cap \sigma$ with $r_p \in [2^i, 2^{i+1})$. Q_σ forms a clique in G , and for each $p \in Q_\sigma$, the disk $D(p)$ covers σ . The neighborhood $N(\sigma)$ is defined as the set of all cells from \mathcal{Q}_i that have distance at most $2^{i+1}n^{1-\alpha}$ from σ . We have $|N(\sigma)| = O(n^{2-2\alpha})$. Let $P_\sigma \subseteq P$ be the points that lie in cells of $N(\sigma)$. For every $i = 0, \dots, L$ and for every $\sigma \in \mathcal{Q}_i$ with $Q_\sigma \neq \emptyset$, we fix an arbitrary representative point $q_\sigma \in Q_\sigma$. For every point $p \in P$, we store for every $i \in \{0, \dots, L\}$ a sorted list of all cells $\sigma \in \mathcal{Q}_i$ with $p \in P_\sigma$ such that q_σ can be reached from p and a list of all cells $\sigma \in \mathcal{Q}_i$ with $p \in P_\sigma$ such that q_σ can reach p . A point in P appears in at most $O(n^{2-2\alpha} \log \Psi)$ point sets P_σ , so the total space requirement is $O(n^{3-2\alpha} \log \Psi)$.

Performing a Query. Let $p, q \in P$ be given. To decide whether p can reach q , we first check all $O(n^\alpha \log n)$ points in S . If there is an $s \in S$ such that p reaches s and such that s reaches q , we return YES. Otherwise, for $i = 0, \dots, L$, we walk through the lists of cells whose representative point is reachable from p at level i and through the list of cells whose representative point can reach q at level i to check whether they contain a common element. Since the lists are sorted, this can be done in time linear in the list size, as in merge sort. If any of these pairs of lists contains a common cell, we return YES. Otherwise, we return NO.

For correctness, first note that we return YES only if there is a path from p to q . Now assume that there is a path π from p to q . If π has more than $n^{1-\alpha}$ vertices, then by Lemma 4.9, the sample S hits π w.h.p., and the algorithm returns YES. Otherwise, let r be the vertex of π with the largest radius, and let i be such that $r_r \in [2^i, 2^{i+1})$. Let σ be the cell of \mathcal{Q}_i that contains r . Since π has at most $n^{1-\alpha}$ vertices and each edge of π has length at most 2^{i+1} , the path π lies in $N(\sigma)$. In particular, both p and q are contained in cells of $N(\sigma)$. Since $r \in Q_\sigma$ and since Q_σ forms a clique in G , the representative point q_σ of σ can be reached from p and can reach q . By the symmetry of neighborhood definition, σ is contained in the list of reachable cells from p and in the lists of cells that can reach q . This common cell will be detected when checking the corresponding lists for p and q at level i .

Time and Space Requirements. For long paths we need $O(n^\alpha \log n)$ time: for every $s \in S$ we test in $O(1)$ time whether p can reach s and whether s can reach q . For short paths

there are $O(\log \Psi)$ levels, and at each level we step through two lists of size $O(n^{2-2\alpha})$. Since we assume $\log \Psi = O(\log n)$, the tradeoff for the query time is at $\alpha = 2/3$, yielding $Q(n) = O(n^{2/3} \log n)$. The same α is the tradeoff for the space usage, which is $O(n^{5/3} \log n)$.

For the preprocessing, we first compute the reachability arrays for each $s \in S$. To do so, we build the spanner H for G from Section 2.2 in time $O(n \log n)$. Then, for each $s \in S$ we do a BFS search in H and its transposed graph. This gives all vertices that s can reach and that can be reached by s in $O(n^{3/2} \log n)$ total time. Now, we do the preprocessing for short paths. For each $i = 0, \dots, L$ and each cell $\sigma \in \mathcal{Q}_i$ that has a representative q_σ we do the following: consider the points P_σ . We compute the spanner H_σ from Section 2.2 for P_σ . For each q_σ , we do a BFS search in H_σ and its transposed graph starting from q_σ . This gives all $p \in P_\sigma$ that reach q_σ and that are reachable from q_σ . The running time is dominated by constructing the spanners. Since each point $p \in P$ is contained in $O(n^{2-2\alpha} \log \Psi) = O(n^{2/3} \log n)$ different P_σ , and since constructing H_σ takes $O(|P_\sigma|(\log \Psi + \log |P_\sigma|))$ time, the preprocessing time for the short paths is $O(n^{5/6} \log^2 n)$.

Acknowledgements. We thank Paz Carmi and Günter Rote for valuable comments.

References

- 1 Jochen Alber and Jirí Fiala. Geometric separation and exact solutions for the parameterized independent set problem on disk graphs. *J. Algorithms*, 52(2):134–151, 2004.
- 2 Azzedine Boukerche. *Algorithms and Protocols for Wireless Sensor Networks*. Wiley Series on Parallel and Distributed Computing). Wiley-IEEE Press, 1st edition, 2008.
- 3 Sergio Cabello and Miha Ježić. Shortest paths in intersection graphs of unit disks. *Comput. Geom.*, 48(4):360–367, 2015.
- 4 Paul Callahan and Rao Kosaraju. A decomposition of multidimensional point sets with applications to k -nearest-neighbors and n -body potential fields. *J. ACM*, 42(1):67–90, 1995.
- 5 Paz Carmi, 2014. Personal communication.
- 6 Timothy M. Chan. A dynamic data structure for 3-D convex hulls and 2-D nearest neighbor queries. *J. ACM*, 57(3):Art. 16, 15, 2010.
- 7 Brent N. Clark, Charles J. Colbourn, and David S. Johnson. Unit disk graphs. *Discrete Math.*, 86(1-3):165–177, 1990.
- 8 Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.
- 9 Martin Fürer and Shiva Prasad Kasiviswanathan. Spanners for geometric intersection graphs with applications. *J. Comput. Geom.*, 3(1):31–64, 2012.
- 10 Sariel Har-Peled. *Geometric Approximation Algorithms*. AMS, 2011.
- 11 Jacob Holm, Eva Rotenberg, and Mikkel Thorup. Planar Reachability in Linear Space and Constant Time. *CoRR*, arXiv:1411.5867, 2014.
- 12 Hiroshi Imai, Masao Iri, and Kazuo Murota. Voronoi Diagram in the Laguerre Geometry and its Applications. *SICOMP*, 14(1):93–105, 1985.
- 13 D. Kirkpatrick. Optimal Search in Planar Subdivisions. *SICOMP*, 12(1):28–35, 1983.
- 14 G. Narasimhan and M. Smid. *Geometric spanner networks*. Cambridge Univ. Press, 2007.
- 15 David Peleg and Liam Roditty. Localized spanner construction for ad hoc networks with variable transmission range. *TOSN*, 7(3), 2010.
- 16 P. v. Rickenbach, R. Wattenhofer, and A. Zollinger. Algorithmic Models of Interference in Wireless Ad Hoc and Sensor Networks. *IEEE ACM T NETWORK*, 17(1):172–185, 2009.
- 17 Andrew Chi-Chih Yao. On Constructing Minimum Spanning Trees in k -Dimensional Spaces and Related Problems. *SICOMP*, 11(4):721–736, 1982.

Recognition and Complexity of Point Visibility Graphs

Jean Cardinal¹ and Udo Hoffmann^{*2}

- 1 Université libre de Bruxelles (ULB)
Brussels, Belgium
jcardin@ulb.ac.be
- 2 TU Berlin
Berlin, Germany
uhoffman@math.tu-berlin.de

Abstract

A *point visibility graph* is a graph induced by a set of points in the plane, where every vertex corresponds to a point, and two vertices are adjacent whenever the two corresponding points are *visible* from each other, that is, the open segment between them does not contain any other point of the set.

We study the recognition problem for point visibility graphs: given a simple undirected graph, decide whether it is the visibility graph of some point set in the plane. We show that the problem is complete for the existential theory of the reals. Hence the problem is as hard as deciding the existence of a real solution to a system of polynomial inequalities. The proof involves simple substructures forcing collinearities in all realizations of some visibility graphs, which are applied to the algebraic universality constructions of Mnëv and Richter-Gebert. This solves a longstanding open question and paves the way for the analysis of other classes of visibility graphs.

Furthermore, as a corollary of one of our construction, we show that there exist point visibility graphs that do not admit any geometric realization with points having integer coordinates.

1998 ACM Subject Classification I.3.5 Computational geometry

Keywords and phrases point visibility graphs, recognition, existential theory of the reals

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.171

1 Introduction

Visibility between geometric objects is a cornerstone notion in discrete and computational geometry, that appeared as soon as the late 1960s in pioneering experiments in robotics [17]. Visibility is involved in major themes that helped shape the field, such as art gallery and motion planning problems [5, 8, 21]. However, despite decades of research on those topics, the combinatorial structures induced by visibility relations in the plane are far from understood.

Among such structures, *visibility graphs* are arguably the most natural. In general, a visibility graph encodes the binary, symmetric visibility relation among sets of objects in the plane, where two objects are visible from each other whenever there exists a straight line of sight between them that does not meet any obstacle. More precisely, a *point visibility graph* associated with a set P of points in the plane is a simple undirected graph $G = (P, E)$ such that two points of P are adjacent if and only if the open segment between them does not

* Supported by the Deutsche Forschungsgemeinschaft within the research training group “Methods for Discrete Structures” (GRK 1408).



contain any other point of P . Note that the points play both roles of vertices of the graph and obstacles. In what follows, we will use the abbreviation PVG for point visibility graph.

1.1 Our results

We consider the *recognition* problem for point visibility graphs: given a simple undirected graph $G = (V, E)$, does there exist a point set P such that G is isomorphic to the visibility graph of P ? More concisely, the problem consists of deciding the property of being a point visibility graph of some point set.

As is often the case for geometric graphs, the recognition problem appears to be intractable under usual complexity-theoretic assumptions. We actually characterize the problem as complete for the existential theory of the reals; hence recognizing point visibility graphs is as hard as deciding the existence of a solution to an arbitrary system of polynomial inequalities over the reals. Equivalently, this amounts to deciding the emptiness of a semialgebraic set. This complexity class is intimately related to fundamental results on *oriented matroids* and *pseudoline arrangements* starting with the insights of Mnëv on the algebraic universality properties of these structures [20]. The notation $\exists\mathbb{R}$ has been proposed recently by Schaefer [27] to refer to this class, motivated by the continuously expanding collection of problems in computational geometry that are identified as complete for it.

The only known inclusion relations for $\exists\mathbb{R}$ are $NP \subseteq \exists\mathbb{R} \subseteq PSPACE$. It is known from the Tarski-Seidenberg Theorem that the first-order theory of real closed fields is decidable, but polynomial space algorithms for problems in $\exists\mathbb{R}$ have been proposed only much more recently by Canny [4].

Whenever a graph is known to be a point visibility graph, the description of the point set as a collection of pairs of integer coordinates constitutes a natural certificate. Since it is not known whether $\exists\mathbb{R} \subseteq NP$, we should not expect such a certificate to have polynomial size. In fact, we show that there exist point visibility graphs all realizations of which have an irrational coordinate, and point visibility graphs that require doubly exponential coordinates in any realization.

1.2 Related work and Connections

The recognition problem for point visibility graphs has been explicitly stated as an important open problem by various authors [14], and is listed as the first open problem in a recent survey from Ghosh and Goswami [9].

A linear-time recognition algorithm has been proposed by Ghosh and Roy for *planar* point visibility graphs [10]. For general point visibility graphs they showed that recognition problem lies in $\exists\mathbb{R}$. More recently, Roy [26] published an ingenious and rather involved NP-hardness proof for recognition of arbitrary point visibility graphs. Our result clearly implies NP-hardness as well, and, in our opinion, has a more concise proof.

Structural aspects of point visibility graphs have been studied by Kára, Pór, and Wood [14], Pór and Wood [24], and Payne et al. [23]. Many fascinating open questions revolve around the *big-line-big-clique* conjecture, stating that for all $k, \ell \geq 2$, there exists an n such that every finite set of at least n points in the plane contains either k pairwise visible points or ℓ collinear points.

Visibility graphs of polygons are defined over the vertices of an arbitrary simple polygon in the plane, and connect pairs of vertices such that the open segment between them is completely contained in the interior of the polygon. This definition has also attracted a lot of interest in the past twenty years. Ghosh gave simple properties of visibility graphs of polygons and

conjectured that they were sufficient to characterize visibility graphs [6, 7]. These conjectures have been disproved by Streinu [31] via the notion of *pseudo-visibility* graphs, or visibility graphs of *pseudo-polygons* [22]. A similar definition is given by Abello and Kumar [1]. Roughly speaking, the relation between visibility and pseudo-visibility graphs is of the same nature as that between arrangements of straight lines and pseudolines. Although, as Abello and Kumar remark, these results somehow suggest that the difficulty in the recognition task is due to a stretchability problem, the complexity of recognizing visibility graphs of polygons remains open, and it is not clear whether the techniques described in this paper can help characterizing it. The influential surveys and contributions of Schaefer about $\exists\mathbb{R}$ -complete problems in computational geometry form an ideal point of entry in the field [27, 28]. Among such problems, let us mention recognition of segment intersection graphs [15], recognition of unit distance graphs and realizability of linkages [13, 28], recognition of disk and unit disk intersection graphs [19], computing the rectilinear crossing number of a graph [3], simultaneous geometric graph embedding [16], and recognition of d -dimensional Delaunay triangulations [2].

1.3 Outline of the paper

In Section 2, we provide two simple visibility graph constructions, the *fan* and the *generalized fan*, all geometric realizations of which are guaranteed to preserve a specified collection of subsets of collinear points. The proofs are elementary and only require a series of basic observations.

In Section 3, we give two applications of the fan construction. In the first, we show that there exists a point visibility graph that does not have any geometric realization on the integer grid. In other words, all geometric realizations of this point visibility graph are such that at least one of the points has an irrational coordinate. Another application of the fan construction follows, where we show that there are point visibility graphs each grid realization of which require coordinates of values $2^{2^{\frac{3}{n}}}$ where n denotes the number of vertices of the point visibility graph.

The main result of the paper is given in Section 4. We first recall the main notions and tools used in the results from Mněv [20], Shor [29], and Richter-Gebert [25] for showing that realizability of abstract order types is complete for the existential theory of the reals. We then combine these tools with the generalized fan construction to produce families of point visibility graphs that can simulate arbitrary arithmetic computations over the reals.

1.4 Notations

For the sake of simplicity, we slightly abuse notations and do not distinguish between a vertex of a point visibility graph and its corresponding point in a geometric realization. We denote by $G[P']$ the induced subgraph of a graph $G = (P, E)$ with the vertex set $P' \subseteq P$. For a point visibility realization R we denote by $R[P']$ the induced subrealization containing only the points P' . The PVG of this subrealization is in general not an induced subgraph of G . By $N(p)$ we denote the open neighbourhood of a vertex p .

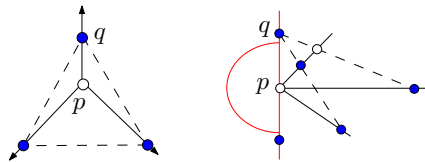
The line through two points p and q is denoted by $\ell(p, q)$ and the open segment between p and q by \overline{pq} . We will often call \overline{pq} the *sightline* between p and q , since p and q see each other iff $\overline{pq} \cap P = \emptyset$. We call two sightlines $\overline{p_1q_1}$ and $\overline{p_2q_2}$ non-crossing if $\overline{p_1q_1} \cap \overline{p_2q_2} = \emptyset$.

For each point p all other points of G lie on $\deg(p)$ many rays $R_1^p, \dots, R_{\deg(p)}^p$ originating from p .

2 Point visibility graphs preserving collinearities

We first describe constructions of point visibility graphs, all the geometric realizations of which preserve some fixed subsets of collinear points.

2.1 Preliminary observations



■ **Figure 1** (Lemma 1) Left: a point sees points on consecutive rays with small angle. Right: a vertex of $\deg(q) = 1$ in $G[N(p)]$ lies on the boundary of an empty halfspace.

In the realization of a PVG, the point p sees exactly $\deg(p)$ many vertices, hence all other points lie on $\deg(p)$ rays of origin p .

► **Lemma 1.** *Let $q \in N(p)$ be a degree-one vertex in $G[N(p)]$. Then all points lie on one side of the line $\ell(p, q)$. Furthermore, the neighbor of q lies on the ray that forms the smallest angle with \overline{qp} .*

Proof. If the angle between two consecutive rays is smaller than π , then every vertex on one ray sees every vertex on the other ray. Hence one of the angles incident to q is at least π and the neighbour of q lies on the other incident ray. ◀

► **Corollary 2.** *If $G[N(p)]$ is an induced path, then the order of the path and the order of the rays coincide.*

Proof. By Lemma 1 the two endpoints of the path lie on rays on the boundary of empty halfspaces. Thus all other rays form angles which are smaller than π , and thus they see their two neighbors of the path on their neighboring rays. ◀

► **Observation 3.** *Let $q, q \neq p$, be a point that sees all points of $N(p)$. Then q is the second point (not including p) on one of the rays emerging from p .*

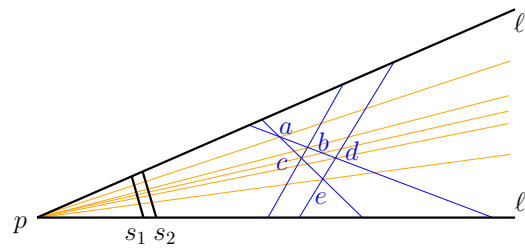
Proof. Assume q is not the second point on one of the rays. Then q cannot see the first point on its ray which is a neighbor of p . ◀

This also shows the following observation.

► **Observation 4.** *Let $q, q \neq p$, be a point that is not the second point on one of the rays from p and sees all but one (r) of the neighbors of p . Then q lies on the ray of r .*

2.2 Fans and generalized fans

We have enough tools by now to show the uniqueness of a PVG obtained from the following construction, which is depicted in Figure 2. Consider a set S of segments between two lines ℓ and ℓ' intersecting in a point p . For each intersection of a pair of segments, construct a ray of origin p and going through this intersection point. Add two segments s_1 and s_2 between ℓ and ℓ' , such that s_1 is the closest and s_2 is the second closest segments to p .



■ **Figure 2** A fan: a vertex is placed on each intersection of two lines/segments.

We now put a point on each intersection of the segments and rays and construct the PVG of this set of points. We call this graph the *fan* of S . Since we have the choice of the position of the segments s_1 and s_2 we can avoid any collinearity between a point on s_1 or s_2 and points on other segments, except for the obvious collinearities on one ray. Thus every point sees all points on s_1 except for the one of the ray it lies on.

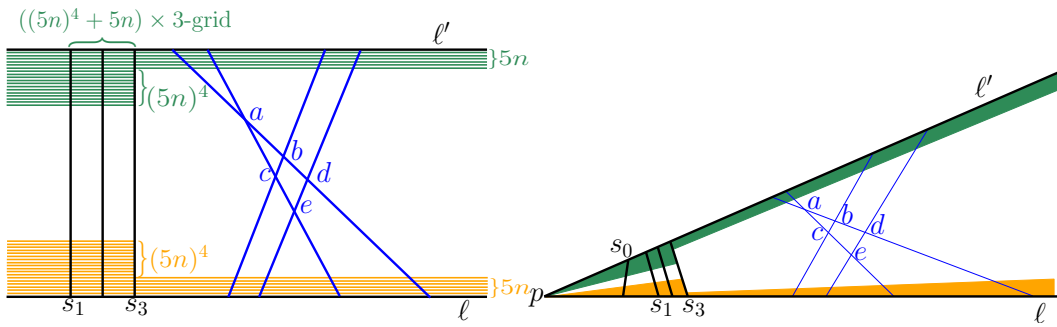
► **Lemma 5.** *All realizations of a fan preserve collinearities between points that lie on one segment and between points that lie on one ray.*

Proof. We first show that the distribution of the points onto the rays of p is unique. By construction the points on s_2 see all the points on s_1 , which are exactly the neighbors of p . Thus by Observation 3 the points from s_2 are the second points of a ray. Since there is exactly one point for each ray on s_2 , all the other points are not second points on a ray. By construction each of the remaining points sees all but one point of s_1 . Observation 4 gives a unique ray a point lies on. The order of the rays is unique by Corollary 2. On each ray the order of the points is as constructed, since the PVG of points on one ray is an induced path.

Now we have to show that the points originating from one segment are still collinear. Consider three consecutive rays R_1, R_2, R_3 . We consider a visibility between a point p_1 on R_1 and one point p_3 on R_3 that has to be blocked by a point on R_2 . Let p_2 be the original blocker from the construction. For each point on R_2 that lies closer to p there is a sightline blocked by this point, and for each point that lies further away from p there is a sightline blocked by this point. For each of those points pick one sightline that corresponds to an original segment and $\overline{p_1 p_3}$. This set of sightlines is non-crossing, since the segments only intersect on rays by assumption. So we have a set of non-crossing sightlines and the same number of blockers available. Since the order on each ray is fixed, and the sightlines intersect R_2 in a certain order, the blocker for each sightline is uniquely determined and has to be the original blocker. By transitivity of collinearity all points from the segments remain collinear. ◀

To show the hardness of PVG recognition in the existential theory of the reals in Section 4 we need a unique realization property for the following generalization of a fan.

Consider again two lines ℓ and ℓ' and a set of n segments S located between those lines. We assume for now that ℓ and ℓ' are parallel, i.e., their intersection point p lies on the line at infinity, and horizontal. Now we are not interested in preserving the exact arrangement of the segments S in a PVG, but only in keeping the segments straight, and the order of the segments on ℓ and on ℓ' as described by S . For that purpose we add three parallel and equidistant segments s_1, s_2, s_3 to the left of all segments of S . Below ℓ' and above ℓ we add $5n$ equidistant rays each, that are parallel to ℓ and ℓ' and start on the point at infinity p . Let ε be the distance between two consecutive rays in one bundle. We choose ε such that $(5n)^4\varepsilon$ is smaller than the distance of any intersection of segments in S to ℓ or ℓ' . We call



■ **Figure 3** Left: a bundle of a generalized fan above and below each intersection. Right: the generalized fan with the segment s_0 and the point p .

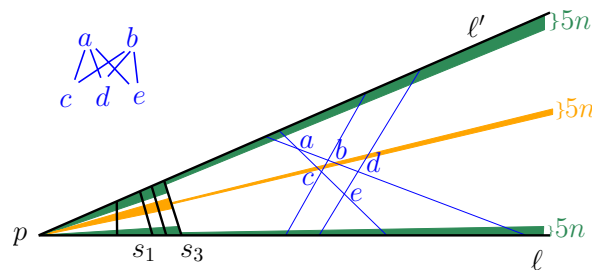
such a set of $5n$ rays a *bundle*. Above the bundle close to ℓ and below the bundle close to ℓ' we add $(5n)^4$ segments starting on s_3 and ending in p . The segments are parallel to the rays of the bundles and are also equidistant with distance ε to their close bundle. The bundles together with the $(5n)^4$ segments forms what we will call the *extended bundle*. The equidistance property is preserved according to the following lemma.

► **Lemma 6.** *Consider a realization of a PVG of an $r \times q$ integer grid, $r \geq 6, q \geq 3$, such that the points of each of the r rows lie on a horizontal line. Then – up to a projective transformation – the horizontal lines are equally spaced, the verticals are parallel, and also equally spaced.*

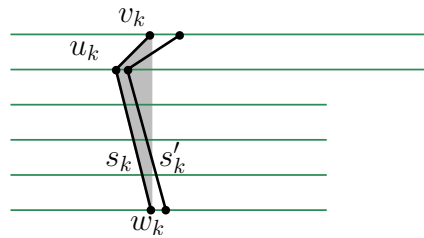
Now we apply a projective transformation, such that the intersection point p of ℓ and ℓ' does not lie on the line at infinity as shown in Figure 3. We add a segment s_0 between ℓ and ℓ' that lies between p and s_1 . Again we take all the intersection points between segments, rays or lines as points and construct the visibility graph of those points. Note that we can add s_0 , such that each point on s_0 sees all points that do not lie on its ray or s_0 . A visibility graph constructed in this way will be called a *generalized fan*. In Lemma 7 we show that all realizations of a generalized fan preserve the collinearities between the points on the segments.

Let us briefly consider the differences between a fan and a generalized fan. In the fan in Figure 2 the vertical order of the intersection points is $a > b > c > d > e$. In contrast, the generalized construction, shown on the left of Figure 3, allows different vertical orders on those points. In Figure 4 we used three bundles instead of two bundles to fix the orders. In the proof of Lemma 7 it will turn out that all realizations for this construction also preserve collinearities. In this case we have a further restriction on the vertical order of the intersection points: the points a and b must lie above the middle bundle, and the points c, d, e must lie below. This restricts the possible vertical orders of intersection points to some linear extensions of the partial order shown in Figure 3. To indicate that a and b lie above c, d and e we introduce the notation $\{a, b\} > \{c, d, e\}$. This notation captures exactly the restriction we can add to the horizontal orders of a fan: given a realization of the segments S between the lines ℓ and ℓ' it is possible to add bundles between some intersection points, partitioning the intersection points of the segments into subsets I_1, \dots, I_k . Now every realization of the PVG respects the vertical order $I_1 > \dots > I_k$ of the intersection points. If $|I_j| = 1$, one line through an intersection point as in Figure 2 can also be used.

► **Lemma 7.** *All realizations of a generalized fan preserve collinearities between points that lie on one segment and between points that lie on one ray.*



■ **Figure 4** A generalized fan with several bundles.

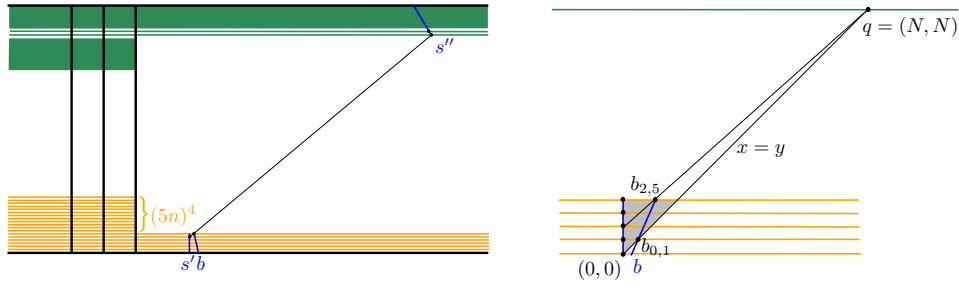


■ **Figure 5** A clockwise orientation of (u_k, v_k, w_k) forces the triple on a right segment s'_k to be oriented clockwise.

Proof. The argument showing that the distribution of the points onto the rays starting at p and the order of the rays remains as constructed is identical to the proof of Lemma 5. So we only have to show that the points from the segments stay collinear. We do this in two steps. In the first one we show that the points on segments within one extended bundle stay collinear. We will use this in a second step to show that the segments in two consecutive bundles stay aligned.

We proceed with the first step. First note that the points from one segment within one bundle stay collinear in each realization by the same arguments as in Lemma 5. The same holds for the points on a segment s_k , $k \in \{0, \dots, 3\}$, and the intersection with the $(5n)^4$ segments. So for the first step we only have to show that the segments s_0, \dots, s_3 in extended bundles stay aligned. Therefore we consider the lowest ray of the bundle close to l' and two neighboring segments. The points on the segments s_k stay collinear on those three rays, because four non-crossing sightlines have to be blocked by four points. Now consider the two lowest rays of the bundle close to l' , and the $(5n)^4$ segments below. Assume that the points on one of the segments s_0, \dots, s_4 do not stay aligned for one s_k . Then the points on s_k that lie on the two lowest rays u_k (lowest) and v_k (second lowest) and the lowest segment w_k form the convex hull of all the points on s_k that lie in between, see Figure 5. In this triangle there are $(5n)^4 - 1$ non-crossing sightlines that have to be blocked. This implies that one of the other segments s_l have to support blockers. If the triple (u_k, v_k, w_k) is oriented clockwise some the blockers have to be supported by a segment s'_k to the right, or by one to the left otherwise. In the clockwise case the three according points on the convex hull of the s'_k have to be oriented clockwise as well. Since a symmetric case holds for the counterclockwise case we obtain a contradiction for the rightmost clockwise or leftmost counterclockwise oriented triple.

So it is left to show that the two subsegments within consecutive bundles stay aligned. We will refer to those subsegments as the upper and the lower part of a segment. First note that the segments s_k , $k \in \{0, \dots, 3\}$ stay aligned in consecutive extensions of a bundle, thus they cannot provide blockers for sightlines between upper and lower part on the other segments.



■ **Figure 6** Left: A blocker on b . Right: The situation after the coordinate transformation.

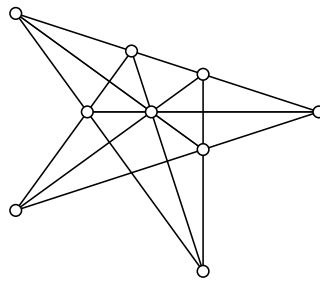
We assume the points from one original segment s are not all collinear in a realization of the fan. We denote by s' and s'' respectively the lower and upper part of s . If s' and s'' are not aligned then one of the two lower points of s'' does not lie on the supporting line of s' . We denote this point by q . Between q and the points on s' there are at least $(5n)^4 - 1$ non-crossing sightlines that have to be blocked. At most n of those sightlines can be blocked from points on the upper bundle, namely the points from the lowest ray if q lies on the second lowest ray. The other blockers lie on the other $n - 1$ lower parts of the segments. From the pigeonhole principle there is a lower part b of a segment that provides at least $\lceil (5n - n - 1)/(n - 1) \rceil = 5$ blockers for sightlines between q and points on s' . We will show that this is not possible.

By first reversing the projective transformation applied in the construction of the generalized fan, and then applying Lemma 6, we can assume that the lines in the lower bundle are parallel and equidistant, as shown in Figure 6. Now we use an affine transformation such that the points of s' have coordinates $(0, i)$ for $i \in \{-k, \dots, r - 1 - k\}$, where k is chosen such that the lowest point blocked by a point on b has coordinates $(0, 0)$. By another linear transformation we can ensure that $q = (N, N)$ for some $N > 0$. We can now use the segments starting from s_3 to give a lower bound on N : the segments above the bundle of s' are also equidistant with the same distance as the lines in the bundle, since the segments extend the grid. Since q lies on a parallel line above those rays we know that $N > (5n)^4$.

The points on b that block visibilities between points on s' from q also have y -coordinates in $\{0, \dots, r - 1 - k\}$, since they lie on lines in the same bundle as s' . Let us assume that the point b_{ij} on b has y -coordinate j and blocks the visibility of $(0, i)$ from q . Then the x -coordinate of b_{ij} is $x = (j - i) \frac{N}{N - i}$. We consider the sets $M := \{(i, j) \mid b_{ij} \text{ is a blocker}\}$ and $M' := \{(j - i) \mid b_{ij} \text{ is a blocker}\}$. We will obtain a contradiction in the following two cases.

Case 1. $|M'| < 3$: In this case there are three points in M with the same value for $j - i$. Those points on b have the coordinates of the form $(\frac{cN}{N-i}, c + i)$ where $c = j - i$ is constant. This is a parameterization of a hyperbola. No three points for $i < N$ on this curve are collinear, which contradicts that they all lie on the segment b .

Case 2: $|M'| \geq 3$: In this case there are three blockers b_0, b_1, b_2 with pairwise different values for $j - i$. Assume without loss of generality that $b_0 = (x_0, j_0)$ blocks $(0, 0)$ from q , $b_1 = (x_1, j_1)$ blocks $(0, i_1)$, and $b_2 = (x_2, j_2)$ blocks $(0, i_2)$. Then the x -coordinates of b_k is given by $x_k = (j_k - i_k) \frac{N}{N - i_k}$. The difference of the x -coordinate of two consecutive points on b is $d_{min} := \frac{x_k - x_0}{j_k - j_0}$. Calculating d_{min} using the expression above once with b_1 and once with b_2 leads to the following equation.



■ **Figure 7** The Perles configuration.

$$\frac{(j_2 - i_2)\frac{N}{N-i_2} - j_0}{j_2 - j_0} = \frac{(j_1 - i_1)\frac{N}{N-i_1} - j_0}{j_1 - j_0}$$

$$\Leftrightarrow (i_1^2 j_0 - i_1^2 j_2 - i_1 j_0 j_2 + i_1 j_1 j_2 - i_2^2 j_0 + i_2^2 j_1 + i_2 j_0 j_1 - i_2 j_1 j_2)N$$

$$+ (-i_1 j_0 + i_1 j_2 + i_2 j_0 - i_2 j_1)N^2 + i_1 i_2 j_0 (j_2 - j_1) = 0$$

Since all coefficients in the last equation are integral we obtain that $i_1 i_2 j_0 (j_2 - j_1)$ is a multiple of N . This is a contradiction to $N > (5n)^4$ since each of the factors is bounded by $5n$ and is nonzero. ◀

3 Drawing point visibility graphs on grids

We give a first simple application of the fan construction.

► **Theorem 8.** *There exists a point visibility graph every geometric realization of which has at least one point with one irrational coordinate.*

Proof. We use the so-called *Perles configuration* of 9 points on 9 lines illustrated in Fig. 7. It is known that for every geometric realization of this configuration in the Euclidean plane, one of the points has an irrational number as one of its coordinate [12]. We combine this construction with the fan construction described in the previous section. Hence we pick two lines ℓ and ℓ' intersecting in a point p , such that all lines of the configuration intersect both ℓ and ℓ' in the same wedge. Note that up to a projective transformation, the point p may be considered to be on the line at infinity and ℓ and ℓ' taken as parallel. We add two non-intersecting segments s_1 and s_2 close to p , that do not intersect any line of the configuration. We then shoot a ray from p through each of the points, and construct the visibility graph of the original points together with all the intersections of the rays with the lines and the two segments s_1, s_2 . From Lemma 5, all the collinearities of the original configuration are preserved, and every realization of the graph contains a copy of the Perles configuration. ◀

Also note that point visibility graphs that can be realized with rational coordinates do not necessarily admit a realization that can be stored in polynomial space in the number of vertices of the graph. To support this, consider a line arrangement \mathcal{A} , and add a point p in an unbounded face of the arrangement, such that all intersections of lines are visible in an angle around p that is smaller than π . Construct rays ℓ and ℓ' through the extremal intersection points and p . From Lemma 5, the fan of this construction gives a PVG that fixes \mathcal{A} . Since there are line arrangements that require integer coordinates of values $2^{2^{\Theta(|\mathcal{A}|)}}$ [11] and the

fan has $\Theta(|\mathcal{A}|^3)$ points we get the following worst-case lower bound on the coordinates of points in a representation of a PVG.

► **Corollary 9.** *There exists a point visibility graph with n vertices every realization of which requires coordinates of values $2^{2^{\Theta(\sqrt[3]{n})}}$.*

4 $\exists\mathbb{R}$ -completeness reductions

The existential theory of the reals ($\exists\mathbb{R}$) is a complexity class defined by the following complete problem. We are given a well-formed quantifier-free formula $F(x_1, \dots, x_k)$ using the numbers 0 and 1, addition and multiplication operations, strict and non-strict comparison operators, Boolean operators, and the variables x_1, \dots, x_k , and we are asked whether there exists an assignment of real values to x_1, \dots, x_k , such that F is satisfied. This amounts to deciding whether a system of polynomial inequalities admits a solution over the reals. The first main result connecting this complexity class to a geometric problem is the celebrated result of Mněv, who showed that *realizability of order types*, or – in the dual – stretchability of pseudoline arrangements, is complete in this complexity class [20]. In what follows, we use the simplified reductions due to Shor [29] and Richter-Gebert [25]. The latter is in turn well explained in a recent manuscript by Matoušek [18]. We refer the curious reader to those references for further details.

The *orientation* of an ordered triple of points (p, q, r) indicates whether the three points form a clockwise or a counterclockwise cycle, or whether the three points are collinear. Let $P = \{p_1, \dots, p_n\}$ and an orientation O of each triple of points in P be given. The pair (P, O) is called an (*abstract*) *order type*. We say that the order type (P, O) is realizable if there are coordinates in the plane for the points of P , such that the orientations of the triples of points match those prescribed by O .

In order to reduce the order type realizability problem to solvability of a system of strict polynomial inequalities, we have to be able to simulate arithmetic operations with order types. This uses standard constructions introduced by von Staudt in his “*algebra of throws*” [30].

4.1 Arithmetics with order types

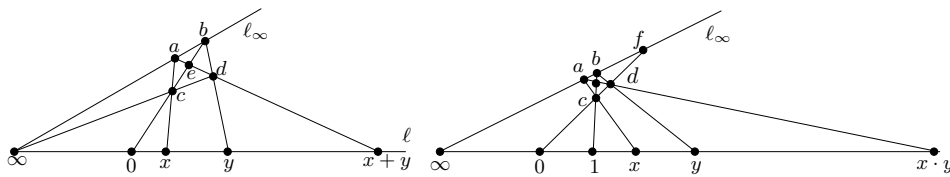
To carry out arithmetic operations using orientation predicates, we associate numbers with points on a line, and use the *cross-ratio* to encode their values.

The cross ratio $(a, b; c, d)$ of four points $a, b, c, d \in \mathbb{R}^2$ is defined as

$$(a, b; c, d) := \frac{|a, c| \cdot |b, d|}{|a, d| \cdot |b, c|},$$

where $|x, y|$ is the determinant of the matrix obtained by writing the two vectors as columns. The two properties that are useful for our purpose is that the cross-ratio is invariant under projective transformations, and that for four points on one line, the cross-ratio is given by $\frac{\overrightarrow{ac} \cdot \overrightarrow{bd}}{\overrightarrow{ad} \cdot \overrightarrow{bc}}$, where \overrightarrow{xy} denotes the oriented distance between x and y on the line.

We will use the cross-ratio the following way: We fix two points on a line and call them 0 and 1. On the line through those points we call the point at infinity ∞ . For a point a on this line the cross-ratio $x := (a, 1; 0, \infty)$ results in the distance between 0 and a scaled by the distance between 0 and 1. Because the cross-ratio is a projective invariant we can fix one line and use the point a for representing the value x . In this way, we have established the coordinates on one line.



■ **Figure 8** Gadgets for addition (left) and multiplication (right) on a line.

For computing on this line, the gadgets for addition and multiplication depicted in Figure 8 can be used. Let us detail the case of multiplication. We are given the points $\infty < 0 < 1 < x < y$ on the line ℓ , and wish to construct a point on ℓ that represents the value $x \cdot y$. Take a second line ℓ_∞ that intersects ℓ in ∞ , and two points a, b on this line. Construct the segments $\overline{by}, \overline{b1}$ and \overline{ax} . Denote the intersection point of \overline{ax} and $\overline{b1}$ by c . Call d the intersection point of \overline{by} and $\ell(0, c)$. The intersection point of ℓ and $\ell(d, a)$ represents the point $x \cdot y =: z$ on ℓ , i.e., $(z, 1; 0, \infty) = (x, 1; 0, \infty) \cdot (y, 1; 0, \infty)$. In a projective realization of the gadget in which the line ℓ_∞ is indeed the line at infinity, the result can be obtained by applying twice the intercept theorem, in the triangles with vertices $0, d, y$ and $0, d, z$, respectively. To add the cross ratios of two points on a line, a similar construction is given in Figure 8.

4.2 The reduction for order types

Using the constructions above we can already model a system of strict polynomial inequalities. However, it is not clear how we can determine the complete order type of the points without knowing the solution of the system. Circumventing this obstacle was the main achievement of Mnëv [20]. We cite one of the main theorems in a simplified version.

► **Theorem 10** ([29],[25]). *Every primary semialgebraic set $V \subseteq \mathbb{R}^d$ is stably equivalent to a semialgebraic set $V' \subseteq \mathbb{R}^n$, with $n = \text{poly}(d)$, for which all defining equations have the form $x_i + x_j = k$ or $x_i \cdot x_j = x_k$ for certain $1 \leq i \leq j < k \leq n$, where the variables $1 = x_1 < x_2 < \dots < x_n$ are totally ordered.*

A *primary semialgebraic set* is a set defined by polynomial equations and strict polynomial inequalities with coefficients in \mathbb{Z} . Although we cannot give a complete definition of *stable equivalence* within the context of this paper, let us just say that two semialgebraic sets V and V' are stably equivalent if one can be obtained from the other by rational transformations and so-called *stable projections*, and that stable equivalence implies *homotopy equivalence*. From the computational point of view, the important property is that V is the empty set if and only if V' is, and that the size of the description of V' in the theorem above is polynomial in the size of the description of V . We call the description of a semialgebraic set V' given in the theorem above the *Shor normal form*.

We can now encode the defining relations of a semialgebraic set given in Shor normal form using abstract order types by simply putting the points $\infty, 0, 1, x_1, \dots, x_n$ in this order on ℓ . To give a complete order type, the orientations of triples including the points of the gadgets and the positions of the gadget on ℓ_∞ have to be specified. This can be done exploiting the fact that the distances between the points a and b of each gadget and their position on ℓ_∞ can be chosen freely. We refer to the references mentioned above for further details. We next show how to implement these ideas to construct a graph G_V associated with a primary semialgebraic set V , such that G_V has a PVG realization if and only if $V \neq \emptyset$.

5 $\exists\mathbb{R}$ -completeness of PVG recognition

The idea to show that PVG recognition is complete in $\exists\mathbb{R}$ is to encode the gadgets described in the previous section in a generalized fan. We therefore consider the gadgets not as a collection of points with given order types, but as a collection of segments between the lines ℓ and ℓ_∞ with given crossing information, i.e., a certain arrangement of the segments of the fan.

We will consider the addition and multiplication gadgets given in Fig. 8, and for a copy g_i of the addition gadget, denote by a_i, b_i, c_i, d_i , and e_i the points corresponding to g_i , and similarly for the multiplication gadget. To formalize the freedom we have in choosing the points a_i and b_i for each addition or multiplication gadget g_i , we make the following two observations. The points of a gadget that do not lie on ℓ are denoted by P_i .

► **Observation 11** ([25],[18]). *The points a_i and b_i can be positioned arbitrarily on ℓ_∞ . The position of the other points of P_i is fully determined by a_i, b_i and the input values on ℓ .*

► **Observation 12** ([25],[18]). *All points of P_i are placed close to a_i if a_i and b_i are placed close to each other. (For each $\varepsilon > 0$ there exists a $\delta > 0$, such $|a_i - b_i| < \delta$ implies $|p - q| < \varepsilon$ for all $p, q \in P_i$.)*

With those two observations in hand, we show we can place the points of the gadgets on ℓ_∞ one by one, such that we have a partial information on the *relative height* of the crossings of the involved segments. This partial information can be combined with the generalized fan construction to force the exact encoding.

Here we need a generalized fan since we cannot obtain the full information of the height all the crossings with the segments of other gadgets, since the position and distance of the other segments of gadgets is influenced by the solution of the inequality system.

For simplicity, we can work in the projective plane. This allows us to apply a projective transformation such that the point ∞ is mapped onto the line at infinity, and the lines ℓ and ℓ_∞ are parallel. Furthermore we can assume ℓ and ℓ_∞ are horizontal lines. In this setting we have to specify a order on the y -coordinate of the intersection points of the segments/the points of the gadgets. Therefore we fix one order of the gadgets g_1, g_2, \dots, g_l on ℓ_∞ .

► **Lemma 13.** *Let V be a nonempty primary semialgebraic set given in Shor normal form and let $g_1, g_{i-1}, g_i, \dots, g_l$ be the gadgets realizing the defining equations, such that g_j is realizing an addition if $j < i$ and a multiplication otherwise. Then there exists a realization such that the order of the y -coordinates of the intersection points is given by*

$$a_1 = \dots = a_l = b_1 = \dots = b_l = f_i = \dots = f_l \quad (1)$$

$$> e_l > d_l > c_l > \dots > e_i > d_i > c_i \quad (2)$$

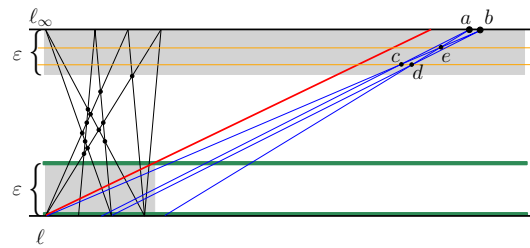
$$> e_{i-1} > c_{i-1} = d_{i-1} > \dots > e_1 > c_1 = d_1 \quad (3)$$

$$> I_2 > \dots > I_l \quad (4)$$

$$> 0 = x_1 = x_2 = \dots = x_k, \quad (5)$$

where I_j denotes the intersections between the segments of the gadget g_k with the segments of the gadgets g_j for $j < k$.

Proof. We fix one solution for the relations defining V . The points on ℓ are fixed realizing this solution. We place the points a_i and b_i such that the other points of the gadgets realize the order of the y -coordinates described in the lemma.



■ **Figure 9** The vertical order of the points in the reduction.

First note that the order of the points within one gadget is determined as described by the construction of the gadgets. The points corresponding to variables are also on ℓ and the points a, b and f all lie on ℓ_∞ . Thus the total relations given in (1) and (5), as well as the relations between each triple of points belonging to one gadget in (2) and (3) are satisfied in all realizations.

We place the points a_i and b_i of the gadgets inductively. Assume that we have placed the first $i - 1$ gadgets such that the inequalities above are satisfied. Now there exists a real ε such that none of the points of the gadgets lies in an ε -neighborhood of ℓ or ℓ_∞ , see Figure 9. For this reason there exists an axis-aligned rectangle of height ε with lower boundary on ℓ , such that every segment drawn so far intersects the upper and the lower boundary of this rectangle (the lower grey box in Figure 9). We now place a_i such that all segments that are constructed for the gadget g_i (blue) intersect the right boundary of this rectangle. This can be achieved by placing a_i further than the intersection point of ℓ_∞ and the supporting line of the diagonal with positive slope of the rectangle (the red segment in Figure 9). This shows that (4) can be satisfied.

To show the inequalities in (2) and (3) hold it remains to check that the points c_i, d_i (and eventually e_i) can be placed in an ε -neighborhood of ℓ_∞ . This can be done, using Observation 12, by placing b_i close to a_i . ◀

► **Theorem 14.** *The recognition of point visibility graphs is $\exists\mathbb{R}$ -complete.*

Proof. For a proof that PVG recognition is in $\exists\mathbb{R}$ we refer to [10]. For the hardness part, the idea of the proof is the following. For a semialgebraic set V we compute the Shor normal form and denote the corresponding primary semialgebraic set by V' . For V' , we can construct the arrangement of pseudosegments that are attached on the lines ℓ and ℓ_∞ . By inverting the projective transformation applied in Lemma 13 we can construct a generalized fan G_V of the pseudosegments between ℓ and ℓ_∞ , such that in any PVG realization the order of the intersection points of the segments satisfies the inequalities in Lemma 13.

The bundles and rays for the generalized fan are added, such that the possible vertical orders are fixed to the ones described in Lemma 13, see Figure 9: We add an orange ray from p through each of the points c_i, d_i and e_i of each gadget $g_i, i \in [l]$. This fixes the inequalities in lines (2)-(3). A green bundle is added before and after each of the sets $I_j, j \in \{2, \dots, l\}$, such that (4) is satisfied.

From this generalized fan we want to construct a point visibility graph G_V . Here we have to be careful with collinearities between points that do not lie on one segment or one ray. Therefore, we show that we can construct the edges and nonedges between points on different segments and different rays, such that they do not restrict *too many* solutions of our strict inequality system. First notice that we can avoid collinearities between points on segments of different gadgets by perturbing the positions of the points a_i, b_i , the exact

position of the bundles, and the distance of the rays within a bundle (we have this freedom in the proof of Lemma 13). So we can assume that the only collinearities of points on different segments appear between segments in one gadget. In the addition gadget we have no three *segments* that intersect in one point. By perturbing the position of the bundles we can avoid collinearities in those gadgets.

In the multiplication gadget we are in the situation that we have three segments $0, 1, x$ (and $0, y, x \cdot y$) that intersect in one point. If the ratio of those three points on ℓ is rational they are (after projective transformations) columns in the integer grid. If those are intersected by a bundle we obtain the points on projective transformation of the integer grid and thus collinearities. The point here is that we can compute during the construction which collinearities appear: the solutions of the original strict inequality system form an open set. In this set we can assume that our solution consists of sufficiently *independent* numbers, e.g. they are algebraically independent over \mathbb{Q} , such that $0, 1, x$ and $0, y, x \cdot y$ only have a rational ratio if x is a coefficient of the inequality system. In this case we can calculate the collinearities. Otherwise, we can perturb the bundles a_i and b_i to avoid collinearities. Hence all collinearities between points on different segments can be computed and do not influence the solvability of the inequality system. This way we can determine all edges of G_V .

The number of vertices of the graph G_V is polynomial in the size of V since calculating the Shor normal form of V gives a description of V' which has size polynomial in the size of V . The number of segments, bundles, rays, and the size of a bundle in the fan are all polynomial in the number of operations in the Shor normal form. All calculations in this construction can be done in polynomial time.

For the $\exists\mathbb{R}$ -hardness it remains to show that the graph G_V is a point visibility graph if and only if V (and thus V') is nonempty. To show that V is nonempty if G_V has a PVG realization we observe that the collinearities from a ray and from a segments stay collinear in each realization by Lemma 7. Thus the gadgets implementing the calculations on ℓ are preserved. Using the cross-ratio as described in Subsection 4.1 a PVG realization encodes a point in V' , and V is nonempty if G_V has a PVG realization.

We show that there exists a PVG realization if V and V' are nonempty. We consider a solution $x \in V'$ and place the points corresponding to the variables on a line ℓ . With points in this position the gadgets implementing the calculations can be realized between ℓ and ℓ_∞ , such that the intersection points of the segments satisfy the order in Lemma 13. \blacktriangleleft

Acknowledgments. We thank an anonymous referee for pointing out an error in the original proof of Lemma 7. The revised proof is largely based on the suggested fix.

References

- 1 James Abello and Krishna Kumar. Visibility graphs and oriented matroids. *Discrete & Computational Geometry*, 28(4):449–465, 2002.
- 2 Karim A. Adiprasito, Arnau Padrol, and Louis Theran. Universality theorems for inscribed polytopes and Delaunay triangulations. *ArXiv e-prints*, 2014.
- 3 Daniel Bienstock. Some provably hard crossing number problems. *Discrete and Computational Geometry*, 6:443–459, 1991.
- 4 John Canny. Some algebraic and geometric computations in PSPACE. In *STOC '88*, pages 460–467. ACM, 1988.
- 5 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008 (third edition).

- 6 Subir K. Ghosh. On recognizing and characterizing visibility graphs of simple polygons. In *1st Scandinavian Workshop on Algorithm Theory (SWAT)*, pages 96–104, 1988.
- 7 Subir K. Ghosh. On recognizing and characterizing visibility graphs of simple polygons. *Discrete & Computational Geometry*, 17(2):143–162, 1997.
- 8 Subir K. Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, 2007.
- 9 Subir K. Ghosh and Partha P. Goswami. Unsolved problems in visibility graphs of points, segments, and polygons. *ACM Computing Surveys (CSUR)*, 46(2):22, 2013.
- 10 Subir K. Ghosh and Bodhayan Roy. Some results on point visibility graphs. In *Algorithms and Computation (WALCOM)*, volume 8344 of *Lecture Notes in Computer Science*, pages 163–175. Springer, 2014.
- 11 Jacob E. Goodman, Richard Pollack, and Bernd Sturmfels. The intrinsic spread of a configuration in \mathbb{R}^d . *Journal of the American Mathematical Society*, pages 639–651, 1990.
- 12 Branko Grünbaum. *Convex Polytopes*, volume 221 (2nd ed.) of *Graduate Texts in Mathematics*. Springer-Verlag, 2003.
- 13 Michael Kapovich and John J. Millson. Universality theorems for configuration spaces of planar linkages. *Topology*, 41:1051–1107, 2002.
- 14 Jan Kára, Attila Pór, and David R. Wood. On the chromatic number of the visibility graph of a set of points in the plane. *Discrete & Computational Geometry*, 34(3):497–506, 2005.
- 15 Jan Kratochvíl and Jirí Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory. Series B*, 62(2):289–315, 1994.
- 16 Jan Kynčl. Simple realizability of complete abstract topological graphs in \mathbb{P} . *Discrete and Computational Geometry*, 45(3):383–399, 2011.
- 17 Tomás Lozano-Pérez and Michael A. Wesley. An algorithm for planning collision-free paths among polyhedral obstacles. *Commun. ACM*, 22(10):560–570, October 1979.
- 18 Jiří Matoušek. Intersection graphs of segments and $\exists\mathbb{R}$. *ArXiv e-prints*, 2014.
- 19 Colin McDiarmid and Tobias Müller. Integer realizations of disk and segment graphs. *Journal of Combinatorial Theory, Series B*, 103(1):114 – 143, 2013.
- 20 Nicolai E. Mněv. 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.
- 21 Joseph O’Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.
- 22 Joseph O’Rourke and Ileana Streinu. Vertex-edge pseudo-visibility graphs: Characterization and recognition. In *Symposium on Computational Geometry*, pages 119–128, 1997.
- 23 Michael S. Payne, Attila Pór, Pavel Valtr, and David R. Wood. On the connectivity of visibility graphs. *Discrete & Computational Geometry*, 48(3):669–681, 2012.
- 24 Attila Pór and David R. Wood. On visibility and blockers. *JoCG*, 1(1):29–40, 2010.
- 25 Jürgen Richter-Gebert. Mněv’s universality theorem revisited. In *Proceedings of the Séminaire Lotharingien de Combinatoire*, pages 211–225, 1995.
- 26 Bodhayan Roy. Point visibility graph recognition is NP-hard. *ArXiv e-prints*, 2014.
- 27 Marcus Schaefer. Complexity of some geometric and topological problems. In *17th International Symposium on Graph Drawing (GD)*, pages 334–344, 2009.
- 28 Marcus Schaefer. Realizability of graphs and linkages. In *Thirty Essays on Geometric Graph Theory*. Springer, 2012.
- 29 Peter W. Shor. Stretchability of pseudolines is NP-hard. *Applied Geometry and Discrete Mathematics—The Victor Klee Festschrift*, 4:531–554, 1991.
- 30 Karl Georg Christian Staudt. *Geometrie der Lage*. Verlag von Bauer und Raspe, 1847.
- 31 Ileana Streinu. Non-stretchable pseudo-visibility graphs. *Comput. Geom.*, 31(3):195–206, 2005.

Geometric Spanners for Points Inside a Polygonal Domain

Mohammad Ali Abam, Marjan Adeli, Hamid Homapour, and Pooya Zafar Asadollahpoor

Department of Computer Engineering, Sharif University of Technology, Iran
abam@sharif.edu, {madeli, homapour, zafar}@ce.sharif.edu

Abstract

Let \mathcal{P} be a set of n points inside a polygonal domain \mathcal{D} . A polygonal domain with h holes (or obstacles) consists of h disjoint polygonal obstacles surrounded by a simple polygon which itself acts as an obstacle. We first study t -spanners for the set \mathcal{P} with respect to the geodesic distance function π where for any two points p and q , $\pi(p, q)$ is equal to the Euclidean length of the shortest path from p to q that avoids the obstacles interiors. For a case where the polygonal domain is a simple polygon (i.e., $h = 0$), we construct a $(\sqrt{10} + \epsilon)$ -spanner that has $O(n \log^2 n)$ edges. For a case where there are h holes, our construction gives a $(5 + \epsilon)$ -spanner with the size of $O(n\sqrt{h} \log^2 n)$.

Moreover, we study t -spanners for the visibility graph of \mathcal{P} ($VG(\mathcal{P})$, for short) with respect to a hole-free polygonal domain \mathcal{D} . The graph $VG(\mathcal{P})$ is not necessarily a complete graph or even connected. In this case, we propose an algorithm that constructs a $(3 + \epsilon)$ -spanner of size $O(n^{4/3+\delta})$. In addition, we show that there is a set \mathcal{P} of n points such that any $(3 - \epsilon)$ -spanner of $VG(\mathcal{P})$ must contain $\Omega(n^2)$ edges.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Geometric Spanners, Polygonal Domain, Visibility Graph

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.186

1 Introduction

Background. Let $\mathcal{G} = (V, E)$ be an undirected edge-weighted graph and let $d_{\mathcal{G}}(p, q)$ be the length of the weighted shortest path from p to q in \mathcal{G} . Let $t \geq 1$ be a real number. The subgraph $\mathcal{S} = (V, E_{\mathcal{S}})$ of \mathcal{G} is called a t -spanner if for any two vertices $p, q \in V$, we have $d_{\mathcal{S}}(p, q) \leq t \cdot d_{\mathcal{G}}(p, q)$. Any path from p to q in \mathcal{S} whose weight is at most $t \cdot d_{\mathcal{G}}(p, q)$ is called a t -path. The dilation or stretch factor of \mathcal{S} is the minimum t for which \mathcal{S} is a t -spanner of \mathcal{G} . The size of \mathcal{S} is defined as the number of edges in $E_{\mathcal{S}}$.

t -spanners have been mostly studied on complete graphs coming from finite metric spaces. Let (\mathcal{P}, d) be a finite metric space where \mathcal{P} is a set of n points. Consider the complete graph \mathcal{G}_c over \mathcal{P} where $wt(p, q) = d(p, q)$ (wt denotes weight) for any two points $p, q \in \mathcal{P}$. For any t -spanner \mathcal{S} of \mathcal{G}_c , we have $d_{\mathcal{S}}(p, q) \leq t \cdot d(p, q)$. Indeed, the spanner \mathcal{S} approximates distances in the metric space up to a factor of t . The t -spanner \mathcal{S} is usually called the t -spanner of the metric space (\mathcal{P}, d) . In this paper, we are interested in spanners in a geometric context, i.e., the metric space comes from a geometric space like the Euclidean space. Here, the graph \mathcal{G}_c is the complete Euclidean graph on \mathcal{P} (i.e., weights are the Euclidean distances). A geometric t -spanner is an Euclidean graph \mathcal{S} on \mathcal{P} such that $d_{\mathcal{S}}(p, q) \leq t \cdot |pq|$ for all points $p, q \in \mathcal{P}$ where $|pq|$ denotes the Euclidean distance between p and q .

In some applications like road networks, when constructing spanners, the main goal is to obtain a small dilation while not using too many edges. However, one may want to obtain



© Mohammad Ali Abam, Marjan Adeli, Hamid Homapour, and Pooya Zafar Asadollahpoor; licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 186–197



Leibniz International Proceedings in Informatics

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

spanners with a number of additional properties such as small weight – weight proportional to the weight of a Minimum Spanning Tree (MST) – and bounded degrees.

Previous work. Althöfer *et al.* [7] were first to study sparse spanners on edge-weighted graphs that have the triangle-inequality property. They showed that for any integer number $t > 0$, there is a $(2t + 1)$ -spanner with $\mathcal{O}(n^{1+1/t})$ edges where n is the number of vertices. This can be applied to any metric space (\mathcal{P}, d) as the complete graph over \mathcal{P} in the metric space has the triangle-inequality property. Geometric spanners have been attracted a lot of attention over the past two decades. It has been shown that for any set of n points in \mathbb{R}^d and any $\varepsilon > 0$, there is a $(1 + \varepsilon)$ -spanner with $\mathcal{O}(n/\varepsilon^{d-1})$ edges – see the recent book by Narasimhan and Smid [13] and references therein for this and many other results on geometric spanners. When the doubling dimension of a metric space is bounded, similar results to the Euclidean setting are possible [12, 14].

Let the points of \mathcal{P} be in a surface $\mathcal{M} \in \mathbb{R}^3$ and let $d_{\mathcal{M}}(p, q)$ be the weight of the shortest (i.e., the minimum weight) path from p to q on \mathcal{M} for any two points $p, q \in \mathcal{P}$. Obviously, $(\mathcal{P}, d_{\mathcal{M}})$ is a metric space and its doubling dimension is not necessarily bounded. Therefore, results to metric spaces with bounded doubling dimension cannot be applied to the metric space $(\mathcal{P}, d_{\mathcal{M}})$ and now the main question is: is it possible to obtain a spanner with a constant stretch factor and a near-linear number of edges for the metric space $(\mathcal{P}, d_{\mathcal{M}})$? Abam *et al.* [3] considered a special case where the surface \mathcal{M} is a plane containing several pillars with width and length of zero but with non-negative height. They assume the points of \mathcal{P} lie at the top of the pillars. This variant can be seen as a set of n weighted points in a plane in which for any two points p and q , their distance is defined to be $wt(p) + |pq| + wt(q)$ where $wt(x)$ is the weight of point x and $|pq|$ is the Euclidean distance of p and q . They presented a $(5 + \varepsilon)$ -spanner with a linear number of edges for any given $\varepsilon > 0$. They also showed that when \mathcal{M} is the boundary of a convex object, it is possible to obtain $(1 + \varepsilon)$ -spanner with a linear number of edges.

Problem statement. Suppose a set \mathcal{P} of n points are given inside a polygonal domain \mathcal{D} which consists of a simple polygon containing h disjoint polygonal holes. The holes and the simple-polygon boundary can be seen as obstacles. Consider the metric space (\mathcal{P}, π) where $\pi(p, q)$ for any points $p, q \in \mathcal{P}$ is equal to the Euclidean length of the shortest path from p to q that avoids the obstacles interiors; the so-called the geodesic distance of p and q . Moreover, let $VG(\mathcal{P})$ be the visibility graph of \mathcal{P} with respect to the polygonal domain \mathcal{D} , i.e., $p, q \in \mathcal{P}$ are connected in $VG(\mathcal{P})$ iff the segment pq avoids the obstacles interiors. Note that $VG(\mathcal{P})$ is not necessarily a complete graph or even a connected graph. In this paper, we investigate the existence of t -spanners with few edges for both the metric space (\mathcal{P}, π) and $VG(\mathcal{P})$. Note that the polygonal domain \mathcal{D} can be seen as a surface. Indeed, obstacles can be seen as walls, tall enough such that any shortest path between two points p and q avoids the walls. Therefore, the metric space (\mathcal{P}, π) is a special case of the metric space $(\mathcal{P}, d_{\mathcal{M}})$ where \mathcal{M} is a surface in \mathbb{R}^3 .

Our results. The first part of our work as explained in Section 2 is devoted to the metric space (\mathcal{P}, π) . For a case where the polygonal domain \mathcal{D} is a simple polygon (i.e., $h = 0$), we construct a $(\sqrt{10} + \varepsilon)$ -spanner that has $\mathcal{O}(n \log^2 n)$ edges. We extend this result to the case where there are h holes. We show that our construction gives a $(5 + \varepsilon)$ -spanner with the size of $\mathcal{O}(n\sqrt{h} \log^2 n)$ for any given $\varepsilon > 0$. The diameter of both spanners is 2. As the second part of our work, in Section 3 we study t -spanners for $VG(\mathcal{P})$. We first show how to obtain

a $(3 + \epsilon)$ -spanner for any given $\epsilon > 0$ of size $O(n^{4/3+\delta})$ for some $\delta > 0$ and then we show that there is a set \mathcal{P} of n points such that any $(3 - \epsilon)$ -spanner of \mathcal{P} must have $\Omega(n^2)$ edges.

2 Spanners for the metric space (\mathcal{P}, π)

Let \mathcal{P} be a set of n points inside a polygonal domain \mathcal{D} which is a simple polygon containing h polygonal disjoint obstacles. Let $\pi(p, q)$ for any two points $p, q \in \mathcal{P}$ be the geodesic distance of p and q with respect to \mathcal{D} . We first present our spanner construction when $h = 0$ in Section 2.1 and then give our general spanner construction in Section 2.2.

2.1 Spanners for points inside a simple polygon

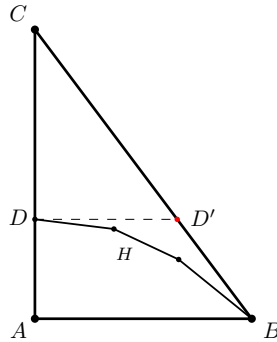
Our spanner construction is based on the SSPD [2, 4] as defined next. For a set Q of n points in \mathbb{R}^d (i.e., the d -dimensional Euclidean space), a pair decomposition of Q is a set of pairs of subsets of Q , such that for every pair of points of $p, q \in Q$ there exists a pair (A, B) in the decomposition such that $p \in A$ and $q \in B$ or vice versa. For a point set A , let $\text{radius}(A)$ be the radius of the minimum enclosing disc of A . An s -Semi-Separated Pair Decomposition (s -SSPD) of Q is a pair decomposition of Q such that for every pair (A, B) , the distance between A and B (i.e. the distance of their minimum enclosing discs) is larger than s times the minimum of the $\text{radius}(A)$ and $\text{radius}(B)$. For a point set Q and a constant $s > 0$, we know there exists an s -SSPD whose weight, $\sum |A| + |B|$ over all pairs, is $O(n \log n)$. The SSPD was introduced to overcome the obesity problem of the Well-Separated Pair Decomposition (WSPD) [9, 15]: there is a set of n points, such that for any WSPD of it, $\sum |A| + |B|$ over all pairs in the WSPD is $\Omega(n^2)$.

Spanner construction. For the given $\epsilon > 0$, we first explain our spanner construction and then prove that the resulting spanner \mathcal{S} is a $(\sqrt{10} + \epsilon)$ -spanner. Our construction is as follows. We partition the simple polygon \mathcal{D} into two simple sub-polygons using a vertical segment ℓ (called the splitting segment) in such a way that each sub-polygon contains at most two-thirds of the points of \mathcal{P} – see [8] for details. For each point $p \in \mathcal{P}$, we compute the point $p_\ell \in \ell$ which has the minimum geodesic distance to p among all points on ℓ . We call p_ℓ the projection of p into ℓ and for a subset A of \mathcal{P} , we define $C_\ell(A)$ to be a point of A whose geodesic distance to ℓ is the smallest. We then compute an s -SSPD for projected points p_ℓ where $s = 4/\epsilon$. Note that some points may have the same projection on ℓ . In this case we treat them as different points while constructing the SSPD. For each pair (A, B) in the SSPD where $\text{radius}(A) \leq \text{radius}(B)$, we add edge $(p, C_\ell(\mathcal{P}(A)))$ to the spanner \mathcal{S} for all points p whose $p_\ell \in A \cup B$ where $\mathcal{P}(A) = \{p \in \mathcal{P} | p_\ell \in A\}$ – recall that an edge (p, q) corresponds to the shortest geodesic path between p and q . We recursively process both simple sub-polygons.

Spanner size. Let $T(n)$ be the size of the resulting spanner \mathcal{S} . Clearly, $T(n) = \sum (|A| + |B|) + T(n_1) + T(n_2)$ where $n_1 + n_2 = n$ and $n_1, n_2 \geq n/3$. Since $\sum (|A| + |B|) = O(n \log n)$ by the SSPD property, we can simply conclude that the spanner size is $O(n \log^2 n)$.

It remains to show that the resulting spanner \mathcal{S} is a $(\sqrt{10} + \epsilon)$ -spanner. We first state the following lemma which plays a key role in our proof showing \mathcal{S} is a $(\sqrt{10} + \epsilon)$ -spanner.

► **Lemma 1.** *Suppose ABC is a right triangle with $\angle CAB = 90$. Let H be a y -monotone path between B and D such that the region bounded by AB , AD , and H is convex where D is some point on edge AC . We have $3|H| + |DC| \leq \sqrt{10}|BC|$ where $|\cdot|$ denotes the Euclidean length.*



■ **Figure 1** A right triangle and a y -monotone convex chain inside it.

Proof. We claim $|H|^2 + |DC|^2 \leq |BC|^2$ and will prove it later. For any two real numbers x and y , we know $(x^2 + y^2)(3^2 + 1^2) \geq (3x + y)^2$. By setting $x = |H|$ and $y = |DC|$, we get $3|H| + |DC| \leq \sqrt{10}|BC|$ as desired.

To prove $|H|^2 + |DC|^2 \leq |BC|^2$, let D' be the point on BC with the same y -coordinate with D . Since H is a convex chain inside triangle $DD'B$ with endpoints D and B , we know

$$|H| \leq |BD'| + |D'D|.$$

Using the above well-known geometric inequality, we have

$$\begin{aligned} |H|^2 + |DC|^2 &\leq (|BD'| + |D'D|)^2 + |DC|^2 \\ &= |BD'|^2 + 2|BD'| \cdot |D'D| + |D'D|^2 + |DC|^2 \\ &= |BD'|^2 + 2|BD'| \cdot |D'D| + |D'C|^2 \\ &\leq |BD'|^2 + 2|BD'| \cdot |D'C| + |D'C|^2 \\ &= (|BD'| + |D'C|)^2 = |BC|^2 \end{aligned}$$



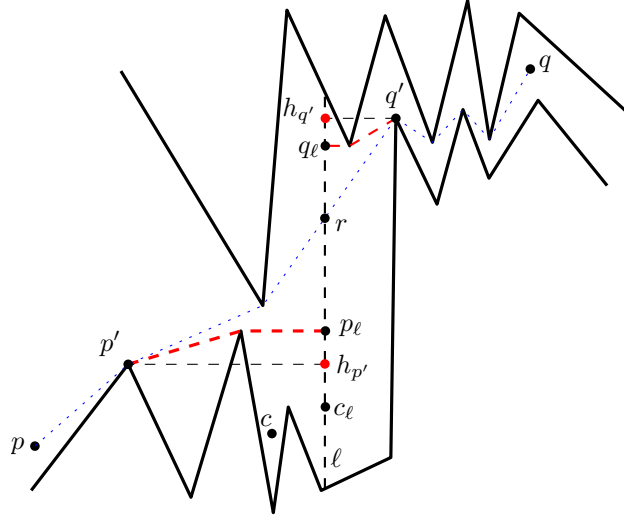
Now, we are ready to prove the main result of this section.

► **Lemma 2.** *The resulting spanner \mathcal{S} of the above construction is a $(\sqrt{10} + \varepsilon)$ -spanner with diameter 2.*

Proof. Any two points $p, q \in \mathcal{S}$ lie at different sides of the splitting segment ℓ at one step of the recursive construction. At this step, there is a semi-separated pair (A, B) that $p_\ell \in A$ and $q_\ell \in B$ or vice versa. WLOG assume $p_\ell \in A$ and $q_\ell \in B$ and moreover assume $\text{radius}(A) \leq \text{radius}(B)$. Let $c = C_\ell(\mathcal{P}(A))$ which of course is a point of \mathcal{P} – see Fig. 2. We recall that among all points whose projections are in A , point c has the minimum geodesic distance to ℓ .

According to our construction at this step of the recursion, edges (p, c) and (q, c) are added to \mathcal{S} . Thus, the length of the shortest path between p and q in \mathcal{S} is at most $\pi(p, c) + \pi(c, q)$. We next show that $\pi(p, c) + \pi(c, q) \leq (\sqrt{10} + \varepsilon)\pi(p, q)$. This implicitly shows the diameter of \mathcal{S} is 2.

Let $\text{SP}(x, y)$ be the shortest path from point x to y with respect to \mathcal{D} for any two points x and y . By the definition of π , the Euclidean length of $\text{SP}(x, y)$ is $\pi(x, y)$. $\text{SP}(p, q)$ definitely intersects ℓ at some point, say r . Let p' (q') be the point at which $\text{SP}(p, q)$ and $\text{SP}(p, p_\ell)$ ($\text{SP}(q, q_\ell)$) get separated – see Fig. 2 to get insight to our notations. It is clear both $\text{SP}(p', p_\ell)$



■ **Figure 2** The splitting segment ℓ partitions the simple polygon into two simple sub-polygons such that each part has at most two-thirds of the points. The projections of points into ℓ are depicted with subscript ℓ .

$SP(q', q_\ell)$ are y -monotone convex chains. $SP(p, q)$ consists of $SP(p, p')$, $SP(p', r)$, $SP(r, q')$ and $SP(q', q)$. We know $\pi(p', r) \geq |p'r|$ and $\pi(q', r) \geq |q'r|$. If we let $h_{p'}$ and $h_{q'}$ be the perpendicular projections of p' and q' on ℓ , in both triangles $q'h_{q'}r$ and $p'h_{p'}r$, the conditions of Lemma 1 hold. All these observations help us to prove the lemma as follows.

Since the distance function π has the triangle-inequality property, we have:

$$\begin{aligned}\pi(p, c) &\leq \pi(p, p_\ell) + |p_\ell c_\ell| + \pi(c_\ell, c) \\ \pi(c, q) &\leq \pi(c, c_\ell) + |c_\ell q_\ell| + \pi(q_\ell, q).\end{aligned}$$

Considering $|c_\ell q_\ell| \leq |c_\ell p_\ell| + |p_\ell r| + |r q_\ell|$ and $\pi(c, c_\ell) \leq \pi(p, p_\ell)$, therefore:

$$\begin{aligned}\pi(p, c) + \pi(c, q) &\leq 3\pi(p, p_\ell) + 2|p_\ell c_\ell| + |p_\ell r| + |r q_\ell| + \pi(q_\ell, q) \\ &= 3\pi(p, p') + 3\pi(p', p_\ell) + 2|p_\ell c_\ell| + |p_\ell r| + |r q_\ell| + \pi(q_\ell, q') + \pi(q', q).\end{aligned}$$

We can apply Lemma 1 to both triangles $q'h_{q'}r$ and $p'h_{p'}r$ and get the following inequalities

$$\begin{aligned}3\pi(p', p_\ell) + |p_\ell r| &\leq \sqrt{10}\pi(p', r) \\ |r q_\ell| + \pi(q_\ell, q') &\leq \sqrt{10}\pi(r, q').\end{aligned}$$

These together yield:

$$3\pi(p', p_\ell) + |p_\ell r| + |r q_\ell| + \pi(q_\ell, q') \leq \sqrt{10}\pi(p', q').$$

Finally, since in the semi-separated pair (A, B) the distance between each two points in A is at most $\frac{2}{s}$ times of the distance between each two points of A and B , we can get:

$$|p_\ell c_\ell| \leq \frac{2}{s}|p_\ell q_\ell| \leq \frac{2}{s}\pi(p, q).$$

If we set $s = \frac{4}{\varepsilon}$, the following inequality holds:

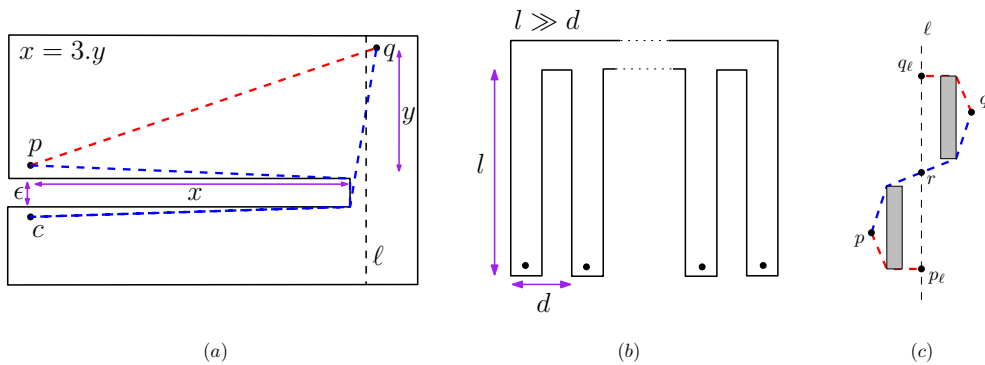


Figure 3 (a) Tight example for the given algorithm in Section 2.1, (b) Any $(2 - \epsilon)$ -spanner in a simple polygonal domain must contain $\Omega(n^2)$ edges, (c) The key property in Lemma 1 does not hold anymore for a polygonal domain with holes.

$$\begin{aligned} \pi(p, c) + \pi(c, q) &\leq 3\pi(p, p') + \sqrt{10}\pi(p', q') + 2|p_\ell c_\ell| + \pi(q', q) \\ &\leq \left(\sqrt{10} + \frac{4}{s}\right)\pi(p, q). \end{aligned}$$



Tight example. As a tight example for our construction, consider the simple polygon in Fig. 3(a) in which $\pi(p, q)$ equals $\sqrt{10}y$ while the shortest path in \mathcal{S} is $10y$.

Lower bound. Consider the simple polygon in Fig. 3(b). When d gets close to 0, $\pi(p, q)$ gets close to $2l$ for any two points p and q . If there is no edge between p and q , the shortest path in \mathcal{S} must go through at least one intermediate vertex, say t . Therefore, the length of the shortest path from p to q , which is at least $\pi(p, t) + \pi(t, q)$, becomes greater than $(2 - \epsilon)\pi(p, q)$ if d is chosen small enough. This implies that to get a $(2 - \epsilon)$ -spanner, we need all edges.

Putting all this together, we get the following theorem.

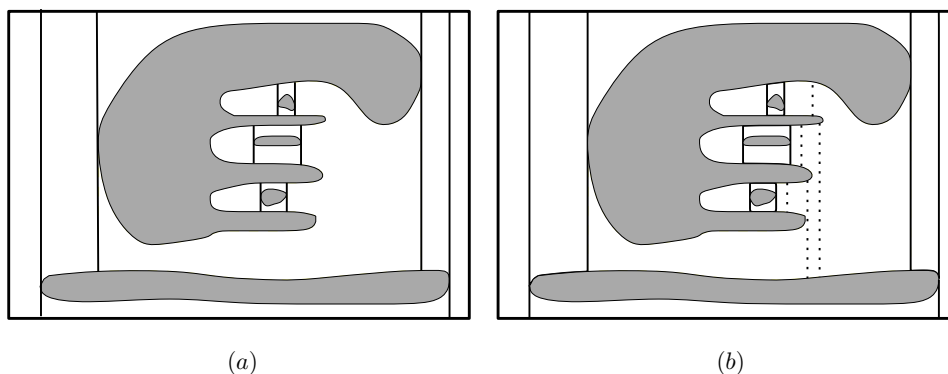
► **Theorem 3.** Let $\epsilon > 0$ be a given real number. Suppose a set \mathcal{P} of n points is given inside a simple polygon \mathcal{D} . There is a $(\sqrt{10} + \epsilon)$ -spanner with diameter 2 of size $\mathcal{O}(n \log^2 n)$ for the metric space (\mathcal{P}, π) . Moreover, there is a set \mathcal{P} of n points such that any $(2 - \epsilon)$ -spanner of the metric space (\mathcal{P}, π) must contain $\Omega(n^2)$ edges.

2.2 Spanners for points inside a polygonal domain with h holes

Suppose the polygonal domain \mathcal{D} contains h disjoint polygonal holes. Our spanner construction is based on the following decomposition.

► **Lemma 4.** The polygonal domain \mathcal{D} with h holes can be decomposed into $\mathcal{O}(h)$ simple polygons using $\mathcal{O}(h)$ vertical segments (called splitting segments) avoiding the holes interiors such that each simple polygon has at most 3 splitting segments on its boundary.

Proof. As the first step, from the leftmost and rightmost points of each obstacle, we draw two vertical extensions; one going downward until an obstacle is hit and one going upward until an obstacle is hit – see Fig. 4(a). This clearly decomposes the polygonal domain into



■ **Figure 4** (a) Planar decomposition of the polygonal domain \mathcal{D} (first step), (b) Decomposing regions with more than three vertical extensions (second step).

$\mathcal{O}(h)$ simple polygons. But one simple polygon may have $m > 3$ vertical extensions on its boundary. In this case, as the second step, we draw $\mathcal{O}(m)$ vertical extensions inside the simple polygon and decompose it into $\mathcal{O}(m)$ simple polygons such that each new simple polygon has at most three vertical extensions on its boundary. To do that, we first draw a vertical extension such that on each of its side there are roughly half of the vertical extensions. We continue recursively on both sides – see Fig. 4(b). The number of the extra vertical extensions satisfies this recursion: $T(m) = 2T(m/2) + 1$, $T(3) = 0$. Therefore, $T(m) = \mathcal{O}(m)$. As each vertical extension of the first step of the construction is adjacent to at most two simple polygons, the total number of the extra extensions is $\mathcal{O}(h)$. ◀

Suppose the decomposition described in Lemma 4 is available to us. We construct a vertex-weighted graph $\mathcal{G}_{\mathcal{D}}$ as follows. We assign a vertex to each simple polygon and associate it with the number of points in \mathcal{P} that are contained in that simple polygon as its weight. We connect two vertices if their corresponding simple polygons are adjacent. Obviously, $\mathcal{G}_{\mathcal{D}}$ is a planar graph with $\mathcal{O}(h)$ vertices. Our divide-and-conquer construction algorithm uses the following well-known theorem for planar graphs.

► **Theorem 5** ([6]). *Suppose $\mathcal{G} = (V, E)$ is a planar vertex-weighted graph with $|V| = m$. Then, there is a (\sqrt{m}) -separator for \mathcal{G} , i.e., V can be partitioned into three sets A , B and C such that (i) $|C| = \mathcal{O}(\sqrt{m})$, (ii) there is no edge between A and B and (iii) $wt(A), wt(B) \leq 2/3wt(V)$, where $wt(X)$ is the weight summation over all vertices in X .*

Theorem 5 can be applied to the graph $\mathcal{G}_{\mathcal{D}}$ as it is a planar graph. In the following, we explain in details how to construct a spanner \mathcal{S} for the metric space (\mathcal{P}, π) .

1. We first construct $\mathcal{G}_{\mathcal{D}}$ and compute its $\mathcal{O}(\sqrt{h})$ -separator. Let A , B and C be the three sets defined in Theorem 5.
2. We collect $\mathcal{O}(\sqrt{h})$ splitting segments into a set H . More precisely, for each vertex of C (we know $|C| = \mathcal{O}(\sqrt{h})$), we add at most the three splitting segments that appear on the boundary of the simple polygon corresponding to the vertex.
3. For each splitting segment ℓ in H , we apply one recursive step of the given algorithm in Section 2.1.
4. We recursively process the induced subgraphs on A and B until one vertex is left. Each vertex at the last level of the recursion corresponds to a simple polygon in the decomposition of Lemma 4. For each such simple polygon, we apply the whole algorithm given in Section 2.1.

Spanner size. Like the argument given in Section 2.1, at each step of the recursion, for each splitting segment, we add $\mathcal{O}(n \log n)$ edges, and in total for $\mathcal{O}(\sqrt{h})$ splitting segments we add $\mathcal{O}(\sqrt{hn} \log n)$ edges. The whole recursive algorithm except at the leaves of the recursion tree, adds $\mathcal{O}(\sqrt{hn} \log^2 n)$ edges. At the leaf v , we add $\mathcal{O}(n_v \log^2 n_v)$ edges where n_v is the number of points inside the corresponding simple polygon. We know $\sum n_v = n$ and therefore, the total added edges at the leaves is $\mathcal{O}(n \log^2 n)$. All this together state that the spanner size is $\mathcal{O}(\sqrt{hn} \log^2 n)$.

Stretch factor. It is tempting to believe that using the argument of Section 2.1, we can show that the spanner \mathcal{S} is a $(\sqrt{10} + \varepsilon)$ -spanner. But unfortunately, a key property that Lemma 1 relies on, does not hold anymore for a polygon domain with holes. This key property is: $\text{SP}(p, r)$ (i.e., the shortest path from p to r) and $\text{SP}(p, p_\ell)$ topologically are the same. When there are holes, this may not happen as depicted in Fig. 3(c). In the figure, $\text{SP}(p, r)$ goes above the specified hole and $\text{SP}(p, p_\ell)$ goes below that hole. Fortunately, we still can show that the spanner \mathcal{S} has a constant stretch factor.

► **Lemma 6.** *The resulting spanner \mathcal{S} of the above construction is a $(5 + \varepsilon)$ -spanner of the metric space (\mathcal{P}, π) .*

Proof. Consider the top level of our recursive construction. The polygonal domain \mathcal{D} is partitioned into three components, one of which is the separator – see Fig. 5. For any two points p and q which are (i) not in the same component or (ii) in the same separator component but in different simple polygons, the shortest paths from p to q intersects at least one of $\mathcal{O}(\sqrt{h})$ splitting segments collected from the separator. Let ℓ be such a splitting segment. Consider the step of the algorithm working on ℓ . There is a semi-separated pair (A, B) such that $p_\ell \in A$ and $q_\ell \in B$ or vice versa. WLOG assume $p_\ell \in A$ and $q_\ell \in B$ and assume $\text{radius}(A) \leq \text{radius}(B)$. If we let $c = C_\ell(\mathcal{P}(A))$, we know edges (p, c) and (q, c) exist in spanner \mathcal{S} . Hence, the shortest path between p and q in \mathcal{S} is at most $\pi(p, c) + \pi(c, q)$. According to the triangle inequality of π , we have:

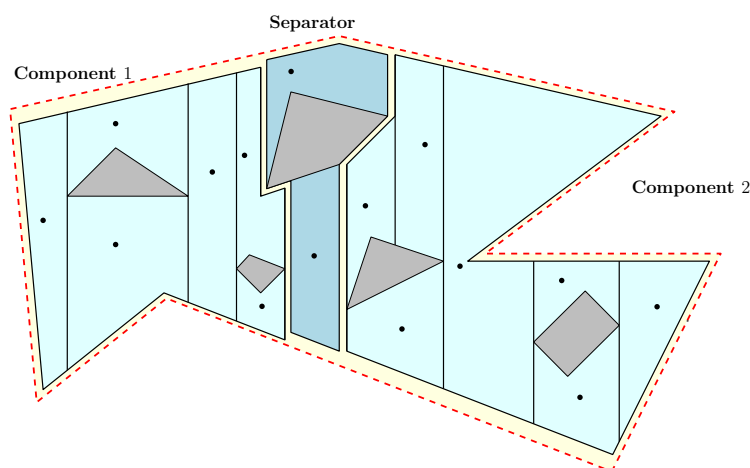
$$\begin{aligned} \pi(p, c) &\leq \pi(p, p_\ell) + |p_\ell c_\ell| + \pi(c_\ell, c) \\ \pi(c, q) &\leq \pi(c, c_\ell) + |c_\ell q_\ell| + \pi(q_\ell, q). \end{aligned}$$

We know:

- $\pi(c, c_\ell)$:
 $\pi(c, c_\ell) \leq \pi(p, p_\ell) \leq \pi(p, q)$
- $\pi(p, p_\ell) + \pi(q_\ell, q)$:
 $\pi(p, p_\ell) + \pi(q_\ell, q) \leq \pi(p, r) + \pi(r, q) = \pi(p, q)$
- $|c_\ell q_\ell|$:
 since $|p_\ell r| \leq \pi(p_\ell, p) + \pi(p, r)$ and $\pi(p, p_\ell) \leq \pi(p, r)$ (the same holds for q and q_ℓ), then :

$$\begin{aligned} |c_\ell q_\ell| &\leq |c_\ell p_\ell| + |p_\ell q_\ell| \\ &\leq \frac{2}{s} \pi |p_\ell q_\ell| + |p_\ell q_\ell| \\ &\leq \left(\frac{2}{s} + 1\right) (|p_\ell r| + |r q_\ell|) \\ &\leq \left(\frac{2}{s} + 1\right) (2\pi(p, q)) \end{aligned}$$
- $|p_\ell c_\ell|$:
 From $c_\ell, p_\ell \in A, q_\ell \in B$ and the SSPD property, we have:

$$|p_\ell c_\ell| \leq \frac{2}{s} |p_\ell q_\ell| \leq \frac{4}{s} \pi(p, q)$$



■ **Figure 5** Any path from one component to another one must intersect the separator's boundaries.

All this together give us:

$$\pi(p, q) \leq \pi(p, c) + \pi(c, q) \leq \left(5 + \frac{8}{s}\right)\pi(p, q)$$

We just need to set $s = \frac{8}{\varepsilon}$. Considering the top level of the recursive construction in the proof can be adjusted to the level at which properties (i) or (ii) are satisfied or both points p and q lie in a simple polygon and their shortest path does not intersect any splitting segments of the separators. In the latter, since we run the whole algorithm of Section 2.1, certainly there is a $(\sqrt{10} + \varepsilon)$ -path between p and q . ◀

To summarize, we get the following theorem.

► **Theorem 7.** *Let $\varepsilon > 0$ be a given real number. Suppose a set \mathcal{P} of n points is given inside a simple polygon \mathcal{D} containing h holes. There is a $(5 + \varepsilon)$ -spanner with diameter 2 of size $\mathcal{O}(n\sqrt{h}\log^2 n)$ for the metric space (\mathcal{P}, π) .*

3 Spanners for the visibility graph

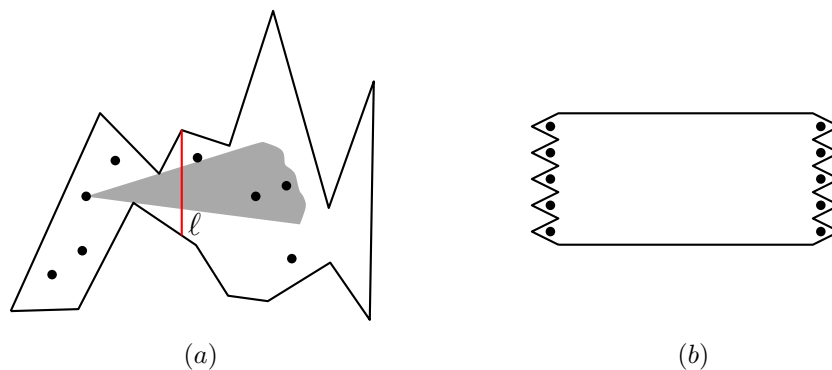
Let \mathcal{P} be a set of n points inside a simple polygon \mathcal{D} (i.e., a polygonal domain without hole). Let $VG(\mathcal{P})$ be the visibility graph of \mathcal{P} , which is not necessarily connected. Here, the goal is to find a t -spanner \mathcal{S} with few edges of $VG(\mathcal{P})$, that is, for any two points $p, q \in \mathcal{P}$, their shortest distance in \mathcal{S} is at most t times their shortest distance in $VG(\mathcal{P})$.

Since $VG(\mathcal{P})$ is a special case of weighted graphs holding triangle-inequality property, by applying the algorithm given in [7] we can get the following spanner.

► **Theorem 8.** *For any integer $t > 0$, there is a $(2t + 1)$ -spanner \mathcal{S} such that the number of edges in \mathcal{S} is $\mathcal{O}(n^{1+1/t})$.*

If we set $t = 1$, the above theorem gives us a 3-spanner of size $\mathcal{O}(n^2)$. We next show that it is possible to get $(3 + \varepsilon)$ -spanner of size $\mathcal{O}(n^{4/3+\delta})$ for any $\varepsilon > 0$.

Spanner construction. We first decompose \mathcal{D} using a splitting segment ℓ into two simple polygons \mathcal{D}_L and \mathcal{D}_R each containing at most $2/3n$ points of \mathcal{P} . Let $VG_\ell(\mathcal{P})$ be the subgraph of $VG(\mathcal{P})$ containing every edge of $VG(\mathcal{P})$ that intersects ℓ . We next explain how to find a



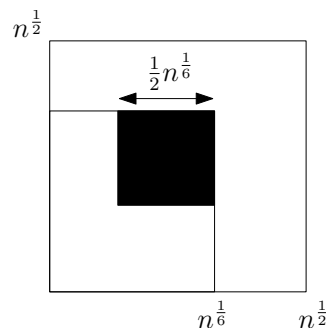
■ **Figure 6** (a) The visibility cone of a point. (b) Any $(3 - \varepsilon)$ -spanner of the visibility graph has size of $\Omega(n^2)$.

$(3 + \varepsilon)$ -spanner of $VG_\ell(\mathcal{P})$ with $O(n^{4/3+\delta})$ edges. By recursing on \mathcal{D}_L and \mathcal{D}_R , we can get a $(3 + \varepsilon)$ -spanner of $VG(\mathcal{P})$ with $O(n^{4/3+\delta})$ edges.

The main idea is to represent $VG_\ell(\mathcal{P})$ which is a bipartite graph, as the union of some complete bipartite graphs and find a spanner for each complete bipartite graph. Let $\sigma(p)$ be the visibility cone of p , that is, all half-lines originating from p and intersecting ℓ – see Fig. 6(a). (p, q) is an edge of $VG_\ell(\mathcal{P})$ if and only if $q \in \sigma(p)$ and $p \in \sigma(q)$. For ease of presentation, we call points in \mathcal{D}_L and \mathcal{D}_R red points and blue points, respectively. We map each $\sigma(p)$ to a segment in the dual plane by the standard transformation [11] where a point (a, b) is mapped to the line $y = ax + b$ and vice versa. It is easy to see that (p, q) is an edge of $VG_\ell(\mathcal{P})$ if and only if the segments corresponding to $\sigma(p)$ and $\sigma(q)$ intersect each other. Therefore, the edges in $VG_\ell(\mathcal{P})$ correspond to the intersection of two segments sets and vice versa. Let us call them red segments (corresponding to the red points) and blue segments (corresponding to the blue points). We then construct a segment-intersection-searching data structure [5] for the red segments, which is a multilevel partition tree, each of whose nodes is associated with a canonical subset of red segments. The total size of canonical subsets is $O(n^{4/3+\delta})$. For every blue segment, all red segments intersecting it can be reported as a union of $O(n^{1/3+\delta})$ pairwise disjoint canonical subsets which is useful to construct a clique cover of VG_ℓ without computing all intersections. Therefore, we can represent $VG_\ell(\mathcal{P})$ as the union of some complete bipartite graphs with the total size $O(n^{4/3+\delta})$. We then compute a $(3 + \varepsilon)$ -spanner of size $O(m \log m)$ for each complete bipartite graph with m vertices as described in [1].

Lower bound. Consider a set of $n/2$ points on a segment whose endpoints are $(0, 0)$ and $(0, \alpha)$ and a set of $n/2$ points on a segment whose endpoints are $(1, 0)$ and $(1, \alpha)$. We can put all n points in a simple polygon as depicted in Fig. 6(b) such that every point on each segment can see any point on the other segment and any two points on a segment cannot see each other. Let p and q be two points on the different segments. For an spanner \mathcal{S} of the visibility graph, if the edge (p, q) does not exist in the spanner, any path between p and q in \mathcal{S} must have at least three edges and since the length of each edge is almost the length of (p, q) – we can choose α small enough depending on ε – the spanner cannot be a $(3 - \varepsilon)$ -spanner. Therefore, the spanner must have every edge of the visibility graph which implies the spanner size is $\Omega(n^2)$.

Putting all together, we get the following result.



■ **Figure 7** Lower bound construction.

► **Theorem 9.** For any given $\varepsilon > 0$, there is a $(3 + \varepsilon)$ -spanner of $VG(\mathcal{P})$ that contains $O(n^{4/3+\delta})$ edges for some $\delta > 0$. Moreover, there is a set \mathcal{P} such that any $(3 - \varepsilon)$ -spanner of the visibility graph $VG(\mathcal{P})$ has size of $\Omega(n^2)$.

► **Remark.** If the polygonal domain \mathcal{D} has h holes, we can apply the technique of Section 2.2 to get a $(3 + \varepsilon)$ -spanner of size $O(\sqrt{h}n^{4/3+\delta})$. Moreover, it is possible to find a set \mathcal{P} of n points such that any $(5 - \varepsilon)$ -spanner must have $\Omega(n^{4/3})$ edges. An instance of the line-point incidence problem [10] with $\Omega(n^{4/3})$ incidences can be used to construct the desired instance. To sketch the overall plan, we introduce two sets A (red points) and B (blue points) inside a polygon domain with holes such that (i) for any $p, p' \in A$ and $q, q' \in B$, $|pq|$ is almost $|p'q'|$ and (ii) two points from A cannot see each other and the same holds for B , and (iii) there is no cycle of length 4 in the bipartite visibility graph and (iv) the number of edges in the visibility graph is $\Omega(n^{4/3})$. All this together mean the girth is at least 6 and all edges have almost the same weight. Therefore, any $(5 - \varepsilon)$ -spanner must contain $\Omega(n^{4/3})$ edges. To get the desired point set, consider a $\sqrt{n} \times \sqrt{n}$ grid as depicted in Fig. 7. The number of grid points (p, q) inside the black square where $GCD(p, q) = 1$ is $\Omega(n^{1/3})$. Look at each of these points as a vector. For each vector, we draw a line parallel to the vector from each grid point. The number of different lines for each vector is $O(n^{2/3})$ and the number of incidences is obviously n . In total we have $O(n)$ lines and $\Omega(n^{4/3})$ incidences. We can look at the lines as blue segments. We also put n red parallel segments in the grid points with the negative slope α and very small length. Now, we dualize the segments to cones with the standard transformation. Let A and B be the dual of red and blue segments respectively – note that points in A and B are apexes of the cones. We can put some obstacles such that for every point in A or B , the dual of the visibility cone is exactly the corresponding segment in our incidence construction. It is easy to see that A and B satisfy the required properties by making α and the scale of the grid smaller.

Acknowledgements. The first author would like to thank Pankaj Agarwal and Mark de Berg who initiated the problem and gave valuable suggestions. Moreover, we would like to thank the anonymous reviewers for their valuable comments.

References

- 1 Mohammad Ali Abam, Paz Carmi, Mohammad Farshi, and Michiel Smid. On the power of the semi-separated pair decomposition. *Computational Geometry*, 46(6):631–639, 2013.
- 2 Mohammad Ali Abam, Mark De Berg, Mohammad Farshi, and Joachim Gudmundsson. Region-fault tolerant geometric spanners. *Discrete & Computational Geometry*, 41(4):556–582, 2009.

- 3 Mohammad Ali Abam, Mark De Berg, Mohammad Farshi, Joachim Gudmundsson, and Michiel Smid. Geometric spanners for weighted point sets. *Algorithmica*, 61(1):207–225, 2011.
- 4 Mohammad Ali Abam and Sariel Har-Peled. New constructions of sspds and their applications. *Computational Geometry*, 45(5):200–214, 2012.
- 5 Pankaj K Agarwal and Micha Sharir. Applications of a new space-partitioning technique. *Discrete & Computational Geometry*, 9(1):11–38, 1993.
- 6 Noga Alon, Paul Seymour, and Robin Thomas. Planar separators. *SIAM Journal on Discrete Mathematics*, 7(2):184–193, 1994.
- 7 Ingo Althöfer, Gautam Das, David Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9(1):81–100, 1993.
- 8 Prosenjit Bose, Jurek Czyzowicz, Evangelos Kranakis, Danny Krizanc, and Anil Maheshwari. Polygon cutting: Revisited. In *Proceedings of Japanese Conference on Discrete & Computational Geometry (JCDCG'98)*, volume 1763 of *LNCS*, pages 81–92. Springer, 1998.
- 9 Paul B Callahan and S Rao Kosaraju. A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM*, 42(1):67–90, 1995.
- 10 Kenneth L Clarkson, Herbert Edelsbrunner, Leonidas J Guibas, Micha Sharir, and Emo Welzl. Combinatorial complexity bounds for arrangements of curves and spheres. *Discrete & Computational Geometry*, 5(1):99–160, 1990.
- 11 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.
- 12 Sariel 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.
- 13 Giri Narasimhan and Michiel Smid. *Geometric spanner networks*. Cambridge University Press, 2007.
- 14 Kunal Talwar. Bypassing the embedding: algorithms for low dimensional metrics. In *Proceedings of Annual ACM symposium on Theory of computing*, pages 281–290, 2004.
- 15 Kasturi R Varadarajan. A divide-and-conquer algorithm for min-cost perfect matching in the plane. In *Proceedings of Annual Symposium on Foundations of Computer Science*, pages 320–331, 1998.

An Optimal Algorithm for the Separating Common Tangents of Two Polygons

Mikkel Abrahamsen*

Department of Computer Science, University of Copenhagen
Universitetsparken 5
DK-2100 Copenhagen Ø
Denmark
miab@di.ku.dk

Abstract

We describe an algorithm for computing the separating common tangents of two simple polygons using linear time and only constant workspace. A tangent of a polygon is a line touching the polygon such that all of the polygon lies to the same side of the line. A separating common tangent of two polygons is a tangent of both polygons where the polygons are lying on different sides of the tangent. Each polygon is given as a read-only array of its corners. If a separating common tangent does not exist, the algorithm reports that. Otherwise, two corners defining a separating common tangent are returned. The algorithm is simple and implies an optimal algorithm for deciding if the convex hulls of two polygons are disjoint or not. This was not known to be possible in linear time and constant workspace prior to this paper.

An outer common tangent is a tangent of both polygons where the polygons are on the same side of the tangent. In the case where the convex hulls of the polygons are disjoint, we give an algorithm for computing the outer common tangents in linear time using constant workspace.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases planar computational geometry, simple polygon, common tangent, optimal algorithm, constant workspace

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.198

1 Introduction

The problem of computing common tangents of two given polygons has received some attention in the case where the polygons are convex. For instance, it is necessary to compute outer common tangents of disjoint convex polygons in the classic divide-and-conquer algorithm for the convex hull of a set of n points in the plane by Preparata and Hong [12]. They give a naïve linear time algorithm for outer common tangents since that suffices for an $O(n \log n)$ time convex hull algorithm. The problem is also considered in various dynamic convex hull algorithms [5, 8, 11]. Overmars and van Leeuwen [11] give an $O(\log n)$ time algorithm for computing an outer common tangent of two disjoint convex polygons when a separating line is known, where each polygon has at most n corners. Kirkpatrick and Snoeyink [9] give an $O(\log n)$ time algorithm for the same problem, but without using a separating line. Guibas et al. [7] give an $\Omega(\log^2 n)$ lower bound on the time required to compute an outer common tangent of two intersecting convex polygons, even if it is known that they intersect in at most two points. They also describe an algorithm achieving that bound.

* Research partly supported by Mikkel Thorup's Advanced Grant from the Danish Council for Independent Research under the Sapere Aude research career programme.



© Mikkel Abrahamsen;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 198–208



Leibniz International Proceedings in Informatics

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

Toussaint [13] considers the problem of computing separating common tangents of convex polygons and notes that the problem occurs in problems related to visibility, collision avoidance, range fitting, etc. He gives a linear time algorithm. Guibas et al. [7] give an $O(\log n)$ time algorithm for the same problem.

All the here mentioned works make use of the convexity of the polygons. If the polygons are not convex, one can use a linear time algorithm to compute the convex hulls before computing the tangents [6, 10]. However, if the polygons are given in read-only memory, it requires $\Omega(n)$ extra bits to store the convex hulls. In this paper, we also obtain linear time while using only constant workspace, i.e. $O(\log n)$ bits. For the outer common tangents, we require the convex hulls of the polygons to be disjoint. There has been some recent interest in constant workspace algorithms for geometric problems, see for instance [1, 2, 3, 4].

The problem of computing separating common tangents is of special interest because these only exist when the convex hulls of the polygons are disjoint, and our algorithm detects if they are not. Thus, we also provide an optimal algorithm for deciding if the convex hulls of two polygons are disjoint or not. This was to the best of our knowledge not known to be possible in linear time and constant workspace prior to our work.

1.1 Notation and some basic definitions

Given two points a and b in the plane, the closed line segment with endpoints a and b is written ab . When $a \neq b$, the line containing a and b which is infinite in both directions is written $\mathcal{L}(a, b)$.

Define the dot product of two points $x = (x_0, x_1)$ and $y = (y_0, y_1)$ as $x \cdot y = x_0y_0 + x_1y_1$, and let $x^\perp = (-x_1, x_0)$ be the counterclockwise rotation of x by the angle $\pi/2$. Now, for three points a , b , and c , we define $\mathcal{T}(a, b, c) = \text{sgn}((b - a)^\perp \cdot (c - b))$, where sgn is the sign function. $\mathcal{T}(a, b, c)$ is 1 if c is to the left of the directed line from a to b , 0 if a , b , and c are collinear, and -1 if c is to the right of the directed line from a to b . We see that

$$\mathcal{T}(a, b, c) = \mathcal{T}(b, c, a) = \mathcal{T}(c, a, b) = -\mathcal{T}(c, b, a) = -\mathcal{T}(b, a, c) = -\mathcal{T}(a, c, b).$$

We also note that if a' and b' are on the line $\mathcal{L}(a, b)$ and appear in the same order as a and b , i.e., $(b - a) \cdot (b' - a') > 0$, then $\mathcal{T}(a, b, c) = \mathcal{T}(a', b', c)$ for every point c .

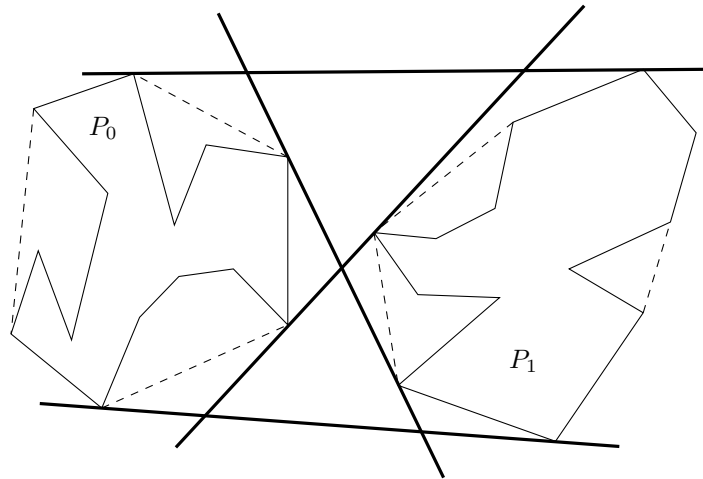
The *left half-plane* $\text{LHP}(a, b)$ is the closed half plane with boundary $\mathcal{L}(a, b)$ lying to the left of directed line from a to b , i.e., all the points c such that $\mathcal{T}(a, b, c) \geq 0$. The *right half-plane* $\text{RHP}(a, b)$ is just $\text{LHP}(b, a)$.

Assume for the rest of this paper that P_0 and P_1 are two simple polygons in the plane with n_0 and n_1 corners, respectively, where P_k is defined by its corners $p_k[0], p_k[1], \dots, p_k[n_k - 1]$ in clockwise or counterclockwise order, $k = 0, 1$. Indices of the corners are considered modulo n_k , so that $p_k[i]$ and $p_k[j]$ are the same corner when $i \equiv j \pmod{n_k}$.

We assume that the corners are in general position in the sense that P_0 and P_1 have no common corners and the combined set of corners $\bigcup_{k=0,1} \{p_k[0], \dots, p_k[n_k - 1]\}$ contains no three collinear corners.

A *tangent* of P_k is a line ℓ such that ℓ and P_k are not disjoint and such that P_k is contained in one of the closed half-planes defined by ℓ . The line ℓ is a *common tangent* of P_0 and P_1 if it is a tangent of both P_0 and P_1 . A common tangent is an *outer common tangent* if P_0 and P_1 are on the same side of the tangent, and otherwise the tangent is *separating*. See Figure 1.

For a simple polygon P , we let $\mathcal{H}(P)$ be the convex hull of P . The following lemma is a well-known fact about $\mathcal{H}(P)$.



■ **Figure 1** Two polygons P_0 and P_1 and their four common tangents as thick lines. The edges of the convex hulls which are not edges of P_0 or P_1 are dashed.

► **Lemma 1.** *For a simple polygon P , $\mathcal{H}(P)$ is a convex polygon and the corners of $\mathcal{H}(P)$ appear in the same cyclic order as they do on P .*

The following lemma states folklore properties of tangents of polygons.

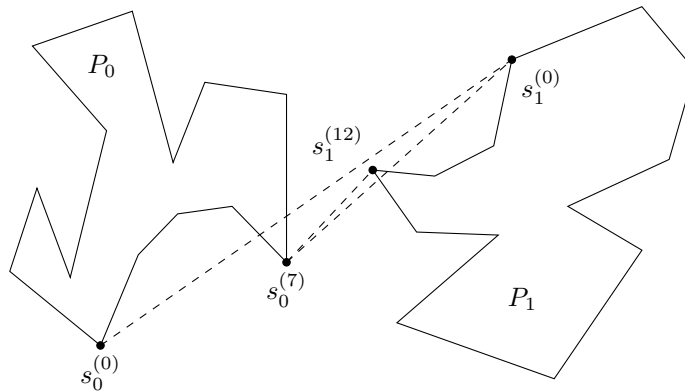
► **Lemma 2.** *A line is a tangent of a polygon P if and only if it is a tangent of $\mathcal{H}(P)$. Under our general position assumptions, the following holds: If one of $\mathcal{H}(P_0)$ and $\mathcal{H}(P_1)$ is completely contained in the other, there are no outer common tangents of P_0 and P_1 . Otherwise, there are two or more. There are exactly two if P_0 and P_1 are disjoint. If $\mathcal{H}(P_0)$ and $\mathcal{H}(P_1)$ are not disjoint, there are no separating common tangents of P_0 and P_1 . Otherwise, there are exactly two.*

2 Computing separating common tangents

In this section, we assume that the corners of P_0 and P_1 are both given in counterclockwise order. We prove that Algorithm 1 returns a pair of indices (s_0, s_1) such that the line $\mathcal{L}(p_0[s_0], p_1[s_1])$ is a separating common tangent with P_k contained in $\text{RHP}(p_{1-k}[s_{1-k}], p_k[s_k])$ for $k = 0, 1$. If the tangent does not exist, the algorithm returns NULL. The other separating common tangent can be found by a similar algorithm if the corners of the polygons are given in clockwise order and ‘= 1’ is changed to ‘= -1’ in lines 3 and 10.

The algorithm traverses the polygons in parallel one corner at a time using the indices t_0 and t_1 . We say that the indices (s_0, s_1) define a *temporary line*, which is the line $\mathcal{L}(p_0[s_0], p_1[s_1])$. We update the indices s_0 and s_1 until the temporary line is the separating common tangent. At the beginning of an iteration of the loop at line 2, we traverse one corner $p_u[t_u]$ of P_u , $u = 0, 1$. If the corner happens to be on the wrong side of the intermediate line, we make the temporary line pass through that corner by updating s_u to t_u and we reset t_{1-u} to $s_{1-u} + 1$. The reason for resetting t_{1-u} is that a corner of P_{1-u} which was on the correct side of the old temporary line can be on the wrong side of the new line and thus needs to be traversed again.

We show that if the temporary line is not a separating common tangent after each polygon has been traversed twice by the loop beginning at line 2, then the convex hulls of the polygons are not disjoint. Therefore, if a corner is found to be on the wrong side of the line defined by



■ **Figure 2** Algorithm 1 running on two polygons P_0 and P_1 . The corners $p_k[s_k^{(i)}]$ are marked and labeled as $s_k^{(i)}$ for the initial values $s_k^{(0)}$ and after each iteration i where an update of s_k happens. The segments $p_0[s_0^{(i)}]p_1[s_1^{(i)}]$ on the temporary line are dashed.

Algorithm 1: SeparatingCommonTangent(P_0, P_1)

```

1  $s_0 \leftarrow 0$ ;  $t_0 \leftarrow 1$ ;  $s_1 \leftarrow 0$ ;  $t_1 \leftarrow 1$ ;  $u \leftarrow 0$ 
2 while  $t_0 < 2n_0$  or  $t_1 < 2n_1$ 
3   if  $\mathcal{T}(p_{1-u}[s_{1-u}], p_u[s_u], p_u[t_u]) = 1$ 
4      $s_u \leftarrow t_u$ 
5      $t_{1-u} \leftarrow s_{1-u} + 1$ 
6    $t_u \leftarrow t_u + 1$ 
7    $u \leftarrow 1 - u$ 
8 for each  $u \leftarrow \{0, 1\}$ 
9   for each  $t \leftarrow \{0, \dots, n_u - 1\}$ 
10    if  $\mathcal{T}(p_{1-u}[s_{1-u}], p_u[s_u], p_u[t]) = 1$ 
11      return NULL
12 return  $(s_0, s_1)$ 

```

(s_0, s_1) in the loop beginning at line 8, no separating common tangent can exist and NULL is returned. Let $s_k^{(i)}$ be the value of s_k after $i = 0, 1, \dots$ iterations of the loop at line 2. We always have $s_k^{(0)} = 0$ due to the initialization of s_k . See Figure 2.

Assume that s_0 is updated in line 4 in iteration i . The point $p_0[s_0^{(i)}]$ is in the half-plane LHP($p_1[s_1^{(i-1)}], p_0[s_0^{(i-1)}]$), but not on the line $\mathcal{L}(p_1[s_1^{(i-1)}], p_0[s_0^{(i-1)}])$. Therefore, we have the following observation.

► **Observation 3.** *When s_k is updated, the temporary line is rotated counterclockwise around s_{1-k} by an angle less than π .*

Assume in the following that the convex hulls of P_0 and P_1 are disjoint so that separating common tangents exist. Let (r_0, r_1) be the indices that define the separating common tangent such that P_k is contained in RHP($p_{1-k}[r_{1-k}], p_k[r_k]$), i.e., (r_0, r_1) is the result we are going to prove that the algorithm returns.

Since $\mathcal{H}(P_k)$ is convex, the temporary line always divides $\mathcal{H}(P_k)$ into two convex parts. If we follow the temporary line from $p_{1-k}[s_{1-k}]$ in the direction towards $p_k[s_k]$, we enter

$\mathcal{H}(P_k)$ at some point x and thereafter leave $\mathcal{H}(P_k)$ again at some point y . We clearly have $x = y$ if and only if the temporary line is a tangent to $\mathcal{H}(P_k)$, since if $x = y$ and the line was no tangent, $\mathcal{H}(P_k)$ would only be a line segment. The part of the boundary of $\mathcal{H}(P_k)$ counterclockwise from x to y is in $\text{RHP}(p_{1-k}[s_{1-k}], p_k[s_k])$ whereas the part from y to x is on $\text{LHP}(p_{1-k}[s_{1-k}], p_k[s_k])$. We therefore have the following observation.

► **Observation 4.** *Let d be the index of the corner of $\mathcal{H}(P_k)$ strictly after y in counterclockwise order. There exists a corner $p_k[t]$ of P_k such that $\mathcal{T}(p_{1-k}[s_{1-k}], p_k[s_k], p_k[t]) = 1$ if and only if $\mathcal{T}(p_{1-k}[s_{1-k}], p_k[s_k], p_k[d]) = 1$.*

Let c_k be the index of the first corner of $\mathcal{H}(P_k)$ when following $\mathcal{H}(P_k)$ in counterclockwise order from y , $c_k = 0, \dots, n_k - 1$. If y is itself a corner of $\mathcal{H}(P_k)$, we have $p_k[c_k] = y$. By observation 4 we see that $\mathcal{T}(p_{1-k}[s_{1-k}], p_k[s_k], p_k[c_k]) \geq 0$ with equality if and only if $p_k[c_k] = p_k[s_k] = y$. Let $c_k^{(0)}$ be c_k when only line 1 has been executed. Consider now the value of c_k after $i = 1, 2, \dots$ iterations of the loop at line 2. Let $c_k^{(i)} = c_k$ and add n_k to $c_k^{(i)}$ until $c_k^{(i)} \geq c_k^{(i-1)}$. This gives a non-decreasing sequence of indices $c_k^{(0)}, c_k^{(1)}, \dots$ of the first corner of $\mathcal{H}(P_k)$ in $\text{LHP}(p_{1-k}[s_{1-k}], p_k[s_k])$. Actually, we prove in the following that we need to add n_k to $c_k^{(i)}$ at most once before $c_k^{(i)} \geq c_k^{(i-1)}$. If $r_k < c_k^{(0)}$ we add n_k to r_k . Thus we have $0 = s_k^{(0)} \leq c_k^{(0)} \leq r_k < 2n_k$.

The following lemma intuitively says that the algorithm does not “jump over” the correct solution and it expresses the main idea in our proof of correctness.

► **Lemma 5.** *After each iteration $i = 0, 1, \dots$ and for each $k = 0, 1$ we have*

$$0 \leq s_k^{(i)} \leq c_k^{(i)} \leq r_k < 2n_k.$$

Proof. We prove the lemma for $k = 0$. From the definition of r_0 , we get that $0 = s_0^{(0)} \leq c_0^{(0)} \leq r_0 < 2n_0$. Since the sequence $s_0^{(0)}, s_0^{(1)}, \dots$ is non-decreasing, the inequality $0 \leq s_k^{(i)}$ is true for every i .

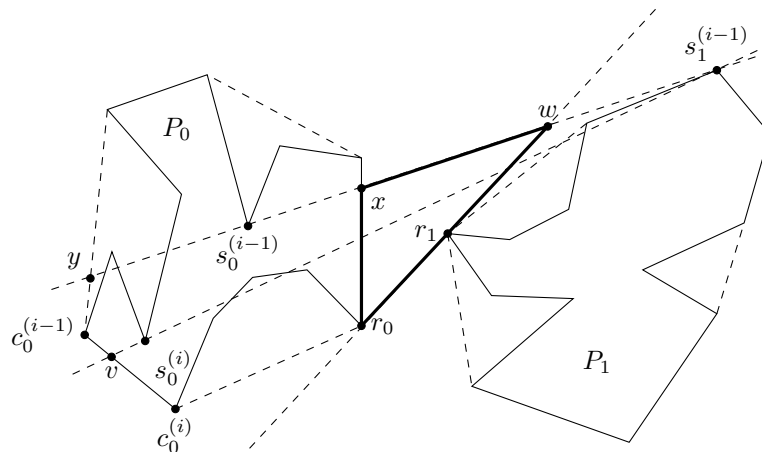
Now, assume inductively that $s_0^{(i-1)} \leq c_0^{(i-1)} \leq r_0$ and consider what happens during iteration i . If neither s_0 nor s_1 is updated, the statement is trivially true from the induction hypothesis, so assume that an update happens.

By the *old temporary line* we mean the temporary line defined by $(s_0^{(i-1)}, s_1^{(i-1)})$ and the *new temporary line* is the one defined by $(s_0^{(i)}, s_1^{(i)})$. The old temporary line enters $\mathcal{H}(P_0)$ at some point x and exits at some point y when followed from $p_1[s_1^{(i-1)}]$. Likewise, let v be the point where the new temporary line exits $\mathcal{H}(P_0)$ when followed from $p_1[s_1^{(i)}]$. The point x exists since the convex hulls are disjoint.

Assume first that the variable u in the algorithm is 0, i.e., a corner of the polygon P_0 is traversed. In this case $s_1^{(i-1)} = s_1^{(i)}$.

We now prove $s_0^{(i)} \leq c_0^{(i)}$. Assume that $p_0[s_0^{(i-1)}] \neq p_0[c_0^{(i-1)}]$. The situation is depicted in Figure 3. In this case $\mathcal{T}(p_1[s_1^{(i-1)}], p_0[s_0^{(i-1)}], p_0[c_0^{(i-1)}]) = 1$. Hence, the update happens when $p_0[c_0^{(i-1)}]$ is traversed or earlier, so $s_0^{(i)} \leq c_0^{(i-1)} \leq c_0^{(i)}$. Assume now that $p_0[s_0^{(i-1)}] = p_0[c_0^{(i-1)}]$. We cannot have $c_0^{(i)} = c_0^{(i-1)}$ since $\mathcal{T}(p_1[s_1^{(i)}], p_0[s_0^{(i)}], p_0[c_0^{(i-1)}]) = -\mathcal{T}(p_1[s_1^{(i-1)}], p_0[s_0^{(i-1)}], p_0[s_0^{(i)}]) = -1$, therefore $c_0^{(i)} > c_0^{(i-1)}$. Consider the corner $p_0[c']$ on $\mathcal{H}(P_0)$ following $p_0[c_0^{(i-1)}]$ in counterclockwise order, $c' > c_0^{(i-1)}$. Due to the minimality of c' , we have $c' \leq c_0^{(i)}$. By Observation 4, $\mathcal{T}(p_1[s_1^{(i-1)}], p_0[s_0^{(i-1)}], p_0[c']) = 1$. Therefore, s_0 must be updated when $p_0[c']$ is traversed or earlier, so $s_0^{(i)} \leq c' \leq c_0^{(i)}$.

For the inequality $c_0^{(i)} \leq r_0$, consider the new temporary line in the direction from $p_1[s_1^{(i-1)}]$ to $p_0[s_0^{(i)}]$. We prove that v is in the part of $\mathcal{H}(P_0)$ from y counterclockwise to r_0 . The point $p_0[s_0^{(i)}]$ is in the polygon Q defined by the segment xy together with the part of



■ **Figure 3** An update of s_0 happens in iteration i from $s_0^{(i-1)}$ to $s_0^{(i)}$ and $p_0[c_0]$ moves forward on $\mathcal{H}(P_0)$ from $p_0[c_0^{(i-1)}]$ to $p_0[c_0^{(i)}]$. The relevant corners are marked and labeled with their indices. The polygon \mathcal{C} from the proof of Lemma 5 is drawn with thick lines.

$\mathcal{H}(P_0)$ from y counterclockwise to x . Therefore, the new temporary line enters and exits Q . It cannot exit through the segment xy , since the old and new temporary lines intersect at $p_1[s_1^{(i-1)}]$, which is in $\mathcal{H}(P_1)$. Therefore, v must be on the part of $\mathcal{H}(P_0)$ from y to x . If r_0 is on the part of $\mathcal{H}(P_0)$ from x counterclockwise to y , then v is on the part from y to r_0 as we wanted.

Otherwise, assume for contradiction that the points appear in the order $y, p_0[r_0], v, x$ counterclockwise along $\mathcal{H}(P_0)$, where $p_0[r_0] \neq v \neq x$. The endpoints of the segment $p_1[s_1^{(i-1)}]x$ are on different sides of the tangent defined by (r_0, r_1) , so the segment intersects the tangent at a point w . The part of $\mathcal{H}(P_0)$ from $p_0[r_0]$ to x and the segments xw and $wp_0[r_0]$ form a simple polygon \mathcal{C} , see Figure 3 for an example. The new temporary line enters \mathcal{C} at the point v , so it must leave \mathcal{C} after v . The line cannot cross $\mathcal{H}(P_0)$ after v since $\mathcal{H}(P_0)$ is convex. It also cannot cross the segment xw at a point after v since the old and the new temporary line cross before v , namely at $p_1[s_1^{(i-1)}]$. The tangent defined by (r_0, r_1) and the new temporary line intersect before v since the endpoints of the segment $p_1[s_1^{(i-1)}]v$ are on different sides of the tangent. Therefore, the line cannot cross the segment $wp_0[r_0]$ at a point after v . Hence, the line cannot exit \mathcal{C} . That is a contradiction.

Therefore, v is on the part of $\mathcal{H}(P_0)$ from y to $p_0[r_0]$ and hence the first corner $p_0[c_0^{(i)}]$ of $\mathcal{H}(P_0)$ after v must be before or coincident with $p_0[r_0]$, so that $c_0^{(i)} \leq r_0$.

Assume now that $u = 1$ in the beginning of iteration i , i.e., a corner of the other polygon P_1 is traversed. In that case, we have $s_0^{(i)} = s_0^{(i-1)} \leq c_0^{(i-1)} \leq c_0^{(i)}$, and we need only prove $c_0^{(i)} \leq r_0$. Observation 3 gives that v is in the part of $\mathcal{H}(P_0)$ from y to x , since the new temporary line is obtained by rotating the old temporary line counterclockwise around $p_0[s_0^{(i-1)}]$ by an angle less than π . That v appears before $p_0[r_0]$ on $\mathcal{H}(P_0)$ counterclockwise from y follows from exactly the same arguments as in the case $u = 0$. This completes the proof. ◀

► **Lemma 6.** *If the temporary line is different from the tangent defined by (r_0, r_1) , then $\mathcal{T}(p_0[s_0], p_1[s_1], p_1[r_1]) = 1$ or $\mathcal{T}(p_1[s_1], p_0[s_0], p_0[r_0]) = 1$.*

Proof. Assume not. There are points of the temporary line on each side of the tangent because it is separating, so the temporary line and the tangent cross each other in a point a .

The point a is on the segment $p_0[r_0]p_1[r_1]$, since otherwise $p_0[r_0]$ and $p_1[r_1]$ would be on the same side of the temporary line, so $\mathcal{T}(p_0[s_0], p_1[s_1], p_1[r_1]) = 1$ or $\mathcal{T}(p_1[s_1], p_0[s_0], p_0[r_0]) = 1$. Choose a point d_R on the temporary line in $\text{RHP}(p_0[r_0], p_1[r_1])$ which is so far away from a that all intersections between the line and the polygons are on the same side of d_R as a . Choose d_L in a similar way in $\text{LHP}(p_0[r_0], p_1[r_1])$. We have $-1 = \mathcal{T}(p_0[r_0], p_1[r_1], d_R) = \mathcal{T}(p_0[r_0], a, d_R) = -\mathcal{T}(d_R, a, p_0[r_0])$, so the supports must appear in the order s_0, s_1 when traveling along the temporary line from d_R towards a for $\mathcal{T}(p_1[s_1], p_0[s_0], p_0[r_0]) \leq 0$ to hold.

We also have that $p_0[s_0]$ is on the segment ad_L since $p_0[s_0] \in \text{LHP}(p_0[r_0], p_1[r_1])$ and $p_1[s_1]$ is on the segment ad_R since $p_1[s_1] \in \text{RHP}(p_0[r_0], p_1[r_1])$. Hence, the order of the supports from d_R towards a is s_1, s_0 . That is a contradiction. \blacktriangleleft

We are now ready to prove that Algorithm 1 has the desired properties.

► **Theorem 7.** *If the polygons P_0 and P_1 have separating common tangents, Algorithm 1 returns a pair of indices (s_0, s_1) defining a separating common tangent such that P_k is contained in $\text{RHP}(p_{1-k}[s_{1-k}], p_k[s_k])$ for $k = 0, 1$. If no separating common tangents exist, the algorithm returns NULL. The algorithm runs in linear time and uses constant workspace.*

Proof. Assume first that separating common tangents do not exist. Then the test in line 10 makes the algorithm return NULL due to some corner $p_u[t]$ on the wrong side of the temporary line.

Assume now that separating common tangents do exist and that the temporary line is not the desired tangent. Without loss of generality, we may assume that $\mathcal{T}(p_1[s_1], p_0[s_0], p_0[r_0]) = 1$ by Lemma 6. Lemma 5 gives that $p_0[r_0]$ will be traversed if no other update of s_0 or s_1 happens. Therefore, an update happens before the loop at line 2 finishes. We conclude that when the loop finishes, the pair (s_0, s_1) defines the separating common tangent as stated.

When an update happens in iteration i of the loop at line 2, the sum $s_0 + s_1$ is increased by a value which is at least $\frac{i-j}{2}$, where $j \geq 0$ was the previous iteration where an update happened. Inductively, we see that the number of iterations is always at most $2(s_0 + s_1) + t_0 - s_0 + t_1 - s_1 \leq 2(t_0 + t_1) \leq 4(n_0 + n_1)$. \blacktriangleleft

3 Computing outer common tangents

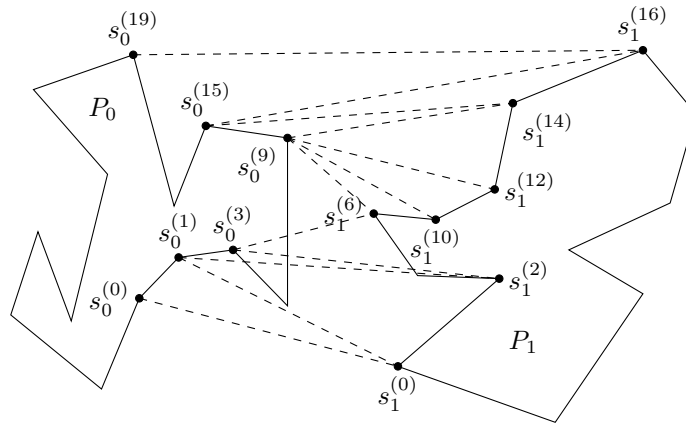
In this section, we assume that two polygons P_0 and P_1 are given such that their convex hulls are disjoint. We assume that the corners $p_0[0], \dots, p_0[n_0 - 1]$ of P_0 are given in counterclockwise order and the corners $p_1[0], \dots, p_1[n_1 - 1]$ of P_1 are given in clockwise order. We say that the *orientation* of P_0 and P_1 is counterclockwise and clockwise, respectively. We prove that Algorithm 2 returns two indices (s_0, s_1) that define an outer common tangent such that P_0 and P_1 are both contained in $\text{RHP}(p_0[s_0], p_1[s_1])$.

As in the case of separating common tangents, we define $s_k^{(i)}$ as the value of s_k after $i = 0, 1, \dots$ iterations of the loop at line 2 of Algorithm 2. See Figure 4. For this algorithm, we get a slightly different analogue to Observation 3:

► **Observation 8.** *When s_k is updated, the temporary line is rotated around s_{1-k} in the orientation of P_{1-k} by an angle less than π .*

Let y be the point where the temporary line enters $\mathcal{H}(P_k)$ when followed from $p_{1-k}[s_{1-k}]$ and x the point where it exits $\mathcal{H}(P_k)$. We have the following analogue of Observation 4.

► **Observation 9.** *Let d be the index of the corner of $\mathcal{H}(P_k)$ strictly after y following the orientation of P_k . There exists a corner $p_k[t]$ of P_k such that $\mathcal{T}(p_0[s_0], p_1[s_1], p_k[t]) = 1$ if and only if $\mathcal{T}(p_0[s_0], p_1[s_1], p_k[d]) = 1$.*



■ **Figure 4** Algorithm 2 running on two polygons P_0 and P_1 . The corners $p_k[s_k^{(i)}]$ are marked and labeled as $s_k^{(i)}$ for the initial values $s_k^{(0)}$ and after each iteration i where an update of s_k happens. The segments $p_0[s_0^{(i)}]p_1[s_1^{(i)}]$ on the temporary line are dashed.

Algorithm 2: OuterCommonTangent(P_0, P_1)

```

1  $s_0 \leftarrow 0; t_0 \leftarrow 1; s_1 \leftarrow 0; t_1 \leftarrow 1; u \leftarrow 0$ 
2 while  $t_0 < 2n_0$  or  $t_1 < 2n_1$ 
3   if  $\mathcal{T}(p_0[s_0], p_1[s_1], p_u[t_u]) = 1$ 
4      $s_u \leftarrow t_u$ 
5      $t_{1-u} \leftarrow s_{1-u} + 1$ 
6    $t_u \leftarrow t_u + 1$ 
7    $u \leftarrow 1 - u$ 
8 return  $(s_0, s_1)$ 
```

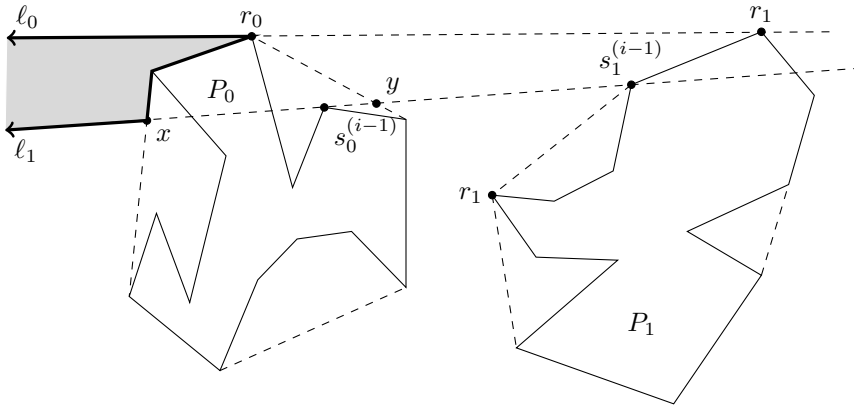
Let c_k be the index of the first corner of $\mathcal{H}(P_k)$ after y following the orientation of P_k , where $p_k[c_k] = y$ if y is itself a corner of $\mathcal{H}(P_k)$. By Observation 9, we have $\mathcal{T}(p_0[s_0], p_1[s_1], p_k[c_k]) \geq 0$ with equality if and only if $p_k[c_k] = p_k[s_k] = y$. Define a non-decreasing sequence $c_k^{(0)}, c_k^{(1)}, \dots$ of the value of c_k after $i = 0, 1, \dots$ iterations as we did for separating tangents. Also, let the indices (r_0, r_1) define the outer common tangent that we want the algorithm to return such that $c_k^{(0)} \leq r_k < 2n_k$. We can now state the analogue to Lemma 5 for outer common tangents.

► **Lemma 10.** *After each iteration $i = 0, 1, \dots$ and for each $k = 0, 1$ we have*

$$0 \leq s_k^{(i)} \leq c_k^{(i)} \leq r_k < 2n_k.$$

Proof. Assume $k = 0$ and the induction hypothesis $s_0^{(i-1)} \leq c_0^{(i-1)} \leq r_0$. The inequality $s_0^{(i)} \leq c_0^{(i)}$ can be proven exactly as in the proof of lemma 5. Therefore, consider the inequality $c_0^{(i)} \leq r_0$ and assume that an update happens in iteration i .

Let the *old temporary line* and the *new temporary line* be the lines defined by the indices $(s_0^{(i-1)}, s_1^{(i-1)})$ and $(s_0^{(i)}, s_1^{(i)})$, respectively. Let y and x be the points where the old temporary line enters and exits $\mathcal{H}(P_0)$ followed from $p_1[s_1^{(i-1)}]$, respectively, and let v be the point where the new temporary line enters $\mathcal{H}(P_0)$. The points y and v exist since the convex hulls of P_0 and P_1 are disjoint.



■ **Figure 5** The area \mathcal{A} from the proof of Lemma 10 in grey. The relevant corners are marked and labeled with their indices.

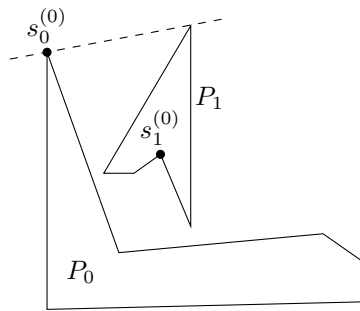
Assume first that the variable u in the algorithm equals 0 when the update happens. We prove that v is in the part of $\mathcal{H}(P_0)$ from y to $p_0[r_0]$ following the orientation of P_0 , which is counterclockwise. The point $p_0[s_0^{(i)}]$ is in the simple polygon Q bounded the part of $\mathcal{H}(P_0)$ from y counterclockwise to x and the segment xy . Therefore, the new temporary line must enter Q to get to $p_0[s_0^{(i)}]$. It cannot enter through xy , since the old and new temporary line cross at $p_1[s_1^{(i-1)}]$ which is not in $\mathcal{H}(P_k)$ by assumption. Therefore, it must enter through the part of $\mathcal{H}(P_0)$ from y to x , so v is in this part. If r_0 is not in the part of $\mathcal{H}(P_0)$ from y to x , it is clearly true that v is in the part from y to $p_0[r_0]$. Otherwise, assume for contradiction that the points appear on $\mathcal{H}(P_0)$ in the order $y, p_0[r_0], v, x$ and $p_0[r_0] \neq v \neq x$. Let ℓ_0 be the half-line starting at $p_0[r_0]$ following the tangent away from $p_1[r_1]$, and let ℓ_1 be the half-line starting at x following the old temporary line away from $p_1[s_1^{(i-1)}]$. The part of $\mathcal{H}(P_0)$ from $p_0[r_0]$ to x and the half-lines ℓ_0 and ℓ_1 define a possibly unbounded area \mathcal{A} outside $\mathcal{H}(P_0)$, see Figure 5. We follow the new temporary line from $p_1[s_1^{(i-1)}]$ towards v . The point $p_1[s_1^{(i-1)}]$ is not in \mathcal{A} and the new temporary line exits \mathcal{A} at v since it enters $\mathcal{H}(P_0)$ at v , so it must enter \mathcal{A} somewhere at a point on the segment $p_1[s_1^{(i-1)}]v$. It cannot enter through $\mathcal{H}(P_0)$ since $\mathcal{H}(P_0)$ is convex. It cannot enter through ℓ_0 since v and $p_1[s_1^{(i-1)}]$ are on the same side of the outer common tangent. It cannot enter through ℓ_1 since the old and new temporary line intersect in $p_1[s_1^{(i-1)}]$, which is not in \mathcal{A} . That is a contradiction, so v is on the part of $\mathcal{H}(P_0)$ from y to $p_0[r_0]$. Hence, the first corner after y is coincident with or before $p_0[r_1]$, i.e., $c_0^{(i)} \leq r_0$.

Assume now that $u = 1$ in the beginning of iteration i so that a corner of the polygon P_1 is traversed. Observation 8 gives that v is on the part of $\mathcal{H}(P_0)$ from y counterclockwise to x . It follows that v appears before $p_0[r_0]$ on $\mathcal{H}(P_0)$ counterclockwise from y from exactly the same arguments as in the case $u = 0$. ◀

We have the following equivalent of Lemma 6 which, however, has a different proof.

► **Lemma 11.** *If the temporary line is different from the tangent defined by (r_0, r_1) , then $\mathcal{T}(p_0[s_0], p_1[s_1], p_0[r_0]) = 1$ or $\mathcal{T}(p_0[s_0], p_1[s_1], p_1[r_1]) = 1$.*

Proof. Assume not. The points $p_0[s_0]$ and $p_1[s_1]$ are both in $\text{RHP}(p_0[r_0], p_1[r_1])$. Therefore, the temporary line cannot be parallel with the tangent, since in that case we would have $\mathcal{T}(p_0[s_0], p_1[s_1], p_0[r_0]) = 1$. Let a be the intersection point between the tangent and the temporary line. The point a cannot be in the interior of the segment $p_0[r_0]p_1[r_1]$,



■ **Figure 6** Two polygons P_0 and P_1 where Algorithm 2 does not work for the initial values of s_0 and s_1 as shown. The correct tangent is drawn as a dashed line.

since in that case, $p_0[r_0]$ and $p_1[r_1]$ would be on different sides of the temporary line, so $\mathcal{T}(p_0[s_0], p_1[s_1], p_0[r_0]) = 1$ or $\mathcal{T}(p_0[s_0], p_1[s_1], p_1[r_1]) = 1$. Assume without loss of generality that a is on the half-line from $p_0[r_0]$ going away from $p_1[r_1]$. Also assume that $p_0[s_0] \neq a$, since otherwise $p_0[s_0] = a = p_0[r_0]$ and $-1 = \mathcal{T}(p_0[r_0], p_1[r_1], p_1[s_1]) = -\mathcal{T}(p_0[s_0], p_1[s_1], p_1[r_1])$. Now, $1 = \mathcal{T}(p_1[r_1], p_0[r_0], p_0[s_0]) = \mathcal{T}(p_1[r_1], a, p_0[s_0]) = -\mathcal{T}(p_0[s_0], a, p_1[r_1])$. This forces $p_1[s_1]$ to be on the segment $p_0[s_0]a$.

From a , the orders of the points are $p_1[s_1], p_0[s_0]$ and $p_0[r_0], p_1[r_1]$ along the temporary line and the tangent, respectively. The points $ap_1[s_1]p_0[r_0]$ form a triangle Δ_0 and $ap_0[s_0]p_1[r_1]$ form a larger triangle Δ_1 containing Δ_0 . The part $\Delta_1 \setminus \Delta_0$ of Δ_1 not in Δ_0 is therefore a quadrilateral $p_0[s_0]p_1[s_1]p_0[r_0]p_1[r_1]$ with all inner angles less than π , so the diagonals $p_0[s_0]p_0[r_0]$ and $p_1[s_1]p_1[r_1]$ cross each other. Hence, the convex hulls of P_0 and P_1 are not disjoint. ◀

We can now prove the stated properties of Algorithm 2 in much the same way as the proof of Theorem 7.

▶ **Theorem 12.** *If the polygons P_0 and P_1 have disjoint convex hulls, Algorithm 2 returns a pair of indices (s_0, s_1) defining an outer common tangent such that P_0 and P_1 are contained in $RHP(s_0, s_1)$. The algorithm runs in linear time and uses constant workspace.*

4 Concluding Remarks

We have described an algorithm for computing the separating common tangents of two simple polygons in linear time using constant workspace. We have also described an algorithm for computing outer common tangents using linear time and constant workspace when the convex hulls of the polygons are disjoint. Figure 6 shows an example where Algorithm 2 does not work when applied to two disjoint polygons with overlapping convex hulls. In fact, if there was no bound on the values t_0 and t_1 in the loop at line 2, the algorithm would update s_0 and s_1 infinitely often and never find the correct tangent. An obvious improvement is to find an equally fast and space efficient algorithm which does not require the convex hulls to be disjoint. An algorithm for computing an outer common tangent of two polygons, when such one exists, also decides if one convex hull is completely contained in the other. Together with the algorithm for separating common tangents presented in Section 2, we would have an optimal algorithm for deciding the complete relationship between the convex hulls: if one is contained in the other, and if not, whether they are disjoint or not. However, keeping in mind that it is harder to compute an outer common tangent of intersecting convex polygons

than of disjoint ones [7], it would not be surprising if it was also harder to compute an outer common tangent of general simple polygons than simple polygons with disjoint convex hulls when only constant workspace is available.

References

- 1 M. Abrahamsen. An optimal algorithm computing edge-to-edge visibility in a simple polygon. In *Proceedings of the 25th Canadian Conference on Computational Geometry, CCCG*, pages 157–162, 2013.
- 2 T. Asano, K. Buchin, M. Buchin, M. Korman, W. Mulzer, G. Rote, and A. Schulz. Memory-constrained algorithms for simple polygons. *Computational Geometry: Theory and Applications*, 46(8):959–969, 2013.
- 3 T. Asano, W. Mulzer, G. Rote, and Y. Wang. Constant-work-space algorithms for geometric problems. *Journal of Computational Geometry*, 2(1):46–68, 2011.
- 4 L. Barba, M. Korman, S. Langerman, and R.I. Silveira. Computing the visibility polygon using few variables. In *Proceedings of the 22nd International Symposium on Algorithms and Computation, ISAAC*, volume 7014 of *Lecture Notes in Computer Science*, pages 70–79. Springer, 2011.
- 5 G.S. Brodal and R. Jacob. Dynamic planar convex hull. In *Proceedings of the 43rd annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 617–626, 2002.
- 6 R.L. Graham and F.F. Yao. Finding the convex hull of a simple polygon. *Journal of Algorithms*, 4(4):324–331, 1983.
- 7 Leonidas Guibas, John Hershberger, and Jack Snoeyink. Compact interval trees: A data structure for convex hulls. *International Journal of Computational Geometry & Applications*, 1(1):1–22, 1991.
- 8 J. Hershberger and S. Suri. Applications of a semi-dynamic convex hull algorithm. *BIT Numerical Mathematics*, 32(2):249–267, 1992.
- 9 D. Kirkpatrick and J. Snoeyink. Computing common tangents without a separating line. In *Proceedings of the 4th International Workshop on Algorithms and Data Structures, WADS*, volume 955 of *Lecture Notes in Computer Science*, pages 183–193. Springer, 1995.
- 10 A.A. Melkman. On-line construction of the convex hull of a simple polyline. *Information Processing Letters*, 25(1):11–12, 1987.
- 11 M.H. Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, 23(2):166–204, 1981.
- 12 F.P. Preparata and S.J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, 1977.
- 13 G.T. Toussaint. Solving geometric problems with the rotating calipers. In *Proceedings of the IEEE Mediterranean Electrotechnical Conference, MELECON*, pages A10.02/1–4, 1983.

A Linear-Time Algorithm for the Geodesic Center of a Simple Polygon

Hee Kap Ahn^{*3}, Luis Barba^{1,2}, Prosenjit Bose¹,
Jean-Lou De Carufel¹, Matias Korman^{4,5}, and Eunjin Oh³

- 1 School of Computer Science, Carleton University, Ottawa, Canada
jit@scs.carleton.ca, jdecaruf@cg.scs.carleton.ca
- 2 Département d'Informatique, Université Libre de Bruxelles, Brussels, Belgium
lbarbaf1@ulb.ac.be
- 3 Department of Computer Science and Engineering, POSTECH,
77 Cheongam-Ro, Nam-Gu, Pohang, Gyeongbuk, Korea
{heekap, jin9082}@postech.ac.kr
- 4 National Institute of Informatics (NII), Tokyo, Japan
korman@nii.ac.jp
- 5 JST, ERATO, Kawarabayashi Large Graph Project

Abstract

Let P be a closed simple polygon with n vertices. For any two points in P , the geodesic distance between them is the length of the shortest path that connects them among all paths contained in P . The geodesic center of P is the unique point in P that minimizes the largest geodesic distance to all other points of P . In 1989, Pollack, Sharir and Rote [Disc. & Comput. Geom. 89] showed an $O(n \log n)$ -time algorithm that computes the geodesic center of P . Since then, a longstanding question has been whether this running time can be improved (explicitly posed by Mitchell [Handbook of Computational Geometry, 2000]). In this paper we affirmatively answer this question and present a linear time algorithm to solve this problem.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Geodesic distance, facility location, 1-center problem, simple polygons

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.209

1 Introduction

Let P be a simple polygon with n vertices. Given two points x, y in P , the *geodesic path* $\pi(x, y)$ is the shortest path contained in P connecting x with y . If the straight-line segment connecting x with y is contained in P , then $\pi(x, y)$ is a straight-line segment. Otherwise, $\pi(x, y)$ is a polygonal chain whose vertices (other than its endpoints) are reflex vertices of P . We refer the reader to [20] for more information on geodesic paths.

The *geodesic distance* between x and y , denoted by $|\pi(x, y)|$, is the sum of the Euclidean lengths of each segment in $\pi(x, y)$. Throughout this paper, when referring to the distance between two points in P , we mean the geodesic distance between them. To ease the description, we assume that each vertex of P has a unique farthest neighbor. This *general position* condition was also assumed by Aronov et al. [2] and can be obtained by applying a slight perturbation to the positions of the vertices [10].

* The work by H.-K. Ahn and E. Oh was supported by the NRF grant 2011-0030044 (SRC-GAIA) funded by the Korea government (MSIP).



Given a point $x \in P$, a (geodesic) *farthest neighbor* of x , is a point $f_P(x)$ (or simply $f(x)$) of P whose geodesic distance to x is maximized.

Let $F_P(x)$ be the function that maps each $x \in P$ to the distance to a farthest neighbor of x (i.e., $F_P(x) = |\pi(x, f(x))|$). A point $c_P \in P$ that minimizes $F_P(x)$ is called the *geodesic center* of P . Similarly, a point $s \in P$ that maximizes $F_P(x)$ (together with $f(s)$) is called a *geodesic diametral pair* and their distance is known as the *geodesic diameter*. Asano and Toussaint [3] showed that the geodesic center is unique (whereas it is easy to see that several geodesic diametral pairs may exist).

In this paper, we show how to compute the geodesic center of P in $O(n)$ time. Due to lack of space, some proofs are omitted. For a full version of this paper refer to [1].

1.1 Previous Work

Since the early 1980s the problem of computing the geodesic center (and its counterpart, the geodesic diameter) has received a lot of attention from the computational geometry community. Chazelle [7] gave the first algorithm for computing the geodesic diameter (which runs in $O(n^2)$ time using linear space). Afterwards, Suri [25] reduced it to $O(n \log n)$ -time without increasing the space constraints. Finally, Hershberger and Suri [14] presented a fast matrix search technique, one application of which is a linear-time algorithm for computing the diameter. The first algorithm for computing the geodesic center was given by Asano and Toussaint [3], and runs in $O(n^4 \log n)$ -time. In 1989, Pollack, Sharir, and Rote [23] improved it to $O(n \log n)$ time. Since then, it has been an open problem whether the geodesic center can be computed in linear time (indeed, this problem was explicitly posed by Pollack et al. [23] and later by Mitchell [20, Chapter 27]).

Several variations of these two problems have been considered. Indeed, the same problem has been studied under different metrics. For example, the L_1 geodesic distance [6], the link distance [9, 15, 24] (where we look for the path with the minimum possible number of bends or *links*), or even rectilinear link distance [21, 22] (a variation of the link distance in which only isothetic segments are allowed). The diameter and center of a simple polygon for both the L_1 and rectilinear link metrics can be computed in linear time (whereas $O(n \log n)$ time is needed for the link distance). Another natural extension is the computation of the diameter and center in polygonal domains (i.e., polygons with one or more holes). Polynomial time algorithms are known for both the diameter [4] and center [5], although the running times are significantly larger (i.e., $O(n^{7.73})$ and $O(n^{12+\epsilon})$, respectively).

1.2 Outline

In order to compute the geodesic center, c_P , Pollack et al. [23] introduce a linear time *chord-oracle*. Given a chord C that splits P into two sub-polygons, this oracle determines which sub-polygon contains c_P . Combining this operation with an efficient search on a triangulation of P , Pollack et al. narrow the search of c_P within a triangle (and find the center using optimization techniques). Their approach however, does not allow them to reduce the complexity of the problem in each iteration, and hence it runs in $\Theta(n \log n)$ time.

The general approach of our algorithm described in Section 6 is similar: partition P into $O(1)$ cells, use an oracle to determine which cell contains c_P , and recurse within the cell. Our approach differs however in two important aspects that allows us to speed-up the algorithm. First, we do not use the chords of a triangulation of P to partition the problem into cells. We use instead a cutting of a suitable set of chords. Secondly, we compute a set Σ of $O(n)$ functions, each defined in a triangular domain contained in P , such that their

upper envelope, ϕ , coincides with F_P . Thus, we can “ignore” the polygon P and focus only on finding the minimum of the function ϕ .

The search itself uses ε -nets and cutting techniques, which guarantee that both the size of the cell containing c_P and the number of functions of Σ defined in it decrease by a constant fraction (and thus leads to an overall linear time algorithm). This search has however two stopping conditions, (1) reach a subproblem of constant size, or (2) find a triangle containing c_P . In the latter case, we show that ϕ is a convex function when restricted to this triangle. Thus, finding its minimum becomes an optimization problem that we solve in Section 7 using cuttings in \mathbb{R}^3 . The key of this approach lies in the computation of the functions in Σ and their triangular domains. Each function $g \in \Sigma$ is defined in a triangular domain Δ contained in P and is associated to a particular vertex w of P . Intuitively speaking, g maps points in Δ to their (geodesic) distance to w . We guarantee that, for each point $x \in P$, there is one function $g \in \Sigma$ defined in a triangle containing x , such that $g(x) = F_P(x)$. To compute these triangles and their corresponding functions, we proceed as follows.

In Section 2, we use the matrix search technique introduced by Hershberger and Suri [14] to decompose the boundary of P , denoted by ∂P , into connected edge-disjoint chains. Each chain is defined by either (1) a consecutive list of vertices that have the same farthest neighbor v (we say that v is *marked* if it has such a chain associated to it), or (2) an edge whose endpoints have different farthest neighbors (such edge is called a *transition edge*).

In Section 3, we consider each transition edge ab of ∂P independently and compute its *hourglass*. Intuitively, the hourglass of ab , H_{ab} , is the region of P between two chains, the edge ab and the chain of ∂P that contains the farthest neighbors of all points in ab . Inspired by a result of Suri [25], we show that the sum of the combinatorial complexities of all hourglasses defined on a transition edge is $O(n)$. (The *combinatorial complexity*—or simply complexity—of a geometric object is the total number of vertices and edges that define it.) In addition, we provide a new technique to compute all these hourglasses in linear time.

In Section 5 we show how to compute the functions in Σ and their respective triangles. We distinguish two cases: (1) Inside each hourglass H_{ab} of a transition edge, we use a technique introduced by Aronov et al. [2] that uses the shortest-path trees of a and b in H_{ab} to construct $O(|H_{ab}|)$ triangles with their respective functions (for more information on shortest-path trees refer to [11]). (2) For each marked vertex v we compute triangles that encode the distance from v . Moreover, we guarantee that these triangles cover every point of P whose farthest neighbor is v . Overall, we compute the $O(n)$ functions of Σ in linear time.

2 Decomposing the boundary

In this section, we decompose ∂P into chains of consecutive vertices that share the same farthest neighbor and edges of P whose endpoints have distinct farthest neighbors.

Using a result from Hershberger and Suri [14], in $O(n)$ time we can compute the farthest neighbor of each vertex of P . Recall that the farthest neighbor of each vertex of P is always a convex vertex of P [3] and is unique by our general position assumption. The (farthest) *Voronoi region* of a vertex v of P is the set of points $R(v) = \{x \in P : F_P(x) = |\pi(x, v)|\}$ (including boundary points).

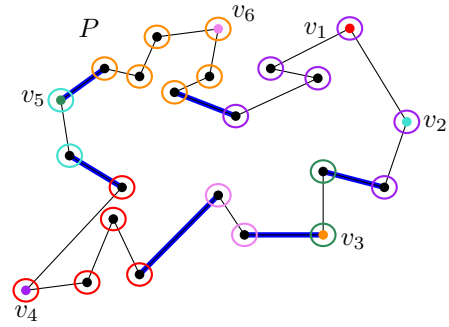
We mark the vertices of P that are farthest neighbors of at least one vertex of P . Let M denote the set of marked vertices of P (clearly this set can be computed in $O(n)$ time after applying the result of Hershberger and Suri). In other words, M contains all vertices of P whose Voronoi region contains at least one vertex of P . Given a vertex v of P , the vertices of P whose farthest neighbor is v appear contiguously along ∂P [2].

Therefore, after computing all these farthest neighbors, we effectively split the boundary into subchains, each associated with a different vertex of M ; see Figure 1.

Given two points x and y on ∂P , let $\partial P(x, y)$ be the polygonal chain that starts at x and follows the boundary of P clockwise until reaching y . We say that three (non-empty) disjoint sets A, B and C contained in ∂P are in *clockwise order* if $B \subset \partial P(a, c)$ for any $a \in A$ and any $c \in C$. (To ease notation, we say that three points $x, y, z \in \partial P$ are in clockwise order if $\{x\}, \{y\}$ and $\{z\}$ are in clockwise order).

Let a and b be the endpoints of a transition edge of ∂P such that b is the clockwise neighbor of a along ∂P . Because ab is a transition edge, we know that $f(a) \neq f(b)$. Recall that we have computed $f(a)$ and $f(b)$ in the previous step and note that $a, b, f(a), f(b)$ are in clockwise order.

For any vertex $v \in \partial P$ such that $f(a) \neq v \neq f(b)$ and $f(a), v, f(b)$ are in clockwise order, we know that there cannot be a vertex u of P such that $f(u) = v$. As proved by Aronov et al. [2, Corollary 2.7.4], if there is a point x on ∂P whose farthest neighbor is v , then x must lie on the open segment (a, b) . In other words, the Voronoi region $R(v)$ restricted to ∂P is contained in (a, b) .



■ **Figure 1** Each vertex of the boundary of P is assigned with a farthest neighbor which is then marked. The boundary is then decomposed into vertex-disjoint chains, each associated with a marked vertex, joined by transition edges (blue) whose endpoints have different farthest neighbors.

3 Hourglasses

For any polygonal chain $C = \partial P(p_0, p_k)$, the *hourglass* of C , denoted by H_C , is the simple polygon contained in P bounded by C , $\pi(p_k, f(p_0))$, $\partial P(f(p_0), f(p_k))$ and $\pi(f(p_k), p_0)$; see Figure 2. We call C and $\partial P(f(p_0), f(p_k))$ the *top* and *bottom* chains of H_C , respectively, while $\pi(p_k, f(p_0))$ and $\pi(f(p_k), p_0)$ are referred to as the *walls* of H_C . We say that the hourglass H_C is *open* if its walls are vertex-disjoint. We say C is a *transition chain* if $f(p_0) \neq f(p_k)$ and neither $f(p_0)$ nor $f(p_k)$ are interior vertices of C . In particular, if an edge ab of ∂P is a transition chain, we say that it is a *transition edge* (see Figure 2).

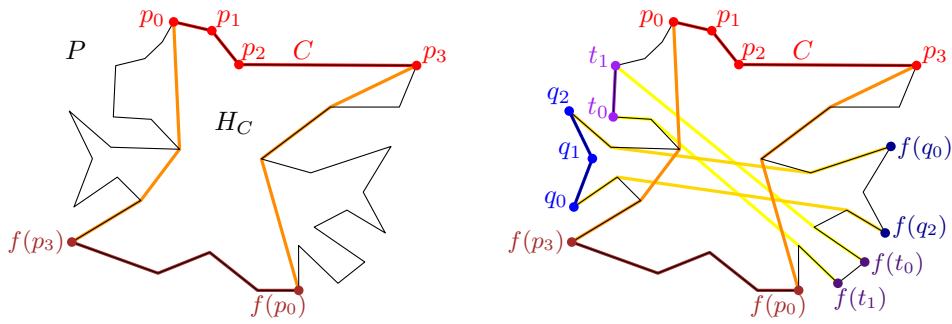
► **Lemma 1** (Restatement of Lemma 3.1.3 of [2]). *If C is a transition chain of ∂P , then the hourglass H_C is an open hourglass.*

In the remainder of the paper, all the hourglasses considered are defined by a transition chain. That is, they are open and their top and bottom chains are edge-disjoint.

The following lemma is depicted in Figure 2 and is a direct consequence of the Ordering Lemma proved by Aronov et al. [2, Corollary 2.7.4].

► **Lemma 2.** *Let C_1, C_2, C_3 be three edge-disjoint transition chains of ∂P in clockwise order. Then, the bottom chains of H_{C_1}, H_{C_2} and H_{C_3} are also edge-disjoint and are in clockwise order.*

Let γ be a geodesic path joining two points on the boundary of P . We say that γ *separates* two points x_1 and x_2 of ∂P if the points of $X = \{x_1, x_2\}$ and the endpoints of γ alternate along the boundary of P (x_1 and x_2 could coincide with the endpoints of γ in degenerate



■ **Figure 2** Given two edge-disjoint transition chains, their hourglasses are open and the bottom chains of their hourglasses are also edge-disjoint. Moreover, these bottom chains appear in the same cyclic order as the top chains along ∂P .

cases). We say that a geodesic path γ separates an hourglass H if it separates the points of its top chain from those of its bottom chain.

► **Lemma 3.** *Let C_1, \dots, C_r be edge-disjoint transition chains of ∂P . Then, there is a set of $t \leq 10$ geodesic paths $\gamma_1, \dots, \gamma_t$ with endpoints on ∂P such that for each $1 \leq i \leq r$ there exists $1 \leq j \leq t$ such that γ_j separates H_{C_i} . Moreover, this set can be computed in $O(n)$ time.*

A chord of P is an edge joining two non-adjacent vertices a and b of P such that $ab \subseteq P$. Therefore, a chord splits P into two sub-polygons.

► **Lemma 4** (Restatement of Lemma 3.4.3 of [2]). *Let C_1, \dots, C_r be a set of edge-disjoint transition chains of ∂P in clockwise order. Then each chord of P appears in $O(1)$ hourglasses among H_{C_1}, \dots, H_{C_r} .*

► **Lemma 5.** *Let x, u, y, v be four vertices of P in clockwise order. Given the shortest-path trees T_x and T_y of x and y in P , respectively, such that T_x and T_y can answer lowest common ancestor (LCA) queries in $O(1)$ time, we can compute the path $\pi(u, v)$ in $O(|\pi(u, v)|)$ time. Moreover, all edges of $\pi(u, v)$, except perhaps one, belong to $T_x \cup T_y$.*

► **Lemma 6.** *Let P be a simple polygon with n vertices. Given k disjoint transition chains C_1, \dots, C_k of ∂P , it holds that*

$$\sum_{i=1}^k |H_{C_i}| = O(n).$$

Proof. Because the given transition chains are edge-disjoint, Lemma 2 implies that the bottom chains of their respective hourglasses are also edge-disjoint. Therefore, the sum of the complexities of all the top and bottom chains of these hourglasses is $O(n)$. To bound the complexity of their walls we use Lemma 4. Since no chord is used more than a constant number of times, it suffices to show that the total number of chords used by all these hourglasses is $O(n)$.

To prove this, we use Lemma 3 to construct $O(1)$ splitting chains $\gamma_1, \dots, \gamma_t$ such that for each $1 \leq i \leq k$, there is a splitting chain γ_j that separates the top and bottom chains of H_{C_i} . For each $1 \leq j \leq t$, let $\mathcal{H}^j = \{H_{C_i} : \text{the top and bottom chain of } H_{C_i} \text{ are separated by } \gamma_j\}$. Since the complexity of the shortest-path trees of the endpoints of γ_j is $O(n)$ [11], and from the fact that the chains C_1, \dots, C_k are edge-disjoint, Lemma 5 implies that the total number

of edges in all the hourglasses of \mathcal{H}^j is $O(n)$. Moreover, because each of these edges appears in $O(1)$ hourglasses among C_1, \dots, C_k , we conclude that

$$\sum_{H \in \mathcal{H}^j} |H| = O(n).$$

Since we have only $O(1)$ splitting chains, our result follows. \blacktriangleleft

3.1 Building hourglasses

Let E be the set of transition edges of ∂P . Given a transition edge $ab \in E$, we say that H_{ab} is a *transition hourglass*. In this section, we present an algorithm that computes each transition hourglass of P in $O(n)$ time.

By Lemma 3 we can compute a set of $O(1)$ separating paths such that for each transition edge ab , the transition hourglass H_{ab} is separated by one (or more) paths in this set. For each endpoint of the $O(1)$ separating paths we compute its shortest-path tree in linear time [8, 11]. In addition, we preprocess these trees in linear time to support LCA queries [13]. Both computations need linear time per endpoint and use $O(n)$ space. Since we do this process for a constant number of endpoints, overall this preprocessing takes $O(n)$ time.

Let γ be a separating path. Note that γ separates the boundary of P into two chains S and S' such that $S \cup S' = \partial P$. Let $\mathcal{H}(\gamma)$ be the set of transition hourglasses separated by γ whose transition edge is contained in S (whenever an hourglass is separated by more than one path, we pick one arbitrarily). Note that we can classify all transition hourglasses into the sets $\mathcal{H}(\gamma)$ in $O(n)$ time (since $O(1)$ separating paths are considered).

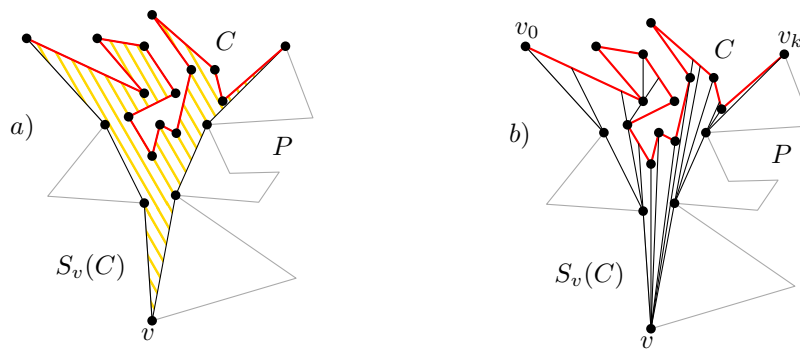
We claim that we can compute all transition hourglasses of $\mathcal{H}(\gamma)$ in $O(n)$ time. By construction, the wall of each of these hourglasses consists of a (geodesic) path that connects a point in S with a point in S' . Let $u \in S$ and $v \in S'$ be two vertices such that $\pi(u, v)$ is the wall of a hourglass in $\mathcal{H}(\gamma)$. Because LCA queries can be answered in $O(1)$ time [13], Lemma 5 allows us to compute this path in $O(|\pi(u, v)|)$ time. Therefore, we can compute all hourglasses of $\mathcal{H}(\gamma)$ in $O(\sum_{H \in \mathcal{H}(\gamma)} |H| + n) = O(n)$ time by Lemma 6. Because only $O(1)$ separating paths are considered, we obtain the following result.

► **Lemma 7.** *The total complexity of the transition hourglasses of all transition edges of P is $O(n)$. Moreover, all these hourglasses can be constructed in $O(n)$ time.*

4 Funnels

Let $C = (p_0, \dots, p_k)$ be a chain of ∂P and let v be a vertex of P not in C . The *funnel* of v to C , denoted by $S_v(C)$, is the simple polygon bounded by C , $\pi(p_k, v)$ and $\pi(v, p_0)$; see Figure 3 (a). Note that the paths $\pi(v, p_k)$ and $\pi(v, p_0)$ may coincide for a while before splitting into edge-disjoint chains. A subset $R \subset P$ is *geodesically convex* if for every $x, y \in R$, the path $\pi(x, y)$ is contained in R . This funnel $S_v(C)$ is then the minimum geodesically convex set that contains v and C . See Lee and Preparata [16] or Guibas et al. [11] for more details on funnels.

► **Lemma 8.** *Let v be a vertex of P and let C be a transition chain such that $R(v) \cap \partial P \subseteq C$ and $v \notin C$. Then, $R(v)$ is contained in the funnel $S_v(C)$*



■ **Figure 3** a) The funnel $S_v(C)$ of a vertex v and a chain C contained in ∂P are depicted. b) The decomposition of $S_v(C)$ into apexed triangles produced by the shortest-path map of v .

4.1 Funnels of marked vertices

Recall that for each marked vertex $v \in M$, we know at least of one vertex on ∂P such that v is its farthest neighbor.

► **Lemma 9.** *Let x be a point in P . If $f(x) = v$ for some marked vertex $v \in M$, then $x \in S_v(C_v)$.*

For any marked vertex v , let u_1, \dots, u_{k-1} be the vertices of P such that $v = f(u_i)$ and assume that u_1, \dots, u_{k-1} are in clockwise order. Let u_0 and u_k be the neighbors of u_1 and u_{k-1} other than u_2 and u_{k-2} , respectively. Note that both u_0u_1 and $u_{k-1}u_k$ are transition edges of P . Thus, we can assume that their transition hourglasses have been computed.

Let $C_v = (u_0, \dots, u_k)$ and consider the funnel $S_v(C_v)$. We call C_v the *main chain* of $S_v(C_v)$ while $\pi(u_k, v)$ and $\pi(v, u_0)$ are referred to as the *walls* of the funnel. Because $v = f(u_1) = f(u_{k-1})$, we know that v is a vertex of both $H_{u_0u_1}$ and $H_{u_{k-1}u_k}$. By definition, we have $\pi(v, u_0) \subset H_{u_0u_1}$ and $\pi(v, u_k) \subset H_{u_{k-1}u_k}$. Thus, we can explicitly compute both paths $\pi(v, u_0)$ and $\pi(v, u_k)$ in $O(|H_{u_0u_1}| + |H_{u_{k-1}u_k}|)$ time. So, overall, the funnel $S_v(C_v)$ can be constructed in $O(k + |H_{u_0u_1}| + |H_{u_{k-1}u_k}|)$ time. Recall that, by Lemma 6, the total sum of the complexities of the transition hourglasses is $O(n)$. In particular, we can bound the total time needed to construct the funnels of all marked vertices by $O(n)$.

Since the complexity of the walls of these funnels is bounded by the complexity of the transition hourglasses used to compute them, by Lemma 7 we get that

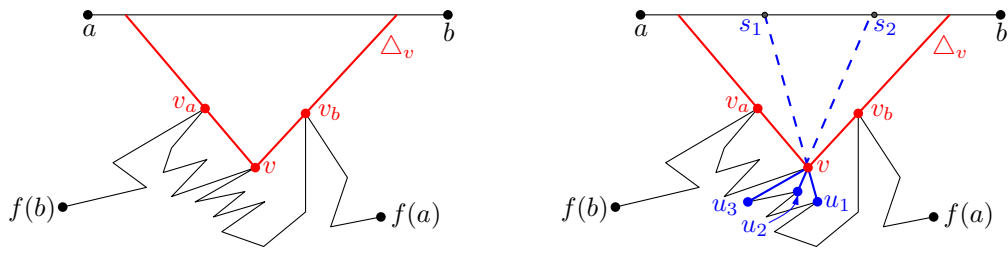
$$\sum_{v \in M} |S_v(C_v)| = O\left(n + \sum_{ab \in E} |H_{ab}|\right) = O(n).$$

► **Lemma 10.** *The total complexity of the funnels of all marked vertices of P is $O(n)$. Moreover, all these funnels can be constructed in $O(n)$ time.*

5 Covering the polygon with apexed triangles

An *apexed triangle* $\Delta = (a, b, c)$ with *apex* a is a triangle contained in P with an associated distance function $g_\Delta(x)$, called the *apex function* of Δ , such that (1) a is a vertex of P , (2) $b, c \in \partial P$, and (3) there is a vertex w of P , called the *definer* of Δ , such that

$$g_\Delta(x) = \begin{cases} -\infty & \text{if } x \notin \Delta, \\ |xa| + |\pi(a, w)| = |\pi(x, w)| & \text{if } x \in \Delta. \end{cases}$$



■ **Figure 4** (left) A vertex v visible from the segment ab lying on the bottom chain of H_{ab} , and the triangle Δ_v which contains the portion of ab visible from v . (right) The children u_1 and u_2 of v are visible from ab while u_3 is not. The triangle Δ_v is split into apexed triangles by the rays going from u_1 and u_2 to v .

In this section, we show how to find a set of $O(n)$ apexed triangles of P such that the upper envelope of their apex functions coincides with $F_P(x)$. To this end, we first decompose the transition hourglasses into apexed triangles that encode all the geodesic distance information inside them. For each marked vertex $v \in M$, we construct a funnel that contains the Voronoi region of v . We then decompose this funnel into apexed triangles that encode the distance from v .

5.1 Inside the transition hourglass

Let ab be a transition edge of P such that b is the clockwise neighbor of a along ∂P . Let B_{ab} denote the open bottom chain of H_{ab} . As noticed above, a point on ∂P can be farthest from a vertex in B_{ab} only if it lies in the open segment ab . That is, if v is a vertex of B_{ab} such that $R(v) \neq \emptyset$, then $R(v) \cap \partial P \subset ab$. In fact, not only is this Voronoi region inside H_{ab} when restricted to the boundary of P , but we can further bound its location and show that $R(v) \subset H_{ab}$. The next result follows directly from Lemma 8.

► **Corollary 11.** *Let v be a vertex of B_{ab} . If $R(v) \neq \emptyset$, then $R(v) \subset H_{ab}$.*

Our objective is to compute $O(|H_{ab}|)$ apexed triangles contained in H_{ab} , each with its distance function, such that the upper envelope of these apex functions coincides with $F_P(x)$ restricted to H_{ab} where it “matters”.

The same approach was already used by Pollack et al. in [23, Section 3]. Given a segment contained in the interior of P , they show how to compute a linear number of apexed triangles such that $F_P(x)$ coincides with the upper envelope of the corresponding apex functions in the given segment. While the construction we follow is analogous, we use it in the transition hourglass H_{ab} instead of the full polygon P . Therefore, we have to specify what is the relation between the upper envelope of the computed functions and $F_P(x)$. We will show that the upper envelope of the apex functions computed in H_{ab} coincides with $F_P(x)$ inside the Voronoi region $R(v)$ of every vertex $v \in B_{ab}$.

Let T_a and T_b be the shortest-path trees in H_{ab} from a and b rooted at a and b , respectively. We can compute these trees in $O(|H_{ab}|)$ time [11]. For each vertex v such that $f(a), v$ and $f(b)$ are in clockwise order, let v_a and v_b be the neighbors of v in the paths $\pi(v, a)$ and $\pi(v, b)$, respectively. We say that a vertex v is *visible* from ab if $v_a \neq v_b$. Note that if a vertex is visible, then the extension of these segments must intersect the top segment ab . Therefore, for each visible vertex v , we obtain a triangle Δ_v as shown in Figure 4.

We further split Δ_v into a series of triangles with apex at v as follows: Let u be a child of v in either T_a or T_b . As noted by Pollack et al., v can be of three types, either (1) u is not

visible from ab (and is hence a child of v in both T_a and T_b); or (2) u is visible from ab , is a child of v only in T_b , and vvu is a left turn; or (3) u is visible from ab , is a child of v only in T_a , and $vavu$ is a right turn.

Let u_1, \dots, u_{k-1} be the children of v of type (2) sorted in clockwise order around v . Let $c(v)$ be the maximum distance from v to any invisible vertex in the subtrees of T_a and T_b rooted at v ; if no such vertex exists, then $c(v) = 0$. Define a function $d_l(v)$ on each vertex v of H_{ab} in a recursive fashion as follows: If v is invisible from ab , then $d_l(v) = c(v)$. Otherwise, let $d_l(v)$ be the maximum of $c(v)$ and $\max\{d_l(u_i) + |u_i v| : u_i \text{ is a child of } v \text{ of type (2)}\}$. Symmetrically, we define a function $d_r(v)$ using the children of type (3) of v .

For each $1 \leq i \leq k - 1$, extend the segment $u_i v$ passed v until it intersects ab at a point s_i . Let s_0 and s_k be the intersections of the extensions of vv_a and vv_b with the segment ab . We define k apexed triangles contained in Δ_v as follows. For each $0 \leq i \leq k - 1$, consider the triangle $\Delta(s_i, v, s_{i+1})$ whose associated apexed (left) function is

$$f_i(x) = \begin{cases} |xv| + \max_{j>i} \{c(v), |vu_j| + d_l(u_j)\} & \text{if } x \in \Delta(s_i, v, s_{i+1}), \\ -\infty & \text{otherwise.} \end{cases}$$

In a symmetric manner, we define a set of apexed triangles induced by the type (3) children of v and their respective apexed (right) functions.

Let g_1, \dots, g_r and $\Delta_1, \dots, \Delta_r$ respectively be an enumeration of all the generated apex functions and apexed triangles such that g_i is defined in the triangle Δ_i . Because each function is determined uniquely by a pair of adjacent vertices in T_a or in T_b , and since these trees have $O(|H_{ab}|)$ vertices, we conclude that $r = O(|H_{ab}|)$.

Note that for each $1 \leq i \leq r$, the apexed triangle Δ_i has two vertices on the segment ab and a third vertex, say a_i , being its apex such that for each $x \in \Delta_i$, $g_i(x) = |\pi(x, w_i)|$ for some vertex w_i of H_{ab} . Recall that w_i is called the definer of Δ_i . Intuitively, Δ_i defines a portion of the geodesic distance function from w_i in a constant complexity region.

► **Lemma 12.** *Given a transition edge ab of P , we can compute a set \mathcal{A}_{ab} of $O(|H_{ab}|)$ apexed triangles in $O(|H_{ab}|)$ time with the property that for any point $p \in P$ such that $f(p) \in B_{ab}$, there is an apexed triangle $\Delta \in \mathcal{A}_{ab}$ with apex function g and definer equal to $f(p)$ such that*

1. $p \in \Delta$ and
2. $g(p) = F_P(p)$.

In other words, Lemma 12 says that no information on farthest neighbors is lost if we only consider the functions of \mathcal{A}_{ab} within H_{ab} . In the next section we construct a set of apexed triangles (and their corresponding apex functions), so as to encode the distance from the vertices of M .

5.2 Inside the funnels of marked vertices

We now proceed to split a given funnel into $O(|S_v(C_v)|)$ apexed triangles that encode the distance function from v . To this end, we use the algorithm described by Guibas et al. [12, Section 2] to compute the shortest-path map of v in $S_v(C_v)$ in $O(|S_v(C_v)|)$ time. This algorithm produces a partition of $S_v(C_v)$ into $O(|S_v(C_v)|)$ interior disjoint triangles with vertices on ∂P , such that each triangle consists of all points in $S_v(C_v)$ whose shortest path to v consists of the same sequence of vertices; see Figure 3 (b). Let Δ be a triangle in this partition and let a be its apex, i.e., the first vertex found along each path $\pi(x, v)$, where

$x \in \Delta$. We define the apex function $g_\Delta(x)$ of Δ as follows:

$$g_\Delta(x) = \begin{cases} |xa| + |\pi(a, v)| & \text{if } x \in \Delta, \\ -\infty & \text{otherwise.} \end{cases}$$

Therefore, for each $x \in \Delta$, $g_\Delta(x) = |\pi(x, v)|$.

► **Lemma 13.** *The shortest-path map of v in $S_v(C_v)$ can be computed in $O(|S_v(C_v)|)$ time and produces $O(|S_v(C_v)|)$ interior disjoint apexed triangles such that their union covers $S_v(C_v)$. Moreover, for each point $x \in R(v)$, there is an apexed triangle Δ with apex function $g(x)$ such that (1) $x \in \Delta$ and (2) $g(x) = F_P(x)$.*

Proof. The above procedure splits $S_v(C_v)$ into $O(|S_v(C_v)|)$ apexed triangles, such that the apex function in each of them is defined as the geodesic distance to v . By Lemma 9, if $x \in R(v)$, then $x \in S_v(C_v)$. Therefore, there is an apexed triangle Δ with apex function $g(x)$ such that $x \in \Delta$ and $g(x) = |\pi(x, v)| = F_P(x)$. Thus, we obtain properties (1) and (2). ◀

6 Prune and search

With the tools introduced in the previous sections, we can describe a prune and search algorithm to compute the geodesic center. The idea of the algorithm is to partition P into $O(1)$ cells using ε -nets, determine in which cell of P the center lies and recurse on that cell as a new subproblem with smaller complexity.

We can discard all apexed triangles that do not intersect the new cell containing the center. Using cuttings to produce this partition of P , we can show that both the complexity of the cell containing the center, and the number of apexed triangles that intersect it decrease by a constant fraction in each iteration of the algorithm. This process is then repeated until either of the two objects has constant descriptive size.

Let τ be the set of all apexed triangles computed in previous sections. Lemmas 6 and 12 bound the number of apexed triangles constructed inside the transition hourglasses, while Lemmas 10 and 13 do so inside the funnels of the marked vertices. We obtain the following.

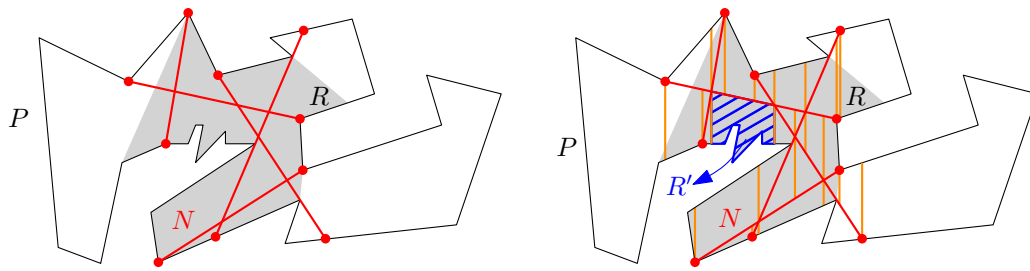
► **Corollary 14.** *The set τ consists of $O(n)$ apexed triangles.*

Let $\phi(x)$ be the upper envelope of the apex functions of the triangles in τ (i.e., $\phi(x) = \max\{g(x) : \Delta \in \tau \text{ and } g(x) \text{ is the apex function of } \Delta\}$). The following result is a direct consequence of Lemmas 12 and 13, and shows that the $O(n)$ apexed triangles of τ not only cover P , but their apex functions suffice to reconstruct the function $F_P(x)$.

► **Lemma 15.** *The functions $\phi(x)$ and $F_P(x)$ coincide in the domain of points of P , i.e., for each $p \in P$, $\phi(p) = F_P(p)$.*

Given a chord C of P , a *half-polygon* of P is one of the two simple polygons in which C splits P . A *k-cell* of P is a simple polygon obtained as the intersection of at most k half-polygons. Because a *k-cell* is the intersection of geodesically convex sets, it is also geodesically convex. The recursive algorithm described in this section takes as input a 4-cell R (initially equal to P) containing the geodesic center of P and the set of apexed triangles of τ that intersect R . In each iteration, it produces a new 4-cell of smaller complexity that intersects just a fraction of the apexed triangles and contains the geodesic center of P . By recursing on this new cell, the complexity of the problem is reduced in each iteration.

Let R be a 4-cell of P containing the geodesic center of P and let τ_R be the set of apexed triangles of τ that intersect R . Let $m_R = \max\{|R|, |\tau_R|\}$, where $|R|$ denotes the



■ **Figure 5** The ε -net N splits R into $O(1)$ sub-polygons that are further refined into a 4-cell decomposition using $O(1)$ ray-shooting queries from the vertices of the arrangement defined by N .

combinatorial complexity of R . Recall that, by construction of the apexed triangles, for each triangle of τ_R at least one and at most two of its boundary segments are chords of P . Let \mathcal{C} be the set containing all chords that belong to the boundary of a triangle of τ_R . Therefore, $|\mathcal{C}| \leq 2|\tau_R| \leq 2m_R$.

To construct ε -nets, we need some definitions (for more information on ε -nets refer to [18]). Let φ be the set of all open 4-cells of P . For each $t \in \varphi$, let $\mathcal{C}_t = \{C \in \mathcal{C} : C \cap t \neq \emptyset\}$ be the set of chords of \mathcal{C} induced by t . Finally, let $\varphi_{\mathcal{C}} = \{\mathcal{C}_t : t \in \varphi\}$ be the family of subsets of \mathcal{C} induced by φ . Consider the set system $(\mathcal{C}, \varphi_{\mathcal{C}})$ (denoted by (\mathcal{C}, φ) for simplicity).

Let $\varepsilon > 0$ (the exact value of ε will be specified later). Because the VC-dimension of the set system (\mathcal{C}, φ) is finite [1], we can compute an ε -net N of (\mathcal{C}, φ) in $O(|\mathcal{C}|/\varepsilon) = O(m_R)$ time [18]. The size of N is $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon}) = O(1)$ and its main property is that any 4-cell that does not intersect a chord of N will intersect at most $\varepsilon|\mathcal{C}|$ chords of \mathcal{C} .

Observe that N partitions R into $O(1)$ sub-polygons (not necessarily 4-cells). We further refine this partition to obtain 4-cells. That is, we shoot vertical rays up and down from each endpoint of N , and from the intersection point of any two segments of N , see Figure 5. Overall, this partitions R into $O(1)$ 4-cells such that each either (i) is a convex polygon contained in P of at most four vertices, or otherwise (ii) contains some chain of ∂P . Since $|N| = O(1)$, the whole decomposition can be computed in $O(m_R)$ time (the intersections between segments of N are done in constant time, and for the ray shooting operations we walk along the boundary of R once).

In order to determine which 4-cell contains the geodesic center of P , we extend each edge of a 4-cell to a chord C . This can be done with two ray-shooting queries (each of which takes $O(m_R)$ time). We then use the chord-oracle from Pollack et al. [23, Section 3] to decide which side of C contains c_P . The only requirement of this technique is that the function $F_P(x)$ coincides with the upper envelope of the apex functions when restricted to C , which is true by Lemma 15 and from the fact that τ_R consists of all the apexed triangles of τ that intersect R . Because the chord-oracle described by Pollack et al. [23, Section 3] runs in time linear in the number of functions defined on C , we can decide in total $O(m_R)$ time in which side of C the geodesic center of P lies. Since our decomposition into 4-cells has constant complexity, we need to perform $O(1)$ calls to the oracle before determining the 4-cell R' that contains the geodesic center of P .

The chord-oracle computes the minimum of $F_P(x)$ restricted to the chord before determining the side containing the minimum. In particular, if c_P lies on any chord bounding R' , then the chord-oracle will find it. Therefore, we can assume that c_P lies in the interior of R' . Moreover, since N is a ε -net, we know that at most $\varepsilon|\mathcal{C}|$ chords of \mathcal{C} intersect R' .

We can show that the complexity of R' also decreases: since $|\mathcal{C}| \leq 2|\tau_R| \leq 2m_R$, at most $2\varepsilon m_R$ apexed triangles intersect R' . Because $F_P(x)$ is defined in each point of R' , Lemma 15

implies that each vertex of R' is covered by at least one apexed triangle of τ_R . Since each apexed triangle can cover at most three vertices, by the pigeonhole principle we conclude that R' can have at most $6\varepsilon m_R$ vertices. Otherwise, an apexed triangle would contain at least four vertices of R' . Thus, if we choose $\varepsilon = 1/12$, we guarantee that both the size of the 4-cell R' and the number of apexed triangles in $\tau_{R'}$ are at most $m_R/2$.

In order to proceed with the algorithm on R' recursively, we need to compute the set $\tau_{R'}$ with the at most $\varepsilon|\mathcal{C}|$ apexed triangles of τ_R that intersect R' (i.e., prune the apexed triangles that do not intersect with R'). For each apexed triangle $\Delta \in \tau_R$, we can determine in constant time if it intersects R' (either one of the endpoints is in $R' \cap \partial P$ or the two boundaries have non-empty intersection in the interior of P). Overall, we need $O(m_R)$ time to compute the at most $\varepsilon|\mathcal{C}|$ triangles of τ_R that intersect R' .

By recursing on R' , we guarantee that after $O(\log m_R)$ iterations, we reduce the size of either τ_R or R' to constant. In the former case, the minimum of $F_P(x)$ can be found by explicitly constructing ϕ in $O(1)$ time. In the latter case, we triangulate R' and apply the chord-oracle to determine which triangle will contain c_P . The details needed to find the minimum of $\phi(x)$ inside this triangle are given in the next section.

► **Lemma 16.** *In $O(n)$ time we can find either the geodesic center of P or a triangle containing the geodesic center.*

7 Finding the center within a triangle

In order to complete the algorithm it remains to show how to find the geodesic center of P for the case in which R' is a triangle. If this triangle is in the interior of P , it may happen that several apexed triangles of τ fully contain R' . Thus, the pruning technique used in the previous section cannot be further applied. We solve this case with a different approach.

Recall that $\phi(x)$ denotes the upper envelope of the apex functions of the triangles in τ , and the geodesic center is the point that minimizes ϕ . The key observation is that, as it happened with chords, the function $\phi(x)$ restricted to R' is convex.

Let $\Delta_1, \Delta_2, \dots, \Delta_m$ be the set of $m = O(n)$ apexed triangles of τ that intersect R' . Let a_i and w_i be the apex and the definer of Δ_i , respectively. Let $g_i(x)$ be the apex function of Δ_i such that

$$g_i(x) = \begin{cases} |xa_i| + \kappa_i & \text{if } x \in \Delta_i, \\ -\infty & \text{otherwise,} \end{cases}$$

where $\kappa_i = |\pi(a_i, w_i)|$ is a constant.

By Lemma 15, $\phi(x) = F_P(x)$. Therefore, the problem of finding the center is equivalent to the following optimization problem in \mathbb{R}^3 :

(P1). Find a point $(x, r) \in \mathbb{R}^3$ minimizing r subject to $x \in R'$ and

$$g_i(x) \leq r, \text{ for } 1 \leq i \leq m.$$

Thus, we need only to find the solution to (P1) to find the geodesic center of P . We use some remarks described by Megiddo in order to simplify the description of (P1) [19].

To simplify the formulas, we square the equation $|xa_i| \leq r - \kappa_i$:

$$\|x\|^2 - 2x \cdot a_i + \|a_i\|^2 = |xa_i|^2 \leq (r - \kappa_i)^2 = r^2 - 2r\kappa_i + \kappa_i^2.$$

And finally for each $1 \leq i \leq m$, we define the function $h_i(x, r)$ as follows:

$$h_i(x, r) = \begin{cases} \|x\|^2 - 2x \cdot a_i + \|a_i\|^2 - r^2 + 2r\kappa_i - \kappa_i^2 & \text{if } x \in \Delta_i, \\ -\infty & \text{otherwise.} \end{cases}$$

Therefore, our optimization problem can be reformulated as:

(P2). Find a point $(x, r) \in \mathbb{R}^3$ such that r is minimized subject to $x \in R'$ and

$$h_i(x, r) \leq 0 \text{ and } r > \max\{\kappa_i\}, \text{ for } 1 \leq i \leq m.$$

Let $h'_i(x, r) = \|x\|^2 - 2x \cdot a_i + \|a_i\|^2 - r^2 + 2r\kappa_i - \kappa_i^2$ be a function defined in the entire plane and let (P2') be an optimization problem analogous to (P2) where every instance of $h_i(x, r)$ is replaced by $h'_i(x, r)$. The optimization (P2') was studied by Megiddo in [19]. We provide some of the intuition used by Megiddo to solve this problem.

Although the functions $h'_i(x, r)$ are not linear, they all have the same non-linear terms. Therefore, for $i \neq j$, we get that $h'_i(x, r) = h'_j(x, r)$ defines a *separating plane*

$$\gamma_{i,j} = \{(x, r) \in \mathbb{R}^3 : 2(\kappa_i - \kappa_j)r - 2(a_i - a_j) \cdot x + \|a_i\|^2 - \|a_j\|^2 - \kappa_i^2 + \kappa_j^2 = 0\}.$$

As noted by Megiddo [19], this separating plane has the following property: If the solution (x, r) to (P2') is known to lie to one side of $\gamma_{i,j}$, then we know that one of the constraints is redundant. Thus, to solve (P2') it sufficed to have a *side-decision oracle* to determine in which side of a plane $\gamma_{i,j}$ the solution lies. Megiddo showed how to implement this oracle in a way that the running time is proportional to the number of constraints [19].

Once we have such an oracle, Megiddo's problem can be solved using a prune and search approach: pair the functions arbitrarily, and consider the set of $m/2$ separating planes defined by these pairs. For some constant t , compute a $1/t$ -cutting in \mathbb{R}^3 of the separating planes. A $1/t$ -cutting is a partition of the plane into $O(t^3) = O(1)$ convex regions each of which is of constant complexity and intersects at most $m/2t$ separating planes. A cutting of planes can be computed in linear time in \mathbb{R}^3 for any $t = O(1)$ [17]. After computing the cutting, determine in which of the regions the minimum lies by performing $O(1)$ calls to the side-decision oracle. Because at least $(t - 1)m/2t$ separating planes do not intersect this constant complexity region, for each of them we can discard one of the constraints as it becomes redundant. Repeating this algorithm recursively we obtain a linear running time.

To solve (P2) we follow a similar approach, but our set of separating planes needs to be extended in order to handle apex functions as they are only defined in the same way as in (P2') in a triangular domain. Note that no vertex of an apexed triangle can lie inside R' .

7.1 Optimization problem in a convex domain

In this section we describe our algorithm to solve the optimization problem (P2). To this end, we pair the apexed triangles arbitrarily to obtain $m/2$ pairs. By identifying the plane where P lies with the plane $Z_0 = \{(x, y, z) : z = 0\}$, we can embed each apexed triangle in \mathbb{R}^3 . A *plane-set* is a set consisting of at most five planes in \mathbb{R}^3 . For each pair of apexed triangles (Δ_i, Δ_j) we define its plane-set as follows: For each chord of P bounding either Δ_i or Δ_j (at most two chords on each triangle), consider the line extending this chord and the vertical extrusion of this line in \mathbb{R}^3 , i.e., the plane containing this chord orthogonal to Z_0 . Moreover, consider the separating plane $\gamma_{i,j}$. The set containing these planes is the plane-set of the pair (Δ_i, Δ_j) .

Let Γ be the union of all the plane-sets defined by the $m/2$ pairs of apexed triangles. Because the plane-set of each pair (Δ_i, Δ_j) consists of at most five planes and contains at least one plane unique to this pair, say $\gamma_{i,j}$, we infer that $m/2 \leq |\Gamma| \leq 5m/2$.

Compute a $1/t$ -cutting of Γ in $O(m)$ time for some constant t to be specified later. Because t is constant, this $1/t$ -cutting splits the space into $O(1)$ convex regions, each bounded by a

constant number of planes [17]. Using a side-decision algorithm (to be specified later), we can determine the region Q of the cutting that contains the solution to (P2). Because Q is the region of a $1/t$ -cutting of Γ , we know that at most $|\Gamma|/t$ planes of Γ intersect Q . In particular, at most $|\Gamma|/t$ plane-sets intersect Q and hence, at least $(t-1)|\Gamma|/t$ plane-sets do not intersect Q . Since $|\Gamma| \geq m/2$, at least $(t-1)m/2t$ plane-sets do not intersect Q .

Let (Δ_i, Δ_j) be a pair such that its plane-set does not intersect Q . Let Q' be the projection of Q on the plane Z_0 . Because the plane-set of this pair does not intersect Q , we know that Q' intersects neither the boundary of Δ_i nor that of Δ_j . Two cases arise:

Case 1. If either Δ_i or Δ_j does not intersect Q' , then we know that their apex function is redundant and we can drop the constraint associated with this apexed triangle.

Case 2. If $Q' \subset \Delta_i \cap \Delta_j$, then we need to decide which constraint to drop. To this end, we consider the separating plane $\gamma_{i,j}$. Notice that inside the vertical extrusion of $\Delta_i \cap \Delta_j$ (and hence in Q), the plane $\gamma_{i,j}$ has the property that if we know which side of it contains the solution, then one of the constraints can be dropped. Since $\gamma_{i,j}$ does not intersect Q as $\gamma_{i,j}$ belongs to the plane-set of (Δ_i, Δ_j) , we can decide which side of $\gamma_{i,j}$ contains the solution to (P2) and drop one of the constraints.

Regardless of the case, if the plane-set of a pair (Δ_i, Δ_j) does not intersect Q , then we can drop one of its constraints. Since at least $(t-1)m/2t$ plane-sets do not intersect Q , we can drop at least $(t-1)m/2t$ constraints. By choosing $t = 2$, we are able to drop at least $(t-1)m/2t = m/4$ constraints. Consequently, after $O(m)$ time, we are able to drop $m/4$ apexed triangles. By repeating this process recursively, we end up with a constant size problem in which we can compute the upper envelope of the functions explicitly and find the solution to (P2) using exhaustive search. Thus, the running time of this algorithm is bounded by the recurrence $T(m) = T(3m/4) + O(m)$ which solves to $O(m)$. Because $m = O(n)$, we can find the solution to (P2) in $O(n)$ time.

It remains to describe the side-decision algorithm. Given a plane γ , we want to decide in which side of γ lies the solution to (P2). To this end, we solve (P2) restricted to γ , i.e., with the additional constraint $(x, r) \in \gamma$. This approach was used by Megiddo [19], the idea is to recurse by reducing the dimension of the problem. Another approach is to use a slight modification of the chord-oracle described by Pollack et al. [23, Section 3].

Once the solution to (P2) restricted to γ is known, we can follow the same idea used by Megiddo [19] to find the side of γ containing the global solution to (P2). That is, we find the apex functions that define the minimum restricted to γ . Since $\phi(x) = F_p(x)$ is locally defined by these functions, we can decide in which side the minimum lies using convexity. We obtain the following result.

► **Lemma 17.** *Let R' be a convex trapezoid contained in P such that R' contains the geodesic center of P . Given the set of all apexed triangles of τ that intersect R' , we can compute the geodesic center of P in $O(n)$ time.*

The following theorem summarizes the result presented in this paper.

► **Theorem 18.** *We can compute the geodesic center of any simple polygon P of n vertices in $O(n)$ time.*

References

- 1 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. *CoRR*, abs/1501.00561, 2015.

- 2 Boris Aronov, Steven Fortune, and Gordon Wilfong. The furthest-site geodesic Voronoi diagram. *Discrete & Computational Geometry*, 9(1):217–255, 1993.
- 3 T. Asano and G.T. Toussaint. Computing the geodesic center of a simple polygon. Technical Report SOCS-85.32, McGill University, 1985.
- 4 Sang Won Bae, Matias Korman, and Yoshio Okamoto. The geodesic diameter of polygonal domains. *Discrete & Computational Geometry*, 50(2):306–329, 2013.
- 5 Sang Won Bae, Matias Korman, and Yoshio Okamoto. Computing the geodesic centers of a polygonal domain. In *Proceedings of CCCG*, 2014.
- 6 Sang Won Bae, Matias Korman, Yoshio Okamoto, and Haitao Wang. Computing the L_1 geodesic diameter and center of a simple polygon in linear time. In *Proceedings of LATIN*, pages 120–131, 2014.
- 7 Bernard Chazelle. A theorem on polygon cutting with applications. In *Proceedings of FOCS*, pages 339–349, 1982.
- 8 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(1):485–524, 1991.
- 9 H.N. Djidjev, A. Lingas, and J.-R. Sack. An $O(n \log n)$ algorithm for computing the link center of a simple polygon. *Discrete & Computational Geometry*, 8:131–152, 1992.
- 10 Herbert Edelsbrunner and Ernst Peter Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Trans. on Graphics*, 9(1):66–104, 1990.
- 11 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(1-4):209–233, 1987.
- 12 Leonidas J Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *Journal of computer and system sciences*, 39(2):126–152, 1989.
- 13 Dov Harel and Robert Endre Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM Journal on Computing*, 13(2):338–355, 1984.
- 14 John Hershberger and Subhash Suri. Matrix searching with the shortest-path metric. *SIAM Journal on Computing*, 26(6):1612–1634, 1997.
- 15 Y. Ke. An efficient algorithm for link-distance problems. In *Proceedings of SoCG*, pages 69–78, 1989.
- 16 Der-Tsai Lee and Franco P Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14(3):393–410, 1984.
- 17 Jiří Matoušek. Approximations and optimal geometric divide-and-conquer. *Journal of Computer and System Sciences*, 50(2):203–208, 1995.
- 18 Jiří Matoušek. Construction of epsilon nets. In *Proc. of SoCG*, pages 1–10. ACM, 1989.
- 19 Nimrod Megiddo. On the ball spanned by balls. *Discrete & Computational Geometry*, 4(1):605–610, 1989.
- 20 J. S. B. Mitchell. Geometric shortest paths and network optimization. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier, 2000.
- 21 B.J. Nilsson and S. Schuierer. Computing the rectilinear link diameter of a polygon. In *Proceedings of CG*, pages 203–215, 1991.
- 22 B.J. Nilsson and S. Schuierer. An optimal algorithm for the rectilinear link center of a rectilinear polygon. *Computational Geometry: Theory and Applications*, 6:169–194, 1996.
- 23 Richard Pollack, Micha Sharir, and Günter Rote. Computing the geodesic center of a simple polygon. *Discrete & Computational Geometry*, 4(1):611–626, 1989.
- 24 S. Suri. *Minimum Link Paths in Polygons and Related Problems*. PhD thesis, Johns Hopkins Univ., 1987.
- 25 Subhash Suri. Computing geodesic furthest neighbors in simple polygons. *Journal of Computer and System Sciences*, 39(2):220–235, 1989.

On the Smoothed Complexity of Convex Hulls*

Olivier Devillers¹, Marc Glisse², Xavier Goaoc³, and Rémy Thomasse⁴

- 1 Inria, Centre de recherche Nancy – Grand Est, France
CNRS, Loria, France
Université de Lorraine, France
- 2 Inria, Centre de recherche Saclay – Île-de-France, France
- 3 LIGM, Université Paris-Est Marne-la-Vallée, France
- 4 Inria, Centre de recherche Sophia Antipolis – Méditerranée, France

Abstract

We establish an upper bound on the smoothed complexity of convex hulls in \mathbb{R}^d under uniform Euclidean (ℓ^2) noise. Specifically, let $\{p_1^*, p_2^*, \dots, p_n^*\}$ be an arbitrary set of n points in the unit ball in \mathbb{R}^d and let $p_i = p_i^* + x_i$, where x_1, x_2, \dots, x_n are chosen independently from the unit ball of radius δ . We show that the expected complexity, measured as the number of faces of all dimensions, of the convex hull of $\{p_1, p_2, \dots, p_n\}$ is $O\left(n^{2-\frac{4}{d+1}}(1+1/\delta)^{d-1}\right)$; the magnitude δ of the noise may vary with n . For $d = 2$ this bound improves to $O\left(n^{\frac{2}{3}}(1+\delta^{-\frac{2}{3}})\right)$.

We also analyze the expected complexity of the convex hull of ℓ^2 and Gaussian perturbations of a nice sample of a sphere, giving a lower-bound for the smoothed complexity. We identify the different regimes in terms of the scale, as a function of n , and show that as the magnitude of the noise increases, that complexity varies monotonically for Gaussian noise but non-monotonically for ℓ^2 noise.

1998 ACM Subject Classification G.3 Probabilistic algorithms

Keywords and phrases Probabilistic analysis, Worst-case analysis, Gaussian noise

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.224

1 Introduction

In this paper we study the *smoothed complexity* [9] of convex hulls, a structure whose importance in computational geometry no longer needs arguing. This smoothed complexity analysis includes two, distinct, technical difficulties. It first requires to study the average complexity of the convex hull of a random perturbation of a given, initial, point set; that is, perform average-case analysis albeit for an atypical probability distribution. It then asks to control the maximum of that expected complexity over all choices of the initial point set. We present new insights on both issues for two noise models: uniform, bounded-radius, Euclidean noise and Gaussian noise.

Motivations. Combinatorial structures induced by geometric data are some of the basic building blocks of computational geometry, and typical examples include convex hulls or Voronoi diagrams of finite point sets, lattices of polytopes obtained as intersections of sets of half-spaces, intersection graphs or nerves of families of balls . . . The size of these structures

* Part of this work is supported by: ANR blanc PRESAGE (ANR-11-BS02-003), Région PACA and Institut Universitaire de France.

usually depends not only on the number n of geometric primitives (points, half-spaces, balls . . .), but also on their relative position: for instance, the number of faces of the Voronoi diagram of n points in \mathbb{R}^d is $\Theta(n)$ if these points lie on a regular grid but $\Theta(n^{\lceil d/2 \rceil})$ when they lie on the moment curve. A simple, conservative, measure is the *worst-case complexity*, which expresses, as a function of n , the maximum complexity over all inputs of size n . For geometric structures, the worst-case bounds are often attained by generic but brittle constructions: the high complexity remains if sufficiently small perturbations are applied, but vanishes under large enough perturbations. One may wonder about the relevance of worst-case bounds in practical situations, where input points come from noisy measurements and are represented using bounded precision. Assessing this relevance requires to quantify the stability of worst-case examples. This is precisely what the *smoothed complexity* captures.

Smoothed complexity model. The smoothed complexity of the convex hull in \mathbb{R}^d is the quantity

$$\max_{p_1^*, p_2^*, \dots, p_n^* \in K} \mathbb{E} [\text{card}(CH(\{p_1^* + x_1, p_2^* + x_2, \dots, p_n^* + x_n\}))]$$

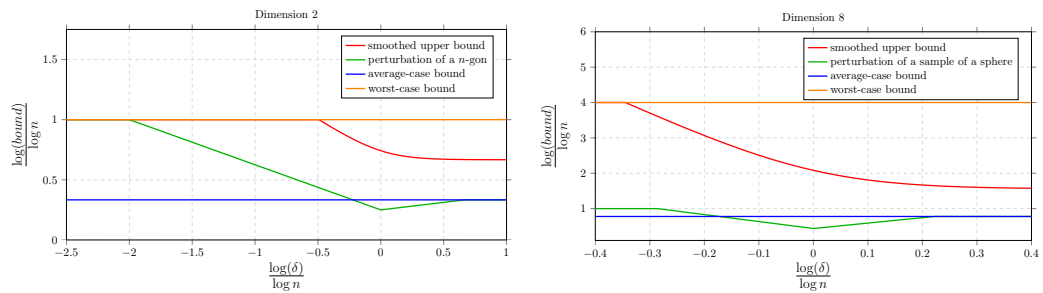
where K is some bounded domain in \mathbb{R}^d of fixed size, $\text{card}(CH(X))$ denotes the combinatorial complexity, *ie* the total number of faces of all dimensions, of the convex hull of X , and x_1, x_2, \dots, x_n are independent random variables, usually identically distributed. The goal is to express this bound as a function of the number n of points and some parameter that describes the amplitude of the perturbations x_i 's. The only examples of smoothed complexity analysis of geometric *structures* (rather than *algorithms*) that we are aware of are some aspects of random polytopes related to the simplex algorithm [9] and visibility maps on terrains [3]. In this paper we consider two types of perturbation, the ℓ^2 **perturbation** where the x_i 's are drawn independently from the ball of radius $\delta > 0$ centered at the origin, and the **Gaussian perturbation** where the x_i 's are drawn independently from the d -dimensional multivariate Gaussian distribution with mean vector $\vec{0}$ and covariance matrix $\sigma^2 I_d$. We will assume that the domain K containing the initial point set is the unit ball centered at the origin, so that the ratio between the initial configuration and the perturbation is entirely contained in the perturbation parameter, δ or σ .

New results. Our first result is the following upper bound (Theorem 7) on the smoothed complexity of the convex hull under ℓ^2 perturbation:

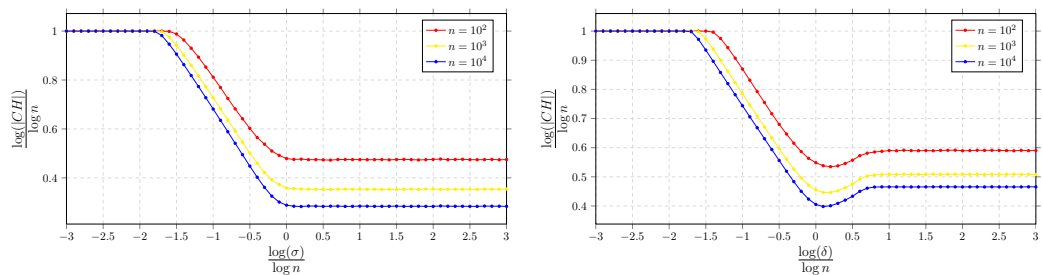
$$\max_{p_1^*, p_2^*, \dots, p_n^* \in K} \mathbb{E} [\text{card}(CH(\{p_1^* + x_1, p_2^* + x_2, \dots, p_n^* + x_n\}))] = O\left(n^{2 - \frac{4}{d+1}} \left(1 + \frac{1}{\delta}\right)^{d-1}\right).$$

(Refer to Figure 1.) For $d = 2$ this bound improves to $O\left(n^{\frac{2}{3}}(1 + \delta^{-\frac{2}{3}})\right)$, *cf* Corollary 9. Here K is the unit ball in \mathbb{R}^d . The bound is asymptotic as $n \rightarrow \infty$ and the constant in the $O()$ depends on d but is independent of δ , which may vary with n . The proof essentially decomposes the initial point set into a “boundary” part and an “interior” part and controls each contribution separately. The classification is very flexible and emerges naturally from a witness-collector mechanism [4] proposed by some of the authors to measure the complexity of random geometric structures.

Going in the other direction, one may wonder which original point sets $\{p_i^*\}_{1 \leq i \leq n} \subset K$ are extremal for the smoothed complexity. In the plane, two natural candidates are the case where the p_i^* 's are all at the origin, and the case where the p_i^* 's form a regular n -gon



■ **Figure 1** A comparison of our smoothed complexity bound of Theorem 7 and two lower bounds, where the initial points are placed respectively at the vertices of a unit-size n -gon (Theorems 10) and in the origin. The left-hand figure is for $d = 2$, the right-hand figure is for $d = 8$, and all bounds are for uniform ℓ^2 perturbation. A data point with coordinates (x, y) means that for a perturbation with δ of magnitude n^x the expected size of the convex hull grows as n^y , subpolynomial terms being ignored. The worst-case bound is given as a reference. The constants in the $O()$ and $\Omega()$ have been ignored as their influence vanishes as $n \rightarrow \infty$ in this coordinate system.



■ **Figure 2** Experimental results for the complexity of the convex hull of a perturbation of the regular n -gon inscribed in the unit circle. Left: Gaussian perturbation of variance σ^2 . Right: ℓ^2 perturbation of amplitude δ . Each data point corresponds to an average over 1000 experiments.

inscribed in K . The former case corresponds to a classical model of random polytopes and is well understood (see below). Experiments for the latter case suggest a surprising difference in the behaviour of ℓ^2 and Gaussian perturbations (refer to Figure 2): while for Gaussian perturbation the expected complexity consistently decreases as the amplitude of the noise increases, for ℓ^2 perturbation some non-monotonicity appears.

Motivated by these observations we performed a complete analysis of the expected complexity of the convex hull of ℓ^2 perturbations of a good sample of the unit sphere and of Gaussian perturbations of a regular n -gon. Our bounds (Theorems 10 and 13) delineate the main regimes in (δ, n) and (σ, n) ; they confirm the existence of the observed non-monotonicity for ℓ^2 perturbation and its absence for Gaussian perturbation, and provide a complete analysis of a candidate lower-bound for the smoothed complexity (see Figure 1).

Related work. This work builds on a previous work by some of the authors to develop a method to derive, with minimum effort, rough estimates on the complexity of some random geometric hypergraphs [4]. The smoothed complexity bound uses ingredients from that witness-collector method in a new way. Theorems 10 and 13 build on one of the case-analysis from that work, extend it to all scales of perturbation and to Gaussian noise, and dispose of extraneous log factors using an idea which we learned from [5] and systematize here (Lemma 3).

The only previous bound on the smoothed complexity of convex hulls is due to Damerow and Sohler [2]. Their main insight is a quantitative version of the following intuitive assertion: if the magnitude of the perturbations is sufficiently large compared to the scale of the initial input, the initial position of the points does not matter and smoothed complexity can be subsumed by some average-case analysis (up to constant factors).¹ A smoothed complexity bound then follows by a simple rescaling argument.²

It should first be noted that the rescaling argument only applies to bound the number of vertices of the convex hull since faces of higher dimension may come from more than one cell. Next, Damerow and Sohler argue that the average-case bound controls the smoothed complexity for *dominating points*; in several situations the dominating points largely outnumber the extreme points, so this bound may be overly conservative. Last, our analysis gives finer bounds than the rescaling argument of Damerow and Sohler alone. Consider for instance the perturbation of the vertices of the unit-size n -gon by a Gaussian noise of standard deviation σ . The rescaling argument yields³ that the number of dominant points is $O\left(\frac{\log(\sigma n)\sqrt{\log n}}{\sigma}\right)$. Our technique bounds the size of the convex hull by $O\left(\frac{\sqrt[4]{\log n}}{\sqrt{\sigma}}\right)$ for $\sigma \in \left[\frac{\log^4 n}{n^2}, \frac{1}{\sqrt{\log n}}\right]$ and $O(\sqrt{\log n})$ for $\sigma > \frac{1}{\sqrt{\log n}}$.

Our work is also related to the classical question of the expected complexity of random polytopes. Starting with the seminal articles of Renyi and Sulanke [7, 8] in the 1960's, a series of works in stochastic geometry led to precise quantitative statements (*eg.* central limit theorems) for models such as convex hulls of points sampled i.i.d. from a Gaussian distribution or the uniform measure on a convex body; we refer the interested reader to the recent survey of Reitzner [6]. Our work departs from this line of research by refining the *model* rather than the *estimates*; to put it bluntly, we content ourselves with $\Theta()$'s in place of central limit theorems but aim for analyzing more complicated probabilistic models where points are not identically distributed and laws are not given explicitly.

2 The Witness-Collector technique

Analyzing the smoothed complexity of convex hulls, or other geometric structures such as Delaunay triangulations, reduces to the following core problem. We are given a range space (\mathbb{R}^d, R) , a finite set $P \subseteq \mathbb{R}^d$ of random independent points, and want to estimate the expected complexity of some geometric hypergraph $H = \{P \cap r : r \in R\}$ induced by R on P . In plain English, a subset $Q \subset P$ is a hyperedge of H if and only if there exists $r \in R$ such that $r \cap P = Q$. When the ranges are the half-spaces delimited by hyperplanes, the set of vertices of any k -dimensional face of the convex hull of P is an element of H of cardinality $k + 1$; the converse is true for the vertices ($k = 0$) and if it may fail for higher dimensional faces, the

¹ Specifically, they show that if n points from a region of diameter r are perturbed by a Gaussian noise of standard deviation $\Omega(r\sqrt{\log n})$ or a ℓ^∞ noise of amplitude $\Omega(r\sqrt[3]{n/\log n})$ then the expected number of dominating point is the same as in the average-case analysis.

² Split the input domain into cells of size $r = O(\sigma/\sqrt{\log n})$, assume that *each* cell contains all of the initial point set, and charge each of them with the average-case bound.

³ Split the original domain into cells of size $r = O(\sigma/\sqrt{\log n})$. The input points are distributed evenly (up to constant factors) between $\Theta(1/r)$ of these cells. Each such cell contains $m = O(rn)$ input points and contributes on average $O(\log m)$ dominating points – here considering dominating points make a difference. Altogether, the expected number of dominating points is $O\left(\frac{\log(rn)}{r}\right) = O\left(\frac{\sqrt{\log n \log(\sigma n)}}{\sigma}\right)$.

overcounting often turns out to be negligible. Our goal is thus to estimate the complexity of $H_{(k+1)}$, the set of hyperedges of H of size $k+1$. From now on we focus on bounding $\text{card}(H_{(k)})$ in the case where R is the set of half-spaces in \mathbb{R}^d .

2.1 Static witness-collector pairs

To estimate the complexity of a geometric hypergraph $H_{(k)}$ we follow a simple and general approach dubbed the *witness-collector* method. The idea is to break down R into a small number of subsets of ranges $R_1 \cup R_2 \cup \dots \cup R_m$ and associate to each R_i two regions, a *witness* W_i and a *collector* C_i , with the following properties:

- (a) W_i contains at least k points of P with high probability,
- (b) C_i contains on average a small number of points of P ,
- (c) if W_i contains k points of P then C_i contains every hyperedge of $\{P \cap r : r \in R_i\}$.

Condition (c) ensures that when a set W_i contains at least k points of P , it witnesses that all hyperedges induced by R_i are collected by C_i . In particular the expected number of hyperedges of H of size k , conditioned on the event that every witness contains at least k points of P , is bounded from above by

$$\mathbb{E} [\text{card}(C_1 \cap P)^k + \text{card}(C_2 \cap P)^k + \dots + \text{card}(C_m \cap P)^k].$$

By (a), the conditioning event fails with small probability, so if that happens we can afford to use the worst-case bound $\binom{\text{card}(P)}{k}$. This bound is expressed in terms of the $\mathbb{E} [\text{card}(C_i \cap P)^k]$ whereas (b) controls $\mathbb{E} [\text{card}(C_i \cap P)]$; this is not an issue as long as the position of the points are independent random variables:

► **Lemma 1** ([4, Lemma 2]). *If $X = \sum_{i=1}^n X_i$, where the X_i are independently distributed random variables with value in $\{0, 1\}$ and $\mathbb{E}[X] \geq 1$ then $\mathbb{E}[X^k] = O(\mathbb{E}[X]^k)$.*

By a Chernoff bound, Condition (a) reduces to controlling the expectation of $\text{card}(W_i \cap P)$:

► **Lemma 2** ([4, Lemma 1]). *Let P be a set of n random points of \mathbb{R}^d independently distributed. If W is a region that contains on average at least $k \log n$ points of P then the probability that W contains less than k points of P is $O(n^{-k})$.*

The simplest use of this approach consists in placing explicitly fixed pairs of witnesses and collectors that “cover” the distribution to analyze (see [4] for several examples). This typically results in bounds containing some extra log factors (coming from Lemma 2).

2.2 Adaptive witness-collector pairs

When using Lemma 2 to ensure Condition (a), we increase the expected size of each $W_i \cap P$ so that *all* witnesses contain enough points for *most* realizations of P . Since we typically need that $W_i \subseteq C_i$, this also overloads the collectors and results in the extra log factors mentioned above. An idea to obtain sharper bounds, first introduced in [5], is to make W_i and C_i random variables depending on the random point set P . By tailoring the witness-collector pairs to each realization of the point set P , very few collectors will need to be large, and those will be negligible in the total.

More formally, we again break down R into a small number of subsets of ranges $R_1 \cup R_2 \cup \dots \cup R_m$ and associate to each R_i a sequence $\{(W_i^j, C_i^j)\}_{j \leq \log^2 n}$ of witness-collector pairs. We replace (a)–(c) by the following conditions for all j :

- (a') $\mathbb{E} \left[\text{card}(W_i^j \cap P) \right] = \Omega(j),$
- (b') $\mathbb{E} \left[\text{card}(C_i^j \cap P) \right] = O(j),$
- (c') if W_i^j contains k points of P then C_i^j contains every hyperedge of $\{P \cap r : r \in R_i\},$
- (d') $W_i^j \subseteq W_i^{j+1},$
- (e') $W_i^j \subseteq C_i^j,$

► **Lemma 3.** *Let (\mathbb{R}^d, R) be a range space, $P \subseteq \mathbb{R}^d$ a set of n random, independent points, and H the hypergraph induced by R on P . Assume that $R = R_1 \cup R_2 \cup \dots \cup R_m$ and that for each $i \in \{1, 2, \dots, m\}$ we have a sequence $\{(W_i^j, C_i^j)\}_{j \leq \log^2 n}$ of witness-collector pairs satisfying (a'), (b'), (c'), (d') and (e') for all i, j . Then $\mathbb{E} \left[\text{card}(H_{(k)}) \right] = O(m).$*

Proof. Let $i \in \{1, 2, \dots, m\}$. We let d_i denote the smallest j such that W_i^j contains at least k points and $C_i = C_i^{d_i}$, or, if no such W_i^j exists, $d_i = \infty$ and $C_i = \mathbb{R}^d$. (So d_i and C_i are random variables depending on P .) All hyperedges of H of size k that are induced by R_i are, by (c') and the definition of d_i , contained in C_i .

We claim that for some $\lambda > 0$, depending only on the constant in the $\Omega()$ in (a'), we have $\mathbb{P}[d_i \geq j] = O(e^{-\lambda j})$ for $j \leq \log^2 n$. Indeed, observe that $\text{card}(W_i^j \cap P)$ is a sum of independently distributed random variables (one per point of P) with values in $\{0, 1\}$. Letting $\alpha_j = \mathbb{E} \left[\text{card}(W_i^j \cap P) \right]$, Chernoff's bound yields that for any $0 < \gamma < 1$,

$$\mathbb{P} \left[\text{card}(W_i^j \cap P) \leq (1 - \gamma)\alpha_j \right] \leq e^{-\frac{\gamma^2 \alpha_j}{2}}.$$

Setting $\gamma = 1 - \frac{k}{\alpha_j}$ we have

$$\mathbb{P}[d_i > j] = \mathbb{P} \left[\text{card}(W_i^j \cap P) \leq k \right] \leq e^{-\frac{\gamma^2 \alpha_j}{2}}$$

and the claim follows by (a').

We also claim that (b') implies that $\mathbb{E} \left[\text{card}(C_i^j \cap P) \mid d_i \geq j \right] = O(j)$. Indeed, working with the complement \bar{C}_i^j of C_i^j , $\mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \right] = \sum_{p \in P} \mathbb{P} \left[p \notin C_i^j \right]$. For any $T \subset P$ we have

$$\mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \mid W_i^j \cap P = T \right] = \sum_{p \in P \setminus T} \mathbb{P} \left[p \notin C_i^j \mid p \notin W_i^j \right] \geq \sum_{p \in P \setminus T} \mathbb{P} \left[p \notin C_i^j \right]$$

by (e'). Thus, $\mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \right] \leq \mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \mid W_i^j \cap P = T \right] + \text{card}(T)$. Total probabilities let us decompose the event $d_i \geq j$ (equivalent, by (d'), to $\text{card}(W_i^{j-1} \cap P) < k$):

$$\begin{aligned} & \mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \mid d_i \geq j \right] \\ &= \sum_{T / \text{card}(T) < k} \mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \mid W_i^{j-1} \cap P = T \right] \mathbb{P} \left[W_i^{j-1} \cap P = T \mid \text{card}(W_i^{j-1} \cap P) < k \right] \\ &\geq \left(\mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \right] - k \right) \sum_{T / \text{card}(T) < k} \mathbb{P} \left[W_i^{j-1} \cap P = T \mid \text{card}(W_i^{j-1} \cap P) < k \right] \\ &= \mathbb{E} \left[\text{card}(\bar{C}_i^j \cap P) \right] - k \end{aligned}$$

Moving back to the complement (P has n points in total),

$$\mathbb{E} \left[\text{card}(C_i^j \cap P) \mid d_i \geq j \right] \leq \mathbb{E} \left[\text{card}(C_i^j \cap P) \right] + k = O(j),$$

and each collector C_i contains on average few points:

$$\begin{aligned} \mathbb{E} [\text{card}(C_i \cap P)] &= \sum_{j=1}^{\log^2 n} \mathbb{E} \left[\text{card}(C_i^j \cap P) \cdot \mathbb{1}_{d_i=j} \right] + \mathbb{E} [n \cdot \mathbb{1}_{d_i=\infty}] \\ &\leq \sum_{j=1}^{\log^2 n} \mathbb{E} \left[\text{card}(C_i^j \cap P) \cdot \mathbb{1}_{d_i \geq j} \right] + \mathbb{E} [n \cdot \mathbb{1}_{d_i=\infty}] \\ &= \sum_{j=1}^{\log^2 n} \mathbb{E} \left[\text{card}(C_i^j \cap P) \mid d_i \geq j \right] \mathbb{P} [d_i \geq j] + n \cdot \mathbb{P} [d_i = \infty] \\ &= \sum_{j=1}^{\log^2 n} O(j)O(e^{-\lambda j}) + nO(n^{-k}) = O(1) \end{aligned}$$

and, by Lemma 1, the number of hyperedges is $\mathbb{E} [\text{card}(H_{(k)})] = O(m)$. ◀

We can turn the $O()$ of Lemma 3 into a $\Theta()$ by using an additional condition:

► **Lemma 4.** *Assume that the conditions of Lemma 3 are satisfied and that the sequence $\{(W_i^j, C_i^j)\}_{j \leq n}$ of witness-collector pairs also satisfies*

(f') *There exist $\gamma > 0$ independent of n and $I \subseteq \{1, 2, \dots, m\}$ of size $\Omega(m)$ such that $\mathbb{P} [W_i^1 \cap W_j^1 \cap P \neq \emptyset] = 0$ for distinct $i, j \in I$, and for $i \in I$, $\mathbb{P} [W_i^1 \cap H_{(1)} \neq \emptyset] \geq \gamma$.*

If each point of P is in at least one hyperedge of size k then $\mathbb{E} [\text{card}(H_{(k)})]$ is $\Theta(m)$.

Proof. Lemma 3 already implies that $\mathbb{E} [\text{card}(H_{(k)})] = O(m)$. Since each vertex of the hypergraph H is in at least one hyperedge of size k , $\text{card}(H_{(k)}) \geq \frac{1}{k} \text{card}(H_{(1)})$. Condition (f') ensures that $\mathbb{E} [\text{card}(H_{(1)})] \geq \gamma \text{card}(I) = \Omega(m)$, which terminates the proof. ◀

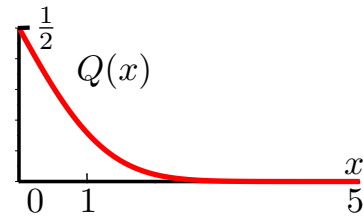
We note that the extra condition of the lemma holds for the hypergraphs that we study in this paper, where the ranges are half-spaces. Indeed, here $H_{(1)}$ is exactly the set of vertices of the convex hull of P , and every vertex belongs to at least one k -dimensional face.

2.3 Example: application to Gaussian polygons

To demonstrate how the witness-collector technique works we give a simple proof of the two-dimensional case of a classical bound on the complexity of Gaussian polytopes [7].

We first recall a few technical properties of Gaussian distributions. The Q -function is defined as the tail probability of the standard Gaussian distribution, so if $X \sim \mathcal{N}(0, 1)$

$$\forall x \in \mathbb{R}, Q(x) = \mathbb{P} [X > x] = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{t^2}{2}} dt$$



We use the following upper and lower bounds:

► **Claim 5.** *For $x > 0$,*

$$\frac{x}{1+x^2} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} < Q(x) < \frac{1}{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} .$$

Proof. The upper bound comes from

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt < \int_x^\infty \frac{t}{\sqrt{2\pi}x} e^{-\frac{t^2}{2}} dt = \int_{\frac{x^2}{2}}^\infty \frac{e^{-t}}{x\sqrt{2\pi}} dt = \frac{1}{\sqrt{2\pi}x} e^{-\frac{x^2}{2}}$$

and the lower bound comes from the fact that

$$\left(1 + \frac{1}{x^2}\right) Q(x) = \int_x^\infty \left(1 + \frac{1}{x^2}\right) \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt > \int_x^\infty \left(1 + \frac{1}{t^2}\right) \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt = \frac{e^{-\frac{x^2}{2}}}{x\sqrt{2\pi}}.$$



We use the so-called Lambert function \mathcal{W}_0 defined as the solution of the functional equation $f(x)e^{f(x)} = x$ [1, Equation (3.1)]. Let us emphasize that for $x \geq 0$ its definition is non-ambiguous and satisfies [1, Equations (4.6) and (4.9)]

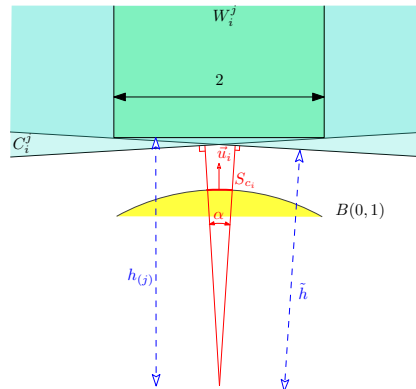
$$\mathcal{W}_0(x) = \log(x) - \log \log(x) + o(1) \tag{1}$$

We can now prove the announced bound:

► **Theorem 6.** *Let $P = \{p_1, p_2, \dots, p_n\}$ be n points i.i.d in \mathbb{R}^2 , $p_i \sim \mathcal{N}(0, I_2)$. For any fixed k , the expected number of k -dimensional faces of the convex hull of P is $\Theta(\sqrt{\log n})$.*

Proof. We break the set of half-planes R into smaller range spaces R_1, \dots, R_m by covering the space of directions, seen as the unit circle $\partial B(0, 1)$, by circular arcs S_{c_1}, \dots, S_{c_m} of angle $\alpha = \Theta\left(\frac{1}{\sqrt{\log n}}\right)$ and inner normals $\vec{u}_1, \dots, \vec{u}_m$. We have $m = \Theta\left(\frac{1}{\alpha}\right)$ arcs for the cover.

We construct each witness as a semi-infinite strip with inner direction \vec{u}_i (see the green region in the figure below). For $i \leq m$ and $j \leq \log^2 n$, the witness W_i^j is defined as the set of points $p = x\vec{v}_i + y\vec{u}_i$ (where (\vec{v}_i, \vec{u}_i) is an orthonormal basis) such that $|x| \leq 1$ and $y > h_{(j)}$, where $h_{(j)} = \sqrt{\mathcal{W}_0\left(\frac{n^2}{j^2}\right)}$ is called the *height* of the witness. The collector C_i^j is defined as the union of the half-planes in R_i that do not contain W_i^j (see the blue region in the figure on the right), so that Condition (c'), (d') and (e') hold.



Every point $p \in P$ writes $p = x_i\vec{v}_i + y_i\vec{u}_i$ with $x_i, y_i \sim \mathcal{N}(0, 1)$ independent. Thus, the probability for p to be in W_i^j is $\mathbb{P}[y_i > h_{(j)}] \mathbb{P}[|x_i| < 1] = \Theta(Q(h_{(j)}))$.

Claim 5 yields $Q(x) \sim_{x \rightarrow \infty} \frac{1}{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$, so

$$Q(h_{(j)}) = Q\left(\sqrt{\mathcal{W}_0\left(\frac{n^2}{j^2}\right)}\right) \sim \frac{1}{\sqrt{2\pi} \sqrt{\mathcal{W}_0\left(\frac{n^2}{j^2}\right)}} e^{-\frac{1}{2} \mathcal{W}_0\left(\frac{n^2}{j^2}\right)} = \frac{1}{\sqrt{2\pi} \sqrt{\mathcal{W}_0\left(\frac{n^2}{j^2}\right) e^{\mathcal{W}_0\left(\frac{n^2}{j^2}\right)}}} = \frac{j}{n\sqrt{2\pi}}$$

and $\mathbb{E}[\text{card}(W_i^j \cap P)] = n\Theta(Q(h_{(j)})) = \Theta(j)$. Condition (a') therefore holds.

To compute the expected number of points in C_i^j , we just compute the expected number of points in one of the extreme half-planes, see the figure above. The height of the left-hand

half-plane is $\tilde{h} = (h_{(j)} - \tan(\frac{\alpha}{2})) \cos(\frac{\alpha}{2}) = h_{(j)} - O(\alpha)$, so the expected number of points in the collector is bounded by $2nQ(\tilde{h})$ and using Equation (1),

$$\begin{aligned} nQ(\tilde{h}) &\leq nQ(h_{(j)}) \frac{Q(\tilde{h})}{Q(h_{(j)})} = jO\left(\frac{e^{-\frac{1}{2}(\tilde{h}^2 - h_{(j)}^2)} h_{(j)}}{\tilde{h}}\right) \\ &= je^{O(h_{(j)}\alpha)} = je^{O(\sqrt{\log \frac{n}{j}}\alpha)} = O(j) \end{aligned}$$

so we obtain $\mathbb{E}[\text{card}(C_i^j \cap P)] = O(j)$. So Condition (b') also holds and Lemma 3 ensures that $\mathbb{E}[\text{card}(H_{(k)})]$ is $O(m) = O(\sqrt{\log n})$.

For the lower bound, observe that for n large enough W_i^1 is inside a wedge of angle $O(\alpha)$ from the origin so a constant fraction of the W_i^1 are disjoint. Moreover, we have $\mathbb{P}[W_i^1 \cap H_{(1)} \neq \emptyset] = 1 - \mathbb{P}[W_i^1 \cap P = \emptyset]$. Since $\mathbb{E}[\text{card}(W_i^1 \cap P)] = \Theta(1)$, Chernoff's bound ensures that for any $0 < \beta < 1$ we have

$$\mathbb{P}[W_i^1 \cap P = \emptyset] \leq \mathbb{P}[\text{card}(W_i^1 \cap P) \leq \beta] \leq e^{-\frac{(1-\beta)^2 \Theta(1)}{2}},$$

so $\mathbb{P}[W_i^1 \cap H_{(1)} \neq \emptyset]$ is bounded from below by a positive constant. Condition (f') of Lemma 4 is thus verified and, by Lemma 4, $\mathbb{E}[\text{card}(H_{(k)})]$ is also $\Omega(m) = \Omega(\sqrt{\log n})$. \blacktriangleleft

3 A smoothed complexity bound for ℓ^2 perturbations

Let $K_x \subseteq \mathbb{R}^d$ denote the ball of radius x centered at the origin. We define the *intersection depth* of a half-space W and a ball $B(p, \rho)$ with center p and radius ρ as $\rho - d(p, W)$.

Let P^* be a set of n points, chosen arbitrarily in K_1 and let P be a random perturbation of P^* obtained by applying to each point, independently, a ℓ^2 perturbation of amplitude δ . We let H denote the geometric hypergraph induced on P by the set R of half-spaces in \mathbb{R}^d . Using the witness-collector technique we prove the following smoothed complexity bound:

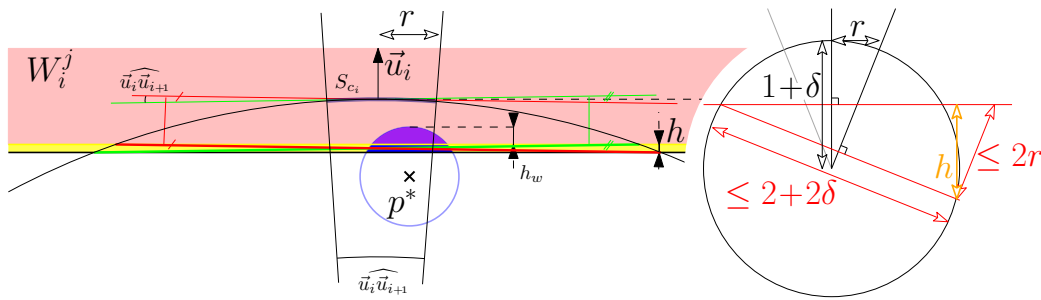
► **Theorem 7.** For any fixed $k \geq 1$, $\mathbb{E}[\text{card}(H_{(k)})] = O\left(n^{2 - \frac{4}{d+1}} \left(1 + \frac{1}{\delta}\right)^{d-1}\right)$

The bound is asymptotic, for $n \rightarrow \infty$, and the constant hidden in the $O()$ depends on k and d , but is uniform in δ . In particular δ can be a function of n . Before we prove Theorem 7 some remarks are in order:

- In dimension 2, the bound asserts that for any input in K_1 , a ℓ^2 noise of amplitude $\delta \gg n^{-1/3}$ suffices to guarantee an expected sub-linear complexity.
- In dimension 3, the bound exceeds the worst-case bound and is thus trivial.
- In dimension d , for any input in K_1 a ℓ^2 noise of amplitude $\delta \gg n^{-4/(d^2-1)}$ suffices to guarantee an expected sub-quadratic complexity.

Proof. We break up the set R of ranges. To that end, we consider a covering Sc_1, Sc_2, \dots, Sc_m of $\partial K_{1+\delta}$ by m spherical caps of radius $r = \delta n^{-\frac{2}{d+1}}$; a minimal-size covering uses $m = O\left(n^{2 - \frac{4}{d+1}} \left(1 + \frac{1}{\delta}\right)^{d-1}\right)$. For $i \in \{1, 2, \dots, m\}$ we consider the set of directions of outer normals to $\partial K_{1+\delta}$ in a point of Sc_i , and let R_i denote the set of half-spaces in \mathbb{R}^d with inner normal in that set.

We next set up, for each R_i , a family $\{(C_i^j, W_i^j)\}_j$ of witness-collector pairs. Let \vec{u}_i denote the normal to $\partial K_{1+\delta}$ in the center of the cap Sc_i . Each witness W_i^j is a half-space with inner-normal \vec{u}_i whose intersection-depth with $K_{1+\delta}$ is set so that it contains on average j



■ **Figure 3** Setup for Claim 8.

points of P . Each collector C_i^j is defined as the union of half-spaces with inner direction in S_{c_i} that do not contain $W_i^j \cap K_{1+\delta}$. This construction readily satisfies Conditions (a'), (c'), (d') and (e'). Moreover, we claim that for any perturbed point $p \in P$ we have (Claim 8):

$$\mathbb{P} [p \in C_i^j] = O\left(\frac{1}{n} + \mathbb{P} [p \in W_i^j]\right)$$

This implies that $\mathbb{E} [\text{card}(C_i^j \cap P)] = O\left(\mathbb{E} [\text{card}(W_i^j \cap P)]\right) = O(j)$ and therefore that our construction also satisfies Condition (b'). The statement of the theorem then follows from Lemma 3. ◀

► **Claim 8.** $\mathbb{P} [p \in C_i^j] = O\left(\frac{1}{n} + \mathbb{P} [p \in W_i^j]\right)$ for any perturbed point $p \in P$.

Proof. Let $p^* \in P^*$ and p its perturbed copy. We fix some indices $1 \leq i \leq m$ and $1 \leq j \leq \lceil \log^2 n \rceil$ and write $w = \mathbb{P} [p \in W_i^j]$ and $c = \mathbb{P} [p \in C_i^j]$. Let ν denote the volume of a $(d-1)$ -dimensional ball of radius 1. The volume of the intersection of a ball of radius δ with a halfspace that cuts it with depth t is

$$f(t, \delta) = \nu \int_0^t (2x\delta - x^2)^{\frac{d-1}{2}} dx$$

(In particular $f(2\delta, \delta) = \text{Vol}(K_\delta)$.) Note that $t \mapsto f(t, \delta)$ is increasing on $[0, 2\delta]$ for any fixed δ . Moreover, for $0 < t \leq \lambda t \leq 2\delta$ we have:

$$f(\lambda t, \delta) = \nu \int_0^t \lambda^{\frac{d-1}{2}} (2x\delta - \lambda x^2)^{\frac{d-1}{2}} \lambda dx \leq \nu \int_0^t \lambda^{\frac{d+1}{2}} (2x\delta - x^2)^{\frac{d-1}{2}} dx = \lambda^{\frac{d+1}{2}} f(t, \delta)$$

Refer to Figure 3-left and let h_w denote the intersection depth at which W_i^j intersects $B(p^*, \delta)$. Observe that $C_i^j \cap P$ is contained in a half-space \tilde{C}_i^j that intersects $K_{1+\delta}$ with depth at most $h_w + h$. Since the diameter of $\tilde{C}_i^j \cap P$ is at most $2 + 2\delta$, considerations on similar triangles (see Figure 3-right) show that $h \leq 2r$. If $h_w \leq 2r$ then we obtain the first

part of the announced bound on c :

$$\begin{aligned} c &\leq \frac{f(2r+h, \delta)}{f(2\delta, \delta)} \leq \frac{f(4\delta n^{-\frac{2}{d+1}}, \delta)}{f(2\delta, \delta)} = \frac{f(4n^{-\frac{2}{d+1}}, 1)}{f(2, 1)} = \frac{1}{f(2, 1)} \int_0^{4n^{-\frac{2}{d+1}}} (2x-x^2)^{\frac{d-1}{2}} dx \\ &\leq \frac{1}{f(2, 1)} \int_0^{4n^{-\frac{2}{d+1}}} (2x)^{\frac{d-1}{2}} dx = O\left(\frac{1}{n}\right). \end{aligned}$$

If $h_w > 2r$ then we can assume that $c > 2w$, as otherwise the claim holds trivially. In particular $h_w \leq \delta$. For n large enough (independently of δ), we also have $h < \delta$ and the depths of intersection of both W_i^j and \tilde{C}_i^j are in the interval $[0, 2\delta]$. We then have

$$c \leq \frac{f(h_w+h, \delta)}{f(2\delta, \delta)} = \frac{f\left(\left(1+\frac{h}{h_w}\right)h_w, \delta\right)}{f(2\delta, \delta)} \leq \left(1+\frac{h}{h_w}\right)^{\frac{d+1}{2}} w \leq 2^{\frac{d+1}{2}} w,$$

the last inequality coming from $h_w > 2r \geq h$. \blacktriangleleft

In two dimensions, this bound can be combined with the rescaling argument of Damerow and Sohler:

► **Corollary 9.** *For $d = 2$, the smoothed complexity of the convex hull of n points placed in the unit disk and perturbed by a ℓ^2 noise of amplitude δ is $O((1 + \delta^{-2/3})n^{2/3})$.*

(This bound implies that in dimension 2, for any input in K_1 , an ℓ^2 noise of amplitude $\delta \gg n^{-1/2}$ suffices to guarantee an expected sub-linear complexity, improving on Theorem 7.)

Proof. We cover K_1 , which contains the initial points, by $\Theta(1/r^2)$ cells of size r . Fix some ordering on these cells and let P_i denote the set of perturbed points whose unperturbed points were initially in the i th cell. We can bound the number of vertices of the convex hull of the perturbed point set by the sum of the number of points on the convex hulls of each of the P_i 's.

So let n_i denote the number of initial points contained in the i th cell. We apply the previous bound, noting that the scale of the initial point set was multiplied by r ; since the combinatorial structure of the convex hull is unchanged by scaling, this is equivalent to multiplying the scale of the noise by $1/r$. The expected number of vertices on the convex hull of P_i is therefore

$$\mathbb{E}[\text{card}(CH(P_i))] = O\left(n_i^{\frac{2}{3}}\left(1 + \frac{r}{\delta}\right)\right)$$

Summing over all cells we get

$$\mathbb{E}[\text{card}(CH(P))] = O\left(\left(1 + \frac{r}{\delta}\right) \sum_{i=1}^{O(r^{-2})} n_i^{\frac{2}{3}}\right)$$

Recall that the n_i sum to n . Using the concavity of $x \mapsto x^{\frac{2}{3}}$, we have

$$\sum_{i=1}^{O(r^{-2})} n_i^{\frac{2}{3}} = O\left(r^{-2}(r^2n)^{2/3}\right) = O\left((n/r)^{2/3}\right)$$

For $\delta \geq 1$ we use the bound $O(n^{2/3})$ from Theorem 7. For $\delta < 1$ we use the previous bound with $r = \delta$. Altogether we obtain that the expected number of vertices of $CH(P)$ is $O((1 + \delta^{-2/3})n^{2/3})$. \blacktriangleleft

4 Perturbing a convex polyhedron by a uniform ℓ^2 noise

We now turn our attention to a class of configurations that are natural candidates to maximize the smoothed complexity of convex hulls in 2 and 3 dimensions. Recall that an (ε, κ) -sample of a surface is a point set such that any ball of radius ε centered on the surface contains between 1 and κ points of the set.

► **Theorem 10.** *Let $P^* = \{p_i^* : 1 \leq i \leq n\}$ be an $(\Theta(n^{\frac{1}{1-d}}), \Theta(1))$ -sample of the unit sphere in \mathbb{R}^d and let $P = \{p_i = p_i^* + \delta x_i\}$ where x_1, x_2, \dots, x_n are independent random variables chosen uniformly in the unit ball. For any fixed k , $\mathbb{E}[\text{card}(H_{(k)})]$ is*

$$\begin{aligned} \Theta(n) \text{ if } \delta \in [0, n^{\frac{2}{1-d}}), & \quad \Theta\left(n^{\frac{d-1}{2d}} \delta^{\frac{(1-d)^2}{4d}}\right) \text{ if } \delta \in (1, n^{\frac{2}{d+1}}), \\ \Theta\left(n^{\frac{d-1}{2d}} \delta^{\frac{1-d^2}{4d}}\right) \text{ if } \delta \in (n^{\frac{2}{1-d}}, 1), & \quad \Theta\left(n^{\frac{d-1}{d+1}}\right) \text{ if } \delta \in (n^{\frac{2}{d+1}}, +\infty). \end{aligned}$$

As for Theorem 7, the bounds are asymptotic, for $n \rightarrow \infty$ and the constants hidden in the $\Theta()$ depend on k and d , but are uniform of δ . In particular, δ can be a function of n . Before we prove Theorem 10 some remarks are in order:

- The first bound merely reflects that a point remains extreme when the noise is small compared to the distance to the nearest hyperplane spanned by points in its vicinity.
- The last bound is of the order of magnitude of the expected number of $(k - 1)$ -faces in the convex hull of n random points chosen independently in a ball of radius δ ; this confirms, and quantifies, the intuition that the position of the original points no longer matters when the amplitude of the noise is really large compared to the scale of the initial configuration.
- The second and third bounds reveal that as the amplitude of the perturbation increases, the expected size of the convex hull does not vary monotonically (see Figure 2): the lowest expected complexity is achieved by applying a noise of amplitude roughly the diameter of the initial configuration.

Proof. Let h be the maximal depth at which a half-space containing k points of P on average intersects $K_{1+\delta}$; such a half-space intersects $\partial K_{1+\delta}$ in a spherical cap of radius $r = \sqrt{2(1+\delta)h - h^2}$ which is $\Theta(\sqrt{(1+\delta)h})$ since $h \leq 1 + \delta$. We break up R in smaller range spaces R_1, R_2, \dots, R_m by covering $\partial K_{1+\delta}$ by spherical caps Sc_1, Sc_2, \dots, Sc_m of radius r , and letting R_i stand for the set of half-spaces in \mathbb{R}^d with inner normal in Sc_i . We need and can take $m = \Theta\left(\left(\frac{1+\delta}{r}\right)^{d-1}\right) = \Theta\left(\left(\frac{1+\delta}{h}\right)^{\frac{d-1}{2}}\right)$.

Let \vec{u}_i denote the normal to $\partial K_{1+\delta}$ in the center of the cap Sc_i . For $j = 1, 2, \dots, \log^2 n$ we define W_i^j as the half-space with inner normal \vec{u}_i and containing on average j points of P . We let C_i^j be the union of half-spaces of R_i that do not contain $W_i^j \cap K_{1+\delta}$. As defined, these pairs of witness-collectors satisfy Conditions (a'), (c'), (d') and (e') of Lemma 3.

First we remark that it is easy to extract from the W_i^1 a family of size $\Omega(m)$ such that the $W_i^1 \cap P$ are disjoint, since $W_i^1 \cap K_{1+\delta}$ is seen from the origin with an angle $\Theta(\frac{1}{m})$. Second, the extremal point in direction \vec{u}_i is in W_i^1 as soon as W_i^1 is non empty. Thus we have $\mathbb{P}[W_i^1 \cap H_{(1)} \neq \emptyset] = 1 - \mathbb{P}[W_i^1 \cap P = \emptyset]$. Since $\mathbb{E}[\text{card}(W_i^1 \cap P)] = 1$, Chernoff's bound ensures that for any $0 < \beta < 1$ we have

$$\mathbb{P}[W_i^1 \cap P = \emptyset] \leq \mathbb{P}[\text{card}(W_i^1 \cap P) \leq \beta] \leq e^{-\frac{(1-\beta)^2}{2}} \leq e^{-\frac{1}{2}} \leq 0.61,$$

so $\mathbb{P}[W_i^1 \cap H_{(1)} \neq \emptyset] \geq 0.39$ and Condition (f') of Lemma 4 is verified.

We claim that $C_i^j \cap K_{1+\delta}$ is contained in the half-space D_i^j with inner normal \vec{u}_i and cutting S_{C_i} in a cap of radius $3r_i^j$, where r_i^j denotes the radius of the cap $W_i^j \cap \partial K_{1+\delta}$. Indeed, for any half-space X , the region $X \cap K_{1+\delta}$ is the convex hull of $X \cap \partial K_{1+\delta}$. It follows that $X \in R_i$ does not contain W_i^j if and only if $X \cap \partial K_{1+\delta}$ does not contain $W_i^j \cap \partial K_{1+\delta}$. This implies that for any $X \in R_i$ the cap $X \cap \partial K_{1+\delta}$ is contained in a cap with same center as $W_i^j \cap \partial K_{1+\delta}$ and radius $3r_i^j$. A half-space cutting out a cap of radius r_x in $\partial K_{1+\delta}$ intersects $K_{1+\delta}$ with depth $h_x = \Theta\left(\frac{r_x^2}{1+\delta}\right)$. Tripling the radius of a cap thus multiplies the depth of intersection by 9. Claim 12 then implies that

$$\mathbb{E} \left[\text{card}(C_i^j \cap P) \right] \leq \mathbb{E} \left[\text{card}(D_i^j \cap P) \right] = O \left(\mathbb{E} \left[\text{card}(W_i^j \cap P) \right] \right) = O(j).$$

By Lemmas 3 and 4 we thus have

$$\mathbb{E} \left[\text{card}(H_{(k)}) \right] = \Theta(m) = \Theta \left(\left(\frac{1+\delta}{h} \right)^{\frac{d-1}{2}} \right).$$

The expressions for the various ranges of δ are then obtained by plugging the expressions for h obtained from Claim 11. \blacktriangleleft

► **Claim 11.** A half-space W such that $\mathbb{E}[\text{card}(W \cap P)] = k$ intersects $K_{1+\delta}$ with depth

$$\begin{aligned} \Theta \left(n^{\frac{2}{1-d}} \right) & \text{ if } \delta \in [0, n^{\frac{2}{1-d}}), & \Theta \left(\delta^{\frac{d+1}{2d}} n^{-\frac{1}{d}} \right) & \text{ if } \delta \in (n^{\frac{2}{1-d}}, n^{\frac{2}{d+1}}), \\ \Theta \left(\delta n^{-\frac{2}{d+1}} \right) & \text{ if } \delta \geq n^{\frac{2}{d+1}}. \end{aligned}$$

Proof. The set of points in ∂K_1 at which we can center a ball of radius δ that intersects $W \cap \partial K_{1+\delta}$ is a spherical cap of radius $\sqrt{2h-h^2} = \Theta(\sqrt{h})$ and $(d-1)$ -dimensional area $\Theta\left(h^{\frac{d-1}{2}}\right)$, if $h \rightarrow 0$, and $\Theta(1)$ otherwise.

By the sampling condition, each ball of radius $n^{\frac{1}{1-d}}$ centered on ∂K_1 contains $\Theta(1)$ points of P^* . In total there are thus $\Theta\left(nh^{\frac{d-1}{2}}\right)$ points $p^* \in P^*$ such that $(p^* + K_\delta) \cap W \neq \emptyset$ if $h \rightarrow 0$, and $\Theta(n)$ otherwise. For the rest of this proof call such a point *relevant*. How much a relevant point contributes to $\mathbb{E}[\text{card}(W \cap P)]$ depends on the magnitude of δ :

If $\delta \leq n^{\frac{2}{1-d}}$ then for at least a constant fraction (depending only on d) of the relevant points p^* , the ball $B(p^*, \delta)$ is contained in W . It follows that $\Theta\left(nh^{\frac{d-1}{2}}\right) \leq k$ and $h = \Theta\left(n^{\frac{2}{d+1}}\right)$.

If $n^{\frac{2}{1-d}} \leq \delta \leq n^{\frac{2}{d+1}}$ then each relevant point p^* contributes at most

$$\frac{\text{Vol}(W \cap (p^* + \delta K))}{\text{Vol}(\delta K)} = O \left(\frac{h(\delta h)^{\frac{d-1}{2}}}{\delta^d} \right) = O \left(\left(\frac{h}{\delta} \right)^{\frac{d+1}{2}} \right)$$

and, again, at least a constant fraction (depending only on d) of the relevant points contribute at least a fraction of that. It follows that $\Theta\left(nh^{\frac{d-1}{2}} \left(\frac{h}{\delta}\right)^{\frac{d+1}{2}}\right) \leq k$. This simplifies into $h = \Theta\left(\delta^{\frac{d+1}{2d}} n^{-\frac{1}{d}}\right)$.

If $\delta \geq n^{\frac{2}{d+1}}$ then again each relevant point p^* contributes $\Theta\left(\left(\frac{h}{\delta}\right)^{\frac{d+1}{2}}\right)$, and the number of relevant points is $\Theta\left(\min\left(\left(nh^{\frac{d-1}{2}}\right), n\right)\right)$. Assuming that the minimum is realized as $\left(nh^{\frac{d-1}{2}}\right)$ yields to $h = \Theta\left(\delta^{\frac{d+1}{2d}} n^{-\frac{1}{d}}\right) \geq \Theta(1)$ meaning that W touches a linear number of $B(p_i^*, \delta)$ and a linear number of points are relevant. Thus, the number of relevant points is $\Theta(n)$, and this gives $h = \Theta\left(n^{-\frac{2}{d+1}} \delta\right)$. \blacktriangleleft

► **Claim 12.** Let W and W' be two half-spaces that intersect $K_{1+\delta}$ with depth h and $9h$ respectively, then $\mathbb{E}[\text{card}(W' \cap P)] = O(\mathbb{E}[\text{card}(W \cap P)])$.

Proof. The proof of Claim 11 shows that the number of points, in all the cases, depends on a polynomial of h . Thus, multiplying the depth by 9 multiplies the expected number of points by a constant (depending only on d). ◀

5 Gaussian perturbation of a polygon

We now investigate the same class of configurations as in Section 4, replacing the uniform ℓ^2 noise by a Gaussian noise. Since the calculations are more involved we only consider the two-dimensional case. Our result is the following:

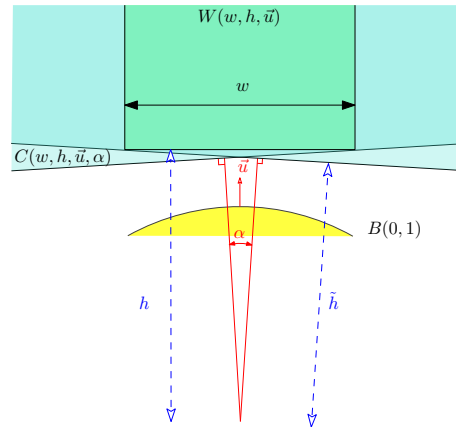
► **Theorem 13.** Let $P^* = \{p_i^*, 0 \leq i < n\}$ be a regular n -gon of radius 1 in \mathbb{R}^2 and let $P = \{p_i = p_i^* + x_i\}$ where x_1, x_2, \dots, x_n are independent Gaussian variables distributed according to $\mathcal{N}(0, \sigma^2 I_2)$. The expected size of the convex hull of P is

$$O(n) \text{ if } \sigma \in \left[0, \frac{\log^4 n}{n^2}\right], \quad O\left(\frac{\sqrt[4]{\log(n\sqrt{\sigma})}}{\sqrt{\sigma}}\right) \text{ if } \sigma \in \left[\frac{\log^4 n}{n^2}, \frac{1}{\sqrt{\log n}}\right],$$

$$O(\sqrt{\log n}) \text{ if } \sigma \in \left[\frac{1}{\sqrt{\log n}}, +\infty\right).$$

(Here also the bound is asymptotic, for $n \rightarrow \infty$, and the constant hidden in the $O()$ depends on k and d , but is uniform in σ . In particular σ can be a function of n .)

Preliminary computations. We proceed as in the proof of Theorem 6 but the tuning of the parameters is more tedious. We decompose the set of half-planes into subsets with normals in a given circle arc of angle α . We let $W(w, h, \vec{u})$ denote the semi-infinite strip with width w and bounded by a half-plane with inner-normal \vec{u} and distance h away from the origin (cf the picture on the right). We also let $C(w, h, \vec{u}, \alpha)$ denote the union of the half-planes with inner normal making an angle at most $\alpha/2$ with \vec{u} and that do not contain $W(w, h, \vec{u})$.



The witnesses and collectors can be adjusted using the following calculations:

► **Lemma 14.** In each of the following situations we have $\mathbb{E}[\text{card}(W(w, h, \vec{u}) \cap P)] = \Theta(j)$ and $\mathbb{E}[\text{card}(C(w, h, \vec{u}, \alpha) \cap P)] = \Theta(\mathbb{E}[\text{card}(W(w, h, \vec{u}) \cap P)])$:

(i) $\frac{j^2 e}{n^2} < \sigma < \sqrt{\mathcal{W}_0\left(\frac{n^2}{j^2}\right)}$, $w = 2\sigma\left(1 + \frac{1}{\sqrt{h-1}}\right)$, $h = 1 + \sigma\sqrt{\frac{3}{2}\mathcal{W}_0\left(\frac{2}{3}\left(\frac{n\sqrt{\sigma}}{j}\right)^{\frac{4}{3}}\right)}$,

and $\alpha = \frac{\sigma}{g+\sqrt{g}}$ with $g = \sigma\sqrt{\frac{3}{2}\mathcal{W}_0\left(\frac{2}{3}(n\sqrt{\sigma})^{\frac{4}{3}}\right)}$, or

(ii) $\sigma \geq \sqrt{\mathcal{W}_0\left(\frac{n^2}{j^2}\right)}$, $w = 2(1+\sigma)$, $h = 1 + \sigma\sqrt{\mathcal{W}_0\left(\frac{n^2}{j^2}\right)}$ and $\alpha = \frac{\sigma}{g+\sqrt{g}}$ with $g = \sigma\sqrt{\mathcal{W}_0(n^2)}$.

Sketch of proof. The witness $W(w, h, \vec{u})$ is a semi-infinite vertical strip and the collector $C(w, h, \vec{u}, \alpha)$ is the union of two half-planes of height \tilde{h} . The proof proceeds by considering two horizontal half-planes H and \tilde{H} at, respectively, distance h and \tilde{h} away from the origin.

Given σ , we set h so that $\text{card}(H \cap P)$ is on average $\Theta(j)$. We call the highest points of P^* *W-relevant*, the threshold being set so that relevant points still contribute $\Theta(j)$ points on average to $\text{card}(H \cap P)$. Letting w_0 denote the width of a vertical strip covering the relevant points of P^* , we set the width w to be $w_0 + \sigma$ so that these relevant points contribute $\Theta(j)$ to $\text{card}(W(w, h, \vec{u}) \cap P)$.

Let q^* be the point with normal \vec{u} on K_1 and $q = q^* + x$ with x distributed according to $\mathcal{N}(0, \sigma^2 I_2)$. We call a point $p^* \in P^*$ *C-relevant* if $\mathbb{P}[p \in \tilde{H}] \geq \mathbb{P}[q \in H]$. We bound from above the contribution of C-irrelevant points to \tilde{H} by some constant times the contribution of all points to the half-plane H . It follows that $\mathbb{E}[\text{card}(P \cap W)] \simeq 2\mathbb{E}[\text{card}(P \cap \tilde{H})] = O(j)$. It remains to tune w and α to obtain the value of \tilde{h} giving the correct amount of relevant points. ◀

Proof of Theorem 13. We cover the space of directions S^1 , envisioned as the unit circle $\partial B(0, 1)$, by circular arcs Sc_1, Sc_2, \dots, Sc_m . Each circle arc Sc_i has center \vec{u}_i and makes an angle $\alpha = \Theta(\frac{1}{m})$ that depends on σ and n . We break up R in smaller range spaces R_1, R_2, \dots, R_m where R_i denotes the set of half-planes with inner normal in Sc_i . We define the witnesses $(W_i^j)_{1 \leq j \leq \log^2 n}$ and the collectors $(C_i^j)_{1 \leq j \leq \log^2 n}$ with the usual goals in mind: W_i^j should have inner normal \vec{u}_i and contain $\Theta(j)$ points on average, and C_i^j is defined as the union of the half-spaces in R_i that do not contain W_i^j .

We first use Lemma 14 to find suitable values of h_j and w_j , that depend on σ and n , such that we can set $W_i^j = W(w_j, h_j, \vec{u}_i)$. We then get, again from Lemma 14, a suitable value of α that ensures that setting $C_i^j = C(w_j, h_j, \vec{u}_i, \alpha)$ satisfies our objectives. This family of witness-collectors satisfies Conditions (a')–(e') so Lemma 3 yields that $\mathbb{E}[CH(P)] = O(\frac{1}{\alpha})$.

We now split the range of σ according to the conditions of Lemma 14 where we set $j = \log^2 n$. Using $\mathcal{W}_0(x) \sim_{x \rightarrow \infty} \log x$ we obtain three regimes:

$$I_1 = \left[0, \frac{\log^4 n}{n^2}\right], \quad I_2 = \left[\frac{\log^4 n}{n^2}, \sqrt{\log n}\right] \text{ and } I_3 = \left[\sqrt{\log n}, +\infty\right].$$

We further split I_2 by observing that for $\sigma \approx \frac{\log^4 n}{n^2}$ the behaviour of $\frac{1}{\alpha}$ is dominated by $\frac{\sqrt{g}}{\sigma} = O\left(\frac{\sqrt[4]{\log(n\sqrt{\sigma})}}{\sqrt{\sigma}}\right)$ whereas for $\sigma \approx \sqrt{\log n}$ it is dominated by $\frac{g}{\sigma} = O(\sqrt{\log n})$. (Inside I_3 , $\frac{1}{\alpha}$ is always dominated by $\frac{g}{\sigma}$.) The switch occurs around the solution $\sigma_0(n)$ of $g = \sqrt{g}$, which solves into $\sigma_0(n) = \Theta\left(\frac{1}{\sqrt{\log n}}\right)$. The upper end of I_2 yields the same behaviour as I_3 , so we merge them to obtain the three regimes of Theorem 13. ◀

References

- 1 Robert M. Corless, Gaston H. Gonnet, D.E.G. Hare, David J. Jeffrey, and Donald E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, 1996. doi:10.1007/BF02124750.
- 2 Valentina Damerow and Christian Sohler. Extreme points under random noise. In *Proc. 12th European Sympos. Algorithms*, pages 264–274, 2004. doi:10.1007/978-3-540-30140-0_25.

- 3 Mark de Berg, Herman Haverkort, and Constantinos P. Tsirogiannis. Visibility maps of realistic terrains have linear smoothed complexity. *Journal of Computational Geometry*, 1:57–71, 2010. [url:jocg.org/index.php/jocg/article/view/12](http://jocg.org/index.php/jocg/article/view/12).
- 4 Olivier Devillers, Marc Glisse, and Xavier Goaoc. Complexity analysis of random geometric structures made simpler. In *Symposium on Computational Geometry*, pages 167–176, 2013. doi:10.1145/2462356.2462362.
- 5 Marc Glisse, Sylvain Lazard, Julien Michel, and Marc Pouget. Silhouette of a random polytope. Research Report 8327, INRIA, 2013. [url:hal.inria.fr/hal-00841374/](http://hal.inria.fr/hal-00841374/).
- 6 Matthias Reitzner. Random polytopes. In *New perspectives in stochastic geometry*, pages 45–76. Oxford Univ. Press, Oxford, 2010.
- 7 Alfréd Rényi and Rolf Sulanke. Über die konvexe Hülle von n zufällig gewählten Punkten I. *Z. Wahrsch. Verw. Gebiete*, 2:75–84, 1963. doi:10.1007/BF00535300.
- 8 Alfréd Rényi and Rolf Sulanke. Über die konvexe Hülle von n zufällig gewählten Punkten II. *Z. Wahrsch. Verw. Gebiete*, 3:138–147, 1964. doi:10.1007/BF00535973.
- 9 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51:385–463, 2004. doi:10.1145/990308.990310.

Finding All Maximal Subsequences with Hereditary Properties

Drago Bokal¹, Sergio Cabello^{*2}, and David Eppstein^{†3}

- 1 Faculty of Natural Sciences and Mathematics, University of Maribor, Slovenia
- 2 Department of Mathematics, FMF, University of Ljubljana, Slovenia
- 3 Computer Science Department, University of California, Irvine, USA

Abstract

Consider a sequence s_1, \dots, s_n of points in the plane. We want to find all maximal subsequences with a given hereditary property \mathcal{P} : find for all indices i the largest index $j^*(i)$ such that $s_i, \dots, s_{j^*(i)}$ has property \mathcal{P} . We provide a general methodology that leads to the following specific results:

- In $O(n \log^2 n)$ time we can find all maximal subsequences with diameter at most 1.
- In $O(n \log n \log \log n)$ time we can find all maximal subsequences whose convex hull has area at most 1.
- In $O(n)$ time we can find all maximal subsequences that define monotone paths in some (subpath-dependent) direction.

The same methodology works for graph planarity, as follows. Consider a sequence of edges e_1, \dots, e_n over a vertex set V . In $O(n \log n)$ time we can find, for all indices i , the largest index $j^*(i)$ such that $(V, \{e_i, \dots, e_{j^*(i)}\})$ is planar.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases convex hull, diameter, monotone path, sequence of points, trajectory

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.240

1 Introduction

The increasing availability of massive amounts of data regarding the spatial movements of smart phones, vehicles, tagged wild animals, ice sheets, etc., has led to an increasing interest in geometric algorithms for *trajectory analysis* [1, 4–6, 9, 13, 16, 17]. Such problems are a natural fit for the *windowed geometry* framework of Bannister et al. [2]: in this framework, a trajectory can be described by a sequence S of points in the plane (the vertices of a polyline), and we wish to develop data structures that can quickly answer queries about the shapes formed by contiguous subsequences of S . These queries may in turn be used for exploratory data analysis of a data set, or as subroutines for higher-level problems such as trajectory segmentation, clustering, or simplification.

In this paper, we consider queries for which the answer is a Boolean value: given a sequence $S = s_1, \dots, s_n$, and a query subsequence $[i, j]$, does the queried subsequence of S have property \mathcal{P} or not? We only consider hereditary properties, i.e., whenever a sequence has property \mathcal{P} , so do all of its subsequences. For example, the property of having a convex hull of area at most 1 is hereditary in this sense. For such problems, the issues of data

* Part of this research was done while visiting IST Austria. Supported by the Slovenian Research Agency, program P1-0297, projects J1-4106 and L7-5459.

† Supported in part by NSF grant 1228639 and ONR grant N00014-08-1-1015.



© D. Bokal, S. Cabello, and D. Eppstein;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 240–254



Leibniz International Proceedings in Informatics

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

structure representation and query time become trivial: we need only store, for each index i , the largest index $j^*(i)$ such that the subsequence $s_i, \dots, s_{j^*(i)}$ has property \mathcal{P} . With this information, the query reduces to a simple comparison of the endpoint j of the query interval with the endpoint $j^*(i)$ of the maximal interval starting at i with property \mathcal{P} . However, the preprocessing stage of this problem, in which we compute each of these values $j^*(i)$, can be highly nontrivial. That is the focus of our contribution: efficient algorithms for finding all of the maximal contiguous subsequences of S with the prescribed property.

Analogous windowed query data structures can be considered as well for non-geometric data, such as sequences of timestamped graph edges [3]. For such data, we may seek the maximal subsequences that have some monotone graph property such as being disconnected, being acyclic, being planar, etc.

1.1 New results

Let $S = s_1, \dots, s_n$ be a sequence of points in the plane. We prove the following results:

- In $O(n \log n \log \log n)$ time we can find, for all indices i , the largest index $j^*(i)$ such that the convex hull of $s_i, \dots, s_{j^*(i)}$ has at most unit area. In the trajectory problems, this models subsequences in which the moving object is either not moving significantly or is traveling close to a straight line.
- In $O(n \log^2 n)$ time we can find, for all indices i , the largest index $j^*(i)$ such that $s_i, \dots, s_{j^*(i)}$ has at most unit diameter. In the trajectory problem, this models subsequences in which the object is not moving significantly.
- In $O(n)$ time we can find, for all indices i , the largest index $j^*(i)$ such that there exists a direction for which the path defined by $s_i, \dots, s_{j^*(i)}$ is monotone. In the trajectory problem, this models subsequences in which the object is moving in some particular direction but may possibly be deviating from a straight line to avoid obstacles.

We develop a methodology, explained in Section 2, that should be useful for many other problems. As another application of these techniques beyond geometry, we show the following result about graph planarity. Let V be a vertex set and let e_1, \dots, e_n be a sequence of edges with endpoints in V . We show how to compute in $O(n \log n)$ time, for all indices i , the largest index $j^*(i)$ such that the graph $(V, \{e_i, \dots, e_{j^*(i)}\})$ is planar.

For the geometric problems, we do not use any heavy machinery and the results are clearly implementable. For graph planarity we use a deep result of Galil, Italiano, and Sarnak [15] that makes our result purely of theoretical interest.

1.2 Comparison with dynamic data structures

For our problems of finding maximal subsequences with property \mathcal{P} , there is a natural alternative approach based on dynamic geometric data structures. Suppose we have a data structure D that can maintain a dynamic set of points, subject to insertions and deletions, and answer queries that ask whether the current set has property \mathcal{P} . Then we may use D to compute the sequence of values $j^*(i)$, using a simple scan, as follows:

- Augment S by a special flag value s_{n+1} that cannot be part of a set with property \mathcal{P} .
- Initialize D to an empty data structure, and set $j = 0$.
- For $i = 1, 2, 3, \dots$ do the following:
 - While the set in D has property \mathcal{P} , increase j by one and insert s_j into D .
 - Set $j^*(i) = j - 1$.
 - Delete s_i from D .

This algorithm performs n insertions and deletions in D and computes all values $j^*(i)$; its time is bounded by $O(n)$ times the time for a single insertion or deletion. However, for the problems we consider, this would be slower than the time bounds we give.

For instance, consider the problem of finding maximal subsequences of points whose convex hull has area at most 1. A natural approach is to use a dynamic data structure that maintains the area of the convex hull under insertions and deletions of points. The data structure by Overmars and van Leeuwen [20] can be easily extended to maintain the area of the convex hull of n points in $O(\log^2 n)$ time per update. We could use this data structure in the scan algorithm above to compute all maximal subsequences in $O(n \log^2 n)$; however, this is slower by a logarithmic factor than our algorithms. Chan [7] has improved the Overmars and van Leeuwen data structure but we are unsure whether this can be adapted to the convex hull area property and it would still be somewhat slower than our algorithm.

Chan [8] shows how to maintain the diameter dynamically in $O(\log^8 n)$ expected amortized time, improving a previous algorithm of Eppstein [11]. This implies that all maximal subsequences of diameter 1 can be computed in $O(n \log^8 n)$ expected time. This is significantly slower than the algorithm we give.

The monotonicity problem depends on the sequence in which the input points are given so it is not possible to express it using data structures based on dynamic point sets. Nevertheless, a similar scan algorithm could be used together with a data structure that detects whether a dynamic set of vectors (the differences of consecutive points in the input sequence) has the property of lying within a halfspace through the origin. This data structural problem can be solved by using a binary search tree (ordered radially around the origin) in logarithmic time per update. However, again, this would be slower than our algorithm.

For graph planarity we would need a dynamic data structure that maintains a planar graph under insertion and deletion of edges. Moreover, we also need to be able to query whether the addition of an edge violates planarity. The best data structure for this takes $O(\sqrt{|V|}) = O(\sqrt{n})$ amortized time per query or operation [12]. Thus, we can find all maximal planar graphs in $O(n^{3/2})$ time, significantly slower than our algorithm. (There are better semi-dynamic data structures for planarity [10], but deletions are costly.)

There has also been research on faster dynamic data structures with restrictions on the update order, such as offline updates in which the entire sequence of updates is known in advance. Here, we do know the order of insertions and deletions, but we do not know how they interlace. In fact, the main substance of the problem is about figuring out when deletions should take place.

1.3 Additional related work

Łącki and Sankowski [19] considered a related windowed query framework for graph problems with an offline sequence of edge updates. As in our problems, queries specify a window within this sequence; however, the goal of a query is to determine whether some, all, or none of the versions of the graph within the window have a given property \mathcal{P} . For the geometric problems that we consider, an analogous type of problem would involve a data set consisting of a sequence of point insertions and deletions, and a query asking whether all, some, or none of the versions of the point set within a window into the query sequence have a given property. However, the graph properties considered by Łącki and Sankowski are different from the geometric and graph properties considered here.

A one-dimensional variant of the windowed diameter problem may be solved in constant time per query, using a range minimum data structure [14] to determine the minimum and maximum value within a query window. Applying this separately to each coordinate would

allow us to determine the L_∞ diameter of a query window. However, this approach does not generalize to Euclidean diameter, and although it can be made to work for the monotone direction problem, it would result in a more complicated solution than the one we give.

2 General strategy

We first review the notation we (ab)use. For any natural numbers a and b , we let $[a, b]$ denote the integer range $\{a, a + 1, \dots, b\}$. Henceforth, n will be used to denote the length of the input sequence. We write $[n]$ instead of $[1, n]$ and use $\mathbb{U} = \{(i, j) \in [n]^2 \mid i \leq j\}$.

Consider a sequence $S = s_1, \dots, s_n$ of points in the plane. For every pair of indices $(i, j) \in \mathbb{U}$ we define the **subsequence** $S[i, j] = s_i, \dots, s_j$. All subsequences considered in this paper are contiguous subsequences. When $j < i$, $S[i, j]$ is the empty sequence. With a slight abuse of notation, we will sometimes treat $S[i, j]$ as a set instead of as a sequence; for example, we will talk about the diameter or the convex hull of $S[i, j]$.

A property \mathcal{P} for subsequences is **hereditary** if it is closed under taking subsequences: if $S[i, j]$ has property \mathcal{P} , then $S[i', j']$ also has property \mathcal{P} for all $i \leq i' \leq j' \leq j$. All properties considered in this paper are hereditary.

Consider a fixed hereditary property \mathcal{P} . We consider a $n \times n$ matrix $A_{\mathcal{P}} = (A_{\mathcal{P}}(i, j))_{(i, j) \in \mathbb{U}}$, defined (only for pairs of indices in \mathbb{U}) by

$$A_{\mathcal{P}}(i, j) = \begin{cases} 1, & \text{if } S[i, j] \in \mathcal{P}, \\ 0, & \text{otherwise.} \end{cases}$$

Values in the bottom triangle $\{(i, j) \mid j < i\}$ are undefined. We want to find for each row i the last index $j^*(i)$ with $A_{\mathcal{P}}(i, j^*(i)) = 1$. When the property \mathcal{P} is clear from the context, we drop the subscript and simply write A instead of $A_{\mathcal{P}}$.

A **rectangle** (of indices) is a subset of indices

$$[a, a + h] \times [b, b + w] = \{(i, j) \in [n]^2 \mid a \leq i \leq a + h, b \leq j \leq b + w\}.$$

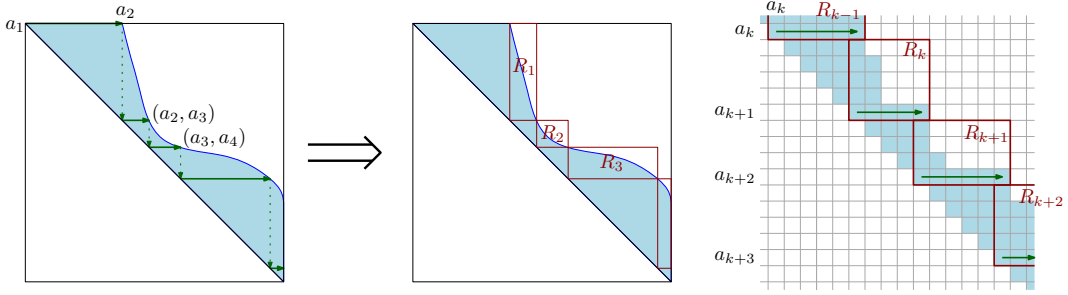
For a rectangle $R = [a, a + h] \times [b, b + w]$, its **height** is $\text{height}(R) = h + 1$ and its **width** is $\text{width}(R) = w + 1$. By **solving a rectangle** $R = [a, a + h] \times [b, b + w]$ we mean finding, for each index $i \in [a, a + h]$, the last nonzero of row i of matrix A that lies inside rectangle R . In general, our algorithms will consider rectangles $[a, a + h] \times [b, b + h]$ with $a + h \leq b$, that is, contained in \mathbb{U} . A rectangle is **anchored at the diagonal** if it is of the type $[a - h, a] \times [a, a + w]$, that is, its bottom left corner lies on the diagonal $\{(i, i) \mid i \in [n]\}$.

We will assume that $A(i, i) = 1$, for all $i \in [n]$; that is, any single point of S always satisfies property \mathcal{P} . (Otherwise $j^*(i)$ is not defined.) Our algorithms will consider rectangles $R = [a, a + h] \times [b, b + w]$ with the property that, for all $i \in [a, a + h]$, $j^*(i) \in [b, b + w]$. That is, these rectangles contain the last nonzero of A in each of their rows. We call a rectangle with this property a **frontier rectangle**. Thus, solving a frontier rectangle is equivalent to finding the values $j^*(i)$ for all $i \in [a, a + h]$.

2.1 Decomposing into anchored rectangles

We are going to use a greedy procedure to reduce the problem to a search within disjoint frontier rectangles anchored at the diagonal that together have $O(n)$ height and width.

Take a sequence of indices $\alpha = a_1, a_2, \dots, n$ such that $a_1 = 1$ and $a_k = j^*(a_{k-1})$, that is, a_k is the largest index with $A(a_{k-1}, a_k) = 1$. (In the special case that $a_k = a_{k-1}$, we redefine a_k to $a_{k-1} + 1$.) This is a greedy decomposition of the sequence into subsequences with



■ **Figure 1** Schema showing the greedy procedure to decompose the problem into (red) frontier rectangles anchored at the diagonal. The blue region denotes entries of matrix A with value 1.

property \mathcal{P} . The subsequences are disjoint, except for the starting and ending points a_k , and maximal with respect to this almost-disjoint property. For each index a_k in α , define the rectangle $R_k = [a_k + 1, a_{k+1}] \times [a_{k+1}, a_{k+2}]$. A schematic view is offered in Figure 1.

Each index $i \in [n]$ appears at most once as a first coordinate and at most twice in the second coordinate of rectangles R_1, R_2, \dots . Therefore

$$\sum_k (\text{height}(R_k) + \text{width}(R_k)) = O(n).$$

By construction, each R_k is a frontier rectangle anchored at the diagonal. Solving the rectangles $R_1, R_2 \dots$ we readily obtain all maximal subsequences with property \mathcal{P} . We summarize.

► **Lemma 2.1.** *Assume that we have the following two subroutines for sequence S and property \mathcal{P} :*

- (a) *Given an index $a \in [n]$, find the largest index $j^*(a)$ such that $S[a, j^*(a)] \in \mathcal{P}$. This takes $T_{\text{greedy}}(j^*(a) - a)$ time for a certain convex function $T_{\text{greedy}}(\cdot)$.*
- (b) *Given a frontier rectangle R of indices anchored at the diagonal, solve R . This takes $T_{\text{rect}}(\text{height}(R) + \text{width}(R))$ time for a certain convex function $T_{\text{rect}}(\cdot)$.*

Then we can find all maximal subsequences with property \mathcal{P} in $O(n) + T_{\text{greedy}}(O(n)) + T_{\text{rect}}(O(n))$ time. ◀

2.2 Solving an anchored rectangle

To solve a frontier rectangle anchored at the diagonal, we are going to use a recursive divide-and-conquer method. The subproblems of this method will be defined by frontier rectangles contained in \mathbb{U} , but not necessarily anchored at the diagonal.

Consider a frontier rectangle $[a, a + h] \times [b, b + w]$ contained in \mathbb{U} . We use a methodology similar to binary search. See Figure 2 for an schematic view. We select an index m halving the interval $[a, a + h]$. Then we find the largest index c such that $A(m, c) = 1$. With this, we infer the following information:

- In the rectangle $[m, a + h] \times [b, c]$, all the values of A are 1.
- In the rectangle $[a, m] \times [c + 1, b + w]$, all the values of A are 0.

We then recurse in the frontier rectangle $[a, m - 1] \times [b, c]$ and in the frontier rectangle $[m + 1, a + h] \times [c, b + w]$. Since in each step we halve the area where we continue the search, we have a recursion of depth $O(\log(w h))$.

However, the subproblems do not really get smaller: to solve the rectangle $R = [a, a + h] \times [b, b + w]$, we have to consider the subsequence $S[a, b + w]$, which has size $w + b - a$. For late

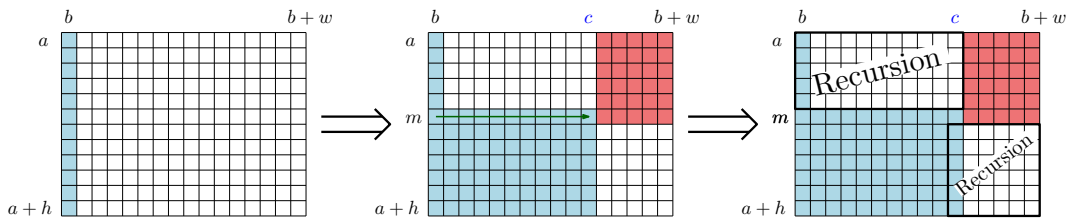


Figure 2 Schema showing the strategy to solve a frontier rectangle. The blue region denotes entries known to have value 1. The red region denotes entries known to have value 0.

subproblems, this may be larger than $w + h$ or wh . However, any of the subsequences $S[i, j]$, where $(i, j) \in R$, can be decomposed into $S[i, a + h]$, $S[a + h, b]$ and $S[b, j]$. The middle sequence $S[a + h, b]$ may be arbitrarily large, but it is a “common factor” to all subsequences $S[i, j]$, $(i, j) \in R$. We replace the subsequence $S[a + h, b]$ by a *sketch* of size $O(w + h)$. The definition of sketch depends on the problem at hand, but the idea is that it should encode the role of $S[a + h, b]$ in the subsequences $S[i, j]$, for all $(i, j) \in R$. In fact, such sketches are also important to find efficiently the index c that controls the division for recursive calls.

To analyze such algorithm, the following technical result will be useful.

► **Lemma 2.2.** *The recursion*

$$T(h, w) = \begin{cases} O(h + w) + T(\lfloor h/2 \rfloor, w') + T(\lfloor h/2 \rfloor, w - w') & \text{if } h \geq 2, \\ O(w) & \text{if } h = 0 \text{ or } 1, \end{cases}$$

where $0 \leq w' \leq w$, implies that $T(h, w) = O((h + w) \log h)$. ◀

3 Directional monotonicity

In this section, we will regard the subsequence $S[i, j]$ as a *polygonal path*. Consider the unit circle \mathbb{S}^1 . For a direction $\vec{u} \in \mathbb{S}^1$, the path $S[i, j]$ is \vec{u} -**monotone**, if it is always increasing in the direction \vec{u} , that is, the scalar product of $\vec{s}_k \vec{s}_{k+1}$ and \vec{u} is positive for each $k \in [i, j - 1]$. The path $S[i, j]$ is **monotone** if it is \vec{u} -monotone for some direction $\vec{u} \in \mathbb{S}^1$. Let $\Theta(i, j) \subset \mathbb{S}^1$ be the set of directions \vec{u} such that $S[i, j]$ is \vec{u} -monotone.

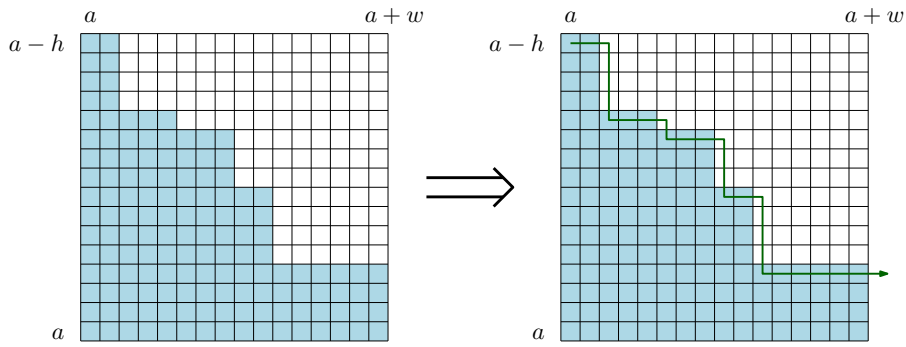
► **Lemma 3.1.** *Given an index $a \in [n]$, we can find the largest index $j^*(a) \in [n]$ such that the path $S[a, j^*(a)]$ is monotone in $O(j^*(a) - a)$ time.*

Proof. Starting with $j = a + 1$, we increment j until we get that $j = n + 1$ or $S[a, j]$ is not monotone, and then return $j - 1$. At each step, we compute the interval $\Theta(a, j)$ in constant time using that $\Theta(a, j) = \Theta(a, j - 1) \cap \{\vec{u} \in \mathbb{S}^1 \mid \langle \vec{s}_{j-1} \vec{s}_j, \vec{u} \rangle > 0\}$. ◀

Lemma 3.1 provides the subroutine needed in Lemma 2.1(a). The next result provides the subroutine needed in Lemma 2.1(b).

► **Lemma 3.2.** *Consider a frontier rectangle R of indices anchored at the diagonal. We can solve the rectangle R in $O(\text{height}(R) + \text{width}(R))$ time.*

Proof. Let R be the rectangle $[a - h, a] \times [a, a + w]$. We compute for each index $j \in [a + 1, a + w]$ the set of directions $\Theta(a, j)$. This is done incrementally in $O(w)$ time. We compute for each index $i \in [a - h, a - 1]$ the set of directions $\Theta(i, a)$. This is done by tracing the *reverse* of the path $S[a - h, a]$ and using the fact that a path is \vec{u} -monotone if and only if its reversal is $(-\vec{u})$ -monotone.



■ **Figure 3** Path followed by (i, j) in the algorithm of Lemma 3.2 to solve a frontier rectangle anchored to the diagonal. The blue region describes query windows that form monotone paths.

Now we just walk along the boundary between monotone and not-monotone in the rectangle R . See Figure 3. Set $i = a - h$ and $j = a + 1$. At each iteration, we compute $\Theta(i, j)$ in constant time using the fact that $\Theta(i, j) = \Theta(i, a) \cap \Theta(a, j)$. If $\Theta(i, j) \neq \emptyset$, we increment j and go to the next iteration. If $\Theta(i, j) = \emptyset$, we deduce that $j^*(i) = j - 1$, increment i , and go to the next iteration. We finish when the pair (i, j) is outside the rectangle R . In this case, we deduce that $j^*(i') = a + w$ for $i' = i, \dots, a$.

The running time is $O(h + w)$ because at each iteration we spend constant time and increment either i or j . ◀

► **Theorem 3.3.** *Let $S = s_1, \dots, s_n$ be a polygonal path in the plane. In $O(n)$ time we can compute, for all indices $i \in [n]$, the largest index $j^*(i)$ such that the polygonal path $s_i, \dots, s_{j^*(i)}$ is monotone.*

Proof. Lemmas 3.1 and 3.2 give the subroutines required in the hypothesis of Lemma 2.1, with running times $T_{greedy}(n) = O(n)$ and $T_{rect}(n) = O(n)$. Lemma 2.1 implies that we can find all maximal subsequences in $O(n)$ time. ◀

Note that in this problem, we did not need the recursive approach discussed in Section 2.2. This is because $\Theta(i, j)$, which plays the role of a sketch, has a constant-size description.

4 Diameter of point sets

For a set of points P in the Euclidean plane, its diameter $\text{diam}(P)$ is the maximum distance between any two points: $\text{diam}(P) = \max_{p, p' \in P} \|p - p'\|$. The diameter of n points can be computed in $O(n \log n)$ time; see for example [21, Chapter 4].

4.1 Sketches

Let P and S be sets of points in the Euclidean plane. A subset $Q \subset P$ is a *diam-sketch of P with respect to S* if

- (i) for each $T \subset S$ we have $\text{diam}(P \cup T) = \text{diam}(Q \cup T)$, and
- (ii) $|Q| = O(|S|)$.

Diam-sketches can be constructed using standard tools and have a certain composition property, as the following lemma explains. See Figure 4 for an example of the construction.

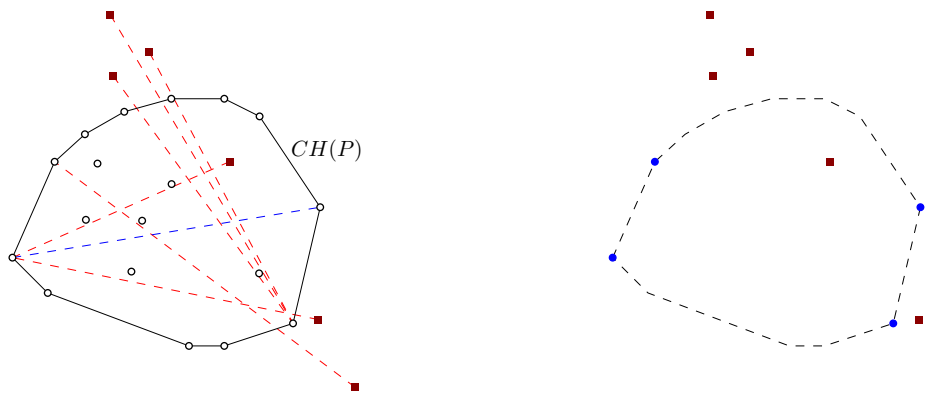


Figure 4 Example of diam-sketch. Left: the dotted black points correspond to P , the squared red points to S . A diametral pair of P and points furthest away from each point of S are shown with dashed segments. Right: the resulting diam-sketch, as constructed in the proof of Lemma 4.1(a).

► **Lemma 4.1.** *Diam-sketches have the following properties.*

- (a) *Given sets P and S of cardinality at most n , we can compute a diam-sketch of P with respect to S in $O(n \log n)$ time.*
- (b) *Let Q be a diam-sketch of P with respect to S , X be a diam-sketch of $Q \cup S_1$ with respect to S_2 , and $S_1 \cup S_2 \subset S$. Then X is a diam-sketch of $P \cup S_1$ with respect to S_2 .*

Proof Sketch. We show (a) giving an explicit construction. We set $Q = \emptyset$, add to Q a diametral pair of P and, for each point $s \in S$, add a point of P that is furthest from s . Such a set Q can be constructed in $O((|P| + |S|) \log(|P|)) = O(n \log n)$ time using standard tools: we compute the diameter of P , build the furthest-point Voronoi diagram VD of P , and locate each point of S in VD . The details are standard. See Figure 4 for an example of the construction. A small case analysis shows that Q is indeed a diam-sketch.

To show (b), consider any $T \subset S_2$. We have to show that $\text{diam}(P \cup S_1 \cup T) = \text{diam}(X \cup T)$. Since Q is a diam-sketch of P with respect to $S \supset S_1 \cup T$ and X is a diam-sketch of $Q \cup S_1$ with respect to $S_2 \supset T$, we have

$$\begin{aligned} \text{diam}(P \cup S_1 \cup T) &= \text{diam}(P \cup (S_1 \cup T)) = \text{diam}(Q \cup (S_1 \cup T)) \\ &= \text{diam}((Q \cup S_1) \cup T) = \text{diam}(X \cup T). \end{aligned} \quad \blacktriangleleft$$

4.2 Algorithms

► **Lemma 4.2.** *Let P be a set of points and $S = s_1, \dots, s_n$ a sequence of points. Assume that we have a diam-sketch Q of P with respect to S . In $O(n \log n)$ time, we can find the largest index j^* such that $\text{diam}(P \cup S[1, j^*]) \leq 1$.*

Proof. We proceed with a binary search. We initialize the search with $\ell = 0$, $r = n$, and $X = Q$. Through the binary search, we maintain the invariant that $\ell \leq j^* \leq r$ and X is a diam-sketch of $P \cup S[1, \ell]$ with respect to $S[\ell + 1, r]$. In an iteration, we set $m = \lceil (\ell + r)/2 \rceil$ and check whether $\text{diam}(X \cup S[\ell + 1, m]) \leq 1$. If the diameter is at most 1, then we continue the search with $\ell = m$, and set X to be a diam-sketch of $X \cup S[\ell, m]$ with respect to $S[m + 1, r]$. If, on the other hand, the diameter is larger than 1, then we continue the search with $r = m - 1$, and set X to be a diam-sketch of X with respect to $S[\ell + 1, m - 1]$. We finish the search when $\ell = r$ by returning ℓ .

Validity of the invariant $\ell \leq j^* \leq r$ follows from the standard argument used for binary search. Validity of the property that X is a diam-sketch of $P \cup S[1, \ell]$ with respect to $S[\ell + 1, r]$ follows by induction and Lemma 4.1(b). When we continue on the right side (setting $\ell = m$), we apply Lemma 4.1(b) with $S_1 = S[\ell, m]$ and $S_2 = S[m + 1, r]$. When we continue on the left side (setting $r = m - 1$), we apply Lemma 4.1(b) with $S_1 = \emptyset$ and $S_2 = S[\ell + 1, m - 1]$.

Correctness of the method follows from the invariant because at the end, when $r = \ell$, we have $\text{diam}(P \cup S[1, \ell]) = \text{diam}(X) \leq 1$ and $r = j^* = \ell$.

For the running time, note that at each step, we handle $O(|X| + r - \ell)$ points. Since X is always a diam-sketch with respect to $S[\ell + 1, r]$, we have $|X| = O(r - \ell)$. At each iteration, we compute the diameter of $O(r - \ell)$ points and the diam-sketch of $O(r - \ell)$ points with respect to a set of size $O(r - \ell)$ using Lemma 4.1(a). This means that we spend $O((r - \ell) \log(r - \ell))$ at each iteration. Since at each iteration the value $r - \ell$ decreases geometrically, we conclude that the total running time is $O(n \log n)$. \blacktriangleleft

► Lemma 4.3. *Consider a frontier rectangle R anchored at the diagonal with height h and width w . We can solve R in $O((h + w) \log^2(h + w))$ time.*

Proof. We give a recursive algorithm. A recursive subproblem is described by a frontier rectangle $[a, a + h] \times [b, b + w]$ contained in \mathbb{U} , and a diam-sketch Q of $S[a + h, b]$ with respect to $S[a, a + h - 1] \cup S[b + 1, b + w]$. The original problem is a problem of such type, where $a + h = b$ and $S[a + h, b] = Q = \{s_b\}$.

If $h = 1$, we use Lemma 4.2 twice, once for each row, to find $j^*(a)$ and $j^*(a + 1)$. For the row $a + 1$ we use Q and the sequence $S[b + 1, b + w]$. For the row a we use Q and the sequence $s_a, S[b + 1, b + w]$. In this case, we need $O(w \log w)$ time. The case $h = 0$ is similar.

Let us now consider the case when $h \geq 2$ and thus the rectangle has at least three rows. We use the divide-and-conquer approach discussed in Section 2.2. Set $m = a + \lfloor h/2 \rfloor$. We find the last index $c \in [b, b + w]$ such that $\text{diam}(S[m, c]) \leq 1$. (Here we are using the property of being a frontier rectangle to infer that $c \geq b$ and thus $(m, c) \in R$.) We have obtained that $j^*(m) = c$. We then recurse on the rectangles $R_1 = [a, m - 1] \times [b, c]$ and $R_2 = [m + 1, a + h] \times [c, b + w]$. Note that R_1 and R_2 are frontier rectangles; see Figure 2. To recurse in the rectangle R_1 , we use a diam-sketch Q_1 of $Q \cup S[m, a + h]$ with respect to $S[a, m - 1] \cup S[b, c]$. Note that Q_1 is a diam-sketch of $S[m, b]$ with respect to $S[a, m - 1] \cup S[b, c]$ because of Lemma 4.1(b), and thus it is the appropriate diam-sketch for the recursive call. Similarly, for the recursion on the rectangle R_2 , we use a diam-sketch Q_2 of $Q \cup S[b, c - 1]$ with respect to $S[m + 1, a + h] \cup S[c, b + w]$. Again, Q_2 is a diam-sketch of $S[a + h, c - 1]$ with respect to $S[m + 1, a + h] \cup S[c, b + w]$ because of Lemma 4.1(b), and it provides appropriate ground for recursion. This finishes the description of the algorithm.

To analyze the running time, note that Q has size $O(h + w)$ because it is a diam-sketch with respect to $h + w$ points. If $h \leq 1$, we spend $O(w \log w)$ time. Let us now look at the case $h > 1$. The index c can be found in $O((h + w) \log(h + w))$ time using Lemma 4.2 with Q and the sequence $S[m, a + h - 1], S[b + 1, b + w]$. The sets Q_1 and Q_2 can be computed in $O((h + w) \log(h + w))$ time using Lemma 4.1(a) and noting that Q has size $O(h + w)$. Thus we spend $O((h + w) \log(h + w))$ time plus the time for recursive calls in R_1 and R_2 . Let $w' = c - b$. The rectangle R_1 has $m - a \leq \lfloor h/2 \rfloor$ rows and $c - b + 1 = w' + 1$ columns, while the rectangle R_2 has $a + h - m \leq \lfloor h/2 \rfloor + 1$ rows and $b + w - c + 1 = w - w' + 1$ columns. Therefore, denoting by $T(h, w)$ the running time for a rectangle with $h + 1$ rows and $w + 1$

columns, we have

$$T(h, w) = \begin{cases} O((h + w) \log(h + w)) + T(\lfloor h/2 \rfloor, w') + T(\lfloor h/2 \rfloor, w - w') & \text{if } h \geq 2, \\ O(w \log(h + w)) & \text{if } h \leq 1. \end{cases}$$

Taking the factor $O(\log(h + w))$ out, Lemma 2.2 implies that $T(h, w) = O((h + w) \log^2(h + w))$. ◀

Buchin et al. [6] give the subroutine needed in Lemma 2.1(a) with $T_{greedy}(n) = O(n \log n)$. An exponential search and Lemma 4.2 can also be used to obtain the same result. Lemma 4.3 gives the subroutine needed in Lemma 2.1(b) with $T_{rect}(n) = O(n \log^2 n)$. Then Lemma 2.1 implies the following.

► **Theorem 4.4.** *Let $S = s_1, \dots, s_n$ be a sequence of points in the plane. In $O(n \log^2 n)$ time, we can compute, for all indices $i \in [n]$, the largest index $j^*(i)$ with $\text{diam}(s_i, \dots, s_{j^*(i)}) \leq 1$.*

5 Area of the convex hull

In this section, we will for simplicity assume general position: no two points have the same x -coordinate and no three points are collinear.

For a point set P , we denote by $CH(P)$ its convex hull. For each point s outside $CH(P)$, let $\tau(s, P)$ be the two points on the boundary of $CH(P)$ that support the tangents to $CH(P)$ through s .

5.1 Sketches

Let P be a set of points and let $S = s_1, \dots, s_n$ be a sequence of points. Let p_{\max} and p_{\min} be the points of P with largest and smallest x -coordinates, respectively. The **CH-sketch of P with respect to the sequence S** is the point set

$$\{p_{\max}, p_{\min}\} \cup \left(\bigcup_{i \in [n]} \tau(s_i, P \cup S[1, i - 1]) \cap P \right).$$

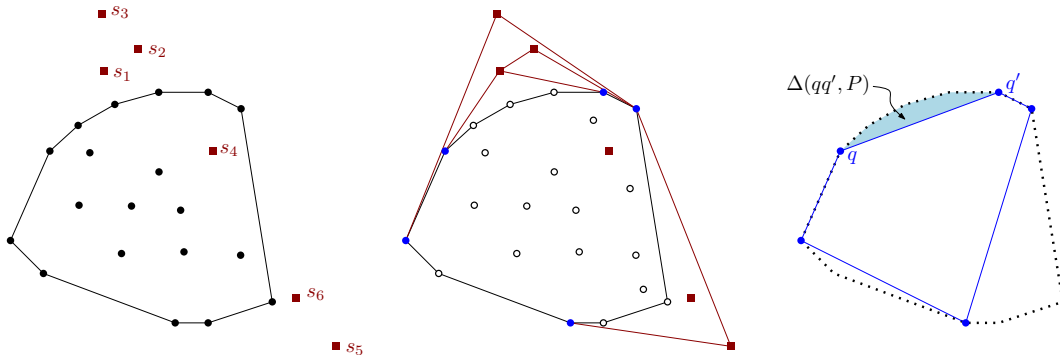
An example is given in Figure 5. The intuition is that the CH-sketch should contain the points of P that support some tangent during the iterative construction of $CH(P \cup S[1, i])$, for $i = 1, \dots, n$. We add p_{\max} and p_{\min} for convenience: we will later maintain the area of the upper hulls of P and the CH-sketch, and it is slightly simpler if their starting and ending points match.

Note that we define CH-sketches with respect to *sequences*, while sketches in the previous section were with respect to sets. We do this to achieve better efficiency.

Let Q denote the CH-sketch of P with respect to S . We have the following straightforward consequences of the definition:

- (i) $|Q| = O(|S|)$ because each point of S contributes at most two points to Q .
- (ii) Each point of Q is a vertex of $CH(P)$ because it is in P and supports a tangent to $CH(P \cup X)$ for some X .
- (iii) For each $i \in [n]$, we have $\tau(s_i, P \cup S[1, i - 1]) = \tau(s_i, Q \cup S[1, i - 1])$, as a point supporting a tangent to $CH(P \cup S[1, i - 1])$ through s_i is either in Q by definition or in $S[1, i - 1]$.

We next discuss some properties about composition of CH-sketches. In our algorithm, we are going to keep two CH-sketches, each with respect to a different sequence. Because of this, some statements become cumbersome.



■ **Figure 5** Left: A set P of points (black dots) with its convex hull and the sequence $S = s_1, \dots, s_6$ (red squares). Center: the sequence of convex hulls $CH(P \cup S[1, i])$ for $i = 1, \dots, 6$. The points of the CH-sketch Q of P with respect to S are marked in solid blue. Right: $CH(Q)$ for the CH-sketch of P and the clipped region $\Delta(qq', P)$ for an edge qq' of $CH(Q)$.

► **Lemma 5.1.** *Let Q_1 be a CH-sketch of P with respect to a sequence $S_1[1, n_1]$. Let Q_2 be a CH-sketch of P with respect to a sequence $S_2[1, n_2]$. Consider indices c_1 and c_2 such that $1 \leq c_1 \leq n_1$ and $1 \leq c_2 \leq n_2$.*

- (a) *If Q' is a CH-sketch of $Q_1 \cup S_1[1, c_1]$ with respect to $S_1[c_1 + 1, n_1]$, then Q' is a CH-sketch of $P \cup S_1[1, c_1]$ with respect to $S_1[c_1 + 1, n_1]$.*
- (b) *If Q' is a CH-sketch of $Q_2 \cup S_1[1, c_1]$ with respect to $S_2[1, c_2]$, then Q' is a CH-sketch of $P \cup S_1[1, c_1]$ with respect to $S_2[1, c_2]$.*

5.2 Clipped regions

Let P be a set of points and let Q be a subset of the vertices of $CH(P)$. Each edge qq' of $CH(Q)$ separates a portion of $CH(P) \setminus CH(Q)$ from $CH(Q)$. We denote such a region by $\Delta(qq', P)$. See Figure 5, right. We use $\Delta(Q, P)$ for the family of **clipped regions** $\{\Delta(qq', P)\}_{qq'}$, where qq' iterates over all edges of $CH(Q)$.

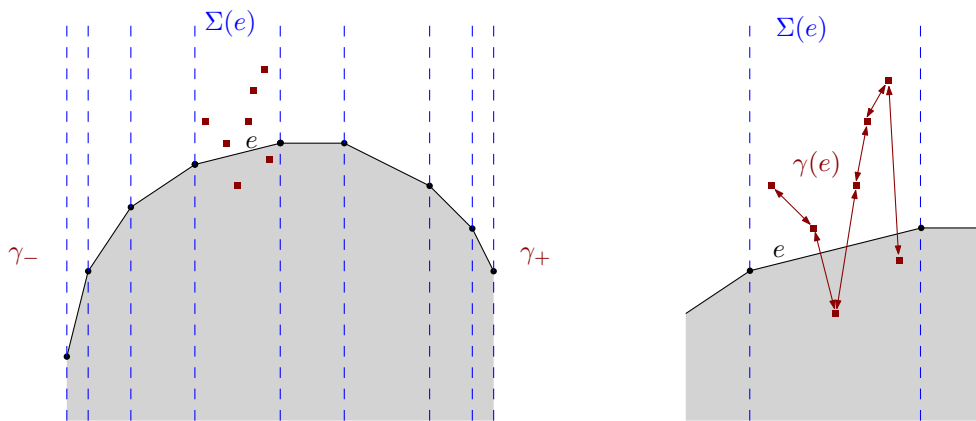
Let Q be a CH-sketch of P with respect to S . Consequence (iii) of the definition of CH-sketch is important to easily obtain the area of $CH(P \cup S[1, i])$ from the area of $CH(Q \cup S[1, i])$ and the area of each of the clipped regions $\Delta(Q, P)$. Indeed, for every index i , a clipped region of $\Delta(Q, P)$ is contained either in $CH(Q \cup S[1, i])$ or in the closure of its complement. Therefore

$$\text{area}(CH(P \cup S[1, i])) = \text{area}(CH(Q \cup S[1, i])) + \sum_{qq'} \text{area}(\Delta(qq', P)),$$

where the sum is taken over all edges qq' of $CH(Q)$ that are also edges of $CH(Q \cup S[1, i])$. See Figure 5 for an example. We use clipped regions to keep track of area difference between $CH(P)$ and $CH(Q)$ though the addition of points of S .

5.3 Algorithms

We are going to keep point sets sorted by their x -coordinates. The sequence S is also going to be kept sorted by x -coordinates. This *does not* mean that the x -coordinates of s_1, s_2, \dots, s_n are increasing when these points are given in sequence order. This means that, besides the sequence S , we have another list where the elements appearing in S are sorted by x -coordinates. For a set or sequence S , we will use $\mathcal{L}_x(S)$ to denote the list containing S sorted by x -coordinate.



■ **Figure 6** Data maintained during the incremental algorithm in the proof of Lemma 5.2.

► **Lemma 5.2.** *Let P be a set and S a sequence of points, both of cardinality at most n . Assume that we have the corresponding lists $\mathcal{L}_x(P)$ and $\mathcal{L}_x(S)$. We can compute a CH-sketch of P with respect to S in $O(n \log \log n)$ time. Moreover, the CH-sketch is obtained sorted by x -coordinate.*

Proof Sketch. We iteratively add the points s_1, \dots, s_n of S , maintain $CH(P \cup S[1, i])$, and mark the points $\tau(s_i, P \cup S[1, i - 1]) \cap P$ to be added to the CH-sketch. We maintain separately the upper and the lower hull of $P \cup S[1, i]$. We discuss only the upper hull.

Through the iterative procedure, we use a list UH that stores the edges of the upper hull of $P \cup S[1, i]$. For each segment e of UH , let $\Sigma(e)$ be the vertical slab contained between the two vertical lines through the endpoints of e . For each edge e in UH , we have a list $\gamma(e)$ of the points of $S[i + 1, n]$ contained in $\Sigma(e)$, sorted by x -coordinate. We have two additional lists, γ_- and γ_+ , that contain the points of $S[i + 1, n]$ to the left and to the right of UH , respectively, also sorted by x -coordinate. See Figure 6. We also maintain a van Emde Boas tree [22] for the vertices of UH . Its purpose is to find, at the time of inserting s_i , the segment e of UH whose slab $\Sigma(e)$ contains s_i . Using the order given by x -coordinates, this is a predecessor query in UH . We can identify each point of $P \cup S$ with its rank in the order given by x -coordinates, and use those ranks as keys for the van Emde Boas tree. Thus, we have an universe of $|P \cup S| = O(n)$ elements and each operation takes $O(\log \log n)$ time.

The data can be initialized in $O(n)$ time computing $CH(P)$ from $\mathcal{L}_x(P)$ and with a simultaneous scanning of UH and $\mathcal{L}_x(S)$. The insertion of a point s_i of S starts locating the edge e such that $\gamma(e)$ contains s_i and the appearance of s_i in $\gamma(e)$. We then have to update the convex hull UH and update the lists making some local operations. The insertion of s_i takes $O(\log \log n + |k_{i-1} - k_i|)$ time, where k_i denotes the number of vertices in the upper hull of $P \cup S[1, i]$. At the end of the insertion, we can obtain the points $\tau(s_i, P \cup S[1, i - 1])$ of the upper hull and mark them for addition to the CH-sketch. Since each point of $P \cup S$ can be deleted at most once, the time over all insertions is $O(n \log \log n)$.

When we have inserted all the points of S , we construct the CH-sketch going through $\mathcal{L}_x(P)$ and selecting those marked for addition to the CH-sketch. We also insert the first and last point of $\mathcal{L}_x(P)$, since they have smallest and largest x -coordinate. ◀

An approach similar to the one used in the proof of Lemma 5.2 can be used to incrementally maintain the area of $CH(Q \cup S[1, i])$, for $i = 1, \dots, n$. If Q is a CH-sketch of P with respect to S , we can then use the area of the clipped regions $\Delta(Q, P)$ to compute the area of

$CH(P \cup S[1, i])$, for $i = 1, \dots, n$. We just have to notice that, for each $i \in [n]$, a clipped region of $\Delta(Q, P)$ is contained in $CH(P \cup S[1, i])$ or in its complement. This leads to the following.

► **Lemma 5.3.** *Let P be a set of points and let $S = s_1, \dots, s_n$ be a sequence of points. Assume that we have a CH-sketch Q of P with respect to S and, for each edge qq' of $CH(Q)$, the area of $\Delta(qq', P)$. Furthermore, assume that we have the corresponding lists $\mathcal{L}_x(Q)$ and $\mathcal{L}_x(S)$. In $O(n \log \log n)$ time, we can find the largest index $j^* \in [n]$ such that $\text{area}(CH(P \cup S[1, j^*])) \leq 1$. ◀*

► **Lemma 5.4.** *Consider a frontier rectangle R anchored at the diagonal with height h and width w . We can solve R in $O((h + w) \log(h + w) \log \log(h + w))$ time.*

Proof. (Sketch) We follow very closely the proof of Lemma 4.3. However, the description of a recursive subproblem is given by:

- (i) a frontier rectangle $[a, a + h] \times [b, b + w]$ contained in \mathbb{U} ;
- (ii) a CH-sketch Q_{ver} of $S[a + h, b]$ with respect to the reversal of $S[a, a + h - 1]$;
- (iii) a CH-sketch Q_{hor} of $S[a + h, b]$ with respect to $S[b + 1, b + w]$;
- (iv) lists $\mathcal{L}_x(Q_{ver})$, $\mathcal{L}_x(Q_{hor})$, $\mathcal{L}_x(S[a, a + h - 1])$, and $\mathcal{L}_x(S[b + 1, b + w])$;
- (v) the convex hull $CH(Q)$, where $Q = Q_{hor} \cup Q_{ver}$; and
- (vi) the area of each of the clipped regions $\Delta(Q, S[a + h, b])$.

Note that the description of such a subproblem has size $O(h + w)$.

We construct the base problem, to start the recursion, in $O((h + w) \log(h + w))$ time as follows. The rectangle $[a, a + h] \times [b, b + w]$ is given as the input, with $a + h = b$. We have $Q_{ver} = Q_{hor} = \{s_b\}$. The lists $\mathcal{L}_x(Q_{ver})$ and $\mathcal{L}_x(Q_{hor})$ have only one element. The lists $\mathcal{L}_x(S[a, a + h - 1])$ and $\mathcal{L}_x(S[b + 1, b + w])$ can be constructed in $O(h \log h)$ and $O(w \log w)$ time, respectively, by just sorting the points from scratch. The remaining data is trivial.

Let us now discuss how we solve a subproblem appearing in the recursion. The case $h \leq 1$ is easier, takes $O(w \log \log w)$ time, and we omit it.

Consider now the case when $h \geq 2$ and thus the rectangle has at least three rows. We use the divide-and-conquer approach discussed in Section 2.2 and already used in Lemma 4.3. Set $m = a + \lfloor h/2 \rfloor$, find the last index $c \in [b, b + w]$ such that $\text{area}(CH(S[m, c])) \leq 1$, and recurse on the rectangles $R_1 = [a, m - 1] \times [b, c]$ and $R_2 = [m + 1, a + h] \times [c, b + w]$. Recall Figure 2.

The value c can be found in $O((h + w) \log \log(h + w))$ time using Lemma 5.3, as follows. We compute the CH-sketch Q_m of $Q_{hor} \cup S[m, a + h - 1]$ with respect to $S[b + 1, b + w]$. Because of Lemma 5.2 and since the input can be obtained sorted by x -coordinate from $\mathcal{L}_x(Q_{hor})$, $\mathcal{L}_x(S[a, a + h - 1])$, and $\mathcal{L}_x(S[b + 1, b + w])$, this can be done in $O((h + w) \log \log(h + w))$ time. Because of Lemma 5.1(b), where S_1 is the reversal of $S[a, a + h - 1]$ and $S_2 = S[b + 1, b + w]$, Q_m is a CH-sketch of $S[a + h, b] \cup S[m, a + h - 1] = S[m, b]$ with respect to $S[b + 1, b + w]$. We can show that the area of the clipped regions $\Delta(Q_m, S[m, b])$ can be obtained in $O((h + w) \log \log(h + w))$ time. Thus Q_m and $S[b + 1, b + w]$ satisfy the hypothesis of Lemma 5.3, as needed to find c .

In $O((h + w) \log \log(h + w))$ time, we can collect the data for the recursive calls. For this, we use Lemma 5.2 to compute CH-sketches of CH-sketches with respect to subsequences. Lemma 5.1 is then used to argue that we are indeed computing the CH-sketches required for the recursive call. We omit the detailed arguments.

Thus, we can construct the recursive subproblems in $O((h + w) \log \log(h + w))$ time. It follows that the time $T(h, w)$ to solve a recursive subproblem with $h + 1$ rows and $w + 1$ rows

is given by

$$T(h, w) = \begin{cases} O((h+w) \log \log(h+w)) + T(\lfloor h/2 \rfloor, w') + T(\lfloor h/2 \rfloor, w-w') & \text{if } h \geq 2, \\ O(w \log \log w) & \text{if } h \leq 1. \end{cases}$$

Lemma 2.2 implies that $T(h, w) = O((h+w) \log(h+w) \log \log(h+w))$. Thus, we can solve all recursive subproblems in $O((h+w) \log(h+w) \log \log(h+w))$, and the result follows. ◀

There are incremental algorithms to maintain the convex hull explicitly in amortized time $O(\log n)$ per insertion; see for example [21, Chapter 3]. Such a procedure gives the subroutine needed in Lemma 2.1(a) with $T_{\text{greedy}}(n) = O(n \log n)$. An exponential search and Lemma 5.3 can also be used to obtain the same result. Lemma 5.4 gives the subroutine needed in Lemma 2.1(b) with $T_{\text{rect}}(n) = O(n \log n)$. Then Lemma 2.1 implies the following.

► **Theorem 5.5.** *Let $S = s_1, \dots, s_n$ be a sequence of planar points. In $O(n \log n \log \log n)$ time, we can compute, for all $i \in [n]$, the largest index $j^*(i)$ such that $CH(\{s_i, \dots, s_{j^*(i)}\})$ has area at most 1.*

6 Planar graphs

In this section, we move away from geometry to discuss graph planarity. This problem provides a neat use of the methodology we presented and an important improvement over the use of dynamic data structures. We only provide a very high-level overview.

Let G be a planar graph and let X be a subset of its vertices. A graph H is a **planar-sketch of G with respect to X** if it satisfies the following conditions:

- $X \subseteq V(H)$,
- H has size $O(|X|)$, and
- for each edge set F with endpoints in X , $G + F$ is planar if and only if $H + F$ is planar.

Galil, Italiano, and Sarnak [15] have shown that such planar-sketches exist and can be computed in linear time. Note that they defined the sketch property for the addition of a single edge ($|F| = 1$). However, Eppstein et al. [12] noted that the same construction works for multiple edges and referred to the sketches as compressed certificates for planarity.

The fact that planar-sketches can be computed in linear time is parallel to Lemma 4.1(a) in this context. One can prove a statement analogous to Lemma 4.1(b) for planar sketches, as follows. If H is a planar-sketch of G with respect to X , F is a set of edges with endpoints in X , $H + F$ is planar, and H' is a planar-sketch of $H + F$ with respect to $Y \subset X$, then H' is a planar sketch of $G + F$ with respect to Y .

Equipped with linear-time planarity testing [18] and the aforementioned linear-time computation of planar-sketches, we can follow the same methodology as in Section 4.2, shaving off a logarithmic factor. Thus, we obtain the subroutine needed in Lemma 2.1(a) with running time $T_{\text{greedy}}(n) = O(n)$, and the subroutine needed in Lemma 2.1(b) with $T_{\text{rect}}(n) = O(n \log n)$. Then Lemma 2.1 implies the following.

► **Theorem 6.1.** *Let $E = e_1, \dots, e_n$ be a sequence of edges. In $O(n \log n)$ time, we can compute, for all indices $i \in [n]$, the largest index $j^*(i)$ such that the graph defined by $e_i + \dots + e_{j^*(i)}$ is planar.* ◀

Acknowledgments. We are grateful to the reviewers for their careful comments.

References

- 1 Boris Aronov, Anne Driemel, Marc J. van Kreveld, Maarten Löffler, and Frank Staals. Segmentation of trajectories for non-monotone criteria. In *SODA 2013*, pages 1897–1911, 2013.
- 2 Michael J. Bannister, William E. Devanny, Michael T. Goodrich, and Joe Simons. Windows into geometric events. In *CCCG 2014*, 2014.
- 3 Michael J. Bannister, Christopher DuBois, David Eppstein, and Padhraic Smyth. Windows into relational events: Data structures for contiguous subsequences of edges. In *SODA 2013*, pages 856–864, 2013.
- 4 Kevin Buchin, Maike Buchin, Marc van Kreveld, Maarten Löffler, Rodrigo I. Silveira, Carola Wenk, and Lionov Wiratma. Median trajectories. *Algorithmica*, 66(3):595–614, 2013.
- 5 Kevin Buchin, Maike Buchin, Marc van Kreveld, and Jun Luo. Finding long and similar parts of trajectories. *Comput. Geom.*, 44(9):465–476, 2011.
- 6 Maike Buchin, Anne Driemel, Marc J. van Kreveld, and Vera Sacristan. Segmenting trajectories: A framework and algorithms using spatiotemporal criteria. *J. Spatial Information Science*, 3(1):33–63, 2011.
- 7 Timothy M. Chan. Dynamic planar convex hull operations in near-logarithmic amortized time. *J. ACM*, 48(1):1–12, 2001.
- 8 Timothy M. Chan. A dynamic data structure for 3-D convex hulls and 2-D nearest neighbor queries. *J. ACM*, 57(3), 2010.
- 9 Chen Chen, Hao Su, Qixing Huang, Lin Zhang, and Leonidas Guibas. Pathlet learning for compressing and planning trajectories. In *SIGSPATIAL'13*, pages 392–395, 2013.
- 10 Giuseppe Di Battista and Roberto Tamassia. On-Line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996.
- 11 David Eppstein. Dynamic Euclidean minimum spanning trees and extrema of binary functions. *Discrete Comput. Geom.*, 13:111–122, 1995.
- 12 David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Thomas H. Spencer. Separator based sparsification. I. Planarity testing and minimum spanning trees. *J. Comput. Syst. Sci.*, 52(1):3–27, 1996.
- 13 David Eppstein, Michael T. Goodrich, and Maarten Löffler. Tracking moving objects with few handovers. In *WADS 2011*, volume 6844 of *LNCS*, pages 362–373. Springer, 2011.
- 14 Johannes Fischer and Volker Heun. Space-efficient preprocessing schemes for range minimum queries on static arrays. *SIAM J. Comput.*, 40(2):465–492, 2011.
- 15 Zvi Galil, Giuseppe F. Italiano, and Neil Sarnak. Fully dynamic planarity testing with applications. *J. ACM*, 46(1):28–91, 1999.
- 16 Joachim Gudmundsson, Jyrki Katajainen, Damian Merrick, Cahya Ong, and Thomas Wolle. Compressing spatio-temporal trajectories. *Comput. Geom.*, 42(9):825–841, 2009.
- 17 Joachim Gudmundsson, Marc van Kreveld, and Bettina Speckmann. Efficient detection of patterns in 2D trajectories of moving points. *GeoInformatica*, 11(2):195–215, 2007.
- 18 John Hopcroft and Robert Tarjan. Efficient planarity testing. *J. ACM*, 21(4):549–568, 1974.
- 19 Jakub Łącki and Piotr Sankowski. Reachability in graph timelines. In *ITCS 2013*, pages 257–268, 2013.
- 20 Mark H. Overmars and Jan van Leeuwen. Maintenance of configurations in the plane. *J. Comput. Syst. Sci.*, 23(2):166–204, 1981.
- 21 Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.
- 22 Peter van Emde Boas. Preserving order in a forest in less than logarithmic time and linear space. *Inf. Process. Lett.*, 6(3):80–82, 1977.

Riemannian Simplices and Triangulations*

Ramsay Dyer, Gert Vegter, and Mathijs Wintraecken

Johann Bernoulli Institute
Rijksuniversiteit Groningen, The Netherlands
{r.h.dyer,g.vegter,m.h.m.j.wintraecken}@rug.nl

Abstract

We study a natural intrinsic definition of geometric simplices in Riemannian manifolds of arbitrary finite dimension, and exploit these simplices to obtain criteria for triangulating compact Riemannian manifolds. These geometric simplices are defined using Karcher means. Given a finite set of vertices in a convex set on the manifold, the point that minimises the weighted sum of squared distances to the vertices is the Karcher mean relative to the weights. Using barycentric coordinates as the weights, we obtain a smooth map from the standard Euclidean simplex to the manifold. A Riemannian simplex is defined as the image of the standard simplex under this barycentric coordinate map. In this work we articulate criteria that guarantee that the barycentric coordinate map is a smooth embedding. If it is not, we say the Riemannian simplex is degenerate. Quality measures for the “thickness” or “fatness” of Euclidean simplices can be adapted to apply to these Riemannian simplices. For manifolds of dimension 2, the simplex is non-degenerate if it has a positive quality measure, as in the Euclidean case. However, when the dimension is greater than two, non-degeneracy can be guaranteed only when the quality exceeds a positive bound that depends on the size of the simplex and local bounds on the absolute values of the sectional curvatures of the manifold. An analysis of the geometry of non-degenerate Riemannian simplices leads to conditions which guarantee that a simplicial complex is homeomorphic to the manifold.

1998 ACM Subject Classification G.1.1 [Numerical analysis] Interpolation, G.1.2 [Numerical analysis] Approximation – linear approximation

Keywords and phrases Karcher means, barycentric coordinates, triangulation, Riemannian manifold, Riemannian simplices

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.255

1 Introduction

The standard definition of a Euclidean simplex as the convex hull of its vertices is not useful for defining simplices in general Riemannian manifolds. Besides the problem that convex hulls are difficult to compute, the resulting objects could not be used as building blocks for triangulations: a minimising geodesic between two points on a shared facet would have to lie within the facet, which is not a realisable constraint in general. A more detailed discussion and references can be found in the full version [9] of this work.

Given the vertices, a geometric Euclidean simplex can also be defined as the domain on which the barycentric coordinate functions are non-negative. This definition *does* extend to general Riemannian manifolds in a natural way. The construction is based on the fact that the barycentric coordinate functions can be defined by a “centre of mass” construction.

* This research has been partially supported by the 7th Framework Programme for Research of the European Commission, under FET-Open grant number 255827 (CGL Computational Geometry Learning). We thank Stefan von Deylen for pointing out the work of Sander [14], and for stimulating discussions. We have also benefited from discussions with Arijit Ghosh.



Suppose $\{v_0, \dots, v_n\} \subset \mathbb{R}^n$, and $(\lambda_i)_{0 \leq i \leq n}$ is a set of non-negative weights that sum to 1. If u is the unique point that minimises the function

$$y \mapsto \sum_{i=0}^n \lambda_i d_{\mathbb{R}^n}(y, v_i)^2, \tag{1}$$

where $d_{\mathbb{R}^n}(x, y) = |x - y|$ is the Euclidean distance, then $u = \sum \lambda_i v_i$, and if the v_i are affinely independent, then the λ_i are the barycentric coordinates of u in the simplex $[v_0, \dots, v_n]$.

We can view a given set of barycentric coordinates $\lambda = (\lambda_0, \dots, \lambda_n)$ as a point in \mathbb{R}^{n+1} . The set Δ^n of all points in \mathbb{R}^{n+1} with non-negative coefficients that sum to 1 is called the *standard Euclidean n -simplex*. Thus the minimisation of the function (1) defines a map from the standard Euclidean simplex to the Euclidean simplex $[v_0, \dots, v_n] \subset \mathbb{R}^n$.

If instead the points $\{v_i\}$ lie in a sufficiently small neighbourhood W in a Riemannian manifold M , then, by using the metric of the manifold instead of $d_{\mathbb{R}^n}$ in Equation (1), we obtain a function $\mathcal{E}_\lambda : W \rightarrow \mathbb{R}$ that has a unique minimum $x_\lambda \in W$. In this way we obtain a mapping $\lambda \mapsto x_\lambda$ from Δ^n to W . We call the image of this map an *intrinsic simplex*, or a *Riemannian simplex*.

Karcher [10] studied such centre of mass constructions extensively in the Riemannian setting, and this kind of averaging technique is often called “Karcher means”. More recently, Rustamov [13] introduced barycentric coordinates on a surface via Karcher means. Sander [14] used the method in arbitrary dimensions to define Riemannian simplices as described above. We are not aware of any published work exploiting this notion of Riemannian simplices prior to that of Rustamov [13] and Sander [14], although the idea was known much earlier [1, § 6.1.5].

Our work is motivated by a desire to develop general sampling density criteria for triangulations of manifolds. To this end we need to establish a property that Sander did not consider. We need to ensure that the map from the Euclidean simplex to the manifold is a smooth embedding. This ensures that the barycentric coordinates mapped to the manifold do in fact provide a local system of coordinates. If the map is not an embedding, we call the Riemannian simplex *degenerate*. Independently, von Deylen [16] has also treated the question of degeneracy of Riemannian simplices. His work includes a detailed analysis of the geometry of the barycentric coordinate map, and several applications. He does not address the problem of sampling density criteria for triangulation.

A Euclidean simplex is non-degenerate if and only if its vertices are affinely independent. We show that a Riemannian simplex is non-degenerate if and only if, for every point in the simplex, the vertices are affinely independent when they are lifted by the inverse of the exponential map to the tangent space at that point.

In a two dimensional manifold this condition is satisfied for a triangle as long as the vertices do not lie on a common geodesic. Similar to the Euclidean case, such a configuration can be avoided by applying an arbitrarily small perturbation to the vertices. However, when the dimension is greater than two, a non-trivial constraint on simplex quality is required; one that cannot be attained by an arbitrarily small perturbation of the vertices.

In order to define a Riemannian simplex, we need the vertices to lie in a geodesically convex set, and this imposes a bound on the edge lengths with respect to an upper bound on the local sectional curvatures. For a surface, this is the only real constraint needed to ensure a non-degenerate simplex. In higher dimensions, we require the simplex size to be constrained also by a lower bound on the sectional curvatures.

Outline and main results

In Section 2 we present the framework for centre of mass constructions, and introduce the barycentric coordinate map and Riemannian simplices. Riemannian simplices are defined (Definition 2) as the image of the barycentric coordinate map, so they are “filled in” geometric simplices. Each of the three subsequent sections is devoted to presenting one of our three main results: conditions for non-degeneracy of Riemannian simplices, Theorem 6; conditions for triangulation, Theorem 11; and the geometric fidelity of the resulting triangulation, Theorem 14.

In Section 3 we establish criteria to ensure that a Riemannian simplex is non-degenerate. In the tangent space at any point in a Riemannian simplex σ_M , there is a Euclidean simplex $\sigma(x)$ that is a natural approximation of σ_M . We give a characterisation of non-degeneracy of σ_M in terms of these Euclidean simplices: σ_M is non-degenerate if and only if $\sigma(x)$ is a non-degenerate Euclidean simplex for every $x \in \sigma_M$ (Proposition 4).

The *thickness* of a Euclidean simplex, defined in Section 3.1, is a measure of its quality, i.e., how far it is from being degenerate. We choose a representative $\sigma(p)$ for some $p \in \sigma_M$ and observe that all the $\sigma(x)$ are geometrically small perturbations of $\sigma(p)$. We then exploit previous results on the stability of Euclidean simplex quality [2, Lemma 8] to establish a simple inequality, relating the thickness of $\sigma(p)$ to the edge lengths of σ_M and a bound on the absolute value of the local sectional curvatures, which when satisfied guarantees that all the $\sigma(x)$ are non-degenerate. It then follows, from the above-mentioned Proposition 4, that σ_M is non-degenerate, and this is Theorem 6.

In Section 4 we develop our criteria for triangulating manifolds. A *triangulation* of a manifold M is a homeomorphism $H: |\mathcal{A}| \rightarrow M$, where \mathcal{A} is an abstract simplicial complex, and $|\mathcal{A}|$ is its carrier (topological realisation).¹ We establish properties of maps whose differentials are all small perturbations of a fixed linear isometry, and use these properties to reveal conditions under which the star of a vertex in a manifold complex will be embedded into M . This allows us to express, in Proposition 8, generic conditions that ensure that a simplicial complex is homeomorphic to M . We then demonstrate that the differential of the barycentric coordinate map can be bounded as required by Proposition 8, and thus arrive at our triangulation criteria expressed in Theorem 11.

The triangulation $H: |\mathcal{A}| \rightarrow M$ is defined by the barycentric coordinate map on each of the simplices. The quantitative aspect of the triangulation criteria is expressed in terms of a scale parameter h which bounds the edge lengths of the Riemannian simplices defined by the triangulation. This bound on h is of the same character as the non-degeneracy criteria: it depends on a *thickness bound* t_0 governing the quality of the simplices involved, and also on a bound on the absolute value of the sectional curvatures.

The complex \mathcal{A} in Theorem 11 naturally admits a piecewise linear metric by assigning edge lengths to the simplices given by the geodesic distance in M between the endpoints. In Section 5 we observe that in order to ensure that this does in fact define a piecewise-flat metric, we need to employ slightly stronger constraints on the scale parameter h . In this case, the complex \mathcal{A} becomes a good geometric approximation of the original manifold, as expressed in Theorem 14, which states that the metric distortion of H is proportional to h^2 .

¹ In fact the triangulations of interest to us have the property that the restriction of H to each simplex in $|\mathcal{A}|$ is a smooth embedding, and also the star of each simplex admits a piecewise linear embedding into \mathbb{R}^n . These additional properties ensure that \mathcal{A} represents the unique piecewise linear structure associated with M . See Thurston [15, Thm 3.10.2], or Munkres [12, Cor. 10.13] for details.

2 Riemannian simplices

In this section we summarise the essential properties of Karcher means, and define Riemannian simplices. We work with an n -dimensional Riemannian manifold M . The centre of mass construction developed by Karcher [10] hinges on the notion of convexity in a Riemannian manifold. A set $B \subseteq M$ is *convex* if any two points $x, y \in B$ are connected by a minimising geodesic γ_{xy} that is unique in M , and contained in B . For $c \in M$, the geodesic ball of radius r centred at c is the set $B_M(c; r)$ of points in M whose distance from c is less than r , and we denote its closure by $\overline{B}_M(c; r)$. If r is smaller than ρ_0 , defined below (4), then $\overline{B}_M(c; r)$ will be convex [5, §6.4].

Recall that the exponential map at $p \in M$ sends a vector v in the tangent space $T_p M$ to the point $\exp_p(v)$ defined by the geodesic of length $|v|$ emanating from p in the direction v . The exponential map is a diffeomorphism when restricted to a ball whose radius is smaller than the *injectivity radius*.

In our context, we are interested in finding a weighted centre of mass of a finite set $\{p_0, \dots, p_j\} \subset B \subset M$, where the containing set B is open, and its closure \overline{B} is convex. The centre of mass construction is based on minimising the function $\mathcal{E}_\lambda : \overline{B} \rightarrow \mathbb{R}$ defined by

$$\mathcal{E}_\lambda(x) = \frac{1}{2} \sum_i \lambda_i d_M(x, p_i)^2, \quad (2)$$

where the $\lambda_i \geq 0$ are non-negative weights that sum to 1, and d_M is the geodesic distance function on M . Karcher's first simple observation is that the minima of \mathcal{E}_λ must lie in the interior of \overline{B} , i.e., in B itself. This follows from considering the gradient of \mathcal{E}_λ :

$$\text{grad } \mathcal{E}_\lambda(x) = - \sum_i \lambda_i \exp_x^{-1}(p_i). \quad (3)$$

At any point x on the boundary of \overline{B} , the gradient vector lies in a cone of outward pointing vectors. It follows that the minima of \mathcal{E}_λ lie in B . The more difficult result that the minimum is unique, Karcher showed by demonstrating that \mathcal{E}_λ is convex. If $B \subseteq M$ is a convex set, a function $f : B \rightarrow \mathbb{R}$ is *convex* if for any geodesic $\gamma : I \rightarrow B$, the function $f \circ \gamma$ is convex. If f has a minimum in B , it must be unique. By Equation (3), it is the point x where

$$\sum_i \lambda_i \exp_x^{-1}(p_i) = 0.$$

Our results will require a bound Λ on the absolute value of the sectional curvatures in M . However, the definition of Riemannian simplices only requires an upper bound on the sectional curvatures, which we denote by Λ_+ . We denote the injectivity radius of M by ι . We have the following result [10, Thm 1.2]:

► **Lemma 1** (Unique centre of mass). *If $\{p_0, \dots, p_j\} \subset B_\rho \subset M$, and B_ρ is an open ball of radius ρ with*

$$\rho < \rho_0 = \min \left\{ \frac{\iota}{2}, \frac{\pi}{4\sqrt{\Lambda_+}} \right\} \quad (4)$$

(if $\Lambda_+ \leq 0$ we take $1/\sqrt{\Lambda_+}$ to be infinite), then on any geodesic $\gamma : I \rightarrow B_\rho$, we have

$$\frac{d^2}{dt^2} \mathcal{E}_\lambda(\gamma(t)) \geq C(\Lambda_+, \rho) > 0, \quad (5)$$

where $C(\Lambda_+, \rho)$ is a positive constant depending only on Λ_+ and ρ . In particular, \mathcal{E}_λ is convex and has a unique minimum in B_ρ , characterised by the vanishing of the gradient (3).

► **Definition 2** (Riemannian simplex). If a finite set $\sigma^j = \{p_0, \dots, p_j\} \subset M$ in an n -manifold is contained in an open geodesic ball B_ρ whose radius, ρ , satisfies Equation (4), then σ^j is the set of vertices of a geometric *Riemannian simplex*, denoted σ_M^j , and defined to be the image of the map

$$\mathcal{B}_{\sigma^j} : \Delta^j \rightarrow M$$

$$\lambda \mapsto \operatorname{argmin}_{x \in \overline{B}_\rho} \mathcal{E}_\lambda(x).$$

We say that σ_M^j is *non-degenerate* if \mathcal{B}_{σ^j} is a smooth embedding; otherwise it is *degenerate*.

► **Remark.** Lemma 1 demands that a Riemannian simplex be contained in a ball whose radius is constrained by ρ_0 . Thus Riemannian simplices always have edge lengths less than $2\rho_0$. If the longest edge length, $L(\sigma_M)$, of σ_M is less than ρ_0 , then σ_M must be contained in the closed ball of radius $L(\sigma_M)$ centred at a vertex. Indeed, any open ball centred at a vertex whose radius is larger than $L(\sigma_M)$, but smaller than ρ_0 , must contain the vertices and have a convex closure. The simplex is thus contained in the intersection of these balls. If $L(\sigma_M) \geq \rho_0$, then a ball of radius $L(\sigma_M)$ need not be convex. In this case we claim only that σ_M is contained in a ball of radius $2\rho_0$ centred at any vertex.

Define an i -face of σ_M^j to be the image of an i -face of Δ^j . Since an i -face of Δ^j may be identified with Δ^i (e.g., by an order preserving map of the vertex indices), the i -faces of σ_M^j are themselves Riemannian i -simplices. In particular, if τ and μ are the vertices of Riemannian simplices τ_M and μ_M , and $\sigma^i = \tau \cap \mu$, then the Riemannian i -simplex σ_M^i is a face of both τ_M and μ_M . The *edges* of a Riemannian simplex are the Riemannian 1-faces. We observe that these are geodesic segments. We will focus on full dimensional simplices. Unless otherwise specified, σ_M will refer to a Riemannian simplex defined by a set σ of $n + 1$ vertices in our n -dimensional manifold M .

The barycentric coordinate map \mathcal{B}_σ is differentiable. This follows from the implicit function theorem, as is shown by Buser and Karcher [5, §8.3.3], for example.

A Riemannian simplex is not convex in general, but by Karcher’s observation it is contained in any open set that contains the vertices and has a convex closure. Thus the simplex is contained in the intersection of such sets.

Equation (4) gives an upper bound on the size of a Riemannian simplex that depends only on the injectivity radius and an *upper* bound on the sectional curvature. For example, in a non-positively curved manifold, the size of a well defined Riemannian simplex is constrained only by the injectivity radius. However, if the dimension n of the manifold is greater than 2, we will require also a *lower* bound on the sectional curvatures in order to ensure that the simplex is non-degenerate.

3 Non-degeneracy criteria

In this section we establish geometric criteria that ensure that a Riemannian simplex is non-degenerate. We first review the properties of Euclidean simplices, including the thickness quality measure, which parameterises how far a simplex is from being degenerate. We observe that we can bound the change in the thickness of a simplex if the edge lengths are perturbed a small amount.

Next we examine the differential of the barycentric coordinate map, and arrive at a characterisation of non-degenerate Riemannian simplices in terms of affine independence (Proposition 4). The Rauch comparison theorem is a central result in Riemannian geometry

which allows us to bound the metric distortion of the exponential map. Combined with the stability of the thickness of Euclidean simplices, this bound on the metric distortion yields conditions which ensure that a Riemannian simplex meets the affine independence characterisation of non-degeneracy, resulting in Theorem 6.

3.1 The stability of Euclidean simplex quality

A Euclidean simplex σ of dimension j is defined by a set of $j + 1$ points in Euclidean space $\sigma = \{v_0, \dots, v_j\} \subset \mathbb{R}^n$. In general we work with abstract simplices, even though we attribute geometric properties to the simplex, inherited from the embedding of the vertices in the ambient space. When we wish to make the dimension explicit, we write it as a superscript, thus σ^j is a j -simplex. Traditional “filled in” geometric simplices are denoted by boldface symbols; $\sigma_{\mathbb{E}} = \text{conv}(\sigma)$ is the convex hull of σ .

A Euclidean simplex $\sigma = \{v_0, \dots, v_j\} \subset \mathbb{R}^n$ has a number of geometric attributes. An i -face of σ is a subset of $i + 1$ vertices, and a $(j - 1)$ face of a j -simplex is a *facet*. The facet of σ that does not have v_i as a vertex is denoted σ_{v_i} . The *altitude* of $v_i \in \sigma$ is the distance from v_i to the affine hull of σ_{v_i} , denoted $a_{v_i}(\sigma)$. The longest edge length is denoted $L(\sigma)$. When there is no risk of confusion, we will omit explicit reference to the simplex, and ignore the distinction between the vertices and their labels. Thus we write L , and a_i instead of $L(\sigma)$ and $a_{v_i}(\sigma)$.

The *thickness* of σ^j , defined as

$$t(\sigma^j) = \begin{cases} 1 & \text{if } j = 0 \\ \min_{v \in \sigma^j} \frac{a_v}{jL} & \text{otherwise.} \end{cases}$$

If $t(\sigma^j) = 0$, then σ^j is *degenerate*. We say that σ^j is t_0 -thick, if $t(\sigma^j) \geq t_0$. If σ^j is t_0 -thick, then so are all of its faces. We write t for the thickness if the simplex in question is clear.

The *barycentric coordinate functions* λ_i associated to σ^j are affine functions on the affine hull of the simplex $\lambda_i: \text{aff}(\sigma^j) \rightarrow \mathbb{R}$ that satisfy $\lambda_i(v_j) = \delta_{ij}$ and $\sum_{i=0}^j \lambda_i = 1$. It is often convenient to choose one of the vertices, v_0 say, of σ to be the origin. We let P be the matrix whose i^{th} column is $v_i - v_0$. Then the barycentric coordinate functions λ_i are linear functions for $i > 0$, and they are dual to the basis defined by the columns of P . This means that if we represent the function λ_i as a row vector, then the matrix Q whose i^{th} row is λ_i satisfies $QP = I_{j \times j}$.

A full dimensional Euclidean simplex σ is non-degenerate, if and only if the corresponding matrix P is non-degenerate. In particular, if σ is full dimensional (i.e., $j = n$), then $Q = P^{-1}$. Suppose $\sigma \subset \mathbb{R}^n$ is an n -simplex. If $\xi \in \mathbb{R}^n$, let $\lambda(\xi) = (\lambda_1(\xi), \dots, \lambda_n(\xi))^{\top}$. Then $\lambda(\xi)$ is the vector of coefficients of $\xi - v_0$ in the basis defined by the columns of P . I.e., $\xi - v_0 = P\lambda(\xi)$.

The quality of a simplex σ is closely related to the quality of P , which can be quantified by means of its *singular values*. In fact, we are only interested in the smallest and largest singular values. The smallest singular value, $s_k(P) = \inf_{|x|=1} |Px|$, vanishes if and only if the matrix P does not have full rank. The largest singular value is the same as the operator norm of P , i.e., $s_1(P) = \|P\| = \sup_{|x|=1} |Px|$. The thickness of σ provides a lower bound [3, Lem. 2.4] on the smallest singular value of P . Specifically, for a j -simplex, we have $s_j(P) \geq \sqrt{j}tL$.

The crucial property of thickness for our purposes is its stability. If two Euclidean simplices with corresponding vertices have edge lengths that are almost the same, then their thicknesses will be almost the same. This allows us to quantify a bound on the smallest singular value of the matrix associated with one of the simplices, given a bound on the other, as shown in the following Lemma [2, Lem. 8]:

► **Lemma 3** (Thickness under distortion). *Suppose that $\sigma = \{v_0, \dots, v_k\}$ and $\tilde{\sigma} = \{\tilde{v}_0, \dots, \tilde{v}_k\}$ are two k -simplices in \mathbb{R}^n such that*

$$|v_i - v_j| - |\tilde{v}_i - \tilde{v}_j| \leq C_0 L(\sigma) \quad \text{with} \quad C_0 = \frac{\eta t(\sigma)^2}{4} \quad \text{and} \quad 0 \leq \eta \leq 1,$$

for all $0 \leq i < j \leq k$. Let P be the matrix whose i^{th} column is $v_i - v_0$, and define \tilde{P} similarly. Then

$$s_k(\tilde{P}) \geq (1 - \eta) s_k(P) \quad \text{and} \quad t(\tilde{\sigma}) \geq \frac{4}{5\sqrt{k}} (1 - \eta) t(\sigma).$$

3.2 The affine independence criterion for non-degeneracy

In this subsection we show that a Riemannian simplex σ_M is non-degenerate if and only if, for any $x \in \sigma_M$, the lift of the vertices by the inverse exponential map yields a non-degenerate Euclidean simplex. The expression for the differential of the barycentric coordinate map obtained in Equation (7) below is the result of a particular case of an argument presented by Buser and Karcher [5, §8.3] in a more general setting.

A Riemannian simplex σ_M is defined by its vertices $\sigma = \{p_0, \dots, p_n\} \subset M$, which are constrained to lie in a convex ball $B_\rho \subseteq M$. For any $x \in B_\rho$ we define a Euclidean simplex $\sigma(x) \subset T_x M$ by $\sigma(x) = \{v_0(x), \dots, v_n(x)\}$, where $v_i(x) = \exp_x^{-1}(p_i)$. The vertices $p_i \in B_\rho$ are considered fixed, but $x \in B_\rho$ is a variable. We continue to use a boldface symbol when we are referring to a simplex as a set of non-negative barycentric coordinates, and normal type refers to the finite vertex set; the convex hull of $\sigma(x)$ is $\sigma_{\mathbb{E}}(x)$.

We work in a domain $U \subseteq \mathbb{R}^n$ defined by a chart $\phi : W \rightarrow U$ with $B_\rho \subseteq W \subseteq M$. Let $\tilde{\sigma} = \phi(\sigma)$ be the image of the vertices of a Riemannian n -simplex $\sigma_M \subset B_\rho$. Label the vertices of $\tilde{\sigma} = \{\tilde{v}_0, \dots, \tilde{v}_n\}$ such that $\tilde{v}_i = \phi(p_i)$, and assume \tilde{v}_0 is at the origin. The affine functions $\lambda_i : u \mapsto \lambda_i(u)$ are the barycentric coordinate functions of $\tilde{\sigma}$. We consider $\text{grad } \mathcal{E}_\lambda$, introduced in Equation (3), now to be a vector field that depends on both $u \in U$ and $x \in B_\rho$. Specifically, we consider the vector field $\nu : U \times B_\rho \rightarrow TM$ defined by

$$\nu(u, x) = - \sum_{i=0}^n \lambda_i(u) v_i(x). \tag{6}$$

Let $b : \tilde{\sigma}_{\mathbb{E}} \rightarrow \sigma_M$ be defined by $b = \mathcal{B}_\sigma \circ \mathcal{L}$, where \mathcal{L} is the canonical linear isomorphism that takes the vertices of $\tilde{\sigma}$ to those of Δ^n , and \mathcal{B}_σ is the barycentric coordinate map introduced in Definition 2. This map is differentiable, by the arguments presented by Buser and Karcher, and $\nu(u, b(u)) = 0$ for all $u \in \tilde{\sigma}_{\mathbb{E}}$. Regarding ν as a vector field along b , its derivative may be expanded as

$$\partial_u \nu + (\nabla \nu) db = 0,$$

where $\partial_u \nu$ denotes the differential of $\nu(u, x)$ with x fixed, $\nabla \nu$ is the covariant differential of $\nu(u, x)$ with u fixed, and db is the differential of b , our barycentric coordinate map onto σ_M , i.e., $db_u : T_u \mathbb{R}^n \rightarrow T_x M$.

Our objective is to exhibit conditions that ensure that db is non-degenerate. It follows from the strict convexity condition (5) of Lemma 1 that the map $\nabla \nu : w \mapsto \nabla_w \nu$ is non-degenerate. Indeed, if $w \in T_x M$ for some $x \in B_\rho$, there is a geodesic $\gamma : I \rightarrow B_\rho$ with $\gamma'(0) = w$, and $\frac{d^2}{dt^2} \mathcal{E}_\lambda(\gamma(t))|_{t=0} = \langle \nabla_w \nu, w \rangle_{\gamma(0)} > 0$. Therefore, we have that

$$db = - (\nabla \nu)^{-1} \partial_u \nu, \tag{7}$$

and thus db is full rank if and only if $\partial_u \nu$ is full rank.

From (6) we observe that when x is fixed, ν is the unique affine map $\mathbb{R}^n \supset U \rightarrow T_x M$, that sends the vertices of $\tilde{\sigma}$ to the corresponding vertices of $\sigma(x)$. In particular, $(\partial_u \nu)_v = (\partial_u \nu)_w$ for all $v, w \in U$. Thus $\partial_u \nu$ is the unique linear map that sends the basis $\{\tilde{v}_i\}$ to $\{(v_i(x) - v_0(x))\}$.

We choose an arbitrary linear isometry to establish a coordinate system on $T_x M$, and let P be the matrix whose i^{th} column is $(v_i(x) - v_0(x))$. Then, if \tilde{P} is the matrix whose i^{th} column is \tilde{v}_i , we obtain [9] the matrix expression for $\partial_u \nu$:

$$\partial_u \nu = -P\tilde{P}^{-1}. \quad (8)$$

From Equation (8) we conclude that $\partial_u \nu$ is full rank if and only if P is of full rank, and this is the case if and only if $\sigma(x)$ is a non-degenerate Euclidean simplex, i.e., its vertices $\{v_i(x)\}$ are affinely independent.

We observe that if db is non-degenerate on σ_M , then b must be injective. Indeed, if $x = b(u)$, then $\{\lambda_i(u)\}$, the barycentric coordinates of u with respect to $\tilde{\sigma}$, are also the barycentric coordinates of the origin in $T_x M$, with respect to the simplex $\sigma(x)$. Thus if $b(u) = x = b(\tilde{u})$, then $\lambda_i(u) = \lambda_i(\tilde{u})$, and we must have $\tilde{u} = u$ by the uniqueness of the barycentric coordinates.

In summary, we have

► **Proposition 4.** *A Riemannian simplex $\sigma_M \subset M$ is non-degenerate if and only if $\sigma(x) \subset T_x M$ is non-degenerate for every $x \in \sigma_M$.*

3.3 Metric distortion of exponential transition

Now we choose the coordinate chart ϕ to be the inverse of the exponential map at some fixed point $p \in B_\rho$. Specifically, we set $\phi = \mathbf{u} \circ \exp_p^{-1} : W \rightarrow \mathbb{R}^m$, where $\mathbf{u} : T_p M \rightarrow \mathbb{R}^m$ is an arbitrary linear isometry that defines the u -coordinate functions in $U = \phi(W)$. The Euclidean simplex $\tilde{\sigma}$ in the coordinate domain can now be identified with $\sigma(p)$.

Our goal now is to estimate the metric distortion incurred when we map a simplex from one tangent space to another via the exponential maps. This will enable us to establish conditions ensuring that $\sigma(x)$ is non-degenerate, based on quality assumptions on $\sigma(p)$. Specifically, we want to bound the difference in the corresponding edge lengths of $\sigma(p)$ and $\sigma(x)$, and since the exponential transition function

$$\exp_x^{-1} \circ \exp_p : T_p M \rightarrow T_x M, \quad (9)$$

maps $\sigma(p)$ to $\sigma(x)$, it suffices to bound the metric distortion of \exp_x^{-1} and \exp_p . This is accomplished by the bounds on the norm of the differential of the exponential map obtained from the Rauch Comparison Theorem (c.f. Buser and Karcher [5, §6.4]). For our purposes the theorem can be stated [9] as:

► **Lemma 5 (Rauch Theorem).** *Suppose the sectional curvatures in M are bounded by $|K| \leq \Lambda$. If $v \in T_p M$ satisfies $|v| = r < \frac{\pi}{2\sqrt{\Lambda}}$, then for any vector $w \in T_v(T_p M) \cong T_p M$, we have*

$$\left(1 - \frac{\Lambda r^2}{6}\right) |w| \leq |(d \exp_p)_v w| \leq \left(1 + \frac{\Lambda r^2}{2}\right) |w|.$$

If $x, p, y \in B_\rho$, with $y = \exp_p(v)$, then $|v| < 2\rho$, and $|\exp_x^{-1}(y)| < 2\rho$, and Lemma 5 tells us that

$$\left\| d(\exp_x^{-1} \circ \exp_p)_v \right\| \leq \left\| (d \exp_x^{-1})_y \right\| \left\| (d \exp_p)_v \right\| \leq 1 + 5\Lambda\rho^2.$$

The image of the line between $v_i(p)$ and $v_j(p)$ in T_pM , under the map $\exp_x^{-1} \circ \exp_p$, is a curve between $v_i(x)$ and $v_j(x)$ in T_xM , whose length is bounded by

$$|v_i(x) - v_j(x)| \leq (1 + 5\Lambda\rho^2) |v_i(p) - v_j(p)|.$$

We can do the same argument the other way, so

$$|v_i(p) - v_j(p)| \leq (1 + 5\Lambda\rho^2) |v_i(x) - v_j(x)|,$$

and we find

$$\begin{aligned} \left| |v_i(x) - v_j(x)| - |v_i(p) - v_j(p)| \right| &\leq 5\Lambda\rho^2(1 + 5\Lambda\rho^2) |v_i(p) - v_j(p)| \\ &\leq 21\Lambda\rho^2 |v_i(p) - v_j(p)| \quad \text{when } \rho < \rho_0. \end{aligned}$$

Letting P be the matrix associated with $\sigma(p)$, and using $C_0 = 21\Lambda\rho^2$, in Lemma 3, we find that the matrix \tilde{P} associated with $\sigma(x)$ in Proposition 4 is non-degenerate if $\sigma(p)$ satisfies a thickness bound of $t_0 > 10\sqrt{\Lambda}\rho$, and we have

► **Theorem 6 (Non-degeneracy criteria).** *Suppose M is a Riemannian manifold with sectional curvatures bounded by $|K| \leq \Lambda$, and σ_M is a Riemannian simplex, with $\sigma_M \subset B_\rho \subset M$, where B_ρ is an open geodesic ball of radius ρ with*

$$\rho < \rho_0 = \min \left\{ \frac{\iota}{2}, \frac{\pi}{4\sqrt{\Lambda}} \right\}.$$

Then σ_M is non-degenerate if there is a point $p \in B_\rho$ such that the lifted Euclidean simplex $\sigma(p)$ has thickness satisfying

$$t(\sigma(p)) > 10\sqrt{\Lambda}\rho.$$

The ball B_ρ may be chosen so that this inequality is necessarily satisfied if

$$t(\sigma(p)) > 10\sqrt{\Lambda}L(\sigma_M), \tag{10}$$

where $L(\sigma_M)$ is the geodesic length of the longest edge in σ_M .

The last assertion follows from the remark following Definition 2: If $L(\sigma_M) < \rho_0$, then σ_M is contained in a closed ball of radius $L(\sigma_M)$ centred at one of the vertices.

4 Triangulation criteria

Suppose we have a finite set of points S in a compact Riemannian manifold M , and an (abstract) simplicial complex \mathcal{A} whose vertex set is S , and such that every simplex in \mathcal{A} defines a non-degenerate Riemannian simplex. When can we be sure that \mathcal{A} triangulates M ? Consider a convex ball B_ρ centred at $p \in S$. We require that, when lifted to T_pM , the simplices near p triangulate a neighbourhood of the origin. If we require that the simplices be small relative to ρ , and triangulate a region that extends to near the boundary of the lifted ball, then Riemannian simplices outside of B_ρ cannot have points in common with the simplices near the centre of the ball, and it is relatively easy to establish a triangulation.

Instead, we aim for finer local control on the geometry. We establish conditions (Lemma 7) that ensure that the complex consisting of simplices incident to p , (i.e., the star of p) is embedded. In order to achieve this, we require finer control on the differential of the map into the manifold than bounds on its singular values.

We are interested in smooth maps from non-degenerate closed Euclidean simplices of dimension n into an n -dimensional manifold M . We will work within coordinate charts, so our primary focus will be on maps of the form $F : \sigma_{\mathbb{E}}^n \rightarrow \mathbb{R}^n$. Requiring that F be smooth on the closed set $\sigma_{\mathbb{E}}^n$ means that its partial derivatives are continuous on $\sigma_{\mathbb{E}}^n$. Equivalently, F can be extended to a smooth map on an open neighbourhood of $\sigma_{\mathbb{E}}^n$. We demand that dF_u is always close to the *same* linear isometry $T : \mathbb{R}^n \rightarrow \mathbb{R}^n$ for all $u \in \sigma_{\mathbb{E}}^n$:

$$\|dF_u - T\| \leq \eta. \tag{11}$$

This is a stronger constraint than can be obtained by a bound of the form $(1 - \eta) |w| \leq |dF_u w| \leq (1 + \eta) |w|$, as in the Rauch theorem (Lemma 5). In this latter case we can only say that $\|dF_u - T_u\| \leq \eta$, where T_u is a linear isometry that depends on u .

A simplicial complex \mathcal{C} is embedded in \mathbb{R}^n if the vertices lie in \mathbb{R}^n and the convex hulls of any two simplices in \mathcal{C} either do not intersect, or their intersection is the convex hull of a simplex in \mathcal{C} . We identify $|\mathcal{C}|$, the carrier of \mathcal{C} , with the union of these geometric simplices; the complex naturally inherits a piecewise flat metric from the embedding.

If p is a vertex in \mathcal{C} , we define the *star of p* to be the subcomplex $\text{star}(p)$ of \mathcal{C} consisting of all simplices that contain p , together with the faces of these simplices. We say that $\text{star}(p)$ is a *full star* if $|\text{star}(p)|$ is a closed topological ball of dimension n with p in its interior, and \mathcal{C} contains no simplices of dimension greater than n .

The *scale* of \mathcal{C} is an upper bound on the length of the longest edge in \mathcal{C} , and is denoted by h . We say that \mathcal{C} is t_0 -*thick* if each simplex in \mathcal{C} has thickness greater than t_0 . The *dimension* of \mathcal{C} is the largest dimension of the simplices in \mathcal{C} . We call a complex of dimension n an *n -complex*. If every simplex in \mathcal{C} is the face of an n -simplex, then \mathcal{C} is a *pure n -complex*.

A map $F : |\mathcal{C}| \rightarrow \mathbb{R}^n$ is *smooth on \mathcal{C}* if for each $\sigma \in \mathcal{C}$ the restriction $F|_{\sigma_{\mathbb{E}}}$ is smooth. This means that $d(F|_{\sigma_{\mathbb{E}}})$ is well defined, and even though dF is not well defined, we will use this symbol when the particular restriction employed is either evident or unimportant. When the underlying complex on which F is smooth is unimportant, we simply say that F is *piecewise smooth*.

The strong constraint on the differential allows us to ensure that thick stars are embedded:

► **Lemma 7** (Embedding a star). *Suppose $\mathcal{C} = \text{star}(p)$ is a t_0 -thick, pure n -complex embedded in \mathbb{R}^n such that all of the n -simplices are incident to a single vertex, p , and $p \in \text{int}(|\mathcal{C}|)$ (i.e., $\text{star}(p)$ is a full star). If $F : |\mathcal{C}| \rightarrow \mathbb{R}^n$ is smooth on \mathcal{C} , and satisfies*

$$\|dF - \text{Id}\| < nt_0 \tag{12}$$

on each n -simplex of \mathcal{C} , then F is an embedding.

The proof [9, Lem. 14] hinges on the fact that thickness provides a lower bound on the angle between a radial ray from p and a facet on the boundary of $\text{star}(p)$. Together with the bound on the differential of F , this enables us to demonstrate that the boundary of $\text{star}(p)$ is embedded by F . Then, since each simplex individually is embedded by F , topological considerations imply that $\text{star}(p)$ itself is embedded by F .

We use this observation to establish conditions that ensure that a map $H : |\mathcal{A}| \rightarrow M$ is a homeomorphism. If H is such that for every vertex in \mathcal{A} , the restriction of H to $|\text{star}(p)|$ is an embedding, then H is a covering map. So if H is injective, it is a triangulation. Injectivity is established by constraining the size of the simplices relative to the injectivity radius of M , and by implicitly constraining the metric distortion associated with H . We obtain the following proposition, which generically models the situation we will work with when we describe a triangulation by Riemannian simplices:

► **Proposition 8** (Triangulation). *Let \mathcal{A} be a manifold simplicial n -complex with finite vertex set S , and M a compact Riemannian manifold with an atlas $\{(W_p, \phi_p)\}_{p \in S}$ indexed by S . Suppose $H : |\mathcal{A}| \rightarrow M$ satisfies:*

1. *For each $p \in S$ the secant map of $\phi_p \circ H$ restricted to $|\text{star}(p)|$ is a piecewise linear embedding $\mathcal{L}_p : |\text{star}(p)| \rightarrow \mathbb{R}^n$ such that each simplex $\sigma \in \mathcal{C}_p = \mathcal{L}_p(\text{star}(p))$ is t_0 -thick, and $|\mathcal{C}_p| \subset B_{\mathbb{R}^n}(\mathcal{L}_p(p); h)$, with $\mathcal{L}_p(p) \in \text{int}(|\mathcal{C}_p|)$. The scale parameter h must satisfy $h < \frac{\iota}{4}$, where ι is the injectivity radius of M .*
2. *For each $p \in S$, $\phi_p : W_p \xrightarrow{\cong} U_p \subset \mathbb{R}^n$ is such that $\bar{B} = \bar{B}_{\mathbb{R}^n}(\mathcal{L}_p(p); \frac{3}{2}h) \subseteq U_p$, and $\|(d\phi_p^{-1})_u\| \leq \frac{4}{3}$, for every $u \in \bar{B}$.*
3. *The map*

$$F_p = \phi_p \circ H \circ \mathcal{L}_p^{-1} : |\mathcal{C}_p| \rightarrow \mathbb{R}^n$$

satisfies

$$\|(dF_p)_u - \text{Id}\| \leq \frac{nt_0}{2}$$

on each n -simplex $\sigma \in \mathcal{C}_p$, and every $u \in \sigma_{\mathbb{E}}$.

Then H is a smooth triangulation of M .

Proof. By Lemma 7, F_p is a homeomorphism onto its image. It follows then that $H|_{|\text{star}(p)|}$ is an embedding for every $p \in S$. Therefore, since $|\mathcal{A}|$ is compact, $H : |\mathcal{A}| \rightarrow M$ is a covering map.

Given $x \in |\mathcal{A}|$, with $x \in \sigma_{\mathbb{E}}$, and p a vertex of $\sigma_{\mathbb{E}}$, let $\tilde{x} = \mathcal{L}_p(x) \in |\mathcal{C}_p|$. Then the bound on dF implies that $|F_p(\tilde{x}) - \mathcal{L}_p(p)| \leq (1 + \frac{nt_0}{2})h \leq \frac{3}{2}h$, so $F_p(\tilde{x}) \in \bar{B}$. Since $\phi_p^{-1} \circ F_p(\tilde{x}) = H(x)$, and

$$|(d\phi_p^{-1})_{F_p(\tilde{x})}(dF_p)_u| \leq \frac{4}{3} \left(1 + \frac{nt_0}{2}\right) \leq 2$$

for any $u \in \sigma_{\mathbb{E}} \subset |\mathcal{C}_p|$, we have that $d_M(H(p), H(x)) \leq 2h$.

Suppose $y \in |\mathcal{A}|$ with $H(y) = H(x)$. Let $\tau \in \mathcal{A}$ with $y \in \tau_{\mathbb{E}}$, and $q \in \tau$ a vertex. Then $d_M(H(p), H(q)) \leq 4h < \iota$. Thus there is a path γ from $H(x)$ to $H(p)$ to $H(q)$ to $H(y) = H(x)$ that is contained in the topological ball $B_M(H(p); \iota)$, and is therefore null-homotopic. Since H is a covering map, this implies that $x = y$. Thus H is injective, and therefore defines a smooth triangulation. ◀

In the context of the barycentric coordinate mapping defining Riemannian simplices, we obtain the desired strong bound on the differential by means of a refinement of the Rauch theorem due to Buser and Karcher [5, §6.4], which for our purposes may be stated as:

► **Lemma 9** (Strong Rauch Theorem). *Assume the sectional curvatures on M satisfy $|K| \leq \Lambda$, and suppose there is a unique minimising geodesic between x and p . If $v = \exp_p^{-1}(x)$, and*

$$|v| = d_M(p, x) = r \leq \frac{\pi}{2\sqrt{\Lambda}},$$

then

$$\|(d\exp_p)_v - T_{xp}\| \leq \frac{\Lambda r^2}{2},$$

where T_{xp} denotes the parallel transport operator along the unique minimising geodesic from p to x .

Given three points $x, y, p \in B_\rho$ in a convex ball, we use further results of Buser and Karcher [5, §6] to obtain a bound on $\|T_{xp} - T_{xy}T_{yp}\|$ with respect to ρ and a bound on the absolute value of the sectional curvatures. This result together with Lemma 9 yields a bound of the desired form (11) on the differential of exponential transition functions:

► **Proposition 10** (Strong exponential transition bound). *Suppose the sectional curvatures on M satisfy $|K| \leq \Lambda$. Let $v \in T_pM$, with $y = \exp_p(v)$. If $x, y \in B_M(p; \rho)$, with*

$$\rho < \frac{1}{2}\rho_0 = \frac{1}{2} \min \left\{ \frac{\iota}{2}, \frac{\pi}{4\sqrt{\Lambda}} \right\},$$

then

$$\|d(\exp_x^{-1} \circ \exp_p)_v - T_{xp}\| \leq 6\Lambda\rho^2.$$

Proposition 10 in turn allows us to obtain the desired form of bound on the differential (7) of the barycentric coordinate map so that we can exploit Proposition 8 to obtain sampling criteria for triangulating a Riemannian manifold, our main result:

► **Theorem 11.** *Suppose M is a compact n -dimensional Riemannian manifold with sectional curvatures K bounded by $|K| \leq \Lambda$, and \mathcal{A} is an abstract simplicial complex with finite vertex set $S \subset M$. Fix a thickness bound $t_0 > 0$, and let*

$$h = \min \left\{ \frac{\iota}{4}, \frac{\sqrt{nt_0}}{6\sqrt{\Lambda}} \right\}. \quad (13)$$

If

1. for every $p \in S$, the vertices of $\text{star}(p)$ are contained in $B_M(p; h)$, and the balls $\{B_M(p; h)\}_{p \in S}$ cover M ;
2. for every $p \in S$, the restriction of the inverse of the exponential map \exp_p^{-1} to the vertices of $\text{star}(p) \subset \mathcal{A}$ defines a piecewise linear embedding of $|\text{star}(p)|$ into T_pM , realising $\text{star}(p)$ as a full star such that every simplex $\sigma(p)$ has thickness $t(\sigma(p)) \geq t_0$,

then \mathcal{A} triangulates M , and the triangulation is given by the barycentric coordinate map on each simplex.

5 The piecewise flat metric

The complex \mathcal{A} described in Theorem 11 naturally inherits a piecewise flat metric from the construction. The length assigned to an edge $\{p, q\} \in \mathcal{A}$ is the geodesic distance in M between its endpoints: $\ell_{pq} = d_M(p, q)$. We first describe conditions which ensure that this assignment of edge lengths does indeed make each $\sigma \in \mathcal{A}$ isometric to a Euclidean simplex. With this piecewise flat metric on \mathcal{A} , the barycentric coordinate map is a bi-Lipschitz map between metric spaces $H : |\mathcal{A}| \rightarrow M$, and we estimate the metric distortion of this map.

If G is a symmetric positive definite $n \times n$ matrix, then it can be written as a Gram matrix, $G = P^T P$ for some $n \times n$ matrix P . Then P describes a Euclidean simplex with one vertex at the origin, and the other vertices defined by the column vectors. The matrix P is not unique, but if $G = Q^T Q$, then $Q = OP$ for some linear isometry O . Thus a symmetric positive definite matrix defines a Euclidean simplex, up to isometry.

If $\sigma = \{p_0, \dots, p_n\} \subset B_\rho$, is the vertex set of a Riemannian simplex σ_M , we define the numbers $\ell_{ij} = d_M(p_i, p_j)$. These are the edge lengths of a Euclidean simplex $\sigma_{\mathbb{E}}$ if and only if the matrix G defined by

$$G_{ij} = \frac{1}{2}(\ell_{0i}^2 + \ell_{0j}^2 - \ell_{ij}^2) \quad (14)$$

is positive definite.

The same kind of argument that bounds the thickness of a simplex subjected to small distortions of its edge lengths, Lemma 3, allows us to ensure that the numbers ℓ_{ij} do define a Euclidean simplex $\sigma_{\mathbb{E}}$ if they are close enough to the edge lengths of a Euclidean simplex, $\sigma(p)$ whose thickness is bounded below. Then, again exploiting the Rauch Theorem 5, we find we need a slightly tighter bound on the scale parameter in order to ensure that \mathcal{A} admits a piecewise flat metric:

► **Proposition 12.** *If the requirements of Theorem 11 are satisfied when the scale parameter (13) is replaced with*

$$h = \min \left\{ \frac{\iota}{4}, \frac{t_0}{6\sqrt{\Lambda}} \right\},$$

then the geodesic distances between the endpoints of the edges in \mathcal{A} define a piecewise flat metric on \mathcal{A} such that each simplex $\sigma \in \mathcal{A}$ satisfies

$$t(\sigma) > \frac{3}{4\sqrt{n}}t_0.$$

In the context of Theorem 11 the barycentric coordinate map on each simplex defines a piecewise smooth homeomorphism $H : |\mathcal{A}| \rightarrow M$. If the condition of Proposition 12 is also met, then \mathcal{A} is naturally endowed with a piecewise flat metric. We wish to compare this metric with the Riemannian metric on M . It suffices to consider an n -simplex $\sigma \in \mathcal{A}$, and establish bounds on the singular values of the differential dH . If $p \in \sigma$, then we can write $H|_{\sigma_{\mathbb{E}}} = b \circ \mathcal{L}_p$, where $\mathcal{L}_p : \sigma_{\mathbb{E}} \rightarrow \sigma_{\mathbb{E}}(p)$ is the linear map that sends $\sigma \in \mathcal{A}$ to $\sigma(p) \in T_pM$.

A bound on the metric distortion of a linear map that sends one Euclidean simplex to another is a consequence of the following (reformulation of [2, Lemma 9]):

► **Lemma 13** (Linear distortion bound). *Suppose that P and \tilde{P} are non-degenerate $k \times k$ matrices such that*

$$\tilde{P}^T \tilde{P} = P^T P + E. \tag{15}$$

Then there exists a linear isometry $\Phi : \mathbb{R}^k \rightarrow \mathbb{R}^k$ such that

$$\|\tilde{P}P^{-1} - \Phi\| \leq \frac{s_1(E)}{s_k(P)^2}.$$

Taking P and \tilde{P} to represent $\sigma_{\mathbb{E}}(p)$ and $\sigma_{\mathbb{E}}$, we can bound $|\mathcal{L}_p|$ and $|\mathcal{L}_p^{-1}|$, and combined with the bounds on db that we have already estimated, we obtain the desired bounds on dH , and we find:

► **Theorem 14** (Metric distortion). *If the requirements of Theorem 11, are satisfied with the scale parameter (13) replaced by*

$$h = \min \left\{ \frac{\iota}{4}, \frac{t_0}{6\sqrt{\Lambda}} \right\},$$

then \mathcal{A} is naturally equipped with a piecewise flat metric $d_{\mathcal{A}}$ defined by assigning to each edge the geodesic distance in M between its endpoints.

If $H : |\mathcal{A}| \rightarrow M$ is the triangulation defined by the barycentric coordinate map in this case, then the metric distortion induced by H is quantified as

$$|d_M(H(x), H(y)) - d_{\mathcal{A}}(x, y)| \leq \frac{50\Lambda h^2}{t_0^2} d_{\mathcal{A}}(x, y),$$

for all $x, y \in |\mathcal{A}|$.

6 Discussion

Traditional demonstrations that smooth manifolds can be triangulated [6, 17, 18] involve establishing a lower bound on simplex quality that is invariant under some kind of refinement operation, and showing that a triangulation will be achieved when the scale parameter is sufficiently small. Theorem 11 provides a means to explicitly quantify “sufficiently small” in this context. Similarly, an analysis of more recent triangulation algorithms in computational geometry [8, 4] could exploit Theorem 11 to quantify a sufficient sampling density.

We refer to the criteria of Theorem 11 as sampling criteria, even though they require a simplicial complex for their definition. Although there is no explicit constraint on the minimal distance between points of S , one is implicitly imposed by the quality constraint on the Riemannian simplices. The required sampling density depends on the quality of the Riemannian simplices, which leaves open the question of what kind of quality of simplices can we hope to attain. A Delaunay complex conforming to the requirements of Theorem 11 can be constructed [2] with the thickness t_0 bounded by $2^{-\mathcal{O}(n^3)}$, and even in flat manifolds, e.g., Euclidean space, the situation is not better in general [7], but in this case, at least in dimension 3, dramatic improvements can be made if the placement of sample points can be structured according to a lattice [11].

More work needs to be done to understand the limitations imposed by the thickness bound t_0 that appears in the density constraint (13), but there is another aspect to the bound that merits more attention. The non-degeneracy criterion established in Theorem 6 demands that the Riemannian simplices be “almost flat”. In other words, if the bound on the absolute value of the sectional curvatures in the neighbourhood is very large, then the simplex must be very small. However, we know that in spaces of constant curvature, where the Riemannian simplex coincides with the usual definition of a simplex as the convex hull of its vertices, the simplices are not constrained to be small. In hyperbolic space the edge lengths of a non-degenerate simplex can be arbitrarily large. It seems that a more refined bound on the scale should depend on the amount the sectional curvatures deviate from some fixed constant, that need not be 0. Given upper and lower bounds Λ_+ and Λ_- on the sectional curvatures, our preliminary unpublished calculations demonstrate a bound on simplex quality for non-degeneracy involving $\Lambda_+ - \Lambda_-$ when $\Lambda_- > 0$. The same analysis in the hyperbolic setting ($\Lambda_+ < 0$) yields a more complicated expression.

References

- 1 M. Berger. *A Panoramic View of Riemannian Geometry*. Springer-Verlag, 2003.
- 2 J.-D. Boissonnat, R. Dyer, and A. Ghosh. Delaunay triangulation of manifolds. Research Report RR-8389, INRIA, 2013. (also: arXiv:1311.0117).
- 3 J.-D. Boissonnat, R. Dyer, and A. Ghosh. The stability of Delaunay triangulations. *IJCGA*, 23(04n05):303–333, 2013. (Preprint: arXiv:1304.2947).
- 4 J.-D. Boissonnat and A. Ghosh. Manifold reconstruction using tangential Delaunay complexes. *Discrete and Computational Geometry*, 51(1):221–267, 2014.
- 5 P. Buser and H. Karcher. *Gromov’s almost flat manifolds*, volume 81 of *Astérisque*. Société mathématique de France, 1981.
- 6 S. S. Cairns. On the triangulation of regular loci. *Annals of Mathematics. Second Series*, 35(3):579–587, 1934.
- 7 S.-W. Cheng, T. K. Dey, H. Edelsbrunner, M. A. Facello, and S. H. Teng. Sliver exudation. *Journal of the ACM*, 47(5):883–904, 2000.
- 8 S.-W. Cheng, T. K. Dey, and E. A. Ramos. Manifold reconstruction from point samples. In *SODA*, pages 1018–1027, 2005.

- 9 R. Dyer, G. Vegter, and M. Wintraecken. Riemannian simplices and triangulations. *Geometriae Dedicata*, 2015. To appear. (Preprint: arXiv:1406.3740).
- 10 H. Karcher. Riemannian center of mass and mollifier smoothing. *Communications on Pure and Applied Mathematics*, 30:509–541, 1977.
- 11 F. Labelle and J. R. Shewchuk. Isosurface stuffing: Fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph.*, 26(3), 2007.
- 12 J. R. Munkres. *Elementary differential topology*. Princeton University press, second edition, 1968.
- 13 R.M. Rustamov. Barycentric coordinates on surfaces. *Eurographics Symposium of Geometry Processing*, 29(5), 2010.
- 14 O. Sander. Geodesic finite elements on simplicial grids. *International Journal for Numerical Methods in Engineering*, 92(12):999–1025, 2012.
- 15 W. P. Thurston. *Three-Dimensional Geometry and Topology*. Princeton University Press, 1997.
- 16 S. W. von Deylen. *Numerische Approximation in Riemannschen Mannigfaltigkeiten mithilfe des Karcher’schen Schwerpunktes*. PhD thesis, Freie Universität Berlin, 2014 (to appear).
- 17 J. H. C. Whitehead. On C^1 -complexes. *Annals of Mathematics*, 41(4), 1940.
- 18 H. Whitney. *Geometric Integration Theory*. Princeton University Press, 1957.

An Edge-Based Framework for Enumerating 3-Manifold Triangulations*

Benjamin A. Burton and William Pettersson

School of Mathematics and Physics, The University of Queensland
Brisbane QLD 4072, Australia
bab@maths.uq.edu.au, william@ewpettersson.se

Abstract

A typical census of 3-manifolds contains all manifolds (under various constraints) that can be triangulated with at most n tetrahedra. Although censuses are useful resources for mathematicians, constructing them is difficult: the best algorithms to date have not gone beyond $n = 12$. The underlying algorithms essentially (i) enumerate all relevant 4-regular multigraphs on n nodes, and then (ii) for each multigraph G they enumerate possible 3-manifold triangulations with G as their dual 1-skeleton, of which there could be exponentially many. In practice, a small number of multigraphs often dominate the running times of census algorithms: for example, in a typical census on 10 tetrahedra, almost half of the running time is spent on just 0.3% of the graphs.

Here we present a new algorithm for stage (ii), which is the computational bottleneck in this process. The key idea is to build triangulations by recursively constructing neighbourhoods of edges, in contrast to traditional algorithms which recursively glue together pairs of tetrahedron faces. We implement this algorithm, and find experimentally that whilst the overall performance is mixed, the new algorithm runs significantly faster on those “pathological” multigraphs for which existing methods are extremely slow. In this way the old and new algorithms complement one another, and together can yield significant performance improvements over either method alone.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory, G.4 Mathematical Software

Keywords and phrases triangulations, enumeration, graph theory

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.270

1 Introduction

In many fields of mathematics, one can often learn much by studying an exhaustive “census” of certain objects, such as knot dictionaries. Our focus here is on censuses of closed 3-manifolds – essentially topological spaces that locally look like \mathbb{R}^3 . Combinatorially, any closed 3-manifold can be represented by a *triangulation*, formed from tetrahedra with faces identified together in pairs [14]. A typical census of 3-manifolds enumerates all 3-manifolds under certain conditions that can be constructed from a fixed number of tetrahedra.

One of the earliest such results was a census of all cusped hyperbolic 3-manifolds which could be built from at most five tetrahedra, by Hildebrand and Weeks [7]; this was later extended to all such manifolds on at most nine tetrahedra [4, 6, 16]. For closed orientable 3-manifolds, Matveev gave the first census of closed orientable prime manifolds on up to six tetrahedra [11]; this has since been extended to 12 tetrahedra [9, 12].

* Partially supported by the Australian Research Council (projects DP1094516, DP110101104). A detailed version of this paper appears at <http://arxiv.org/abs/1412.2169>



Most (if not all) census algorithms in the literature enumerate 3-manifolds on n tetrahedra in two main stages. The first stage is to generate a list of all 4-regular multigraphs on n nodes. The second stage takes each such graph G , and sequentially identifies faces of tetrahedra together to form a triangulation with G as its dual 1-skeleton (for a highly tuned implementation of such an algorithm, see [5]).

There are $|S_3| = 6$ possible maps to use for each such identification of faces. Thus for each graph G , the algorithm searches through an exponential (in the number of tetrahedra) search tree where each leaf in this tree represents a triangulation but not necessarily a 3-manifold triangulation. Much research has focused on pruning this search tree by identifying and removing subtrees which only contain non-3-manifold triangulations [2, 3, 10, 11].

In this paper we describe a different approach to generating a census of 3-manifolds. The first stage remains the same, but in the second stage we build up the neighbourhood of each *edge* in the triangulation recursively, instead of joining together faces one at a time. This is, in a sense, a paradigm shift in census enumeration, and as a result it generates significantly different search trees with very different opportunities for pruning. We implement the new algorithm in a specific setting (potential minimal triangulations), and compare its performance against existing algorithms. We find that this new search framework complements existing algorithms very well, and we predict that a heuristic combination that combines the benefits of this with existing algorithms can significantly speed up census enumeration.

The key idea behind this new search framework is to extend each possible dual 1-skeleton graph to a “fattened face pairing graph”, and then to find particular cycle-based decompositions of these new graphs. We also show how various improvements to typical census algorithms (such as those in [1]) can be translated into this new setting.

2 Definitions and notation

In combinatorial topology versus graph theory, the terms “edge” and “vertex” have distinct meanings. Therefore in this paper, the terms *edge* and *vertex* will be used to mean an edge or vertex of a tetrahedron, triangulation or manifold; and the terms *arc* and *node* will be used to mean an edge or vertex in a graph respectively.

A 3-manifold is a topological space that locally looks like either 3-dimensional Euclidean space (i.e., \mathbb{R}^3) or closed 3-dimensional Euclidean half-space (i.e., $\mathbb{R}_{z \geq 0}^3$). In this paper when we mention 3-manifolds we always mean compact and connected 3-manifolds. When we refer to faces, we are explicitly talking about 2-faces (i.e., facets of a tetrahedron). We represent 3-manifolds combinatorially as triangulations [14]: a collection of tetrahedra (3-simplices) with some 2-faces pairwise identified.

► **Definition 1.** A *general triangulation* is a collection $\Delta_1, \Delta_2, \dots, \Delta_n$ of n abstract tetrahedra, along with bijections $\pi_1, \pi_2, \dots, \pi_m$ where each π_i is an affine map between two faces of tetrahedra, and where each face of each tetrahedron is in at most one such bijection.

We call these affine bijections *face identifications* or simply *identifications*. Note that unlike simplicial complexes, we do allow identifications between two distinct faces of the same tetrahedron. If the quotient space of such a triangulation is a 3-manifold, we will say that the triangulation represents said 3-manifold.

► **Notation 2.** Given a tetrahedron with vertices a, b, c and d , we will define *face a* to be the face opposite vertex a . That is, *face a* is the face consisting of vertices b, c and d . We will sometimes also refer to this as *face bcd* . We will write $abc \leftrightarrow efg$ to mean that face abc is

identified with face efg and that in this identification we have vertex a identified with vertex e , vertex b identified with vertex f and vertex c identified with vertex g .

We will also use the notation ab to denote the edge joining vertices a and b on some tetrahedron. Note that by this notation, the edge ab on a tetrahedron with vertices labelled a , b , c and d will be the intersection of faces c and d .

As a result of the identification of various faces, some edges or vertices of various tetrahedra are identified together. The *degree* of an edge of the triangulation, denoted $\deg(e)$, is defined to be the number of edges of tetrahedra which are identified together to form the edge of the triangulation.

We also need to define the *link* of a vertex before we can discuss triangulations of 3-manifolds.

► **Definition 3.** Given a vertex v in some triangulation, the link of v , denoted $Link(v)$, is the (2-dimensional) frontier of a small regular neighbourhood of v .

We now detail the properties a general triangulation must have to represent a 3-manifold. Recall that we only discuss connected 3-manifolds.

► **Lemma 4.** A general triangulation represents a 3-manifold if the following additional conditions hold:

- the triangulation is connected;
- the link of any vertex in the triangulation is homeomorphic to either a 2-sphere or a disc;
- no edge in the triangulation is identified with itself in reverse.

We will call such a triangulation a *3-manifold triangulation*. It is straight-forward to show that these conditions are both necessary and sufficient for the underlying topological space to be a 3-manifold (possibly with boundary). However in this paper we only consider 3-manifolds without boundary. That is, every face of a tetrahedron will be identified with some other face in a 3-manifold triangulation.

► **Lemma 5.** Given any connected closed triangulation T on n tetrahedra with k vertices where no edge is identified with itself in reverse, the triangulation has $n + k$ edges if and only if the link of each vertex in T is homeomorphic to a 2-sphere.

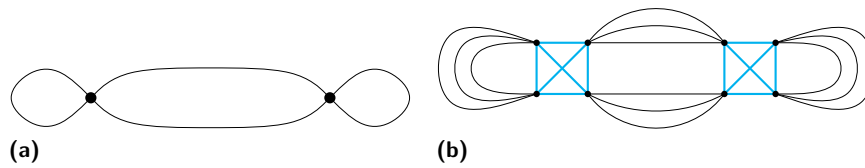
The above result is routine to show. We also need to define the *face pairing graph* of a triangulation. The *face pairing graph* of a triangulation, also known as the dual 1-skeleton, is a graphical representation of the face identifications of the triangulation. Each tetrahedron is associated with a node in the face pairing graph, and one arc joins a pair of tetrahedra for each identification of faces between the two tetrahedra. Note that face pairing graph is not necessarily a simple graph. Indeed, it will often contain both loops (when there is an identification of two distinct faces of the same tetrahedron) and parallel arcs (when there are multiple face identifications between two tetrahedra).

Lastly, we need a few properties of manifolds and triangulations.

► **Definition 6.** A 3-manifold \mathcal{M} is *irreducible* if every embedded 2-sphere in \mathcal{M} bounds a 3-ball in \mathcal{M} .

► **Definition 7.** A 3-manifold \mathcal{M} is *prime* if it cannot be written as a connected sum of two manifolds where neither is a 3-sphere.

► **Definition 8.** A 3-manifold is \mathbb{P}^2 -*irreducible* if it is irreducible and also contains no embedded two-sided projective plane.



■ **Figure 1** The face pairing graph (a) and fattened face pairing graph (b) of a triangulation. Note that the blue arcs are internal arcs, while the black arcs are external arcs.

Prime manifolds are the most fundamental manifolds to work with. We note that prime 3-manifolds are either irreducible, or are one of the orientable direct product $S^2 \times S^1$ or the non-orientable twisted product $S^2 \tilde{\times} S^1$. As these are both well known and have triangulations on two tetrahedra, for any census of minimal triangulations on three or more tetrahedra we can interchange the conditions “prime” and “irreducible”. Any non-prime manifold can be constructed from a connected sum of prime manifolds, so enumerating prime manifolds is sufficient for most purposes. A similar (but more complicated) notion holds for \mathbb{P}^2 -irreducible manifolds in the non-orientable setting. As such, minimal prime \mathbb{P}^2 -irreducible triangulations form the basic building blocks in combinatorial topology.

► **Definition 9.** A 3-manifold triangulation of a manifold \mathcal{M} is *minimal* if \mathcal{M} cannot be triangulated with fewer tetrahedra.

Minimal triangulations are well studied, both for their relevance to computation and for their applications in zero-efficient triangulations [8]. Martelli and Petronio [10] also showed that, with the exceptions S^3 , RP^3 and $L_{3,1}$, the minimal number of tetrahedra required to triangulate a closed, irreducible and \mathbb{P}^2 -irreducible 3-manifold \mathcal{M} is equal to the *Matveev complexity* [12] of \mathcal{M} .

3 Manifold decompositions

In this section we define a fattened face pairing graph, and show how we can represent any general triangulation as a specific decomposition of its fattened face pairing graph. This allows us to enumerate general triangulations by enumerating graph decompositions. We then demonstrate how to restrict this process to only enumerate 3-manifold triangulations.

A *fattened face pairing graph* is an extension of a face pairing graph F which we use in a dual representation of the corresponding triangulation. Instead of one node for each tetrahedron, a fattened face pairing graph contains one node for each face of each tetrahedron. Additionally, a face identification in the triangulation is represented by *three* arcs in the fattened face pairing graph; these three arcs loosely correspond to the three pairs of edges which are identified as a consequence of the face identification.

► **Definition 10.** Given a face pairing graph F , a fattened face pairing graph is constructed by first tripling each arc (i.e., for each arc e in F , add two more arcs parallel to e), and then replacing each node ν of F with a copy of K_4 such that each node of the K_4 is incident with exactly one set of triple arcs that meet ν .

► **Example 11.** Figure 1 shows a face pairing graph and the resulting fattened face pairing graph. The arcs shown in blue are what we call *internal* arcs. Each original node has been replaced with a copy of K_4 and in place of each original arc a set of three parallel arcs have been added.

We will refer to the arcs of each K_4 as *internal arcs*, and the remaining arcs (coming from the triple edges) as *external arcs*. As a visual aid we will draw internal arcs in blue. Each such K_4 represents a tetrahedron in the associated triangulation, and as such we will say that a fattened face pairing graph has n tetrahedra if it contains $4n$ nodes.

Triangulations are often labelled or indexed in some manner. Given any labelling of the tetrahedra and their vertices, we label the corresponding fattened face pairing graph as follows. For each tetrahedron i with faces a , b , c and d , we label the nodes of the corresponding K_4 in the fattened face pairing graph $v_{i,a}$, $v_{i,b}$, $v_{i,c}$ and $v_{i,d}$, such that if face a of tetrahedron i is identified with face b of tetrahedron j then there are three parallel external arcs between $v_{i,a}$ and $v_{j,b}$.

In such a labelling, the node $v_{i,a}$ represents face a of tetrahedron i . Each internal arc $\{v_{i,a}, v_{i,b}\}$ represents the unique edge common to faces a and b of tetrahedron i . Each external arc $\{v_{i,a}, v_{j,b}\}$ represents one of the three pairs of edges of tetrahedra which become identified as a result of identifying face a of tetrahedron i with face b of tetrahedron j . Note that the arc only represents the pair of edges being identified, and does not indicate the orientation of said identification.

We now define *ordered decompositions* of fattened face pairing graphs. Later, we show that there is a natural correspondence between such a decomposition and a general triangulation, and we show exactly how the 3-manifold constraints on general triangulations (see Lemma 4) can be translated to constraints on these decompositions. There is also a natural relationship between such decompositions and *spines* of 3-manifolds, as used by Matveev and others [12]; we touch on this relationship again later in this section.

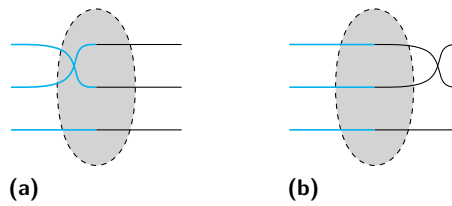
► **Definition 12.** An *ordered decomposition* of a fattened face pairing graph $F = (E, V)$ is a set of closed walks $\{P_1, P_2, \dots, P_n\}$ such that:

- $\{P_1, P_2, \dots, P_n\}$ partition the arc set E ;
- P_i is a closed walk of even length for each i ; and
- if arc e_{j+1} immediately follows arc e_j in one of the walks then exactly one of e_j or e_{j+1} is an internal arc.

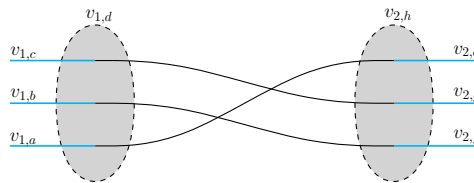
An ordered decomposition of a fattened face pairing graph exactly describes a general triangulation. We outline this idea here by showing how three parallel external arcs can represent an identification of faces. Complete technical details are given in the full version of this paper.

Since the ordered decomposition consists of closed walks of alternating internal and external arcs, the decomposition pairs up the six arcs exiting each node so that each external arc is paired with exactly one internal arc. To help visualise this, we can draw such nodes as larger ellipses, with three external arcs and three internal arcs entering the ellipse, as in Figure 2. Each external arc meets exactly one internal arc inside this ellipse. This only represents how such arcs are paired up in a given decomposition – the node is still incident with all six arcs. We also see in Figure 2 that the fattened face pairing graph can always be drawn such that any “crossings” of arcs only occur between external arcs. Such crossings are simply artefacts of how the fattened face pairing graph is drawn in the plane, and in no way represent any sort of underlying topological twist.

Figure 3 shows a partial drawing of an ordered decomposition of a fattened face pairing graph. In this, we see a set of three parallel external arcs between nodes $v_{1,d}$ and $v_{2,h}$. This tells us that face d of tetrahedron 1 is identified with face h of tetrahedron 2. Additionally, we see that one of the external arcs connects internal arc $\{v_{1,c}, v_{1,d}\}$ with internal arc $\{v_{2,g}, v_{2,h}\}$. This tells us that edge ab of tetrahedron 1 (represented by $\{v_{1,c}, v_{1,d}\}$) is identified with edge



■ **Figure 2** Two close up views of a node of a fattened face pairing graph with the same pairing of arcs. The node itself is represented by the grey ellipse, and all six arcs are incident upon this node. Note how both figures show the same pairing of edges, the only difference is where the “crossing” occurs.



■ **Figure 3** A partial drawing of a fattened face pairing graph.

ef of tetrahedron 2 (represented by $\{v_{2,g}, v_{2,h}\}$). Since we know that face abc is identified with face efg modulo a possible reflection and/or rotation, this tells us that vertex c is identified with vertex g in this face identification. We can repeat this process for the other paired arcs to see that vertex a is identified with vertex e and vertex b is identified with vertex f . The resulting identification is therefore $abc \leftrightarrow efg$.

Repeating this for each set of three parallel external arcs gives the required triangulation. The process is easily reversed to obtain an ordered decomposition from a general triangulation. Complete constructions for both processes are given in the full version of this paper.

Recall that $\deg(e)$ is the number of edges of tetrahedra identified together to form edge e in the triangulation. The following corollary follows immediately from the constructions.

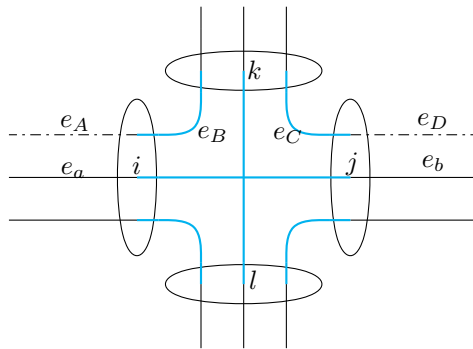
► **Corollary 13.** *Given an ordered decomposition $\{P_1, \dots, P_t\}$, each walk P_i corresponds to exactly one edge e in the corresponding general triangulation. In addition, $|P_i| = 2 \deg(e)$.*

Recall that in a 3-manifold triangulation, no edge may be identified with itself in reverse. In a triangulation one may check this by considering a ring of tetrahedra around some edge. By tracking face identifications through the tetrahedra in the ring, one can determine if the central edge is identified with itself in reverse. The following definition combined with Lemma 15 achieves the same result in our new framework.

► **Definition 14.** Given an ordered decomposition $\mathcal{P} = \{P_1, P_2, \dots, P_t\}$, we can *mark* a walk P_x as follows.

Pick an external arc e_s from P_x . Arbitrarily pick an external arc e_S parallel to e_s , and mark e_S as being “above” e_s . Then let $e_a = e_s$ and $e_A = e_S$ and continue as follows (see Figure 4 for a diagram of the construction):

- Let e_b be the next external arc in P_x after e_a .
- The internal arc preceding e_b joins two nodes. Call these nodes i and j , such that e_b is incident on j .
- Some external arc e_A incident on i must be marked as “above” e_a . Find the closed walk which e_A belongs to. In this closed walk there must exist some internal arc which either



■ **Figure 4** The process used to mark edges as per Definition 14. The dot-dashed arcs are the ones marked as “above”. Recall that the ellipses are whole nodes, the insides of which denote how internal and external arcs are paired up in the decomposition.

immediately precedes or follows e_A through node i . Call this internal arc e_B . Note that the walk containing these two arcs need not be, and often is not, P_x . Arc e_B must be incident to i , and some other node which we shall call k .

- Find the internal arc e_C between nodes k and j , and find the walk P_y that it belongs to. In this walk, one of the arcs parallel to e_b must either immediately precede or follow e_C and be incident upon node j . Call this arc e_D .
- If $e_b = e_s$, and e_D is already marked as being above e_b , we terminate the marking process.
- Otherwise, mark the arc e_D as being above e_b and repeat the above steps, now using e_b in place of e_a , and using e_D in place of e_A .

Note that this process of marking specifically marks one arc as being “above” another. It does not mark arcs as being “above” in general.

To visualise this definition in terms of the decomposition, see Figure 4. The arcs e_a and e_b are part of a closed walk, and we are marking the edges “above” this walk. Arc e_A was arbitrarily chosen. Arc e_B follows e_A , and then we find e_C as the arc sharing one node with e_B and one with e_b . From e_C we can find and mark e_D .

In brief, the walks containing e_A and e_D represent edges of tetrahedra in the triangulation that share triangles with the common edge represented by P_x , and which both sit “above” this common edge (assuming some up/down orientation). Both e_B and e_C are internal arcs of the same tetrahedron and share a common node k , so we know that both these internal arcs represent edges of the same tetrahedron which share a common face k . The external arcs e_A and e_D represent identifications of e_B and e_C respectively with edges of (typically different) adjacent tetrahedra.

► **Lemma 15.** *Take an ordered decomposition containing a walk P_x with arcs marked according to Definition 14, and consider the corresponding triangulation. Then the edge of the triangulation represented by P_x is identified to itself in reverse if and only if there exists some external arc e in P_x that has two distinct external arcs both marked as “above” e .*

Essentially, this condition indicates that the marking procedure cycles through the entire walk twice (marking two parallel arcs as “above” each arc of the walk), as opposed to once (marking only one arc as “above” each arc of the walk). The proof of this lemma is routine, and is given in full in the complete version of this paper.

If a walk P_x in an ordered decomposition can be marked according to Definition 14 such that each external arc e in P_x has exactly one other external arc marked as “above” e , we say that this walk is *non-reversing*.

► **Definition 16.** A *manifold decomposition* is an ordered decomposition of a fattened face pairing graph satisfying all of the following conditions.

- The ordered decomposition contains $n + 1$ closed walks.
- The fattened face pairing graph contains $4n$ nodes.
- Each walk is non-reversing.
- The associated manifold triangulation contains exactly 1 vertex.

► **Theorem 17.** *Up to relabelling, there is a one-to-one correspondence between manifold decompositions of connected fattened face pairing graphs and 1-vertex 3-manifold triangulations.*

Proof. Earlier in this section we described the correspondence between general triangulations and ordered decompositions. All that remains is to show that the extra properties of a manifold decomposition force the corresponding triangulation to be a 3-manifold triangulation. Since the decomposition contains $n + 1$ walks, Corollary 13 tells us the triangulation has $n + 1$ edges. Additionally, each tetrahedron corresponds to four nodes in the fattened face pairing graph, so the triangulation has n tetrahedra and thus by Lemma 5 we see that the link of each vertex is homeomorphic to a 2-sphere. Each walk is non-reversing so Lemma 15 says that no edge in the corresponding triangulation is identified with itself in reverse, and we have the required result. ◀

We now define the notation used to express specific ordered decompositions. The notation is defined such that it can also be interpreted as a *spine code* (as used by Matveev’s Manifold Recognizer [13]), and that the spine generated from such a spine code is a dual representation of the same combinatorial object represented by the manifold decomposition. For more detail on spine codes, see [12].

► **Notation 18.** *Take an ordered decomposition of a fattened face pairing graph with $4n$ nodes, and label each set of three parallel external arcs with a distinct value taken from the set $\{1, \dots, 2n\}$ (so two external arcs receive the same label if and only if they are part of the same triple of parallel arcs). Assign an arbitrary orientation to each set of three parallel external arcs. For each walk in the ordered decomposition:*

1. Create an empty ordered list.
2. Follow the external arcs in the walk.
 - a. If an external arc is traversed in a direction consistent with its orientation, add $+i$ to the end of the corresponding ordered list.
 - b. If instead the arc in the walk is traversed in the reverse direction, add $-i$ to the end of the list.
 - c. Continue until the first external arc in the walk is reached.

Note that this notation only records the external arcs, and does not record any internal arcs in walks.

We can also reconstruct the face pairing graph (and therefore the fattened face pairing graph) from this notation (in particular, we can reconstruct the internal arcs). The method essentially uses the fact that each external arc represents some identification of two faces (and three parallel external arcs will represent the same identification of two faces), and so we can use the orientation of each arc to distinguish between the two faces in each identification and thereby build up the face pairing graph.

An implementation note: it is trivial, given a fattened face pairing graph and a “partial” ordered decomposition in which all the internal arcs are missing, to reconstruct the complete ordered decomposition. For the theoretical discussions in this paper we work with the

full ordered decompositions, but in the implementation we only store the sequential list of external arcs as in Notation 18.

4 Algorithm and improvements

In this section we give various improvements that may be used when enumerating manifold decompositions (i.e., 3-manifold triangulations). These are based on known theoretical results in 3-manifold topology, combined with suitable data structures.

Enumeration algorithms [1, 2, 5, 9, 11, 12] in 3-manifold topology often focus on closed, minimal, irreducible and \mathbb{P}^2 -irreducible 3-manifold triangulations. These properties were all defined in Section 2. For brevity, we say that a triangulation (or manifold decomposition) has such a property if and only if the underlying manifold has the property. In both this section and Section 5, we will restrict our algorithm to this same setting. This highlights the usefulness of our algorithm, and allows us to demonstrate how existing results can be translated into our new framework.

Many existing algorithm implementations in the literature [5, 12] build triangulations by identifying faces pairwise (or taking combinatorially equivalent steps, such as annotating edges of special spines [12]). The algorithm we give here essentially constructs the neighbourhood of each *edge* of the triangulation one at a time. Therefore the search tree traversed by our new algorithm is significantly different than that traversed by other algorithms. This is highlighted experimentally by the results given in Section 5.

4.1 Algorithm

The basis of our implementation is a simple backtracking approach to enumerate manifold decompositions. Walks are built up one arc at a time, and recursion ensures that every possible manifold decomposition is found. However, this approach is not tractable for any interesting values of n , and so we introduce the following improvements.

4.2 Limiting the size of walks

The following results are taken from [1], though in the orientable case similar results were known earlier by other authors [9, 12].

► **Lemma 19.** (2.1 in [1]) *No closed minimal triangulation has an edge of degree three that belongs to three distinct tetrahedra.*

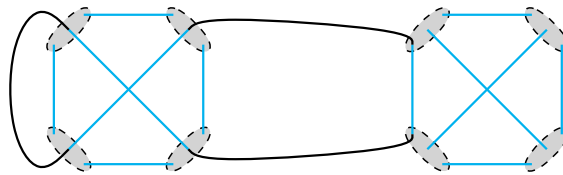
► **Lemma 20.** (2.3 and 2.4 in [1]) *No closed minimal \mathbb{P}^2 -irreducible triangulation with ≥ 3 tetrahedra contains an edge of degree ≤ 2 .*

Given that the degree of an edge e of a triangulation is the number of tetrahedron edges which are identified to form e , these results translate to manifold decompositions as follows.

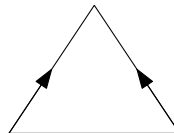
► **Corollary 21.** *No closed minimal \mathbb{P}^2 -irreducible manifold decomposition with ≥ 3 tetrahedra contains a walk which itself contains less than three external arcs.*

► **Corollary 22.** *No closed minimal manifold decomposition contains a walk which itself contains exactly three internal arcs representing edges on distinct tetrahedra (i.e., belonging to three distinct K_4 subgraphs).*

The above results are direct corollaries, as it is simple to translate the terms involved and the results are simple enough to implement in an algorithm. In the backtracking algorithm,



■ **Figure 5** The only possible walk containing 3 internal arcs not all from distinct tetrahedra in a fattened face pairing graph on more than 1 tetrahedron. Only the external arcs used in the walk are shown, other external arcs are not shown.



■ **Figure 6** A one-face cone formed by identifying the two marked edges.

this means we can implement a check on the number of arcs in a walk before adding the walk to the decomposition. This is implementable as a constant time check if the length of the current partial walk is stored.

Additionally, for a census of 1-vertex triangulations on n tetrahedra, a manifold decomposition must contain exactly $n + 1$ walks. If the algorithm has completed k walks, then there are $n + 1 - k$ walks left to complete. We use this idea in the following improvement.

By Corollary 22 a closed walk in a manifold decomposition which contains three internal arcs must contain two internal arcs belonging to the same K_4 , as in Figure 5. We modify our algorithm to enumerate all such closed walks first. Each such walk is either present or absent in any manifold decomposition. For each possible combination of such walks, we fix said walks and then run the search on the remaining arcs. All other walks must now contain at least four external arcs, so during the census on n tetrahedra if the algorithm has completed k walks and there are less than $4(n + 1 - k)$ unused external arcs we know that the partial decomposition cannot be completed to a 1-vertex manifold decomposition.

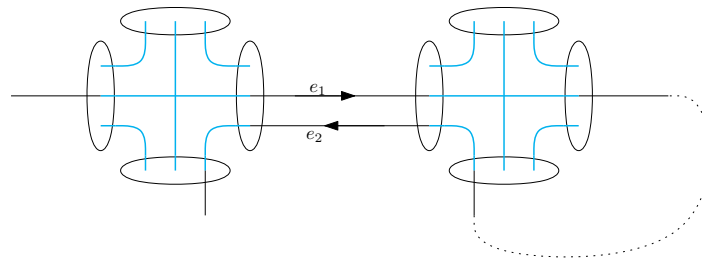
► **Improvement 23.** *For each K_4 in the given graph, determine if two of its internal arcs can be used together in a walk containing exactly three internal arcs. If this is possible, add said walk to the set S . Then, for each subset $s \subseteq S$, use s as a starting set of walks and attempt to complete the ordered decomposition. If during the enumeration process k walks have been completed and there are less than $4(n + 1 - k)$ unused external arcs, prune the search tree at this point.*

4.3 Avoiding cone faces

For some properties of minimal triangulations, it is not clear that the corresponding tests can be implemented cheaply. Here, we identify further results from the literature that enable fast implementations in our setting. The following was shown in [1].

► **Lemma 24.** (2.8 in [1]) *Let T be a closed minimal \mathbb{P}^2 -irreducible triangulation containing ≥ 3 tetrahedra. Then no single face of T has two of its edges identified to form a cone as illustrated in Figure 6.*

For manifold decompositions, our translation of this result also requires the underlying manifold to be orientable in order to give a fast algorithmic test.



■ **Figure 7** The depicted walk cannot occur in a closed minimal \mathbb{P}^2 -irreducible orientable manifold decomposition as external arcs e_1 and e_2 are used in opposite directions. The dotted lines indicates the walk continues through undrawn parts of the fattened face pairing graph.

► **Lemma 25.** *Let D be a closed minimal \mathbb{P}^2 -irreducible manifold decomposition of an orientable manifold containing ≥ 3 tetrahedra. Then no walk of D can use two parallel external arcs in opposite directions (as seen in Figure 7).*

A complete proof appears in the full version of this paper. The proof assigns orientations to corresponding tetrahedra, and then tracks orientations of the edges of tetrahedra to show that if a one-face cone is present then an edge must be identified with itself in reverse. This result leads to the following.

► **Improvement 26.** *When enumerating orientable manifold decompositions, if an external arc e is to be added to some walk W , and e is parallel to another external arc e' which itself is in W , check whether e and e' will be used in opposite directions. If so, do not use e at this point; instead backtrack and prune the search tree.*

4.4 One vertex tests

Definition 16 requires that the associated manifold only have one vertex. We test this by tracking properties of the vertex links as the manifold decomposition (i.e., triangulation) is built up. Specifically, while the manifold decomposition is still being constructed, no vertex link may be a closed surface.

► **Improvement 27.** *When building up a manifold decomposition, track how many “frontier edges” remain around each vertex link. If any vertex links are closed off before the manifold decomposition is completed, backtrack and prune the current subtree of the search space.*

The number of frontier edges of each vertex link, as well as which vertex links are identified together, are tracked via a union-find data structure. The data structure is slightly tweaked to allow back tracking (see [2] for details), storing the number of frontier edges at each node. For more details on the union-find algorithm in general, see [15].

4.5 Canonicity and Automorphisms

When running a search, many equivalent manifold decompositions will be found. These decompositions may differ in the order of the walks found, or two walks might have different starting arcs or directions. For example, the two walks (a, b, c) and $(-b, -a, -c)$ are equivalent. The second starts on a different arc, and traverses the walk backwards, but neither of these change the manifold decomposition. Additionally, the underlying face pairing graph often has non-trivial automorphism group.

To eliminate such duplication, we only search for *canonical* manifold decompositions. We use the obvious definition for a canonical walk (lowest-index arc is written first and is used in the positive direction).

There are two points in the algorithm where we might test for canonical decompositions.

► **Improvement 28.** *Every time an external arc is added to a walk, check if the current decomposition is canonical. If not, disregard this choice of arc and prune the search tree.*

► **Improvement 29.** *Every time a walk is completed, check if the current decomposition is canonical. If not, disregard this choice of arc and prune the search tree.*

Unfortunately, checking if a (possibly partial) decomposition is canonical is not computationally cheap. Experimental results showed that using Improvement 29 was significantly faster than using Improvement 28 as fewer checks for canonicity were made.

5 Results and Timing

In this section we detail the results from testing the algorithm. We test the manifold decomposition algorithm and its improvements from Section 4 against the existing implementation in *Regina*. Our algorithm (and indeed all known enumeration algorithms) are exponential in the number of nodes on a given graph. As a result, testing is limited to graphs of at most 10 nodes. Recall also that we are testing the enumeration of closed, minimal, irreducible and \mathbb{P}^2 -irreducible 3-manifold triangulations.

Regina is a suite of topological software and includes state of the art algorithms for census enumerations in various settings, including non-orientable and hyperbolic manifolds [3, 4]. *Regina* and its source code are freely available, which facilitates comparable implementations and fair testing. *Regina* also filters out invalid triangulations as a final stage, which allows us to test the efficiency of our various improvements by enabling or disabling them independently. Like other census algorithms in the literature, *Regina* builds triangulations using the traditional framework by identifying faces two at a time.

In testing, we measure time to begin when either algorithm is given some textual representation of a face pairing graph, and ending when all triangulations are found. That is, testing times include the calculation of automorphisms (for both *Regina* and our new algorithms), as well as the construction of the fattened face pairing graph.

We find that while *Regina* outperforms our new algorithms overall, there are non-trivial subcases for which our algorithm runs an order of magnitude faster. Importantly, in a typical census on 10 tetrahedra, *Regina* spends almost half of its running time on precisely these subcases. This shows that our new algorithm has an important role to play: it complements the existing framework by providing a means to remove some of its most severe bottlenecks. Section 5.2 discusses these cases in more detail.

These observations are, however, in retrospect: what we do not have is a clear indicator in advance for which algorithm will perform best for any given subcase.

Recall that a full census enumeration involves generating all 4-regular multigraphs, and then for each such graph G , enumerating triangulations with face pairing graph G . In earlier sections we only dealt with individual graphs, but for the tests here we ran each algorithm on all 4-regular multigraphs of a given order n .

In the following results, we use the term MD to denote our basic algorithm, using improvements 23, 27 and 29. For enumerating orientable manifolds only, we also use Improvement 26 and denote the corresponding algorithm as MD-o. Experimentation indicated that Improvement 27 was computationally expensive, and so we also tested algorithm MD*

■ **Table 1** Running time of *Regina* and the manifold decomposition (MD) algorithms when searching for manifold decompositions on n tetrahedra.

(a) Running times (in seconds) for the general setting.			(b) Running times (in seconds) for the orientable setting		
n	<i>Regina</i>	MD	n	<i>Regina</i>	MD-o
7	29	80	7	< 1	25
8	491	2453	8	147	535
9	11 288	79 685	9	3 499	13 161
10	323 530	3 406 211	10	90 969	430 162

(using only Improvements 23 and 29) and algorithm MD*-o (using Improvements 23 26 and 29). Note that these last two algorithms may find ordered decompositions which are not necessarily manifold decompositions, but we can easily filter these out once the enumeration is complete.

The algorithms were tested on a cluster of Intel Xeon L5520s running at 2.27GHz. Times given are total CPU time; that is, a measure of how long the test would take single-threaded on one core. The algorithms themselves, when run on all 4-regular multigraphs on n nodes, are trivially parallelisable which allows each census to complete much faster by taking advantage of available hardware.

We note that, as expected, the census results are consistent across both algorithms.

5.1 Aggregate tests

In the general setting, where we allow orientable and non-orientable triangulations alike, Table 1a highlights that *Regina* outperforms MD when summed over all face pairing graphs. The difference seems to grow slightly as n increases, pointing to the possibility that more optimisations in this setting are possible.

We suspect that tracking the orientability of vertex links is giving *Regina* an advantage here (see [2], Section 5). Tracking orientability is more difficult with ordered decompositions, as the walks are built up one at a time – each external arc represents an identification of edges, but does not specify the orientation of this identification. Thus orientability cannot be tested until at least two of any three parallel external arcs are used in walks.

We also compare MD-o to *Regina*, where we ask both algorithms to only search for orientable triangulations. Both algorithms run significantly faster than in the general setting (demonstrating that Improvement 26 is a significant improvement). Table 1b shows that *Regina* outperforms MD-o roughly by a factor of four. This appears to be constant, and here we expect MD to be comparable to *Regina* after more careful optimisation (such as *Regina*'s own algorithm has enjoyed over the past 13 years [1, 2]).

To test Improvement 27 (the one-vertex test), we compare MD* and MD*-o against MD and MD-o respectively. The timing data in Tables 2b and 2a shows that MD* and MD*-o outperformed MD and MD-o, demonstrating that Improvement 27 actually slows down the algorithm. We verified that Improvement 27 is indeed discarding unwanted triangulations – the problem is that tracking the vertex links is too computationally expensive. Algorithms MD* and MD*-o instead enumerate these unwanted triangulations and test for one vertex after the fact, discarding multiple vertex triangulations after they have been explicitly constructed. The cost of this is included in the timing results, which confirms that such an “after the fact” verification process is indeed faster than the losses incurred by Improvement 27.

■ **Table 2** Running times of MD, MD*, MD-o, MD*-o when searching for manifold decompositions on n tetrahedra.

<p>(a) Running times (in seconds) for the general setting.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>n</th> <th>MD</th> <th>MD*</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>80</td> <td>71</td> </tr> <tr> <td>8</td> <td>2 453</td> <td>1 875</td> </tr> <tr> <td>9</td> <td>79 685</td> <td>58 743</td> </tr> <tr> <td>10</td> <td>3 406 211</td> <td>1 624 025</td> </tr> </tbody> </table>	n	MD	MD*	7	80	71	8	2 453	1 875	9	79 685	58 743	10	3 406 211	1 624 025	<p>(b) Running times (in seconds) for the orientable setting.</p> <table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>n</th> <th>MD-o</th> <th>MD*-o</th> </tr> </thead> <tbody> <tr> <td>7</td> <td>25</td> <td>16</td> </tr> <tr> <td>8</td> <td>535</td> <td>446</td> </tr> <tr> <td>9</td> <td>13 161</td> <td>10 753</td> </tr> <tr> <td>10</td> <td>430 162</td> <td>291 544</td> </tr> </tbody> </table>	n	MD-o	MD*-o	7	25	16	8	535	446	9	13 161	10 753	10	430 162	291 544
n	MD	MD*																													
7	80	71																													
8	2 453	1 875																													
9	79 685	58 743																													
10	3 406 211	1 624 025																													
n	MD-o	MD*-o																													
7	25	16																													
8	535	446																													
9	13 161	10 753																													
10	430 162	291 544																													

■ **Table 3** Running time in seconds of MD* and *Regina* on particular graphs on 10 nodes.

Graph ID	<i>Regina</i>	MD*
48 308	2476	142
48 083	2487	192
48 288	2164	118
47 332	2141	229
47 333	2003	134
47 520	2083	221
46 914	2108	302

5.2 Individual graph tests

It is on individual (and often pathological) face pairing graphs that the new algorithm shines. Recall that the census enumeration problem requires running an enumeration algorithm on all connected 4-regular multigraphs of a given order. Table 3 shows the running time of both *Regina* and MD* on a cherry-picked sample of such graphs on 10 tetrahedra.

From these we can see that on some particular graphs, MD* outperforms *Regina* by an order of magnitude. While these graphs were cherry-picked, they do display the shortfalls of *Regina*. There are 48432 4-regular multigraphs on 10 nodes, and it takes *Regina* 89.9 CPU-hours to complete this census. Of these 48432 graphs, 48242 are processed in under 300 seconds each. In contrast, it takes *Regina* 43.6 CPU-hours to process the remaining 190 graphs. This accounts for 48.5% of the running time of *Regina*'s census on 10 tetrahedra triangulations.

Running these “pathological” graphs through MD takes 12.1 CPU-hours, for a saving of 31.5 CPU-hours. This would reduce the running time of the complete census from 89 hours to 58 hours, a 35% improvement. With a priori knowledge of running times, passing each graph to the faster of either *Regina* or MD* can give a 44% improvement in the running time of the complete census.

6 Discussion

Although its performance is inconsistent, it is significant that our new framework performs an order of magnitude faster than existing algorithms on those subcases where existing algorithms struggle. If we had an effective heuristic that could determine which algorithm (*Regina* or MD*) to use for any given a 4-valent graph, then this could speed up the census

enumeration process significantly. Identifying such a heuristic is the subject of ongoing work.

Beyond the enumeration problem, this framework may also help us find families of “forbidden subgraphs”: subgraphs which, if present in some 4-valent graph G , indicate that there is *no* minimal 3-manifold triangulation with G as its dual 1-skeleton [1]. Such graphs G can be omitted entirely from the census enumeration, and so identifying (with proof) forbidden subgraphs can lead to further significant improvements in census running times.

References

- 1 Benjamin A. Burton. Face pairing graphs and 3-manifold enumeration. *J. Knot Theory Ramifications*, 13(8):1057–1101, 2004.
- 2 Benjamin A. Burton. Enumeration of non-orientable 3-manifolds using face-pairing graphs and union-find. *Discrete & Computational Geometry*, 38(3):527–571, 2007.
- 3 Benjamin A. Burton. Detecting genus in vertex links for the fast enumeration of 3-manifold triangulations. In *ISSAC 2011: Proceedings of the 36th International Symposium on Symbolic and Algebraic Computation*, pages 59–66. ACM, 2011.
- 4 Benjamin A. Burton. The cusped hyperbolic census is complete. *arXiv:1405.2695*, 2014.
- 5 Benjamin A. Burton, Ryan Budney, and William Pettersson. Regina: Software for 3-manifold topology and normal surface theory. Licensed under GPLv2, 1999-2013.
- 6 Patrick J. Callahan, Martin V. Hildebrand, and Jeffrey R. Weeks. A census of cusped hyperbolic 3-manifolds. *Math. Comp.*, 68(225):321–332, 1999. With microfiche supplement.
- 7 Martin Hildebrand and Jeffrey Weeks. A computer generated census of cusped hyperbolic 3-manifolds. In *Computers and mathematics*, pages 53–59. Springer, New York, 1989.
- 8 William Jaco and J. Hyam Rubinstein. 0-efficient triangulations of 3-manifolds. *J. Differential Geom.*, 65(1):61–168, 2003.
- 9 Bruno Martelli and Carlo Petronio. Three-manifolds having complexity at most 9. *Experimental Mathematics*, 10(2):207–236, 2001.
- 10 Bruno Martelli and Carlo Petronio. A new decomposition theorem for 3-manifolds. *Illinois J. Math.*, 46:755–780, 2002.
- 11 Sergei V. Matveev. Computer recognition of three-manifolds. *Experiment. Math.*, 7(2):153–161, 1998.
- 12 Sergei V. Matveev. *Algorithmic topology and classification of 3-manifolds*, volume 9 of *Algorithms and Computation in Mathematics*. Springer, Berlin, second edition, 2007.
- 13 Sergei V. Matveev et al. Manifold recognizer, 2014. <http://www.matlas.math.csu.ru/?page=recognizer>.
- 14 Edwin E. Moise. Affine structures in 3-manifolds. V. The triangulation theorem and Hauptvermutung. *Ann. of Math. (2)*, 56:96–114, 1952.
- 15 Robert Sedgewick. *Algorithms in C++*. Addison-Wesley, Reading, MA, 1992.
- 16 Morwen Thistlethwaite. Cusped hyperbolic manifolds with 8 tetrahedra. <http://www.math.utk.edu/~morwen/8tet/>, October 2010.

Order on Order Types*

Alexander Pilz¹ and Emo Welzl²

- 1 Institute for Software Technology, Graz University of Technology, Austria
apilz@ist.tugraz.at
- 2 Department of Computer Science, Institute of Theoretical Computer Science
ETH Zürich, Switzerland
emo@inf.ethz.ch

Abstract

Given P and P' , equally sized planar point sets in general position, we call a bijection from P to P' *crossing-preserving* if crossings of connecting segments in P are preserved in P' (extra crossings may occur in P'). If such a mapping exists, we say that P' *crossing-dominates* P , and if such a mapping exists in both directions, P and P' are called *crossing-equivalent*. The relation is transitive, and we have a partial order on the obtained equivalence classes (called *crossing types* or *x-types*). Point sets of equal order type are clearly crossing-equivalent, but not vice versa. Thus, x-types are a coarser classification than order types. (We will see, though, that a collapse of different order types to one x-type occurs for sets with triangular convex hull only.)

We argue that either the *maximal* or the *minimal x-types* are sufficient for answering many combinatorial (existential or extremal) questions on planar point sets. Motivated by this we consider basic properties of the relation. We characterize order types crossing-dominated by points in convex position. Further, we give a full characterization of minimal and maximal abstract order types. Based on that, we provide a polynomial-time algorithm to check whether a point set crossing-dominates another. Moreover, we generate all maximal and minimal x-types for small numbers of points.

1998 ACM Subject Classification G.2.1 Combinatorics, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases point set, order type, planar graph, crossing-free geometric graph

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.285

*Dedicated to Jacob E. Goodman and Richard Pollack
on the occasion of their eightieth birthdays.*

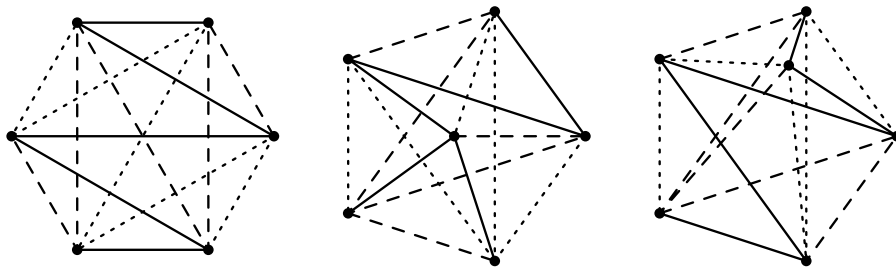
1 Introduction

Let us start right away with an illustrating example, which did indeed motivate our study. We came across the following nice open question, which was considered in [7] and investigated further in [3]: Given a complete geometric graph (edges as straight segments) on $2m$ points in general position in the plane, is it always possible to partition the edges into m crossing-free spanning trees? For addressing such problems, the concept of order types¹ is ubiquitously

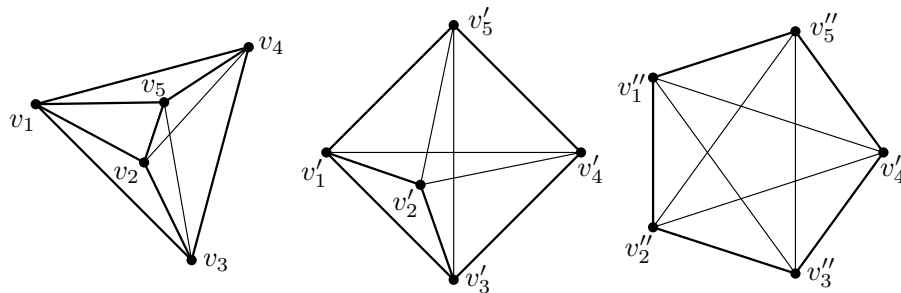
* A.P. is supported by the ESF EUROCORES programme EuroGIGA – ComPoSe, Austrian Science Fund (FWF): I 648-N18. E.W. acknowledges support from EuroCores/EuroGiga/ComPoSe Swiss National Science Foundation (SNF) 20GG21 134318/1.

¹ The reader not familiar with the notion of order types is referred to the end of this section.





■ **Figure 1** The three maximal order types for 6 points with a partition of the complete geometric graph into three crossing-free spanning-trees (see [3, Figure 8]).

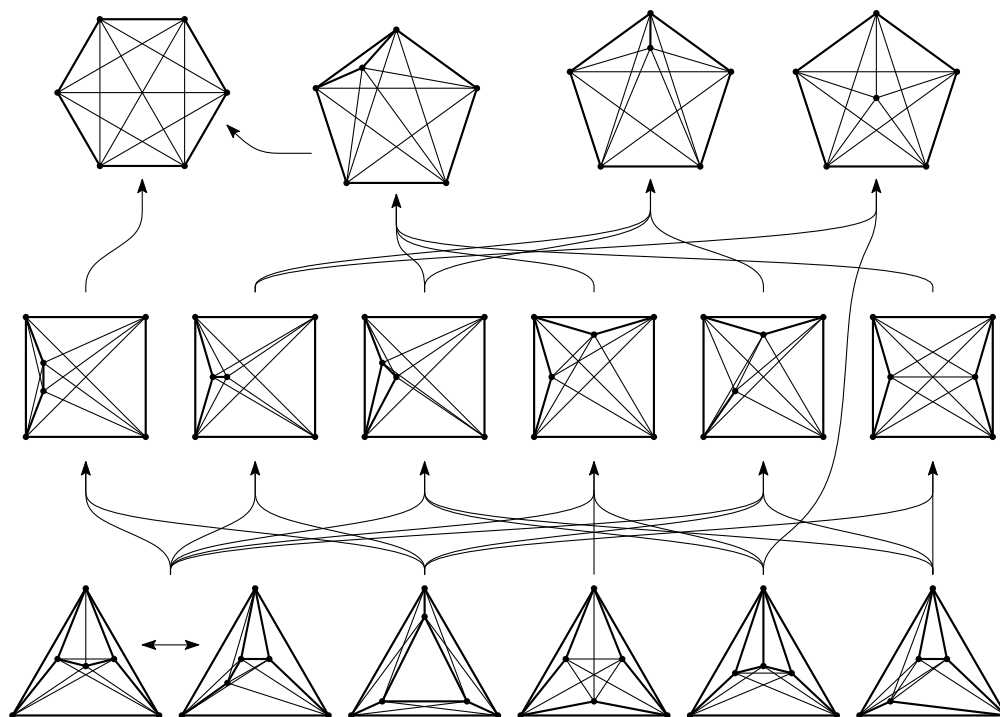


■ **Figure 2** Points sets P , P' , and P'' on five points. The mappings $p \mapsto p'$ and $p' \mapsto p''$ are x -preserving, therefore $P \leq_x P' \leq_x P''$. $P <_x P' <_x P''$ follows from the increasing number of crossings.

used in Discrete and Computational Geometry, as it allows for classifying the infinite number of point sets of a given size into a finite number of equivalence classes, capturing combinatorial properties such as which pairs of spanned line segments cross and which points define the set’s convex hull. Checking the question at hand for 4 points is easy, since there are only two order types. For 6 points, there are already 16 order types to consider. By what we study below, we claim, though, that the partitions for the three order types given in Figure 1 constitute already a complete proof of the fact for 6 points. This is because the three order types are maximal w.r.t. crossing pairs of edges – a notion to be rendered more precisely and not to be confused with the maximum number of crossings, as achieved by the convex position order type only. In fact, in our example, convex position does allow a partition into crossing-free spanning paths, while this is not true for some other order type of six points. By using the techniques presented herein, we were able to experimentally confirm that such a partition exists for any point set of up to ten points, using the reduced set of order types.

We proceed to basic definitions. Given two equally sized point sets P and Q in general position in the plane, a bijection $P \rightarrow Q, p \mapsto p'$, is called *crossing-preserving* (or *x -preserving*) if whenever the segment pq crosses the segment rs (for points p, q, r and s in P) then $p'q'$ crosses $r's'$. If such a mapping exists, we say that Q *crossing-dominates* (*x -dominates*) P , in symbols $Q \geq_x P$ or $P \leq_x Q$. If such mappings exist in both directions, then P and Q are called *crossing-equivalent* (*x -equivalent*), in symbols $P \sim_x Q$. Finally, if $Q \geq_x P$ but Q and P are not x -equivalent, then we say that Q *strictly crossing-dominates* (*strictly x -dominates*) P , in symbols $Q >_x P$ or $P <_x Q$. The relation is transitive and it induces a partial order on the obtained equivalence classes (called *crossing types* or *x -types*). An order type or crossing type is called *crossing-maximal* (*x -maximal*), if for a set P of that type there is no point set that strictly x -dominates P ; accordingly for *crossing-minimal*.

Figure 2 shows three 5-point sets $P, P',$ and P'' with x -preserving mappings ($p \mapsto p'$ and $p' \mapsto p''$, respectively) witnessing $P \leq_x P' \leq_x P''$. Obviously, $P \leq_x Q$ entails that the



■ **Figure 3** The 16 order types for 6 points with the Hasse-diagram for the x-dominance relation.

complete geometric graph on Q has at least as many crossings as the complete graph for P . We can therefore conclude, in fact, $P <_x P' <_x P''$.

Figure 3 displays 16 point sets representing the order types on six points, with the Hasse-diagram for the x-dominance relation. There, we can make the following observations.

- There are two order types (at the lower left of Figure 3) that merge to one x-type. We will show that such a collapse happens only if the order types have three extreme points.
- The number of maximal order types is 3, all these maximal order types are also x-types. There are six minimal order-types and five minimal x-types. Exploiting the basic properties we develop, we were able to determine (by a computer program) the values in Table 1. Two of these necessary basic properties are listed next.
- If Q strictly x-dominates P in Figure 3, then Q has strictly more extreme points than P . We will show that this is true in general.
- There are only four order types that are *not* x-dominated by sets in convex position. We will develop a necessary and sufficient criterion for x-dominance by sets in convex position. This makes this property easy to check without explicitly providing an x-preserving mapping (the property is that there has to be a Hamiltonian cycle of so-called unavoidable edges). For sets not dominated by sets in convex position, we give a detailed characterization, and show how to efficiently obtain an x-preserving mapping between two point sets, if one exists.

It often suffices to check x-minimal or x-maximal x-types. In Combinatorial Geometry, one is often concerned with estimating the minimum or maximum number of certain (mainly plane) geometric graphs a point set admits. Such combinatorial questions usually depend only on the set's order type. A complete enumeration of the order types of small point sets

■ **Table 1** The number of small crossing-minimal and crossing-maximal order types.

#	order types [1, 5]	crossing-maximal	crossing-minimal
4	2	1 50%	1 50%
5	3	1 33%	1 33%
6	16	3 19%	6 38%
7	135	17 13%	49 36%
8	3'315	489 15%	1'179 36%
9	158'817	28'103 18%	55'278 35%
10	14'309'547	2'866'895 20%	4'888'160 34%
11	2'334'512'907	[503'727'394, 504'463'503] 22%	[787'697'700, 787'720'845] 34%

by Aichholzer, Aurenhammer, and Krasser [1] allows for investigating such problems for small instances. Our concept of crossing types enables us to only consider a subset of all order types for several combinatorial problems. We give an incomplete list of examples.

- The number of crossing-free graphs of a certain type (e.g., spanning trees, polygonizations, or perfect matchings) is minimized for a maximal x-type and it is maximized for a minimal x-type.
- Similarly, the smallest number $e_k(n)$ such that any graph with at least $e_k(n)$ edges contains $k + 1$ pairwise disjoint edges (see, e.g., [16]) is determined by a maximal x-type.
- The maximal number of points in convex position is minimized for a minimal x-type.
- The size of the largest crossing family (see e.g., [6]) is minimized for a minimal x-type.
- The minimal number of crossings in a straight line drawing of the complete graph is realized on some minimal crossing type.

Indeed, point sets in convex position minimize the number of various classes of plane graphs [4]. However, there are classes where the crossing-preserving mapping does not maintain membership in the class, the most obvious example being triangulations. Hence, the number of triangulations may not be minimized by a maximal x-type.

Order types, orientation-equivalence. Given a sequence pqr of three distinct non-collinear points, we define its *orientation* ∇pqr as $+1$ if the sequence (p, q, r) traverses the triangle bounding the convex hull of $\{p, q, r\}$ (denoted by Δpqr) in counterclockwise direction and as -1 if this orientation is clockwise. Given two equally sized point sets P and Q in general position, a bijection $P \rightarrow Q$, $p \mapsto p'$, is called *order-preserving* if there is an $\varepsilon \in \{-1, +1\}$ such that $\nabla p'q'r' = \varepsilon \nabla pqr$ for all sequences pqr of three distinct points in P . If such a mapping exists, we say that P and Q are *order-equivalent*. The resulting equivalence classes are called *order types* [9] (and “being of the same order type” is mostly used for what we called here “order-equivalent”). For every natural number n there is only a finite number of order types; complete databases are available up to $n = 11$, see [1, 5].

Segment pq crosses segment rs iff both $\nabla pqr \cdot \nabla pqs = -1$ (i.e., r and s lie on different sides of the line through p and q) and $\nabla rsp \cdot \nabla rsq = -1$. Hence, order-equivalent sets are also x-equivalent (as we have indicated already, there are examples that show the reverse implication not to be true).

Given a point set, the orientation of each point triple is clearly defined by the containment of points in the half-planes given by the supporting lines of all point pairs. In a *generalized configuration of points*, these supporting lines of point pairs are replaced by supporting pseudo-lines (i.e., bi-infinite simple Jordan curves such that each pair of pseudo-lines intersects once – in a crossing, not tangentially). The orientation of a point triple is defined by the half-planes

given by these supporting pseudo-lines. See, e.g., [10] for a formal definition (in the projective plane). This concept generalizes order types to *abstract order types*. An abstract order type of a point set (with straight supporting lines) is *realizable*.

We will see that there are point sets that are x -dominated by an abstract order type, but not by a realizable one. For abstract order types, we will obtain a complete characterization of the generalized configurations of points x -dominating and x -dominated by a given one. Since it is $\exists\mathbb{R}$ -complete to decide whether an abstract order type is realizable [12] (see also [15]), there is not much hope for obtaining the same result for (realizable) order types.

Rotation systems. Let P be a point set of n points in general position. For any point $p \in P$, consider a ray r_p starting at p . When rotating r_p counterclockwise around p , the points $P \setminus \{p\}$ are traversed by r_p in a fixed circular order, called the *rotation* of p . The *rotation system* of P is the set of the rotations of all points of P . Similar to order types, we consider two rotation systems to be equivalent if one can be obtained from the other by relabeling and mirroring. The following result (whose origin will be discussed later in this paragraph) gives a tight relation between the rotation system and crossing-equivalence.

► **Corollary 1** (Kynčl [11, Proposition 6]). *Two point sets have the same rotation system iff they are crossing-equivalent.*

However, our main concern in this work is not crossing-equivalence, but rather the partial order on the set of all order types defined by crossing dominance.

Clearly, the order type of P determines its rotation system. The reverse problem, i.e., reconstructing the order type of P when given only its rotation system, has been considered in connection with applications in robotics (see, e.g., [17]). Wismath [18] gives a simple example of a rotation system on four elements that can be obtained by two different point sets with labels. (He then describes a method to reconstruct the point set if additional information is available.) However, when disregarding the labels in Wismath's example, the two different point sets have the same order type. An example of two different order types producing the same rotation system is given by the two point sets in the lower left corner of Figure 3. Aichholzer et al. [2] show that, essentially, the order type can be reconstructed from the rotation system if there are more than three extreme points, or if the extreme points are given. Their reconstruction method is applicable even if an unknown number of rotations have been reversed. Further, they give tight bounds on the number of (labeled) order types with a common rotation system based on properties of the point set. (We will revisit these properties in Section 5.)

Let K_P be the complete geometric graph on the point set P . The rotation system of P determines the order in which the edges emanate from each vertex of K_P . Straight-edge drawings of the complete graph are generalized by so-called *good drawings*. In a good drawing of a graph, vertices are represented by distinct points, and edges are drawn as simple Jordan arcs, where two edges intersect in at most one single point that may be their common endpoint or a proper crossing. Kynčl [11] shows that, for good drawings of the complete graph, a valid set of crossing edge pairs fully determines the rotation system of the good drawing, and that the rotation system determines whether two edges cross. Corollary 1 is therefore a special case of that result.²

² In general, the rotation system does not determine the crossings for non-complete graphs. The problem of determining the crossing number of a graph with a given rotation system is NP-complete [13].

Further related work. A complementary topic to characterizing crossing-maximal sets is the one for finding universal point sets. An n -universal point set admits a straight-line embedding of any planar graph with n vertices. Cardinal, Hoffmann, and Kusters [8] showed that for $n \leq 10$ there exists an n -universal point set of size n , and that for $n \geq 15$ no such set can exist. There is a certain relation to crossing-minimal sets, but observe that our setting is more constrained, as there is a bijection between the vertices/points.

Notation. The function $\text{conv}(P)$ denotes the convex hull and $\text{extr}(P)$ denotes the set of extreme points in a set P of points. Let Conv_n denote the order type of all sets of n points in convex position (i.e., sets P with $|\text{extr}(P)| = |P| = n$). Throughout the paper, let $p \mapsto p'$ be an x -preserving mapping from a finite set P of points in the plane (in general position) to another point set $P' = \{p' \mid p \in P\}$ in general position. For $A \subseteq P$, we write A' for $\{p' \mid p \in A\}$.

2 Crossing-Dominance, Convex Position, and Inner Points

In a set of n points in convex position, every 4-tuple of points determines exactly one crossing pair of segments. Hence there are $\binom{n}{4}$ such crossing pairs, which is obviously the largest possible number for n points. Therefore, no set can strictly x -dominate a set in Conv_n , and Conv_n is an x -maximal order type. We characterize the sets that are x -dominated by sets in convex position. For that purpose, given a set P in general position, we call a pair $\{p, q\}$ of two distinct points in P *unavoidable* if no segment determined by two points in P crosses the segment pq . The term “unavoidable” stems from the fact that every triangulation of P must use all unavoidable pairs as edges. Clearly, the edges of the convex hull give rise to such unavoidable pairs, but other possibilities occur. In fact, the number of unavoidable pairs in a set of n points can be as large as $2n - 2$, see [14].

► **Theorem 2.** $P \leq_x Q \in \text{Conv}_n$ iff the unavoidable pairs in P contain a Hamiltonian cycle.

Proof. Suppose $P \leq_x Q \in \text{Conv}_n$ with $Q = \{q_0, q_1, \dots, q_{n-1}\}$ such that points of Q appear in order $(q_0, q_1, \dots, q_{n-1})$ along the boundary of $\text{conv}(Q)$ in counterclockwise order, and let $P = \{p_0, p_1, \dots, p_{n-1}\}$ such that $p_i \mapsto q_i$ is x -preserving. Then all pairs $\{p_i, p_{(i+1) \bmod n}\}$, $i = 0, 1, \dots, n-1$, have to be unavoidable, since only unavoidable pairs can map to unavoidable pairs under an x -preserving mapping. Hence, a Hamiltonian cycle of unavoidable pairs exists for P . For the other direction, suppose the unavoidable pairs in P allow a Hamiltonian cycle. Let $P = \{p_0, p_1, \dots, p_{n-1}\}$ with pairs $\{p_i, p_{(i+1) \bmod n}\}$, $i = 0, 1, \dots, n-1$, forming a spanning cycle of unavoidable pairs. Note that the geometric realization of this cycle is a simple polygon that is crossed by no segment connecting points in P . That is, every segment connecting points in P is either completely inside or completely outside the polygon (or part of the polygon). Suppose segment $p_i p_j$ crosses $p_k p_\ell$. Then they have to be (i) either both inside the unavoidable polygon or both outside the unavoidable polygon and (ii) the appearance of points $\{p_i, p_j\}$ and $\{p_k, p_\ell\}$ alternate along the unavoidable cycle. Since two segments in a convex polygon cross iff their endpoints alternate along the convex polygon, an x -preserving mapping from P to Q readily follows. ◀

For point sets not dominated by Conv_n , we can identify the following property, which gives a rather strong condition for x -dominance.

► **Theorem 3.** Suppose $P \leq_x P'$ and there exists point $p' \notin \text{extr}(P')$. Then the rotation of p in P is equivalent to the rotation of p' in P' .

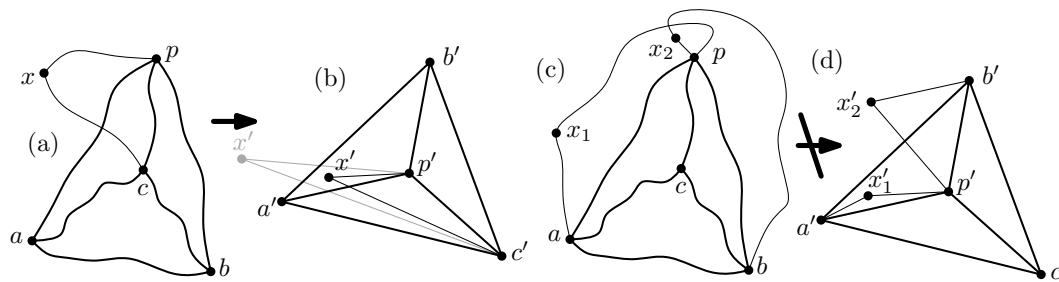


Figure 4 For a point p' in the interior of a triangle, the rotation is equivalent to the one of its preimage. (a) The path from p via x to c has to cross an edge of the triangle Δpab . This is only possible in P' if x' is at the same relative position in the rotation around p' . (b) If the images of x_1 and x_2 would change their relative position in the rotation, we would lose a crossing between the paths (p, x_1, a) and (p, x_2, b) .

Proof. The statement is obviously true for four points; suppose therefore that $|P| \geq 5$. Since p' is an inner point of P' , there exists at least one triangle $\Delta a'b'c'$ that contains p' . The complete graph on $\{a, b, c, p\}$ is also crossing-free. (We do not know whether p is inside Δabc , but this is not needed for our argument. To prevent confusion by geometric artifacts, one may even consider K_P to be projected onto a sphere.) Suppose there is a point x that is, w.l.o.g, separated from c by a and b in the rotation around p . Then the path (p, x, c) has to leave the triangle Δabp by crossing one of its edges (see Figure 4 (a)). This crossing also has to be present in P' and this is only possible if x' is also separated from c' by a' and b' in the rotation around p' , as shown in Figure 4 (b). Hence, the rotation around p is the same w.r.t. (a, b, c) for every fifth point x . We are therefore left with the case where there are two points x'_1 and x'_2 , separated, w.l.o.g., from c' by a' and b' ; the situation is the same for their preimages in P' by the previous arguments. Let the subsequence in the rotation of p' be (a', x'_1, x'_2, b') , and suppose it is (a, x_2, x_1, b) in P . Then the path (p, x_1, a) intersects the path (p, x_2, b) , as sketched in Figure 4 (c). But the images of such paths, shown in Figure 4 (d), are always non-intersecting in P' , a contradiction. ◀

3 Crossing-Dominance Needs More Extreme Points

The x -dominance relation exhibits the following monotonicity properties.

► **Proposition 4.** (1) If $P \leq_x Q$, then the complete straight line drawing K_Q of the complete graph on Q has at least as many crossing pairs of edges as K_P does. (2) If $P \leq_x Q$, then $|\text{extr}(P)| \leq |\text{extr}(Q)|$.

Proof. (1) follows directly from the definition of an x -preserving mapping. For (2) remember that a triangulation of a point set (in general position) with n points and h extreme points has exactly $3n - 3 - h$ edges. Now consider an x -preserving bijection $p \mapsto p'$ from P to Q and some triangulation of Q , which has $3n - 3 - |\text{extr}(Q)|$ edges. The preimage of this triangulation is a crossing-free graph on P which is contained in some triangulation of P with $3n - 3 - |\text{extr}(P)|$ edges. Therefore, $3n - 3 - |\text{extr}(Q)| \leq 3n - 3 - |\text{extr}(P)|$. ◀

The main purpose of this section is to shed some extra light on property (2). In particular we will show that (i) $P <_x Q$ implies $|\text{extr}(P)| < |\text{extr}(Q)|$ (Theorem 13). Moreover, given an x -preserving mapping from P to Q , (ii) the inverse is also x -preserving iff $|\text{extr}(P)| = |\text{extr}(Q)|$

(Theorem 14) and (iii) the mapping is order-preserving iff $\text{extr}(P)' = \text{extr}(Q)$ (Theorem 10). We will see that a triangular convex hull is a situation that needs special attention.

Switching to crossing. We discriminate the relative position of two distinct non-crossing segments pq and rs on points in a set S in general position as follows: They are called *incident* if they share an endpoint, they are called *parallel* if the lines supporting them intersect in a point outside of both segments, and they are called *stabbing* if the line of one of the two segments crosses the other segment. “Crossing”, “parallel”, “stabbing”, and “incident” exhaust all possibilities for two segments connecting points in general position.

Note that in an x -preserving mapping the image of non-crossing segments can of course be crossing – not so for parallel segments, though.

► **Lemma 5.** (1) If pq and rs are parallel, then $p'q'$ and $r's'$ are parallel. (2) If pq and rs are non-crossing and $p'q'$ and $r's'$ are crossing, then pq and rs are stabbing (i.e., $\{p, q, r, s\}$ is not in convex position).

Diagonals stay. A segment pq connecting two points p and q in a point set S is called a *diagonal of S* if p and q are extreme points in S and pq is not an edge of the convex hull of S .

► **Lemma 6.** If pq is a diagonal of P , then $p'q'$ is a diagonal of P' .

Proof. Note that pq is crossed by some segment rs (take any two points in $P \setminus \{p, q\}$ on opposite sides of the line through pq). Hence, $p'q'$ cannot be an edge of the convex hull. Therefore, if $p'q'$ is not a diagonal of P' , then the line containing $p'q'$ must intersect some edge $a'b'$ (in its interior) of the convex hull of P' . Note that $p'q'$ and $a'b'$ do not cross, so pq and ab do not cross, and therefore a and b must lie on the same side of the line h containing pq (because p and q are extreme). Now, since pq is a diagonal, there must be a point c on the other side of this line h ; we know that both ac and bc cross pq . But it is not possible that both $a'c'$ and $b'c'$ cross $p'q'$, since a' and b' lie on opposite sides of the line through p' and q' ; a contradiction to $p \mapsto p'$ being x -preserving. ◀

If there are at least four extreme points then every extreme point participates in a diagonal. Therefore, an x -preserving mapping maps extreme points to extreme points.

► **Corollary 7.** If $|\text{extr}(P)| \geq 4$ then $(\text{extr}(P))' \subseteq \text{extr}(P')$.

The aforementioned argument cannot be used if there are only three extreme points in P . In fact, the implication in the corollary simply is not true in this case (see the x -equivalent sets in Figure 3).

Jumping out of a triangle. As we have learned, if pq and rs do not cross, but $p'q'$ and $r's'$ do, then pq and rs must be stabbing and therefore p, q, r, s is not in convex position. W.l.o.g., let p be in $\text{conv}(\{q, r, s\})$. If, indeed, $p'q'$ and $r's'$ cross, then clearly $p' \notin \text{conv}(\{q', r', s'\})$. This “jumping out of a triangle” immediately has further implications for the location of p' , as the following lemma states.

► **Lemma 8.** If $p \in \text{conv}(\{q, r, s\})$ and $p' \notin \text{conv}(\{q', r', s'\})$ then $p' \notin \text{conv}(A')$ with A being the set of points in P not in the interior of $\text{conv}(\{q, r, s\})$. In particular, $p' \notin \text{conv}(\text{extr}(P)')$.

Proof. Consider some point $t \in P$ not in the interior of $\text{conv}(\{q, r, s\})$, i.e., this point lies either outside $\text{conv}(\{q, r, s\})$ or is one of the points q, r , or s . Now consider p' , which is outside $\text{conv}(\{q', r', s'\})$ and therefore a line h separating p' from $\text{conv}(\{q', r', s'\})$ exists.

Clearly, if t is among q, r, s , then h separates p' from t' . If t is outside $\text{conv}(\{q, r, s\})$, then pt crosses one of the segments qr, rs , or sq , say it is qr . Now $p't'$ must cross $q'r'$; therefore p' and t' must be on opposite sides of h and this line h separates all of A from p . ◀

Observe now that if $P' >_x P$, then the mapping must turn a non-crossing pair into a crossing pair (otherwise P and P' are x-equivalent). Hence some point p has to leave some triangle under the x-preserving mapping and therefore P' must have some new extreme point u' (not necessarily p' itself), i.e., $u' \in \text{extr}(P')$ but $u' \notin \text{extr}(P)$. In combination with Corollary 7 this yields the following.

▶ **Corollary 9.** *If $|\text{extr}(P)| \geq 4$ and $P' >_x P$, then $(\text{extr}(P))' \subsetneq \text{extr}(P')$.*

Hence, the number of extreme points has to increase. The following theorem is an important implication of the “jumping out of a triangle” observation.

▶ **Theorem 10.** *The x-preserving mapping $p \mapsto p'$ is order-preserving iff $\text{extr}(P)' = \text{extr}(P')$.*

Corollary 7 and Theorem 10 imply that order types and crossing types coincide for point sets with at least 4 extreme points. This can also be seen by combining Corollary 1 and the fact that, given the rotation system and the extreme points of a point set, its order type is determined (as shown in [2]).

▶ **Corollary 11.** *If $P \sim_x P'$ and $|\text{extr}(P)| \geq 4$, then P and P' are of the same order type.*

In Figure 3 we have seen an example witnessing that the condition $|\text{extr}(P)| \geq 4$ is essential in Corollary 11. It is evident that $|\text{extr}(P)| = 3$ needs special attention.

Three extreme points. While we have examples where $(\text{extr}(P))' \subseteq \text{extr}(P')$ is not true (if $|\text{extr}(P)| = 3$), we can still show that $P' >_x P$ always implies $|\text{extr}(P')| > |\text{extr}(P)|$.

▶ **Lemma 12.** *If $|\text{extr}(P)| = |\text{extr}(P')| = 3$ for sets P and P' with $P \leq_x P'$, then $P \sim_x P'$.*

We can finally conclude (from Corollary 9 and Lemma 12) that strict x-dominance goes with a strictly larger set of extreme points.

▶ **Theorem 13.** *If $P <_x P'$, then $|\text{extr}(P)| < |\text{extr}(P')|$.* ◀

It remains to show x-equivalence for related sets with the same number of extreme points.

▶ **Theorem 14.** *The inverse of the x-preserving mapping $p \mapsto p'$ is x-preserving iff $|\text{extr}(P)| = |\text{extr}(P')|$.*

Proof. If the inverse of $p \mapsto p'$ is x-preserving, then $P \sim_x P'$ and we know that $|\text{extr}(P)| = |\text{extr}(P')|$ holds by Proposition 4. If the inverse of $p \mapsto p'$ is not x-preserving, then there has to be a crossing in P' that is not present in the preimage P , i.e., there are strictly more crossings in $K_{P'}$ than in K_P . Therefore, P' strictly x-dominates P and, by Theorem 13, we have $|\text{extr}(P)| < |\text{extr}(P')|$, contradicting the assumption that $|\text{extr}(P)| = |\text{extr}(P')|$. ◀

4 (Sufficient) Conditions for Crossing-Dominance

In Section 2 we gave a characterization of the point sets dominated by Conv_n . Together with Theorem 13, this immediately gives us the following result.

▶ **Corollary 15.** *If $|\text{extr}(P)| = |P| - 1$, then P is x-maximal iff it has no Hamiltonian cycle of unavoidable edges.*

While for (realizable) point sets not dominated by Conv_n we cannot give such a complete characterization, we can give properties that witness x -maximality of a point set, based on unavoidable edges. Consider two point sets $P \leq_x P'$ with $\text{extr}(P)' \subsetneq \text{extr}(P')$. Then there is an edge ab of $\text{conv}(P)$ such that a' and b' are not consecutive on the boundary of $\text{conv}(P')$. Let $C' = (a', \dots, b')$ be the chain from a' to b' on the boundary of $\text{conv}(P')$ whose preimage C does not contain any points of $\text{extr}(P)$ except from a and b . Clearly, the edges of C have to be unavoidable in P . We call such a chain of unavoidable edges C between a and b an *unavoidable detour* and call the points in $C \setminus \{a, b\}$ its *elements*. Using Corollary 9 we immediately get the following result.

► **Theorem 16.** *If P with $|\text{extr}(P)| \geq 4$ has no unavoidable detour, then it is x -maximal.* ◀

This further implies

► **Theorem 17.** *For any given number m , there exists a number n such that among all order types of size n there are at least m crossing-maximal ones.*

General properties of unavoidable detours. Since unavoidable detours are fundamental for x -dominance, we identify some of their properties. For the following lemmas, let P be a point set containing an unavoidable detour C between two distinct extreme points a and b (recall that a and b are neighbored on the convex hull boundary of P and observe that there cannot exist a chain of unavoidable edges between two non-neighbored extreme points that does not use other extreme points of the set).

► **Lemma 18.** *The region bounded by the cycle $C \cup ab$ does not contain any point of $P \setminus C$.*

► **Lemma 19.** *In the rotation of any point $p \in P \setminus C$, the elements of C occur in the order defined by C and are consecutive among $C \cup \text{extr}(P)$.*

► **Lemma 20.** *All points of $P \setminus C$ are on the same side of any two points $p, q \in C$.*

► **Lemma 21.** *No two points in $P \setminus C$ have a supporting line intersecting C more than once.*

Unavoidable detours and x -dominating sets. We can construct examples where not only the elements of C jump out of a triangle. However, the points that jump out of a triangle are not arbitrary.

► **Lemma 22.** *Suppose we have two point sets $P \leq_x P'$ with an unavoidable detour $C = (a, \dots, b)$ s.t. $\text{extr}(P') = (\text{extr}(P) \cup C)'$. Let J be the set of points that jump out of a triangle in that mapping. Then the line defined by a point $j \in J$ and any other point $p \in P \setminus C$ intersects ab .*

► **Theorem 23.** *If a point set P has an unavoidable detour then it is x -dominated by an abstract order type (that may not be realizable by a point set).*

Proof. Let C be any unavoidable detour in P between two extreme points a and b . We construct a generalized configuration P' of points that x -dominates P such that $\text{extr}(P')$ consists of $\text{extr}(P)'$ plus the images of the elements of C . We transform $P \setminus C$ and its set of supporting lines into a generalized configuration of points. Since C is an unavoidable detour, all points of $P \setminus C$ are on the same side of any two points of C by Lemma 20. We replace C by a Jordan arc between a and b . This pseudo-segment intersects exactly those supporting lines of $P \setminus C$ as the initial edge ab , since any supporting line of $P \setminus C$ intersected C at most once by Lemma 21. Therefore, it can be extended to a pseudo-line intersecting each

supporting line exactly once along the initial supporting line of ab . The supporting lines that intersected C now intersect the pseudo-segment $a'b'$ in the same order as along C . We can therefore place the convex chain C' and the relevant parts of its supporting pseudo-lines arbitrarily close to the pseudo-segment. Again, an extension of the supporting lines is done appropriately along the initial supporting line of ab , making the resulting point set and its pseudo-line arrangement a valid generalized configuration of points that x -dominates P . ◀

In contrast to that, we have the following result, which implies that realizability of abstract order types is crucial in connection with x -maximal point sets.

► **Theorem 24.** *There are point sets that have at least four extreme points and an unavoidable detour, but are still x -maximal.*

The construction in the proof of Theorem 23 gives, in general, one out of many abstract order types that dominate P , depending on where in $\text{conv}(C')$ we place the points of $J' \setminus C'$. However, we have the following restriction.

► **Proposition 25.** *Suppose we have two point sets $P \leq_x P'$ with $|\text{extr}(P)| \geq 4$ and an unavoidable detour $C = (a, \dots, b)$ s.t. $\text{extr}(P') = (\text{extr}(P) \cup C)'$. Then $P \setminus C$ and $P' \setminus C'$ have the same order type.*

In the previous statements, we considered only single unavoidable detours. However, the results can again be applied to the dominating set if it contains an unavoidable detour. While this is fine when working with abstract order types, keep in mind that there may be non-realizable dominating abstract order types that are again dominated by a realizable one.

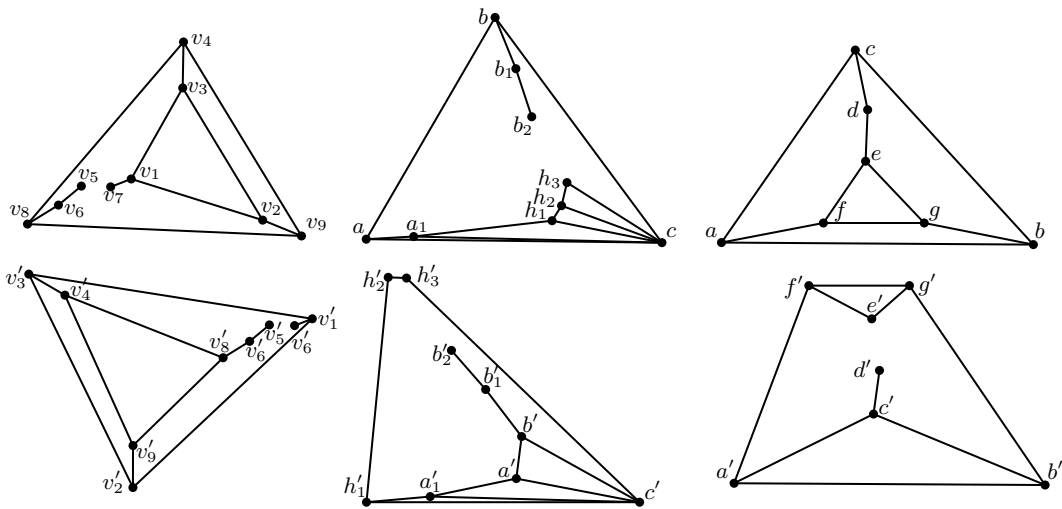
5 Different Extreme Points in the Dominating Set

While in Section 4 we gave a characterization of crossing-domination for the case where all images of the extreme points are again extreme points of the dominating set, we consider now, for $P \leq_x P'$, the case $\text{extr}(P)' \not\subseteq \text{extr}(P')$. Corollary 9 tells us that we have $|\text{extr}(P)| = 3$. We classify the different cases by the number of common extreme points (see Figure 5 for examples). For two of the cases, we will exploit the properties of crossing-equivalent subsets.

Properties of crossing-equivalent sets. It became obvious that the case $|\text{extr}(P')| = 3$ needs special attention. For that we try to understand the scenario when P and P' have a triangular convex hull, but some extreme points of P do not map to extreme points in P' .

As already discussed, crossing-equivalence of two point sets implies that they have the same rotation system. Point sets with the same rotation system have been discussed by Aichholzer et al. [2]. They show that if two order types with the same rotation system have no common extreme point (under the given bijection) these two order types are the only ones in that equivalence class, as there are exactly two triangles of unavoidable edges that can be transformed to the convex hull. (They give an upper bound of $|P| - 1$ on the number of different (labeled) order types in that equivalence class when there are common extreme points.) The following lemma states, in a different formulation, a result that is also given in [2].

► **Lemma 26.** *If $\text{extr}(P') = \{a', b', c'\} \neq (\text{extr}(P))'$, then $|\text{extr}(S)| = 3$ for all sets S with $\{a, b, c\} \subseteq S \subseteq P$. More precisely, P can be partitioned into three sets $P_a = \{a_0, \dots, a_{n_a}\}$, $P_b = \{b_0, \dots, b_{n_b}\}$, and $P_c = \{c_0, \dots, c_{n_c}\}$, such that $a_0 = a$, $b_0 = b$, and $c_0 = c$ and for all nonnegative integers i, j, k with $i \leq n_a$, $j \leq n_b$, and $k \leq n_c$, $\text{conv}(\{a_i, b_j, c_k\}) \cap P = \{a_0, \dots, a_i\} \cup \{b_0, \dots, b_j\} \cup \{c_0, \dots, c_k\}$.*



■ **Figure 5** x -Preserving mappings where only zero (left), one (middle), and two (right) extreme points stay the same. Unavoidable edges are drawn.

Observe that this constrains the position of the points to a high extent. Still, we can actually take an instance of any order type, scale it appropriately, and use it as one of the three subsets when constructing such a point set. The following theorem shows that this characterization also provides a construction for all sets in the equivalence class, meaning that if one of them is realizable then all of them are realizable.

► **Theorem 27.** *If there exists a triangle Δabc in a point set P such that P can be partitioned into three sets as in Lemma 26, then we can construct a point set P' with $P \sim_x P'$ and $\text{extr}(P') = \{a', b', c'\}$.*

No common extreme point. For the case $\text{extr}(P) \cap \text{extr}(P') = \emptyset$, we have

► **Theorem 28.** *If, for $P \leq_x P'$, $\text{extr}(P) \cap \text{extr}(P') = \emptyset$, then $|\text{extr}(P')| = 3$ and $P \sim_x P'$.*

Therefore, the two point sets have a different order type but the same rotation system. As already mentioned, there are only two order types in such an equivalence class [2].

One common extreme point. Let $\text{extr}(P) = \{a, b, c\}$ with $a', b' \notin \text{extr}(P')$, i.e., c is the only extreme point of P whose image is also an extreme point of P' . We denote the cycle on the boundary of $\text{conv}(P')$ by $H' = (h'_0, \dots, h'_k)$ and define $h'_0 = c$. Further, we consider the triangle Δabc and the cycle H to be oriented counterclockwise. Since the segment between c and any other point of P must not cross the chain (h_1, \dots, h_k) , the region where the other points of P can be placed is partitioned into two disjoint parts. Let $A \subset P$ be the points to the right of h_0h_1 (in particular, $a \in A$), and let $B \subset P$ be the points to the left of h_0h_k (implying $b \in B$). Observe that the cycle H may not be in convex position, but the radial order of its interior points around both a and b is h_1, \dots, h_k .

► **Lemma 29.** *The interior of the convex hull of h'_1, \dots, h'_k is empty.*

Since, by Lemma 29, we obtain a triangular convex hull when removing the vertices h_2, \dots, h_{k-1} , Lemma 12 directly gives us the following result.

► **Corollary 30.** *If the convex hulls of P and P' share exactly one point, then the rotations around all vertices remain the same when removing the elements h_2, \dots, h_{k-1} and their images from P and P' , respectively.*

Corollary 30 is a powerful tool for inspecting the structure of P and P' . It allows us to apply Lemma 26 to see that there is a hierarchy among the points of P in A and B . For example, in the rotation around a , all points of $B \setminus \{b\}$ are between b and h_k , and vice versa.

► **Corollary 31.** *No point of $P' \setminus \{c'\}$ is on the same side of $a'b'$ as c' .*

Also, the triangle Δabc behaves as in the case of crossing-equivalent sets and therefore its image is unavoidable when removing the points $\{h_2, \dots, h_{k-1}\}$.

► **Corollary 32.** *There is no point $p' \in P' \setminus H'$ s.t. the edge $c'p'$ intersects the edge $a'b'$.*

Let us partition A into subsets A_i s.t. A_i consists of the points in A that are in the interior of the triangle $\Delta ah_i h_{i+1}$, and do the same for B . Note that there is a line ℓ that separates A and B from H .

► **Lemma 33.** *Let i be the highest index such that $A_i \neq \emptyset$. Analogously, let j be the lowest index such that $B_j \neq \emptyset$. Then $i \leq j$.*

Corollary 30, in combination with Lemma 26, tells us that, after removing, say, a , the resulting subset has again a triangular convex hull; therefore the previous lemma also holds for the new extreme point a_1 . Hence, we get

► **Corollary 34.** *Let i be the highest index such that the supporting line of two points $a_k, a_l \in A$ intersects the edge $h_i h_{i+1}$, and let j be the lowest index such that the supporting line of two points $b_q, b_r \in B$ intersects the edge $h_j h_{j+1}$. Then $i \leq j$.*

It is easy to construct examples where there are multiple sets that strictly dominate P and that have the image of H as extreme points by having different rotations around the images of c . Still, Corollary 34 completes the characterization of the sets P and P' up to stretchability by similar reasoning as in the proof of Theorem 23.

Two common extreme points. If the convex hull boundary of P' contains two images of extreme points of P , then there has to be an unavoidable detour connecting such points a and b . This unavoidable detour gives the description of an abstract order type that strictly dominates P by Theorem 23, just like for sets with more extreme points. However, as the three order types with five points show (e.g., by swapping the labels v'_1 and v'_2 in Figure 2), we might not have $\text{extr}(P)' \subset \text{extr}(P')$. There may therefore be several abstract order types that strictly dominate P , and these may not be constructed the same way as in the proof of Theorem 23. However, we know the following.

► **Theorem 35.** *Let $\text{extr}(P) = \{a, b, c\}$ and $P \leq_x P'$. If $a', b' \in \text{extr}(P')$ and $c' \notin \text{extr}(P')$, then $a'b'$ is an edge of $\text{conv}(P')$.*

6 Algorithms and the Order Type Data Base

So far, we did not address the algorithmic problem of deciding whether a point set crossing-dominates another. We do so in this section. Further, we explain how we used the properties of for x -dominance to extract all crossing-maximal and crossing-minimal order types for up to 11 points. Theorem 3 is the key result for checking whether a point set not in convex position x -dominates another. Since Theorem 2 also gives a description of sets dominated by Conv_n , we can devise fast algorithms for checking x -dominance.

► **Lemma 36.** *The existence of a Hamiltonian cycle in the set of unavoidable edges of a point set P can be decided in polynomial time.*

► **Theorem 37.** *Given two point sets P and P' of size n , it can be decided in polynomial time whether $P \leq_x P'$.*

As a practical result of our work, we generated files containing exactly the realizations of the crossing-maximal and crossing-minimal order types for up to 10 points. For 11 points, such files were also extracted, but are likely to contain a small fraction of false-positives, which could not be filtered out by our methods due to the vast number of order types of size 11. We sketch our approach that allowed us to eventually generate these data bases within few CPU hours by extracting them from the Point Set Order Type Data Base (see [1, 5]).

Theorem 23 states that every point set P containing an unavoidable detour is dominated by an abstract order type. We use this to quickly find a point set in the Order Type Data Base that dominates a given one. First, we enumerate all cycles that consist of unavoidable edges that contain all extreme points of P . For each such cycle, we consider the abstract order type that has this cycle as convex hull boundary; all other point triples are oriented in the same way as in P . By the proof of Theorem 23, this abstract order type x -dominates P . (Note that this does not produce all possible abstract order types x -dominating P ; we do not get those sets where points with non-extreme image jump out of a triangle.) We get the lexicographically smallest λ -matrix as a fingerprint (see [9]) from each such abstract order type and search for this matrix in the data base. Using this method, most of the non-maximal sets could be identified quickly (we can perform a binary search for the matrix in the data base). For some of the sets, no realizable order type could be found this way. Therefore, for up to 10 points, we used a second iteration in which these sets were checked against all other ones that were not identified to be non-maximal in the first iteration. Further, all sets with triangular convex hull that contain a cycle of unavoidable edges of length at least four not violating the conditions of Corollary 34 were checked in this phase. Similarly, the following characterization allows us to identify crossing-minimal sets.

► **Theorem 38.** *Let $P, |\text{extr}(P)| \geq 4$, be a set of points such that, for every sequence $H = (h_i, \dots, h_j), |H| \geq 3$, of consecutive points on the convex hull boundary of P , $\text{conv}(H)$ contains a pair $p, q \in P \setminus H$ s.t. pq does not stab $h_i h_j$. Then P is crossing-minimal.*

For sets that have at least one chain C , $|C| \leq |\text{extr}(P)| - 1$, on the boundary of the convex hull that may be an unavoidable detour in a dominated set, we obtain a corresponding abstract order type. In such an abstract order type, the chain C is made reflex (which is, in general, not the only possibility), and for all points inside $\text{conv}(C)$, the triple orientations change accordingly. This way, many dominated order types can be found quickly.

Calculating the λ -matrix of an implicitly given abstract order type is a rather involved and therefore error-prone task, and even obtaining a bijection between two point sets using Theorem 3 is not completely fail-safe. However, when given the bijection between two point sets, we can, in a brute-force way, compare all 4-tuples to check crossing-dominance, a comparatively simple task. Once given the point set that witnesses non-maximality or non-minimality of another one, these sets can be compared quickly. This separate check was used to verify the resulting data.

Acknowledgements. We want to thank Oswin Aichholzer and Thomas Hackl for valuable discussions, in particular on handling the Order Type Data Base and on partitioning edges into spanning trees for 10 points, as well as for setting up computations on the Graz University

of Technology HPC system in early stages. The authors would like to acknowledge the use of HPC resources provided by the ZID of Graz University of Technology. This work was initiated during the 10th European Research Week on Geometric Graphs, held in St. Karl, Switzerland, from September 9 to September 13, 2013.

References

- 1 Oswin Aichholzer, Franz Aurenhammer, and Hannes Krasser. Enumerating order types for small point sets with applications. *Order*, 19(3):265–281, 2002.
- 2 Oswin Aichholzer, Jean Cardinal, Vincent Kusters, Stefan Langerman, and Pavel Valtr. Reconstructing point set order types from radial orderings. In *ISAAC 2014*, volume 8889 of *LNCS*, pages 15–26. Springer, 2014.
- 3 Oswin Aichholzer, Thomas Hackl, Matias Korman, Marc van Kreveld, Maarten Löffler, Alexander Pilz, Bettina Speckmann, and Emo Welzl. Packing plane spanning trees and paths in complete geometric graphs. In *Proc. 26th Canadian Conference on Computational Geometry (CCCG 2014)*, 2014.
- 4 Oswin Aichholzer, Thomas Hackl, Birgit Vogtenhuber, Clemens Huemer, Ferran Hurtado, and Hannes Krasser. On the number of plane graphs. In *Proc. 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2006)*, pages 504–513. ACM Press, 2006.
- 5 Oswin Aichholzer and Hannes Krasser. Abstract order type extension and new results on the rectilinear crossing number. *Comput. Geom.*, 36(1):2–15, 2007.
- 6 Boris Aronov, Paul Erdős, Wayne Goddard, Daniel J. Kleitman, Michael Klugerman, János Pach, and Leonard J. Schulman. Crossing families. *Combinatorica*, 14(2):127–134, 1994.
- 7 Prosenjit Bose, Ferran Hurtado, Eduardo Rivera-Campo, and David R. Wood. Partitions of complete geometric graphs into plane trees. *Comput. Geom.*, 34(2):116–125, 2006.
- 8 Jean Cardinal, Michael Hoffmann, and Vincent Kusters. On universal point sets for planar graphs. In *Computational Geometry and Graphs - Thailand-Japan Joint Conference (TJJCCGG 2012)*, volume 8296 of *LNCS*, pages 30–41. Springer, 2012.
- 9 Jacob E. Goodman and Richard Pollack. Multidimensional sorting. *SIAM J. Comput.*, 12(3):484–507, 1983.
- 10 Jacob E. Goodman and Richard Pollack. Semispaces of configurations, cell complexes of arrangements. *J. Combin. Theory Ser. A*, 37(3):257–293, 1984.
- 11 Jan Kynčl. Enumeration of simple complete topological graphs. *Eur. J. Comb.*, 30(7):1676–1685, 2009.
- 12 Nicolai E. Mněv. The universality theorems on the classification problem of configuration varieties and convex polytope varieties. In *Topology and Geometry—Rohlin Seminar*, volume 1346 of *Lecture Notes in Math.*, pages 527–544. Springer, 1988.
- 13 Michael J. Pelsmajer, Marcus Schaefer, and Daniel Štefankovič. Crossing numbers of graphs with rotation systems. *Algorithmica*, 60(3):679–702, 2011.
- 14 Gerhard Ringel. Extremal problems in the theory of graphs. In *Theory of Graphs and its Applications (Smolenice, 1963)*, pages 85–90. Publ. House Czechoslovak Acad. Sci., Prague, 1964.
- 15 Marcus Schaefer. Complexity of some geometric and topological problems. In *Graph Drawing*, volume 5849 of *LNCS*, pages 334–344. Springer, 2009.
- 16 Géza Tóth and Pavel Valtr. Geometric graphs with few disjoint edges. In *Symposium on Computational Geometry*, pages 184–191, 1998.
- 17 Benjamín Tovar, Luigi Freda, and Steven M. LaValle. Learning combinatorial map information from permutations of landmarks. *I. J. Robotic Res.*, 30(9):1143–1156, 2011.
- 18 Stephen K. Wismath. Point and line segment reconstruction from visibility information. *Int. J. Comput. Geometry Appl.*, 10(2):189–200, 2000.

Limits of Order Types*

Xavier Goaoc¹, Alfredo Hubard², Rémi de Joannis de Verclos³,
Jean-Sébastien Sereni⁴, and Jan Volec⁵

- 1 LIGM, Université Paris-Est Marne-la-Vallée, France
- 2 GEOMETRICA, Inria Sophia-Antipolis, France
- 3 G-SCOP, CNRS-INPG, Grenoble, France
- 4 LORIA, CNRS, France
- 5 Department of mathematics, ETH Zürich, Switzerland

Abstract

The notion of limits of dense graphs was invented, among other reasons, to attack problems in extremal graph theory. It is straightforward to define limits of order types in analogy with limits of graphs, and this paper examines how to adapt to this setting two approaches developed to study limits of dense graphs.

We first consider *flag algebras*, which were used to open various questions on graphs to mechanical solving via semidefinite programming. We define flag algebras of order types, and use them to obtain, via the semidefinite method, new lower bounds on the density of 5- or 6-tuples in convex position in arbitrary point sets, as well as some inequalities expressing the difficulty of sampling order types uniformly.

We next consider *graphons*, a representation of limits of dense graphs that enable their study by continuous probabilistic or analytic methods. We investigate how planar measures fare as a candidate analogue of graphons for limits of order types. We show that the map sending a measure to its associated limit is continuous and, if restricted to uniform measures on compact convex sets, a homeomorphism. We prove, however, that this map is not surjective. Finally, we examine a limit of order types similar to classical constructions in combinatorial geometry (Erdős-Szekeres, Horton . . .) and show that it cannot be represented by any somewhere regular measure; we analyze this example via an analogue of Sylvester’s problem on the probability that k random points are in convex position.

1998 ACM Subject Classification G.2.m Discrete Mathematics, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases order types, Limits of discrete structures, Flag algebras, Erdos-Szekeres, Sylvester’s problem

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.300

1 Introduction

The order type of a point set is a combinatorial encoding of the respective positions of its elements that suffices to determine many of its properties. For instance, the order type determines the halving lines or more generally the k -sets of the point set, which graphs admit crossing-free straight line drawings with vertices supported on that point set, the structure of

* This work was partially supported by the ANR projects PRESAGE (ANR-11-BS02-003) and HEREDIA (ANR-10-JCJC-0204-01), by the Institut Universitaire de France, by the SNSF grant 200021-149111 and by the Advanced Grant of the European Research Council GUDHI (Geometric Understanding in Higher Dimensions).

its simplicial depth partition, etc. Order types have received continued attention in discrete and computational geometry since the 1980's and are known to be rather intricate objects, for instance difficult to axiomatise [13].

In this paper, we report on an effort to apply to order types ideas from the theories of *dense graph limits* developed by Borgs, Chayes, Lovász, Sós, Szegedy and Vesztergombi and *flag algebras* developed by Razborov. While order types can be defined for points in d dimensions, in topological spaces, possibly with alignment, etc, all point sets considered in this paper are finite subsets of the euclidean plane, with no aligned triple.

Order types. Formally, order types are defined as follows. Define the *orientation* of a triangle pqr in the plane as *clockwise* (CW) if r lies to the right of the line pq oriented from p to q and *counter-clockwise* (CCW) if r lies to the left of that oriented line. (So the orientation of qpr is different from that of pqr .) We say that two planar point sets P and Q *have the same order type* if there exists a bijection $f : P \rightarrow Q$ that preserves orientations: for any triple of pairwise distinct points $p, q, r \in P$ the triangles pqr and $f(p)f(q)f(r)$ have the same orientation. The relation of having the same order type is easily checked to be an equivalence relation; the equivalence class, for this relation, of a finite point set P is called the *order type* of P . A point set P with order type ω is called a *realization* of ω .

When convenient, we extend to order types any notion that can be defined on a set of points and does not depend on a particular choice of realization. For instance we define the *size* of an order type ω as the cardinality $|\omega|$ of any of its realization. We adopt the convention that there is exactly one order type of each of the sizes 0, 1 and 2. We used the comprehensive list of all the order types of size up to 11, which was made available by Oswin Aichholzer¹ based on his work with Aurenhammer and Krasser [2] on the enumeration of order types. Throughout this paper, all non-trivial facts we use with reference on order types of small size can be traced back to that resource. We let \mathcal{O} denote the set of order types and \mathcal{O}_n the set of order types of size n .

Convergent sequences and limits of order types. We define the *density* $p(\omega, \omega')$ of an order type ω in another order type ω' as the probability that $|\omega|$ random points chosen uniformly from a point set realizing ω' have order type ω . (Observe that this probability depends solely on the order types and not on the choice of realization.) We say that a sequence $\{\omega_n\}_{n \in \mathbb{N}}$ of order types *converges* if the size $|\omega_n|$ goes to infinity as n goes to infinity, and if for any fixed order type ω the sequence of densities $p(\omega, \omega_n)$ converges. The *limit* of a convergent sequence of order types $\{\omega_n\}_{n \in \mathbb{N}}$ is the map

$$\begin{cases} \mathcal{O} & \rightarrow [0, 1] \\ \omega & \mapsto \lim_{n \rightarrow \infty} p(\omega, \omega_n) \end{cases}$$

A standard compactness argument reveals that limits of order types abound. Indeed, for each element ω_n in a sequence of order types, the map $\omega \in \mathcal{O} \mapsto p(\omega, \omega_n)$ can be seen as a point in $[0, 1]^{\mathbb{N}}$, which is compact by Tychonoff's theorem. Any sequence of order types whose size go to infinity therefore contains a convergent subsequence, and many extremal properties of point sets can be expressed in terms of limits of order types.

¹ <http://www.ist.tugraz.at/aichholzer/research/rp/triangulations/order-types/>

Problems and results. Let \diamond_k denote the order type of k points in convex position, $\text{conv}_k(n)$ the minimum number of convex k -gons in a set of n points in the plane, and $c_k = \lim_{n \rightarrow \infty} \text{conv}_k(n) / \binom{n}{k}$ their minimum density. Determining $\text{conv}_k(n)$ and c_k are classical problems in discrete geometry; see *eg* [6, Section 8.4, Problem 1]. Our first results are the following new lower bounds:

► **Proposition 1.** $c_5 \geq 0.0608516$ and $c_6 \geq 0.0018311$.

The best upper bounds we are aware of on these numbers are $c_5 \leq 0.0625$ and $c_6 \leq 0.005822$ and we are not aware of previously known lower bounds. We prove Proposition 1 by a reformulation of limits of order types as positive homomorphisms from a so-called flag algebra of order types into \mathbb{R} (see Proposition 8); this point of view allows a semidefinite programming formulation of the search for inequalities satisfied by limits of order types. Specifically, we argue that for any limit of order type ℓ

$$\ell(\diamond_5) \geq 0.0608516 \quad \text{and} \quad \ell(\diamond_6) \geq 0.0018311.$$

The number c_4 corresponds to the celebrated rectilinear crossing number of the complete graph and has been extensively investigated; the best lower bound we could obtain on c_4 via flag algebra is $c_4 \geq 0.37843917$, which is inferior to the best known bound $c_4 \geq 277/729 \approx 0.3799$ (the best known upper bound being $83247328/218791125 \approx 0.3804$). We refer the interested reader to the survey of Abrego, Fernandez-Merchant and Salazar [1].

Probabilistic constructions are sometimes effective ways of finding extremal combinatorial structures, a textbook example being the lower bound on Ramsey numbers for graphs. It is of course easy to generate a random order type, for instance by sampling i.i.d. some measure over \mathbb{R}^2 . It is not clear, however, how well such a method samples the space of order types, and hence how effective it would be to test conjectures and search for extremal examples (see *eg* [6, p 326]). Sampling order types of a given size *uniformly* looks difficult, as suggested by the lack of closed formulas for counting them, but we know of no formal justification of the hardness of this problem. As it turns out, limits of order types can also be defined as families of probability distributions on order types with certain internal consistencies (see Proposition 6) and our second result, also obtained by the semidefinite method of flag algebras, shows that a broad class of random generation method must exhibit some bias:

► **Proposition 2.** *For any limit of order types ℓ there exist two order types ω_1, ω_2 of size 6 such that $\ell(\omega_1) > 1.8208$ $\ell(\omega_2) > 0$.*

This inevitability of bias applies in particular to the random generation of order types by independent sampling of points from *any* measure over \mathbb{R}^2 . Specifically, let μ be a finite measure over \mathbb{R}^2 and for any order type ω let $p(\omega, \mu)$ denote the probability that $|\omega|$ random points chosen independently from $\frac{1}{\mu(\mathbb{R}^2)}\mu$ realize ω ; if every line is negligible for μ then $\ell_\mu : \omega \mapsto p(\omega, \mu)$ is a limit of order types (Lemma 7) and Proposition 2 applies.

Some hard problems in extremal graph theory were solved by representing limits of graphs by continuous functions, called *graphons*; a celebrated example is the application of large deviations principles to random Erdős-Renyi graphs $G(n, p)$ conditioned on the rare event of having triangle density q^3 for some $q > p$ [10], or on having a fixed degree sequence. At the heart of these results lies the fact that the relation between graphons and limits of graphs is not only a bijection, but an actual homeomorphism when both spaces are equipped with the adequate topologies. Since every finite measure μ over \mathbb{R}^2 (for which lines are negligible)

defines a limit of order types ℓ_μ , it is natural to wonder if such measures can represent all limits of order types, and whether this representation can be made an homeomorphism.

Let \mathcal{L} denote the space of limits of order types, endowed with the topology of the metric

$$d(\ell_1, \ell_2) := \sum_{i=1}^{\infty} 2^{-i} |\ell_1(\omega_i) - \ell_2(\omega_i)|, \tag{1}$$

where $\{\omega_1, \omega_2, \dots\}$ is some arbitrary enumeration of the set of order types. We first show that the map $\mu \mapsto \ell_\mu$, from the space of finite measures over \mathbb{R}^2 for which every line is negligible, equipped with the topology of the weak convergence, into \mathcal{L} , is continuous (Proposition 10). We next consider the special case of restrictions of the Lebesgue measure (the area) to compact convex sets with non empty interior (*convex bodies*). Let \mathcal{K} denote the quotient of the space of convex bodies by affine transforms: if K is a convex body, $[K] \in \mathcal{K}$ is the class of convex bodies affinely equivalent to K . We equip \mathcal{K} with the Banach-Mazur distance² d_{BM} , and remark that if K is a convex body and μ_K is the uniform measure on K then the limit of order types ℓ_{μ_K} depends only on $[K]$. We prove:

► **Theorem 3.** *Let K and K' be two planar convex bodies.*

- (i) *If for any $\omega \in \mathcal{O}$ we have $p(\omega, \mu_K) = p(\omega, \mu_{K'})$ then K and K' are affinely equivalent.*
- (ii) *For any $\omega \in \mathcal{O}$ we have $|p(\omega, \mu_K) - p(\omega, \mu_{K'})| \leq 2|\omega|d_{BM}(K, K')$.*

As a consequence, the map $[K] \in \mathcal{K} \mapsto \ell_{\mu_K} \in \mathcal{L}$ is a homeomorphism to its image.

The type of rigidity expressed by Theorem 3 extends to a broader class of measures (see the journal version).

We next show that there exists a limit of order types that cannot be represented, in the sense defined above, by a measure. The gist of the construction is to consider a sequence of measures whose weak limit (in the measure sense) contains a Dirac mass. Specifically, for any real $t \in (0, 1)$, let \odot_t be a probability distribution over \mathbb{R}^2 supported on two concentric circles, with radii 1 and t , respectively. Each of the two circles has \odot_t -measure 1/2, distributed proportionally to the length on that circle. We denote by ℓ_{\odot_t} the limit of order types associated to \odot_t (cf Lemma 7) and we let ℓ_\odot be the limit of a convergent sub-sequence of $\{\ell_{\odot_{1/n}}\}_{n \in \mathbb{N}^*}$. Here we prove:

► **Proposition 4.** *If μ is a compactly supported measure over \mathbb{R}^2 then there exists $\omega \in \mathcal{O}$ such that $p(\omega, \mu) \neq \ell_\odot(\omega)$.*

The proof that the compactness assumption can be removed is postponed to the journal version.

We finally examine a variation on constructions of Erdős and Szekeres [7] and Horton [8] to construct a limit of order types that no measure that is somewhere regular can represent. We first define inductively a sequence $\{P_n\}_{n \in \mathbb{N}}$ of point sets. The set P_0 consists of a single point. Assuming P_n has been constructed, we let P_{n+1} to be the union of two congruent copies of P_n , P_n^0 and P_n^1 , so that the following is true: any point in P_n^1 lies above every line spanned by two points from P_n^0 , any point from P_n^0 lies below every line spanned by two points from P_n^1 , and the least x coordinate of a point in P_n^1 is greater than the greatest x coordinate of a point in P_n^0 . We then let ω_n denote the order type of P_n and let ℓ_H denote the limit of some convergent subsequence of $\{\omega_n\}$.

² Recall that d_{BM} is $d_{BM}([K], [K']) := \ln \left(\inf \{r : r \in \mathbb{R}^+, \exists A \in GA(2, \mathbb{R}) : K \subset AK' \subset rK\} \right)$ where rK denotes a scaling of K by a factor r ; we abuse the terminology here as it is a distance only for symmetric convex sets.

► **Proposition 5.** *If μ is a measure over \mathbb{R}^2 that is, on an open set of positive μ -measure, absolutely continuous to either the Lebesgue measure or the length measure on a C^2 curve then there exists $k \geq 4$ such that $p(\diamond_k, \mu) > \ell_H(\diamond_k)$.*

Our proof hinges on the fact that when $k \rightarrow \infty$, $\ell_H(\diamond_k)$ decays faster than $p(\diamond_k, \mu)$ for any of the measures considered. For perspective, recall that it is known that the rectilinear crossing number equals the infimum, over all open sets $U \subset \mathbb{R}^2$ with finite Lebesgue measure, of $p(\diamond_4, \mu_U)$, where μ_U is the Lebesgue measure restricted to U [12].

2 Limits of order types

Order types can be understood as equivalence classes of chirotopes under the action of permutations (see below). As such, they are an example of *models* in the language of Razborov [11], and the theory of limits of order types is a special case of Razborov’s work. In this section, we give a geometric presentation of the various faces of limits of order types. We intend the presentation to be as self-contained as possible, and refer to general results of Razborov when needed.

Limits as probability distributions on order types. The *split probability* $p(\omega', \omega''; \omega)$, where $\omega', \omega'', \omega$ are order types, is the probability that a random partition of a point set realizing ω into two classes of sizes $|\omega'|$ and $|\omega''|$, chosen uniformly among all such partitions, produces two sets with respective order types ω' and ω'' . (In particular $p(\omega', \omega''; \omega) = 0$ if $|\omega| \neq |\omega_1| + |\omega_2|$.)

Fix two order types $\omega', \omega'' \in \mathcal{O}$, consider a converging sequence $\{\omega_n\}_{n \in \mathbb{N}}$ of order types, and let n_0 be such that $|\omega_n| \geq |\omega'| + |\omega''|$ for any $n \geq n_0$. For any $n \geq n_0$ let

$$\alpha_n = p(\omega', \omega_n)p(\omega'', \omega_n) \quad \text{and} \quad \beta_n = \sum_{\omega \in \mathcal{O}_{|\omega'|+|\omega''|}} p(\omega', \omega''; \omega)p(\omega, \omega_n).$$

Now, fix some point set P with order type ω_n . On the one hand, α_n equals the probability that two independent events both happens: (i) that a set P' of $|\omega'|$ random points chosen uniformly from P have order type ω' , and (ii) that another set P'' of $|\omega''|$ random points chosen uniformly from P have order type ω'' . On the other hand, observe that β_n equals the probability that (i) and (ii) happen *and* that P' and P'' are disjoint. The difference $|\alpha_n - \beta_n|$ is therefore bounded from above by the probability that P' and P'' intersect. Bounding from above the probability that P' and P'' have an intersection of one or more elements by the expected size of $P' \cap P''$, we have

$$\left| p(\omega', \omega_n)p(\omega'', \omega_n) - \sum_{\omega \in \mathcal{O}_{|\omega'|+|\omega''|}} p(\omega', \omega''; \omega)p(\omega, \omega_n) \right| \leq \mathbb{E}(|P' \cap P''|) = \frac{|\omega'| |\omega''|}{|\omega_n|}. \quad (2)$$

Taking $n \rightarrow \infty$ in (2) we see that every limit of order types ℓ satisfies

$$\forall \omega', \omega'' \in \mathcal{O}, \quad \ell(\omega')\ell(\omega'') = \sum_{\omega \in \mathcal{O}_{|\omega'|+|\omega''|}} p(\omega', \omega''; \omega)\ell(\omega). \quad (3)$$

These internal consistency relations provide the following alternative characterization of limits as families of distributions on order types:

► **Proposition 6** (Lovasz and Szegedy [9, Theorem 2.2], Razborov [11, Theorem 3.3]). *A function $\ell : \mathcal{O} \rightarrow \mathbb{R}$ is a limit of order types if and only if it satisfies Condition (3) and for every $n \in \mathbb{N}$ the restriction $\ell|_{\mathcal{O}_n}$ is a probability distribution on \mathcal{O}_n .*

Limits from measures over \mathbb{R}^2 . As spelled out in the paragraph following Proposition 2, measures over \mathbb{R}^2 provide examples of limits of order types.

► **Lemma 7.** *The map $\ell_\mu : \omega \in \mathcal{O} \mapsto p(\omega, \mu)$ is a limit of order types if and only if μ is a measure for which every line is negligible.*

Proof. Assume that ℓ_μ is a limit of order types and let $\{\omega_n\}_{n \in \mathbb{N}}$ be a sequence converging to μ . Let \cdot denote the order type of size 3. We have

$$p(\cdot, \mu) = \ell_\mu(\cdot) = \lim_{n \rightarrow \infty} p(\cdot, \omega_n) = 1$$

so three random points chosen independently from $\frac{1}{\mu(\mathbb{R}^2)}\mu$ are aligned with probability 0, and every line is negligible for μ .

Conversely, assume that μ is a measure for which every line is negligible. For every $n \geq 3$ the restriction of ℓ_μ to \mathcal{O}_n is a probability distribution. Moreover, for any order types $\omega', \omega'' \in \mathcal{O}$ we have

$$Pr_\mu(\omega')Pr_\mu(\omega'') = \sum_{\omega \in \mathcal{O}_{|\omega'|+|\omega''|}} Pr_\mu(\omega)p(\omega', \omega''; \omega)$$

since the union of two independent random sets of sizes $|\omega_1|$ and $|\omega_2|$ has size $|\omega_1| + |\omega_2|$ almost surely. Proposition 6 implies that ℓ_μ is a limit of order types. ◀

Limits as positive algebra homomorphisms. Let $\{\omega_n\}_{n \in \mathbb{N}}$ be a sequence of order types converging to a limit ℓ . Let $\omega \in \mathcal{O}$, let $k \geq |\omega|$ and let n_0 be large enough so that $|\omega_n| \geq k$ for $n \geq n_0$. A simple conditioning argument yields that for any $n \geq n_0$,

$$p(\omega, \omega_n) = \sum_{\omega' \in \mathcal{O}_k} p(\omega, \omega')p(\omega', \omega_n).$$

Indeed, the probability that a random sample realizes ω is the same if we sample uniformly $|\omega|$ points from a realization of ω_n , and if we sample k points uniformly from that realization, then select a subset of $|\omega|$ of these k points uniformly. It follows that any limit ℓ of order types satisfies:

$$\forall \omega \in \mathcal{O}, \forall k \geq |\omega|, \quad \ell(\omega) = \sum_{\omega' \in \mathcal{O}_k} p(\omega, \omega')\ell(\omega'). \tag{4}$$

Now, let $\mathbb{R}\mathcal{O}$ be the set of all finite formal linear combinations of elements of \mathcal{O} with real coefficients and consider the quotient vector space

$$\mathcal{A} = \mathbb{R}\mathcal{O}/\mathfrak{K} \quad \text{where} \quad \mathfrak{K} = \text{vect} \left\{ \omega - \sum_{\omega' \in \mathcal{O}_{|\omega|+1}} p(\omega, \omega')\omega' : \omega \in \mathcal{O} \right\}.$$

We define a product on \mathcal{O} by

$$\forall \omega_1, \omega_2 \in \mathcal{O}, \quad \omega_1 \times \omega_2 = \sum_{\omega \in \mathcal{O}_{|\omega_1|+|\omega_2|}} p(\omega_1, \omega_2; \omega)\omega \tag{5}$$

and extend it linearly to $\mathbb{R}\mathcal{O}$. This extension is compatible with the quotient by \mathfrak{K} [11, Lemma 2.4] and therefore turns \mathcal{A} into an algebra.

We call an algebra homomorphism from \mathcal{A} to \mathbb{R} *positive* if it maps every element of \mathcal{O} to a non-negative real, and denote by $\text{Hom}^+(\mathcal{A}, \mathbb{R})$ the set of positive algebra homomorphism from \mathcal{A} to \mathbb{R} . (Note that any algebra homomorphism sends \cdot , the order-type of size one, to the real 1 as it is the neutral element for the product on order types.)

► **Proposition 8** ([11, Theorem 3.3b]). *A map $f : \mathcal{O} \rightarrow \mathbb{R}$ is a limit of order types if and only if its linear extension is compatible with the quotient by \mathfrak{K} and defines a positive homomorphism from \mathcal{A} to \mathbb{R} .*

We write that an element of \mathcal{A} is non-negative when its image under any positive homomorphism is non-negative. The algebra \mathcal{A} allows us to compute effectively with density relations that hold for every limit ℓ .

► **Example 9.** Let us denote by \cdot the order type on one point, by $\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \end{smallmatrix}$ and $\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix}$ the two order types of size four and by $\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix}$, $\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix}$, and by $\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix}$ the three order types of size five, seen as elements of \mathcal{A} . From Identity (4) we get

$$\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} = \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} + \frac{3}{5} \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} + \frac{1}{5} \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} \quad \text{and} \quad \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} + \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} + \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} = \cdot \tag{6}$$

Since for any limit of order types ℓ we have $\ell(\cdot) = 1$, the above easily implies that $\ell(\diamond_4) \geq 1/5$. Using again Identity (4), and the non-negativity of $\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix}$ we then obtain:

$$\frac{2}{5} \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} \geq \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} - \frac{3}{5} (\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} + \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} + \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix}) = \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} - \frac{3}{5} \cdot$$

and $\ell(\diamond_5) \geq \frac{5}{2}\ell(\diamond_4) - \frac{3}{2}$ for any limit of order types ℓ .

3 The semidefinite method for order types

Let us give an intuition of how the *semidefinite method* works on an example. A simple (mechanical) examination of 6405 order types reveals that $p(\diamond_4, \omega) \geq 19/70$ for any $\omega \in \mathcal{O}_8$. With Identity (4) this implies $\begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} \geq 19/70 \cdot$ or equivalently $c_4 \geq 19/70 > 0.2714$. Observe that for any $C \in \mathcal{A}$ and any (linear extension of a) limit of order types ℓ we have $\ell(C \times C) = \ell(C)^2 \geq 0$ by Proposition 8. We thus have at our command an infinite source of inequalities to consider to try and improve the above bounds. For instance, a tedious but elementary computation yields that

$$\left(\frac{6}{25} \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} - \frac{11}{125} \begin{smallmatrix} \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \end{smallmatrix} \right)^2 + \frac{298819}{1093750} \sum_{\omega \in \mathcal{O}_8} \omega = \sum_{\omega \in \mathcal{O}_8} a_\omega \omega,$$

where $a_\omega \leq p(\diamond_4, \omega)$ for every $\omega \in \mathcal{O}_8$. This implies that $\ell(\diamond_4) \geq 298819/1093750 > 0.2732$ for any limit of order types ℓ . The search for interesting combinations of such inequalities can be done by semidefinite programming.

3.1 Improving the semidefinite method via rooting and averaging

The effectiveness of the semidefinite method for limits of graphs was greatly enhanced by considering partially labelled graphs. We unfold here a similar machinery, using some blend of order types and chirotopes.

Partially labelled point sets, flags, σ -flags and \mathcal{A}^σ . A point set *partially labelled* by a finite set \mathcal{Z} (the *labels*) is a finite point set P together with some injective map $L : \mathcal{Z} \rightarrow P$; we will write this (P, \mathcal{Z}, L) when we need to make explicit the set of labels and the label map. We say that two partially labelled point sets (P, \mathcal{Z}, L) and (P', \mathcal{Z}, L') have the same *flag* if there exists a bijection $\phi : P \rightarrow P'$ that preserves both the orientation and the labelling, the latter meaning that $\phi(L(i)) = L'(i)$ for any $i \in \mathcal{Z}$. The relation of having the same flag is an

equivalence relation, and a *flag* is an equivalence class for this relation. Again, we call any partially labelled point set a realization of its equivalence class, and the size $|\tau|$ of a flag τ is the cardinality of any of its realizations.

We call a flag where all the points are labelled, *ie* where $|P| = |\mathcal{Z}|$ in some realization (P, \mathcal{Z}, L) , a \mathcal{Z} -*chirotope*. (When $\mathcal{Z} = [k] = \{1, 2, \dots, k\}$ a \mathcal{Z} -chirotope coincides with the classical notion of chirotope.) Discarding the unlabelled part of a flag τ with label set \mathcal{Z} yields some \mathcal{Z} -chirotope σ called the *root* of τ . We call a flag with root σ a σ -*flag* and we denote by \mathcal{X}^σ the set of σ -flags. The *unlabelling* τ^\emptyset of a flag τ with realization (P, \mathcal{Z}, L) is the order type of P .

Let \mathcal{Z} be a set of labels and σ a \mathcal{Z} -chirotope. We define densities and split probabilities for σ -flags like for order types. Namely, let τ, τ' and τ'' be σ -flags realized, respectively, by (P, \mathcal{Z}, L) and (P', \mathcal{Z}, L') and (P'', \mathcal{Z}, L'') . The *density* of τ in τ' is the probability that for a random subset S of size $|P| - |\mathcal{Z}|$, chosen uniformly in $P' \setminus L'(\mathcal{Z})$, the partially labelled set $(S \cup L'(\mathcal{Z}), \mathcal{Z}, L')$ has flag τ . The *split probability* $p(\tau, \tau'; \tau'')$ is the probability that for a random subset S of size $|P| - |\mathcal{Z}|$, chosen uniformly in $P'' \setminus L''(\mathcal{Z})$, the partially labelled set $(S \cup L''(\mathcal{Z}), \mathcal{Z}, L'')$ and $(P'' \setminus S, \mathcal{Z}, L'')$ have, respectively, flags τ and τ' .

We can finally define an algebra of σ -flags as for order types. We equip the quotient vector space

$$\mathcal{A}^\sigma = \mathbb{R}\mathcal{X}^\sigma / \mathfrak{K}^\sigma \quad \text{where} \quad \mathfrak{K}^\sigma = \text{vect} \left\{ \omega - \sum_{\omega' \in \mathcal{X}_{|\omega|+1}^\sigma} p(\omega, \omega') \omega' : \omega \in \mathcal{X}^\sigma \right\}$$

by the linear extension of the product defined on \mathcal{X}^σ by $\tau \times \tau' = \sum_{\tau'' \in \mathcal{X}_{|\tau|+|\tau'|-|\sigma|}^\sigma} p(\tau, \tau'; \tau'') \tau''$.

Rooted homomorphisms and averaging. The use of the \mathcal{A}^σ 's to study \mathcal{A} relies on three tools which we now introduce. We first define an *embedding* of a \mathcal{Z} -chirotope in an order type ω as a σ -flag with root σ and unlabelling ω . We use *random embeddings* with the following distribution in mind: fix some point set realizing ω , consider the set I of injections $f : \mathcal{Z} \rightarrow P$ such that (P, \mathcal{Z}, f) is a σ -flag, choose some injection f_r from I uniformly at random, and consider the flag of (P, \mathcal{Z}, f_r) . We call this the *labelling* distribution on embeddings of σ in ω .

Next, we associate to any convergent sequence of order types $\{\omega_n\}_{n \in \mathbb{N}}$, and for any \mathcal{Z} -chirotope σ , a probability distribution on $\text{Hom}^+(\mathcal{A}^\sigma, \mathbb{R})$. For any $n \in \mathbb{N}$, the labelling distribution on embeddings of σ in ω_n defines a probability distribution \mathbf{P}_n^σ on mappings from \mathcal{A}^σ to \mathbb{R} ; specifically, for each embedding θ_n of σ in ω_n we consider the map

$$f_{\theta_n} : \begin{cases} \mathcal{A}^\sigma & \rightarrow \mathbb{R} \\ \tau & \mapsto p(\tau, \theta_n) \end{cases}$$

and assign to it the same probability, under \mathbf{P}_n^σ , as the probability of θ_n under the labelling distribution. Since $p(\omega, \omega_n)$ converges as $n \rightarrow \infty$ for every $\omega \in \mathcal{O}$, the sequence $\{\mathbf{P}_n^\sigma\}_{n \in \mathbb{N}}$ weakly converges to a Borel probability measure on $\text{Hom}^+(\mathcal{A}^\sigma, \mathbb{R})$ [11, Theorems 3.12 and 3.13] which we denote by \mathbf{P}_ℓ^σ . Moreover, if $\ell(\sigma^\emptyset) > 0$ then the homomorphism induced by ℓ determines the probability distribution \mathbf{P}_ℓ^σ [11, Theorem 3.5].

We finally define, for any \mathcal{Z} -chirotope σ , an *averaging* (or downward) operator $[\cdot]_\sigma : \mathcal{A}^\sigma \rightarrow \mathcal{A}$ as the linear operator defined on the elements of $\tau \in \mathcal{X}^\sigma$ by $[\tau]_\sigma := p_\tau^\sigma \cdot \tau^\emptyset$, where p_τ^σ is the probability that a random embedding of σ to τ^\emptyset (for the labelling distribution) equals τ . Here are a few examples of σ -flags, where $\sigma = 123$ is the CCW chirotope of size 3:

$$\left[\begin{array}{cc} 3 \cdot & \\ 1 \cdot & \cdot 2 \end{array} \right]_{123} = \frac{1}{2} \cdot \quad \left[\begin{array}{cc} \cdot & 3 \cdot \\ 1 \cdot & \cdot 2 \end{array} \right]_{123} = \frac{1}{6} \cdot \cdot \quad \left[\begin{array}{ccc} 3 \cdot & & \\ 1 \cdot & \cdot & \cdot 2 \end{array} \right]_{123} = \frac{1}{8} \cdot \cdot \cdot$$

For any given \mathcal{Z} -chirotope σ and a limit of order types ℓ , we have the following important identity [11, Lemma 3.11]:

$$\forall \tau \in \mathcal{A}^\sigma, \quad \ell(\llbracket \tau \rrbracket_\sigma) = \ell(\llbracket \sigma \rrbracket_\sigma) \int_{\phi^\sigma \in \text{Hom}^+(\mathcal{A}^\sigma, \mathbb{R})} \phi^\sigma(\tau) d\mathbf{P}_\ell^\sigma. \quad (7)$$

In particular, $\ell(\llbracket C^\sigma \rrbracket_\sigma) \geq 0$ for any $C^\sigma \in \mathcal{A}^\sigma$ such that $\phi^\sigma(C^\sigma) \geq 0$ almost surely for $\phi^\sigma \in \text{Hom}^+(\mathcal{A}^\sigma, \mathbb{R})$, relatively to \mathbf{P}_ℓ^σ ; for any limit of order types ℓ and any \mathcal{Z} -chirotope σ we therefore have

$$\forall C^\sigma \in \mathcal{A}^\sigma, \quad \ell\left(\llbracket (C^\sigma)^2 \rrbracket_\sigma\right) \geq 0. \quad (8)$$

3.2 The semidefinite method for order types

The operator $\llbracket \cdot \rrbracket_\sigma$ is linear, so for every $\phi \in \text{Hom}^+(\mathcal{A}, \mathbb{R})$, any $A_1^\sigma, A_2^\sigma, \dots, A_I^\sigma \in \mathcal{A}^\sigma$, and any non-negative reals z_1, z_2, \dots, z_I , we have

$$\phi\left(\left\llbracket \sum_{i \in [I]} z_i \cdot (A_i^\sigma)^2 \right\rrbracket_\sigma\right) \geq 0.$$

For any finite set of flags $S \subseteq \mathcal{O}^\sigma$ and for any real, symmetric, positive semidefinite matrix M of size $|S| \times |S|$, we have $\phi(\llbracket v_S^T M v_S \rrbracket_\sigma) \geq 0$, where v_S is the vector in $(\mathcal{A}^\sigma)^{|S|}$ whose i th coordinate equals the i th element of S (for some given order). This recasts the search for a good “positive” quadratic combination as a semidefinite programming problem.

Let N be an integer, $f = \sum_{\omega \in \mathcal{O}_N} f_\omega \omega$ some target function, and $\sigma_1, \dots, \sigma_k$ a finite list of chirotopes so that $|\sigma_i| \equiv N \pmod{2}$. For each $i \in [k]$, let v_i be the $|\mathcal{X}_{(N+|\sigma_i|)/2}^{\sigma_i}|$ -dimensional vector with i th coordinate equal to the i th element of $\mathcal{X}_{(N+|\sigma_i|)/2}^{\sigma_i}$. We look for a real b as large as possible subject to the constraint that there exists k real, symmetric, positive semidefinite matrices M_1, M_2, \dots, M_k , where M_i has size $|v_i| \times |v_i|$, so that

$$\forall \omega \in \mathcal{O}_N, \quad f_\omega \geq a_\omega \quad \text{where} \quad \sum_{\omega \in \mathcal{O}_N} a_\omega \omega = \sum_{i \in [k]} \llbracket v_i^T M_i v_i \rrbracket_{\sigma_i} + b \sum_{\omega \in \mathcal{O}_N} \omega. \quad (9)$$

The values of the a_ω 's are determined by b , the entries of the matrices M_1, M_2, \dots, M_k , the splitting probabilities $p(\tau', \tau''; \tau)$, where $\tau', \tau'' \in \mathcal{X}_{(N+|\sigma_i|)/2}^{\sigma_i}$ and $\tau \in \mathcal{X}_N^{\sigma_i}$, and the probabilities $p_\tau^{\sigma_i}$, where $\tau \in \mathcal{O}_N^{\sigma_i}$. Moreover, finding the maximum value of b and the entries of the matrices M_i can be formulated as a semidefinite program.

Effective semidefinite programming for flags of order types. In order to use a semidefinite programming software for finding a solution of programs in the form of (9), it is enough to generate the sets \mathcal{O}_N and $\mathcal{X}_N^{\sigma_i}$, the split probabilities $p(\tau', \tau''; \tau)$, where $\tau', \tau'' \in \mathcal{X}_{(N+|\sigma_i|)/2}^{\sigma_i}$ and $\tau \in \mathcal{X}_N^{\sigma_i}$, and the probabilities $p_\tau^{\sigma_i}$, where $\tau \in \mathcal{O}_N^{\sigma_i}$.

We generated the sets and the values by brute force up to $N = 8$. The only non-trivial algorithmic step is deciding whether two order types, represented by point sets, are equivalent. This can be done by computing some canonical ordering of the points that turn two point sets with the same order type into point sequences with the same chirotope. Aloupis et al. [4] recently proposed an algorithm performing that in time $O(n^2)$; the method we implemented takes time $O(n^2 \log n)$ and seems to be folklore (we learned it from Pocchiola and Pilaud). For solving the semidefinite program itself, we used a library called CSDP [5]. The input data for CSDP was generated using a mathematical software SAGE [14].

Setting up the semidefinite programs. In the rest of this section we work with $N = 8$ and use chirotopes labelled $\sigma_1, \sigma_2, \dots, \sigma_{24}$ where σ_1 the empty chirotope, σ_2 the only chirotope of size two, σ_3 and σ_4 the two chirotopes of size 4 depicted on the left, and $\sigma_5, \dots, \sigma_{24}$ a fixed set of 20 chirotopes of size 6 so that $\mathcal{O}_6 = \{\sigma_5^\emptyset, \dots, \sigma_{24}^\emptyset\}$; note that since $|\mathcal{O}_6| = 20$, what follows will not depend on the choices made in labelling $\sigma_5, \dots, \sigma_{24}$. The vectors v_1, v_2, \dots, v_{24} described in the previous paragraph for this choice of N and σ_i 's have lengths 2, 44, 468, 393, 122, 112, 114, 101, 101, 103, 106, 103, 103, 120, 102, 108, 94, 90, 91, 91, 95, 95, 92, 104, respectively.

Computations proving Propositions 1 and 2. We solved two semidefinite programs with the above choice of parameters for $f = \sum_{\omega \in \mathcal{O}_8} p(\diamond_5, \omega)$ and $f = \sum_{\omega \in \mathcal{O}_8} p(\diamond_6, \omega)$ and obtained real symmetric positive semidefinite matrices M_1, \dots, M_{24} and M'_1, \dots, M'_{24} with rational entries so that

$$\sum_{\omega \in \mathcal{O}_8} p(\diamond_5, \omega)\omega \geq \sum_{i \in [24]} \llbracket v_i^T M_i v_i \rrbracket_{\sigma_i} + \frac{15715211616602583691}{258254417031933722624} \sum_{\omega \in \mathcal{O}_8} \omega,$$

and

$$\sum_{\omega \in \mathcal{O}_8} p(\diamond_6, \omega)\omega \geq \sum_{i \in [24]} \llbracket v_i^T M'_i v_i \rrbracket_{\sigma_i} + \frac{67557324685725989}{36893488147419103232} \sum_{\omega \in \mathcal{O}_8} \omega.$$

The lower bounds on c_5 and c_6 then follow from Identity (4).

Assume (without loss of generality) that $\mathcal{O}_6 = \{\omega_{6,1}, \omega_{6,2}, \dots, \omega_{6,20}\}$. Solving two semidefinite programs, we obtained real symmetric positive semidefinite matrices M_1, \dots, M_{24} and M'_1, \dots, M'_{24} as well as non-negative rational values d_1, \dots, d_{20} and d'_1, \dots, d'_{20} so that

$$\sum_{j \in [20]} d_j \left(\omega_{6,j} - \frac{1}{32} \sum_{\omega \in \mathcal{O}_8} \omega \right) + \sum_{i \in [24]} \llbracket v_i^T M_i v_i \rrbracket_{\sigma_i} < 0$$

and

$$\sum_{j \in [20]} d'_j \left(-\omega_{6,j} + \frac{1}{18} \sum_{\omega \in \mathcal{O}_8} \omega \right) + \sum_{i \in [24]} \llbracket v_i^T M'_i v_i \rrbracket_{\sigma_i} < 0.$$

They imply that there is no $\ell \in \text{Hom}^+(\mathcal{A}, \mathbb{R})$ such that, respectively $\ell(\omega) \geq 1/32$ for every $\omega \in \mathcal{O}_6$, or such that $\ell(\omega) \leq 1/18$ for every $\omega \in \mathcal{O}_6$. Together this proves Proposition 2 with an imbalance bound of $32/18 > 1.77$. The better bound of Proposition 2 is obtained by a refinement of this approach where the order types with minimum and maximum probability are prescribed; this requires solving over 700 semidefinite programs.

The numerical values of the entries of all the matrices M_1, \dots, M_{24} and coefficients d_1, \dots, d_{20} mentioned above can be downloaded from the web page <http://honza.ucw.cz/proj/orderotypes/>. In fact, the matrices M_1, \dots, M_{24} are not stored directly, but as an appropriate non-negative sum of squares, which makes the verification of positive semidefiniteness trivial. To make an independent verification of our computations easier, we created sage scripts called “verify_prop*.sage”, available from the same web page.

4 Representation of limits by measures

Let \mathcal{L} denote the space of limits of order types endowed with the topology of the distance given by Equation (1). Let \mathcal{M} denote the space of finite measures over \mathbb{R}^2 for which every line is negligible, equipped with the topology of the weak convergence³.

► **Proposition 10.** *The map $\mu \in \mathcal{M} \mapsto \ell_\mu \in \mathcal{L}$ is continuous.*

Proof. For $k \geq 1$ and any measure μ over \mathbb{R}^2 we let μ^k denote the k -fold product measure over \mathbb{R}^{2k} . For any order type ω we let $\mathcal{R}_\omega \subset \mathbb{R}^{2|\omega|}$ denote the space of all realizations of ω , that is \mathcal{R}_ω contains all $2|\omega|$ -tuples $(x_1, y_1, x_2, y_2, \dots, x_{|\omega|}, y_{|\omega|})$ such that the points $(x_1, y_1), (x_2, y_2), \dots, (x_{|\omega|}, y_{|\omega|})$ realize ω . Observe that $p(\omega, \mu) = \mu^k(\mathcal{R}_\omega)$.

Let $\{\mu_n\}_{n \in \mathbb{N}}$ be a sequence of measures in \mathcal{M} weakly converging to a measure $\mu \in \mathcal{M}$. For any k , the k -fold product measures μ_n^k converge weakly to μ^k . Moreover, for every order type ω the boundary $\partial\mathcal{R}_\omega$ consists solely of planar point sets with at least one aligned triple. The measure $\mu^k(\partial\mathcal{R}_\omega)$ is therefore bounded from above by the probability that $|\omega|$ random points sampled from μ contains at least three aligned points. Since every line is negligible for μ , this ensures that $\mu^k(\partial\mathcal{R}_\omega) = 0$ and therefore for any ω , $\ell_{\mu_n}(\omega) = \mu_n^k(\mathcal{R}_\omega) \rightarrow \mu^k(\mathcal{R}_\omega) = \ell_\mu(\omega)$. ◀

In the rest of this section we prove Theorem 3, which strengthens Proposition 10 for uniform measures on convex bodies, and prove Proposition 4 and 5.

4.1 Proof of Theorem 3

The gist of our proof is to relate a convex set K to the limit of order types ℓ_K induced by the measure μ_K through a family of positive algebra homomorphism $\phi_{P, \mu_K, P'}(\tau) \in \text{Hom}^+(\mathcal{A}^{\sigma'}, \mathbb{R})$ defined for any point sequences P and P' .

For two chirotopes σ, σ' we write $\sigma' \triangleright \sigma$ and say that σ' extends σ if there exists sequences of points $P = \{p_1, p_2, \dots, p_n\}$ and $P' = \{p'_1, p'_2, \dots, p'_{n'}\}$ so that P has chirotope σ and the sequence $P \cup P' := \{q_1, q_2, \dots, q_{n+n'} : q_i = p_i \text{ for } i \leq n \text{ and } q_i = p'_{i-n} \text{ for } i > n\}$ has chirotope σ' . Let μ be a measure over \mathbb{R}^2 for which lines are negligible. For any σ' -flag τ we let $\phi_{P, \mu, P'}(\tau)$ denote the probability that $|\tau| - |\sigma'|$ random unlabeled points chosen independently from μ define, together with $P \cup P'$, a partially labelled sequence realizing τ . The map $\tau \in \mathcal{A}^{\sigma'} \mapsto \phi_{P, \mu, P'}(\tau)$ is easily seen to be a positive algebra homomorphism from $\mathcal{A}^{\sigma'}$ to \mathbb{R} . For a fixed P and varying P' such that $n' = |P'|$, we define a map

$$\phi_{P, \mu, \cdot} : \begin{cases} (K)^{n'} & \rightarrow \bigcup_{\sigma' \triangleright \sigma; |\sigma'| = |\sigma| + n'} \text{Hom}^+(\mathcal{A}^{\sigma'}, \mathbb{R}) \\ P' & \mapsto \{\tau \mapsto \phi_{P, \mu, P'}(\tau)\} \end{cases}$$

where we assume that τ is a σ' -flag and $P \cup P'$ have chirotope σ' . (For the sake of the presentation, we write $\phi_{P, \mu, t}$ in place of $\phi_{P, \mu, \{t\}}$ when applying $\phi_{P, \mu, \cdot}$ to a singleton.) The key fact about this map is that if we push forward $\mu^{n'}$ through $\phi_{P, \mu, \cdot}$ it induces a probability distribution on $\bigcup_{|\sigma'| = |\sigma| + n', \sigma' \triangleright \sigma} \text{Hom}^+(\mathcal{A}^{\sigma'}, \mathbb{R})$ that turns out, due to a theorem of Razborov, to be essentially determined by ℓ_μ . We will denote by \mathbf{Q} a set of n' random points chosen independently from μ , and by $\phi_{P, \mu, \mathbf{Q}}$ the random homomorphism corresponding to the push forward of $\mu^{n'}$.

³ A sequence $\{\mu_n\}_{n \in \mathbb{N}}$ of measures weakly converges to a measure μ if $\mu_n(A) \rightarrow \mu(A)$ for every measurable set A such that $\mu(\partial A) = 0$, where ∂ stands for the topological boundary.

We first argue that the geometry of K , up to affine transformation, can be retrieved from these homomorphisms since they encode ratios of triangle areas that determine certain barycentric coordinates.

► **Lemma 11.** *Let K be a convex body, $\{t_1, t_2, t_3, t\} \subset K$. For any triangle T' supported in $\{t_1, t_2, t_3, t\}$, the ratio of the area of T' to the area of $t_1 t_2 t_3$ is determined by the values of $\phi_{\{t_1, t_2, t_3\}, \mu_K, t}$ on σ -flags of size 5, where σ is the chirotope of $\{t_1, t_2, t_3, t\}$.*

Proof. The relative area of a triangle T' with respect to a triangle T is the quotient $\frac{\text{area}(T')}{\text{area}(T)}$. Let us begin with the case in which $t \in \text{conv}(T)$ with $T = \{t_1, t_2, t_3\}$. The point t subdivides T into 3 triangles. Without loss of generality, let τ be the σ -flag corresponding to appending a point t' inside the triangle $\{t, t_2, t_3\}$. By definition $\phi_{T, \mu_K, t}(\tau) = \frac{\text{area}(t, t_2, t_3)}{\text{area}(t_1, t_2, t_3)}$. When t belongs to any of the six remaining regions defined by the lines spanned by $\{t_1, t_2, t_3\}$, a triangle of the form $\{t, t_2, t_3\}$ is divided into two triangles by T , and as before we can determine the relative area of each of these triangles and their sum provides the relative area of $\{t, t_2, t_3\}$. ◀

We next show that measures that induce the same limit give rise to equivalent families of homomorphisms (due to lack of space we defer the proof to the journal version).

► **Lemma 12.** *Let μ and μ' be two measures in \mathbb{R}^2 for which lines are negligible. Let \mathbf{Q} be a set of m random points chosen independently from μ , and \mathbf{Q}' be a set of m random points chosen independently from μ' . If $\ell_\mu = \ell_{\mu'} = \ell$ then for every chirotope σ such that $\ell([\sigma]_\sigma) > 0$, there exist sequences of points P and P' with chirotope σ such that $\phi_{P, \mu, \mathbf{Q}} = \phi_{P', \mu', \mathbf{Q}'}$.*

We now have all the ingredients to prove Theorem 3.

Proof of Theorem 3. We begin by proving the consequence of (i) and (ii). The space (\mathcal{K}, d_{BM}) is a compact Hausdorff space, so (ii) implies that $\mathcal{L}_{\mathcal{K}}$ is compact and (i) implies that the map is a bijection with its image. Any continuous bijection from a Hausdorff space to a compact space is a homeomorphism.

We now prove (ii). Let $d_{TV}(\mu_1, \mu_2) := \sup_A |\mu_1(A) - \mu_2(A)|$, where the supremum is taken among all measurable sets A , denote the total variation distance between two probability measures μ_1 and μ_2 . It is classical that $d_{TV}(\mu_1^k, \mu_2^k) \leq k d_{TV}(\mu_1, \mu_2)$ so in particular $|p(\omega, \mu_K) - p(\omega, \mu_{K'})| \leq |\omega| d_{TV}(\mu_K, \mu_{K'})$. Hence it is enough to show that $d_{TV}(\mu_K, \mu_{gK'}) \leq 2d_{BM}(K, K')$ for some nondegenerate affine transformation g . Without loss of generality we can assume that $K \subset K' \subset rK$ where $r = e^{d_{BM}(K, K')}$. Since $K \subset K'$ the supremum $\sup_A |\mu_K(A) - \mu_{K'}(A)|$ is attained by $A = K$. Indeed, for every measurable set A , the signed measure $\mu_K(A) - \mu_{K'}(A) = \frac{\text{area}(A \cap K)}{\text{area } K} - \frac{\text{area}(A \cap K')}{\text{area } K'}$ does not decrease by substituting A by $A' = A \cap K$, and among subsets of K this signed measure does not decrease by substituting A by a superset. Hence $d_{TV}(\mu_K, \mu_{K'}) = 1 - \frac{\text{area}(K)}{\text{area}(K')} \leq 1 - \frac{\text{area}(K)}{\text{area}(rK)} = 1 - \frac{1}{r^2} \leq 2 \ln r$. The last inequality is true provided $r \leq 1$, which is the case.

Finally we prove item (i). Let K and K' be two convex bodies such that $\ell_K = \ell_{K'}$. By Lemma 12, there exists triangles T and T' such that $\phi_{T, \mu_K, \mathbf{t}} = \phi_{T', \mu_{K'}, \mathbf{t}'}$, where \mathbf{t} and \mathbf{t}' are points chosen uniformly at random from K and K' respectively. Define the signed area of an ordered triangle as its area multiplied by its orientation (i.e. it is positive if the triangle is CCW oriented and negative otherwise) and denote it by area^* . Remark that the relative signs of the triangles depend only on the chirotope σ' of $\{t_1, t_2, t_3, t\}$. By Lemma 11, for every $t \in K$ the homomorphism $\phi_{T, \mu_K, t} \in \cup_{|\sigma'|=4} \text{Hom}^+(\mathcal{A}^{\sigma'}, \mathbb{R})$ is enough to reconstruct the relative area^* with respect to T of each triangle in t_1, t_2, t_3, t . Using barycentric coordinates and T as an affine basis:

$$t = \frac{\text{area}^*(t, t_2, t_3)}{\text{area}^*(t_1, t_2, t_3)} t_1 + \frac{\text{area}^*(t_1, t, t_3)}{\text{area}^*(t_1, t_2, t_3)} t_2 + \frac{\text{area}^*(t_1, t_2, t)}{\text{area}^*(t_1, t_2, t_3)} t_3,$$

we recover t from $\phi_{T, \mu_K, t}$. Writing t in this way for every homomorphism in the support of $\phi_{T, \mu_K, t}$ we reconstruct the convex body K . Analogously, writing t' using T' as an affine basis and $\phi_{T', \mu_{K'}, t'}$ to compute the relative areas for every homomorphism in the support of $\phi_{T', \mu_{K'}, t'}$, we reconstruct K' . Since $\phi_{T', \mu_{K'}, t'}$ and $\phi_{T, \mu_K, t}$ are identical, K' is the image of K under the affine map taking T to T' . \blacktriangleleft

4.2 Proof of Proposition 4

It is perhaps tempting, when searching for a measure representing a given limit ℓ , to take a sequence of random order types \mathbf{r}_n from ℓ , with $\lim_{n \rightarrow \infty} |\omega_n| = \infty$, take for each n a realization P_n of ω_n and expect that the empirical measure $\mu_{P_n} := \frac{1}{|P_n|} \sum_{s \in P_n} \delta_s$ converges to a measure representing ℓ . The next lemma gives necessary and sufficient conditions for this approach to work (due to space constraint we defer the proof to the journal version):

► **Lemma 13.** *Let ℓ be a limit of order types. There exists a measure μ for which lines are negligible and such that $P(\omega, \mu) = \ell(\omega)$ for all $\omega \in \mathcal{O}$ if and only if there exists a sequence of point sets $\{P_n\}_{n \in \mathbb{N}}$ whose order types converge to ℓ and such that for any $\epsilon > 0$ the following two conditions hold:*

- (i) *there exists $R > 0$ such that for n large enough, all but at most a fraction ϵ of P_n lies within distance R from the origin.*
- (ii) *for any line $h \subset \mathbb{R}^2$, there exists $\delta > 0$ such that for n large enough, the fraction of points from P_n within distance δ from h is at most ϵ ,*

The condition of Lemma 13 is both necessary and sufficient, and allows us to prove that ℓ_{\odot} cannot be represented by a compactly supported measure.

Proof of Proposition 4. Let \mathbf{R}_n^t be a point set of size $N = n^2$ sampled according to \odot_t . Order the points of \mathbf{R}_n^t on the boundary of $\partial(\text{conv}(\mathbf{R}_n^t))$ following the counterclockwise orientation. Denote this set by $\text{out}(\mathbf{R}_n^t) := \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_m\}$ and order its complement in some arbitrary fashion and denote it by $\text{in}(\mathbf{R}_n^t) := \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_{N-m}\}$.

For each point $s_i \in \text{out}(\mathbf{R}_n^t)$ consider the total order on $\mathbf{R}_n^t \setminus \{s_i\}$ induced by rotating a semiline about s_i , starting with the semiline at s_{i+1} . This order is called *the local sequence* of s_i . It is well known and not hard to show that it is a chirotope invariant. In this case, the local sequence of $s_i \in \text{conv}(\mathbf{r}_n)$, is $(s_{i+1}, s_{i+2}, \dots, s_j, t_{k_1} t_{k_2}, \dots, t_{k_{\text{in}(\mathbf{r}_n)}} s_{j+1} s_{j+2}, \dots, s_{i-1})$, where the order of the points in $\text{in}(\mathbf{r}_n)$ depends on i , but this will be irrelevant. Denote by $j: \text{out}(\mathbf{R}_n^t) \rightarrow \text{out}(\mathbf{R}_n^t)$ a function that assigns to s_i the last element of $\text{out}(\mathbf{R}_n^t)$ in its local sequence before it reaches the elements of $\text{in}(\mathbf{R}_n^t)$. Since the number of points in $\text{in}(\mathbf{R}_n^t)$ is distributed like a binomial with N trials and probability $\frac{1}{2}$, for each i the triangle $\text{conv}(s_i, s_{j(i)}, s_{j(i)+1})$ contains the points of $\text{in}(\mathbf{R}_n^t)$ with probability at least $1 - \frac{1}{2^{N+2}} f(t)$, where $f(t)$ is a continuous function that approaches 1 as t approaches 0. By the union bound this happens for all i with probability at least $1 - \frac{N}{2^{N+2}} f(t)$. Let $|j(i) - i|$ be the number of vertices on $\text{out}(\mathbf{R}_n^t)$ on a counterclockwise walk on $\partial(\text{conv}(\mathbf{R}_n^t))$. For each i , the random variable $|j(i) - i|$ is distributed like a binomial with N trials and probability $\frac{1}{4}$. Hoeffding inequality implies that there exists an absolute constant $C > 0$ such that,

$$\Pr \left[\left| |j(i) - i| - \frac{N}{4} \right| \geq C \sqrt{N \log N} \right] = O \left(\frac{1}{N^2} \right).$$

By the union bound,

$$\Pr \left[\forall i : \left| |j(i) - i| - \frac{N}{4} \right| > C \sqrt{N \log N} \right] = O \left(\frac{1}{N} \right).$$

We can conclude that with high probability the image of j contains more than $\Omega(\sqrt{\frac{N}{\log N}})$ points and that each triangle of the form $\text{conv}(s_i, s_{j(i)}, s_{j(i)+1})$ contains in (\mathbf{R}_n^t) .

Now assume for contradiction that μ is a compactly supported measure representing ℓ_\odot . Let \mathbf{r}_n be a random order type of size $N = n^2$ chosen according to ℓ_\odot . Let \mathbf{R}_n be a set of N points sampled uniformly and independently from μ . Since μ represents ℓ_\odot the order type of \mathbf{R}_n is distributed like \mathbf{r}_n . Let \mathbf{r}_n^t be the random order type of \mathbf{R}_n^t . Define $\text{out}(\mathbf{R}_n) := \{s'_1, s'_2, \dots, s'_{m'}\}$ and $\text{in}(\mathbf{R}_n) := \{t'_1, t'_2, \dots, t'_{N-m'}\}$, analogously as we did for $\text{out}(\mathbf{R}_n^t)$ and $\text{in}(\mathbf{R}_n^t)$ for \mathbf{R}_n . Since the distributions of order types \mathbf{r}_n^t and \mathbf{r}_n can be made arbitrarily close in total variation distance by making t small enough, we can conclude that with high probability, for each i , $\text{conv}[s'_i, s'_{j(i)}, s'_{j(i)+1}]$ contains in (\mathbf{R}_n) .

On the other hand, if the support of μ has finite perimeter, then the sum of the lengths of the edges of $\text{out}(\mathbf{R}_n)$ is also finite, hence the infimal length among such edges approaches zero as n approaches infinity. Let i_0 be such that the edge $s_{j(i_0)}, s_{j(i_0)+1}$ realizes the infimal length. Let h be the line spanned by s_{i_0} and $s_{j(i_0)}$. Given $\epsilon < \frac{1}{8}$, there exists $\delta(\epsilon) > 0$ such that $\mu(h + B(\delta)) < \epsilon/2$ and hence, by the law of large numbers $\mu_{\mathbf{R}_n}(h + B(\delta)) < \epsilon$ almost surely. But we showed that $\text{conv}[s'_{i_0}, s'_{j(i_0)}, s'_{j(i_0)+1}]$ contains in (\mathbf{R}_n) with high probability, which implies that $\mu_{\mathbf{R}_n}(h + B(\delta)) > \frac{1}{2} - \epsilon$ with high probability, which is a contradiction. ◀

4.3 Proof of Proposition 5

Recall that \diamond_k is the order type of k points in convex position. It is folklore that any set of n points contains at least $\frac{k^{3k/2}}{4k^2} \binom{n}{k}$ subsets of k points in convex position, so for any limit of order types ℓ we must have $\ell(\diamond_k) \geq \frac{k^{3k/2}}{4k^2}$ (due to space constraint we defer the proof to the journal version). We first show that this bound is essentially attained by ℓ_H :

► **Lemma 14.** $\ell_H(\diamond_k) \leq 2^{-\frac{k^2}{2} + k \log k}$.

Proof. Define a k -cup to be a sequence of points lying on the graph of a convex function, and a k -cap to be a sequence of points lying on the graph of a concave function. Let $q_+(k, P_n)$ be the fraction of k -tuples of P_n forming a k -cup, and $q_-(k, P_n)$ be the fraction of k -tuples of P_n forming a k -cap. Since a k -tuple in convex position contains either a $\frac{k}{2}$ -cup or a $\frac{k}{2}$ -cap the union bound gives $p(\diamond_k, \omega_n) \leq q_+(\frac{k}{2}, P_n) + q_-(\frac{k}{2}, P_n)$. By symmetry is enough to bound $q_+(k, P_n)$. Denote by $Q_+(k, P_n)$ the number of k -cups in P_n . Since every k -cup containing points from P_n^0 and P_n^1 contains at most one point from P_n^1 ,

$$Q_+(k, P_{n+1}) \leq Q_+(k-1, P_n^0) |P_n^1| + Q_+(k, P_n^0) + Q_+(k, P_n^1).$$

Note that $Q_+(3, P_n) \leq \binom{2^n}{3}$ and $Q_+(k, P_0) \leq 1$. By induction on $n+k$ we get that $Q_+(k, P_n) \leq 2^{nk - \frac{k^2}{2}}$. With Stirling's formula, we thus have $q_+(k, P_n) \leq 2^{-\frac{k^2}{2} + k \log k}$ for n large enough. ◀

We next bound from below $p(\diamond_k, \mu)$ under some regularity assumptions on μ . These bounds are up to an undetermined constant; the fact that the rate of decay of $p(\diamond_k, \mu)$ is by an order of magnitude slower than that of $\ell_H(\diamond_k)$ is enough, however, to ensure that for any such μ there exists some n such that $p(\diamond_k, \mu) \neq \ell_H(\diamond_k)$, thus proving Proposition 5.

► **Lemma 15.** *Let μ be a measure over \mathbb{R}^2 for which lines are negligible.*

- (i) *If there exists an open set of positive μ -measure on which μ is absolutely continuous to the Lebesgue measure then $p(\diamond_k, \mu) \geq 2^{-2k \log k + O(k)}$.*
- (ii) *If there exists an open set of positive μ -measure on which μ is absolutely continuous to the length measure on a C^2 curve then $p(\diamond_k, \mu) \geq 2^{-O(k)}$.*

The number of different order types in the plane is $2^{4n \log n}$, up to multiplicative factors of order $2^{o(n \log n)}$ [3, Section 4]. Notice that the asymptotic bounds presented on $p(\diamond_k, \mu)$ both for smooth curves and for the Lebesgue measure, imply that there exists a sequence of order types ω_k such that $\frac{\ell_\mu(\omega_k)}{\ell_\mu(\diamond_k)}$ approaches zero as k approaches infinity. On the other hand, the bounds for $\ell_H(\diamond_k)$ imply that there exists an order type such that $\frac{\ell_H(\omega_k)}{\ell_H(\diamond_k)}$ approaches infinity.

References

- 1 B. Abrego, S. Fernandez-Merchant, and G. Salazar. The rectilinear crossing number of k_n : Closing in (or are we?). In *Thirty Essays on Geometric Graph Theory*, pages 5–18. Springer, 2013.
- 2 O. Aichholzer, F. Aurenhammer, and H. Krasser. Enumerating Order Types for Small Point Sets with Applications. In *Proc. 17th Ann. ACM Symp. Computational Geometry*, pages 11–18, Medford, Massachusetts, USA, 2001.
- 3 N. Alon. The number of polytopes, configurations and real matroids. *Mathematika*, 33:62–71, 1986.
- 4 G. Aloupis, J. Iacono, S. Langerman, Ö. Özkan, and S. Wührer. The complexity of order type isomorphism. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA’14*, pages 405–415, 2014.
- 5 B. Borchers. CSDP, A C library for semidefinite programming. *Optimization Methods and Software*, 11(1–4):613–623, 1999.
- 6 P. Brass, W. Moser, and J. Pach. *Research Problems in Discrete Geometry*. Springer, 2005.
- 7 P. Erdős and G. Szekeres. On some extremum problems in elementary geometry. *Eotvos Sect. Math*, 3–4:53–62, 1962.
- 8 J. D. Horton. Sets with no empty convex 7-gons. *Canad. Math. Bull.*, 26:482–484, 1983.
- 9 L. Lovász and B. Szegedy. Limits of dense graph sequences. *J. Combin. Theory Ser. B*, 96(6):933–957, 2006.
- 10 E. Lubetzky and Y. Zhao. On replica symmetry of large deviations in random graphs. *Random Structures & Algorithms*, 2014. doi: 10.1002/rsa.20536.
- 11 A. A. Razborov. Flag algebras. *J. Symbolic Logic*, 72(4):1239–1282, 2007.
- 12 E. Scheinerman and H. Wilf. The rectilinear crossing number of a complete graph and sylvester’s “four point problem” of geometric probability. *Amer. Math. Monthly*, 101:939–943, 1994.
- 13 P. W. Shor. Stretchability of pseudolines is NP-hard. In *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 531–554. Amer. Math. Soc., 1991.
- 14 W. A. Stein et al. *Sage Mathematics Software (Version 6.1)*. The Sage Development Team, 2013. <http://www.sagemath.org>.

Combinatorial Redundancy Detection*

Komei Fukuda¹, Bernd Gärtner², and May Szedlák²

- 1 Department of Mathematics and Department of Computer Science
Institute of Theoretical Computer Science, ETH Zürich
CH-8092 Zürich, Switzerland
komei.fukuda@math.ethz.ch
- 2 Department of Computer Science
Institute of Theoretical Computer Science, ETH Zürich
CH-8092 Zürich, Switzerland
{gaertner,may.szedlak}@inf.ethz.ch

Abstract

The problem of detecting and removing redundant constraints is fundamental in optimization. We focus on the case of linear programs (LPs) in dictionary form, given by n equality constraints in $n + d$ variables, where the variables are constrained to be nonnegative. A variable x_r is called *redundant*, if after removing $x_r \geq 0$ the LP still has the same feasible region. The time needed to solve such an LP is denoted by $LP(n, d)$.

It is easy to see that solving $n + d$ LPs of the above size is sufficient to detect all redundancies. The currently fastest practical method is the one by Clarkson: it solves $n + d$ linear programs, but each of them has at most s variables, where s is the number of nonredundant constraints.

In the first part we show that knowing all of the finitely many dictionaries of the LP is sufficient for the purpose of redundancy detection. A dictionary is a matrix that can be thought of as an enriched encoding of a vertex in the LP. Moreover – and this is the combinatorial aspect – it is enough to know only the signs of the entries, the actual values do not matter. Concretely we show that for any variable x_r one can find a dictionary, such that its sign pattern is either a redundancy or nonredundancy certificate for x_r .

In the second part we show that considering only the sign patterns of the dictionary, there is an output sensitive algorithm of running time $\mathcal{O}(d \cdot (n + d) \cdot s^{d-1} \cdot LP(s, d) + d \cdot s^d \cdot LP(n, d))$ to detect all redundancies. In the case where all constraints are in general position, the running time is $\mathcal{O}(s \cdot LP(n, d) + (n + d) \cdot LP(s, d))$, which is essentially the running time of the Clarkson method. Our algorithm extends naturally to a more general setting of arrangements of oriented topological hyperplane arrangements.

1998 ACM Subject Classification G.2.1 Combinatorics

Keywords and phrases system of linear inequalities, redundancy removal, linear programming, output sensitive algorithm, Clarkson’s method

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.315

1 Introduction

The problem of detecting and removing redundant constraints is fundamental in optimization. Being able to understand redundancies in a model is an important step towards improvements of the model and faster solutions.

* Research supported by the Swiss National Science Foundation (SNF Project 200021_150055 / 1).



© Komei Fukuda, Bernd Gärtner, and May Szedlák;
licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG’15).

Editors: Lars Arge and János Pach; pp. 315–328



Leibniz International Proceedings in Informatics

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

In this paper, we focus on redundancies in systems of linear inequalities. We consider systems of the form

$$\begin{aligned} x_B &= b - Ax_N \\ x_B &\geq 0 \\ x_N &\geq 0 \end{aligned} \tag{1}$$

where B and N are disjoint finite sets of variable indices with $|B| = n$, $|N| = d$, $b \in \mathbb{R}^B$ and $A \in \mathbb{R}^{B \times N}$ are given input vector and matrix. We assume that the system (1) has a feasible solution. Any consistent system of linear equalities and inequalities can be reduced to this form.

A variable x_r is called *redundant* in (1) if $x_B = b - Ax_N$ and $x_i \geq 0$ for $i \in B \cup N \setminus \{r\}$ implies $x_r \geq 0$, i.e., if after removing constraint $x_r \geq 0$ from (1) the resulting system still has the same feasible region. Testing redundancy of x_r can be done by solving the linear program (LP)

$$\begin{aligned} \text{minimize} \quad & x_r \\ \text{subject to} \quad & x_B = b - Ax_N \\ & x_i \geq 0, \quad \forall i \in B \cup N \setminus \{r\}. \end{aligned} \tag{2}$$

Namely, a variable x_r is redundant if and only if the LP has an optimal solution and the optimal value is nonnegative.

Let $LP(n, d)$ denote the time needed to solve an LP of form (2). Throughout the paper, we are working in the real RAM model of computation, where practical algorithms, but no polynomial bounds on $LP(n, d)$ are known. However, our results translate to the standard Turing machine model, where they would involve bounds of the form $LP(n, d, \ell)$, with ℓ being the bit size of the input. In this case, $LP(n, d, \ell)$ can be polynomially bounded. The notation $LP(n, d)$ abstracts from the concrete representation of the LP, and also from the algorithm being used; as a consequence, we can also apply it in the context of LPs given by the signs of their dictionaries.

By solving $n + d$ linear programs, $\mathcal{O}((n + d) \cdot LP(n, d))$ time is enough to detect all redundant variables in the real RAM model, but it is natural to ask whether there is a faster method. The currently fastest practical method is the one by Clarkson with running time $\mathcal{O}((n + d) \cdot LP(s, d) + s \cdot n \cdot d)$ [4]. This method also solves $n + d$ linear programs, but each of them has at most s variables, where s is the number of nonredundant variables. Hence, if $s \ll n$, this *output-sensitive* algorithm is a major improvement.

A related (dual) problem is the one of finding the extreme points among a set P of n points in \mathbb{R}^d . A point $p \in P$ is *extreme* in P , if p is not contained in the convex hull of $P \setminus \{p\}$. It is not hard to see that this problem is a special case of redundancy detection in linear systems.

Specialized (and output-sensitive) algorithms for the extreme points problem exist [14, 6], but they are essentially following the ideas of Clarkson's algorithm [4]. For fixed d , Chan uses elaborate data structures from computational geometry to obtain a slight improvement over Clarkson's method [2].

In this paper, we study the *combinatorial* aspects of redundancy detection in linear systems. The basic questions are: What kind of information about the linear system do we need in order to detect all redundant variables? With this restricted set of information, how fast can we detect all of them? Our motivation is to explore and understand the boundary between geometry and combinatorics with respect to redundancy. For example, Clarkson's method [4] uses *ray shooting*, an intrinsically geometric procedure; similarly, the dual extreme

points algorithms [14, 6] use scalar products. In a purely combinatorial setting, neither ray shooting nor scalar products are well-defined notions, so it is natural to ask whether we can do without them.

We will show that our results solely depend on the finite combinatorial information given by the signed dictionaries, i.e., the size is bounded by a function of d and n only. A dictionary can be thought of as an encoding of the associated arrangements of hyperplanes, the corresponding signed dictionary only contains the signs of the encoding (see Section 2). On the other hand Clarkson's algorithm depends on the input data A and b .

Our approach is very similar to the combinatorial viewpoint of linear programming pioneered by Matoušek, Sharir and Welzl [13] in form of the concept of *LP-type problems*. The question they ask is: how quickly can we *optimize*, given only combinatorial information? As we consider redundancy detection and removal as important towards efficient optimization, it is very natural to extend the combinatorial viewpoint to also include the question of redundancy. The results that we obtain are first steps and leave ample space for improvement. An immediate theoretical benefit is that we can handle redundancy detection in structures that are more general than systems of linear inequalities; most notably, our results naturally extend to the realm of *oriented matroids* [1].

Statement of Results

The first point that we will make is that for the purpose of redundancy testing, it is sufficient to know all the finitely many dictionaries associated with the system of inequalities (1). Moreover, we show that it is sufficient to know only the *signed* dictionaries, i.e., the *signs* of the dictionary entries. Their actual numerical values do not matter.

In Theorem 2, we give a characterization of such a redundancy certificate. More precisely, we show that, for every redundant variable x_r there exists at least one signed dictionary such that its sign pattern is a redundancy certificate of x_r . Similarly, as shown in Theorem 4, for every nonredundant variable there exists a nonredundancy certificate. Such a single certificate can be detected in time $LP(n, d)$ (see Section 4.3). The number of dictionaries needed to detect *all* redundancies depends on the LP and can vary between constant and linear in $n + d$ [10, Appendix].

In a second part, we present a Clarkson-type, output-sensitive algorithm that detects all redundancies in running time $\mathcal{O}(d \cdot (n + d) \cdot s^{d-1} LP(s, d) + d \cdot s^d \cdot LP(n, d))$ (Theorem 5). Under some general position assumptions the running time can be improved to $\mathcal{O}((n + d) \cdot LP(s, d) + s \cdot LP(n, d))$, which is basically the running time of Clarkson's algorithm. In these bounds, $LP(n, d)$ denotes the time to solve an LP to which we have access only through signed dictionaries. As in the real RAM model, no polynomial bounds are known, but algorithms that are fast in practice exist.

In general our algorithm's running time is worse than Clarkson's, but it only requires the combinatorial information of the system and not its actual numerical values. If the feasible region is not full dimensional (i.e. not of dimension d), then a redundant constraint may become nonredundant after the removal of some other redundant constraints. To avoid these dependencies of the redundant constraints we assume full dimensionality of the feasible region. Because of our purely combinatorial characterizations of redundancy and nonredundancy, our algorithm works in the combinatorial setting of oriented matroids [1], and can be applied to remove redundancies from oriented topological hyperplane arrangements.

2 Basics

Before discussing redundancy removal and combinatorial aspects in linear programs, we fix the basic notation on linear programming –such as dictionaries and pivots operations – and review finite pivot algorithms. (For further details and proofs see e.g. [3, Part 1], [7, Chapter 4].)

2.1 LP in Dictionary Form

Throughout, if not stated otherwise, we always consider *linear programs* (LPs) of the form

$$\begin{aligned} &\text{minimize} && c^T x_N \\ &\text{subject to} && x_B = b - Ax_N \\ &&& x_E \geq 0, \end{aligned} \tag{3}$$

where $E := B \cup N$ and as introduced in (1), B and N are disjoint finite sets of variable indices with $|B| = n$, $|N| = d$, $b \in \mathbb{R}^B$ and $A \in \mathbb{R}^{B \times N}$ are given input vector and matrix. An LP of this form is called LP in *dictionary form* and its *size* is $n \times d$. The set B is called a (initial) *basis*, N a (initial) *nonbasis* and $c^T x_N$ the *objective function*.

The *feasible region* of the LP is defined as the set of $x \in \mathbb{R}^E$ that satisfy all constraints, i.e., the set $\{x \in \mathbb{R}^E | x_B = b - Ax_N, x_E \geq 0\}$. A feasible solution \bar{x} is called *optimal* if for every feasible solution x , $c^T \bar{x} \leq c^T x$. The LP is called *unbounded* if for every $k \in \mathbb{R}$, there exists a feasible solution x , such that $c^T x \leq k$. If there exists no feasible solution, the LP is called *infeasible*.

The *dictionary* $D(B) \in \mathbb{R}^{B \cup \{f\} \times N \cup \{g\}}$ of an LP (3) w.r.t. a basis B is defined as

$$D := D(B) = \begin{bmatrix} 0 & c^T \\ b & -A \end{bmatrix},$$

where f is the index of the first row and g is the index of the first column. For each $i \in B \cup \{f\}$ and $j \in N \cup \{g\}$, we denote by d_{ij} its (i, j) entry, by D_i the row indexed by i , and by D_j the column indexed by j .

Hence by setting $x_f := c^T x_N$, we can rewrite (3) as

$$\begin{aligned} &\text{minimize} && x_f \\ &\text{subject to} && x_{B \cup \{f\}} = Dx_{N \cup \{g\}} \\ &&& x_E \geq 0, \\ &&& x_g = 1. \end{aligned} \tag{4}$$

Whenever we do not care about the objective function, we may set $c = 0$, and with abuse of notation, set $D = [b, -A]$.

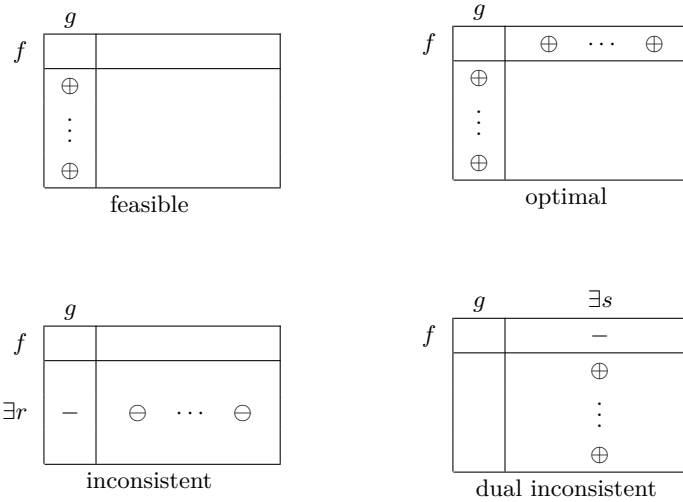
The *basic solution* w.r.t. B is the unique solution \bar{x} to $x_{B \cup \{f\}} = Dx_{N \cup \{g\}}$ such that $\bar{x}_g = 1$, $\bar{x}_N = 0$ and hence $\bar{x}_{B \cup \{f\}} = D.g$.

The *dual LP* of LP (4) is defined as

$$\begin{aligned} &\text{minimize} && y_g \\ &\text{subject to} && y_{N \cup \{g\}} = -D^T y_{B \cup \{f\}} \\ &&& y_E \geq 0, \\ &&& y_f = 1. \end{aligned} \tag{5}$$

It is useful to define the following four different types of dictionaries (and bases) as shown in the figure below, where "+" denotes positivity, " \oplus " nonnegativity and similarly "-" negativity and " \ominus " nonpositivity.

A dictionary D (or the associated basis B) is called *feasible* if $d_{ig} \geq 0$ for all $i \in B$. A dictionary D (or the associated basis B) is called *optimal* if $d_{ig} \geq 0$, $d_{fj} \geq 0$ for all $i \in B, j \in N$. A dictionary D (or the associated basis B) is called *inconsistent* if there exists $r \in B$ such that $d_{rg} < 0$ and $d_{rj} \leq 0$ for all $j \in N$. A dictionary D (or the associated basis B) is called *dual inconsistent* if there exists $s \in N$ such that $d_{fs} < 0$ and $d_{is} \geq 0$ for all $i \in B$.



The following proposition follows from standard calculations.

► **Proposition 1.** *For any LP in dictionary form the following statements hold.*

1. *If the dictionary is feasible then the associated basic solution is feasible.*
2. *If the dictionary is optimal, then the associated basic solution is optimal.*
3. *If the dictionary is inconsistent, then the LP is infeasible.*
4. *If the dictionary is dual inconsistent, then the dual LP is infeasible. If in addition the LP is feasible, then the LP is unbounded.*

2.2 Pivot Operations

We now show how to transform the dictionary of an LP into a modified dictionary using elementary matrix operation, preserving the equivalence of the associated linear system. This operation is called a *pivot operation*.

Let $r \in B, s \in N$ and $d_{rs} \neq 0$. Then it is easy to see that one can transform $x_{B \cup \{f\}} = Dx_{N \cup \{g\}}$ to an equivalent system (i.e., with the same solution set) :

$$x_{B' \cup \{f\}} = D'x_{N' \cup \{g\}},$$

where $B' = B \setminus \{r\} \cup \{s\}$ ($N' = N \setminus \{s\} \cup \{r\}$, respectively) is a new (non)basis and

$$d'_{ij} = \begin{cases} \frac{1}{d_{rs}} & \text{if } i = s \text{ and } j = r \\ -\frac{d_{rj}}{d_{rs}} & \text{if } i = s \text{ and } j \neq r \\ \frac{d_{is}}{d_{rs}} & \text{if } i \neq s \text{ and } j = r \\ d_{ij} - \frac{d_{is} \cdot d_{rj}}{d_{rs}} & \text{if } i \neq s \text{ and } j \neq r \end{cases} \quad (i \in B' \cup \{f\} \text{ and } j \in N' \cup \{g\}). \tag{6}$$

We call a dictionary *terminal* if it is optimal, inconsistent or dual inconsistent. There are several finite pivot algorithms such as the simplex and the criss-cross method that transform any dictionary into one of the terminal dictionaries [16, 17, 11],[5, Section 4]. This will be discussed further in Section 4.3.

3 Combinatorial Redundancy

Consider an LP in dictionary form as given in (3). Then $x_r \geq 0$ is *redundant*, if the removal of the constraint does not change the feasible solution set, i.e., if

$$\begin{aligned} & \text{minimize} && c^T x_N \\ & \text{subject to} && x_B = b - Ax_N \\ & && x_i \geq 0, \quad \forall i \in E \setminus \{r\}, \end{aligned} \tag{7}$$

has the same feasible solution set as (3). Then the variable x_r and the index r are called *redundant*.

If the constraint $x_r \geq 0$ is not redundant it is called *nonredundant*, in that case the variable x_r and the index r are called *nonredundant*.

It is not hard to see that solving $n + d$ LPs of the same size as (7) suffices to find all redundancies. Hence running time $\mathcal{O}((n + d) \cdot LP(n, d))$ suffices to find all redundancies, where $LP(n, d)$ is the time needed to solve an LP of size $n \times d$. Clarkson showed that it is possible to find *all* redundancies in time $\mathcal{O}((n + d) \cdot LP(s, d) + s \cdot n \cdot d)$, where s is the number of nonredundant variables [4]. In case where $s \ll n$ this is a major improvement. To be able to execute Clarkson's algorithm, one needs to assume full dimensionality and an interior point of the feasible solution set. In the LP setting this can be done by some preprocessing, including solving a few ($\mathcal{O}(d)$) LPs [9, Section 8].

In the following we focus on the combinatorial aspect of redundancy removal. We give a combinatorial way, the *dictionary oracle*, to encode LPs in dictionary form, where we are basically only given the signs of the entries of the dictionaries. In Section 4 we will show how the signs suffice to find all redundant and nonredundant constraints of an LP in dictionary form.

Consider an LP of form (3). For any given basis B , the *dictionary oracle* returns a matrix

$$D^\sigma = D^\sigma(B) \in \{+, -, 0\}^{B \times N \cup \{g\}}, \text{ with } d_{ij}^\sigma = \text{sign}(d_{ij}), \forall i \in B, j \in N \cup \{g\}.$$

Namely, for basis B , the oracle simply returns the matrix containing the signs of $D(B)$, without the entries of the objective row f .

4 Certificates

We show that the dictionary oracle is enough to detect all redundancies and nonredundancies of the variables in E . More precisely for every $r \in E$, there exists a basis B such that $D^\sigma(B)$ is either a redundancy or nonredundancy certificate for x_r . We give a full characterization of the certificates in Theorems 2 and 4. The number of dictionaries needed to have *all* certificates depend on the LP. See [10, Appendix] for examples where constantly many suffice and where linearly many are needed.

For convenience throughout we make the following assumptions, which can be satisfied with simple preprocessing.

1. The feasible region of (3) is full dimensional (and hence nonempty).
2. There is no $j \in N$ such that $d_{ij} = 0$ for all $i \in B$.

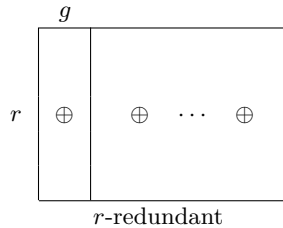
In Section 4.3 we will see that both the criss-cross and the simplex method can be used on the dictionary oracle for certain objective functions. Testing whether the feasible solution set is empty can hence be done by solving one linear program in the oracle setting. As mentioned in the introduction the full-dimensionality assumption is made to avoid dependencies between

the redundant constraints. This can be achieved by some preprocessing on the LP, including solving a few $O(d)$ LPs [9].

It is easy to see that if there exists a column j such that $d_{ij} = 0$ for all $i \in B$, then x_j is nonredundant and we can simply remove the column.

4.1 A Certificate for Redundancy in the Dictionary Oracle

We say a that basis B is r -redundant if $r \in B$ and $D_r^\sigma \geq 0$ i.e. if $D^\sigma(B)$ is as given in the figure below.



Since the r -th row of the dictionary represents $x_r = d_{rg} + \sum_{j \in N} d_{rj}x_j$, $x_r \geq 0$ is satisfied as long as $x_j \geq 0$ for all $j \in N$. Hence $x_r \geq 0$ is redundant for (3).

► **Theorem 2 (Redundancy Certificate).** *An inequality $x_r \geq 0$ is redundant for the system (3) if and only if there exists an r -redundant basis.*

Proof. We only have to show the “only if” part.

Suppose $x_r \geq 0$ is redundant for the system (3). We will show that there exists an r -redundant basis.

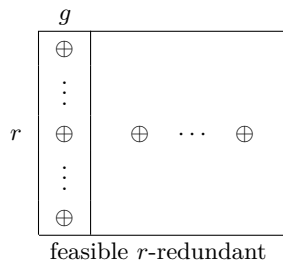
Consider the LP minimizing the variable x_r subject to the system (3) without the constraint $x_r \geq 0$. Since $x_r \geq 0$ is redundant for the system (3), the LP is bounded. By assumption 1 and the fact that every finite pivot algorithm terminates in a terminal dictionary the LP has an optimal dictionary.

If the initial basis contains r , then we can consider the row associated with r as the objective row. Apply any finite pivot algorithm to the LP. Otherwise, r is nonbasic. By assumption 2, one can pivot on the r -th column to make r a basic index. This reduces the case to the first case.

Let’s consider an optimal basis and optimal dictionary for the LP where x_r is the objective function. Since it is optimal, all entries d_{rj} for $j \in N$ are nonnegative. Furthermore, d_{rg} is nonnegative as otherwise we would have found a solution that satisfies all constraints except $x_r \geq 0$, implying nonredundancy of x_r . ◀

From the proof of Theorem 2 the following strengthening of Theorem 2 immediately follows.

► **Corollary 3.** *An inequality $x_r \geq 0$ is redundant for the system (3) if and only if there exists a feasible r -redundant basis.*



4.2 A Certificate for Nonredundancy in the Dictionary Oracle

Similarly as in the redundancy case, we introduce a certificate for nonredundancy using the dictionary oracle. A basis B is called r -nonredundant if B is feasible, $r \in N$ and $d_{tg} = 0$ implies $d_{tr} \leq 0$ for all $t \in B$ i.e. $D^\sigma(B)$ is of the following form.

g	r
+	
⋮	
+	
0	⊖
⋮	⋮
0	⊖

r -nonredundant

► **Theorem 4 (Nonredundancy Certificate).** *An inequality $x_r \geq 0$ is nonredundant for the system (3) if and only if there exists an r -nonredundant basis.*

Before proving the theorem, we observe the following.

1. Unlike in the redundancy certificate an r -nonredundant basis needs to be feasible. To verify the correctness of a nonredundancy certificate we need to check between n and $2n$ entries, which is typically much larger than the $d + 1$ entries we need for the redundant case.
2. If the g -column of a feasible basis does not contain any zeros, then all nonbasic variables are nonredundant. In general when $x_r \geq 0$ is nonredundant, not necessarily every feasible basis B with $r \in N$ is r -nonredundant. Consider the system:

$$\begin{aligned} x_3 &= x_1 + x_2 \\ x_1, x_2, x_3 &\geq 0. \end{aligned}$$

Then the basis $\{3\}$ is not a certificate of nonredundancy of x_1 , as $d_{31}^g = +$ in the associated dictionary. On the other hand, the basis $\{2\}$ is 1-nonredundant:

	g	1	2		g	1	3	
3	0	+	+		2	0	-	+

Proof of Theorem 4. Let (LP) be of form (3) and suppose that $x_r \geq 0$ is nonredundant. Then it follows that for ϵ small enough $x_r \geq -\epsilon$ is nonredundant in

$$\begin{aligned} &\text{minimize} && x_r \\ &\text{subject to} && x_B = b - Ax_N \\ & && x_i \geq 0, \quad \forall i \in B \cup N \setminus \{r\} \\ & && x_r \geq -\epsilon. \end{aligned} \tag{8}$$

Note that this LP can easily be transformed to an LP of form (3) by the straight forward variable substitution $x'_r = x_r + \epsilon$.

LP (8) attains its minimum at $-\epsilon$ and hence there exists an optimal dictionary where r is nonbasic. Let B be such a feasible optimal basis of (LP^ϵ) with $r \in N$. We show that if we choose ϵ small enough, B is r -nonredundant in (LP) .

Let B_1, B_2, \dots, B_m be the set of all bases (feasible and infeasible) of (LP) , that have r as a nonbasic variable. Choose $\epsilon > 0$ such that

$$\epsilon < \min \left\{ \frac{d_{tg}}{d_{tr}} \mid t \in B_i : d_{tg}, d_{tr} < 0; i = 1, 2, \dots, m \right\}.$$

If the right hand side (RHS) is undefined, we choose any $\epsilon < \infty$.

Geometrically this means that if for $t \in B_i$ $x_t \geq 0$ is violated in the basic solution w.r.t. B_i in (LP) , then it is still violated in the corresponding basic solution (LP^ϵ) . Let D and D^ϵ be the dictionaries w.r.t. B in (LP) and (LP^ϵ) respectively.

D and D^ϵ only differ in their entries of column g , where

$$d_{tg}^\epsilon = d_{tg} - \epsilon \cdot d_{tr}, \forall t \in B. \tag{9}$$

We need to show that B is r -nonredundant in (LP) . To show that B is a feasible basis we need that $d_{tg} \geq 0$ for all $t \in B$. If $d_{tr} \geq 0$, then this is clear. In the case where $d_{tr} < 0$ it follows that $\epsilon \geq \frac{d_{tg}}{d_{tr}}$ and hence $d_{tg} \geq 0$ by choice of ϵ . Hence B is feasible and if $d_{tg} = 0$, then by equation (9) it follows that $d_{tr} \leq 0$. Therefore B is r -nonredundant.

For the other direction let B be r -nonredundant and D and D^ϵ the corresponding dictionaries in (LP) and (LP^ϵ) , respectively. Choose $\epsilon > 0$ such that

$$\epsilon \leq \min \left\{ \frac{d_{tg}}{d_{tr}} \mid t \in B : d_{tg}, d_{tr} > 0 \right\}.$$

If the RHS is undefined, we choose any $\epsilon < \infty$.

We claim that for such an ϵ , B is still feasible for (LP^ϵ) and hence $x_r \geq 0$ is nonredundant. Again the two dictionaries only differ in row g , where

$$d_{tg}^\epsilon = d_{tg} - \epsilon \cdot d_{tr}, \forall t \in B.$$

In the case where $d_{tg} = 0$, it follows that $d_{tg}^\epsilon \geq 0$ by r -nonredundancy. If $d_{tg} > 0$, then

$$d_{tg}^\epsilon = d_{tg} - \epsilon \cdot d_{tr} \geq d_{tg} - \min \left\{ \frac{d_{t'g}}{d_{t'r}} \mid t' \in B : d_{t'g}, d_{t'r} > 0 \right\} \frac{d_{t'g}}{d_{t'r}} \cdot d_{tr} \geq 0.$$



4.3 Finite Pivot Algorithms for Certificates

In this section we discuss how to design finite pivot algorithms for the dictionary oracle model. Both the criss-cross method and the simplex method can be used for the dictionary oracle to find redundancy and nonredundancy certificates. A finite pivot algorithm chooses in every step a pivot according to some given rule and terminates in an optimal, inconsistent or dual inconsistent basis in a finite number of steps. Note that both the criss-cross method and the simplex method may not be polynomial in the worst case, but are known to be fast in practice [12, 15]. Furthermore there exists no known polynomial algorithm to solve an LP given by the dictionary oracle. Fukuda conjectured that the randomized criss-cross method is an expected polynomial time algorithm [8].

By the proof of Theorem 2, in order to find a redundancy certificate in (3) it is enough to solve (3) with objective function x_r . Similarly by the proof of Theorem 4, for a nonredundancy certificate it is enough to solve the ϵ -perturbed version (8).

For the criss-cross method, the pivot rule is solely dependent on the signs of the dictionary entries and not its actual values [7, Chapter 4], [11]. Standard calculations show that the signs in the ϵ -perturbed dictionary (for $\epsilon > 0$ small enough) are completely determined by the signs of the original dictionary. We recall that the dictionary oracle does not output the objective row, but since we minimize in direction of x_r the signs of the objective row are completely determined. (If r is basic then the objective row has the same entries as the r -th row and if r nonbasic then $d_{fr} = +$ and all other entries of the objective row are

zero.) Therefore the dictionary oracle is enough to decide on the pivot steps of the criss-cross method.

For the simplex method with the smallest index rule, we are given a feasible basis and the nonbasic variable of the pivot element is chosen by its sign only [3, Part 1 Section 3]. The basic variable of the pivot is chosen as the smallest index such that feasibility is preserved after a pivot step. Using the dictionary oracle one can test the at most n possibilities and choose the appropriate pivot.

5 An Output Sensitive Redundancy Detection Algorithm

Throughout this section, we denote by S' the set of nonredundant indices and by R' the set of redundant indices. Denote by $LP(n, d)$ the time needed to solve an LP. By the discussion in Section 4.3, for any x_r , $r \in E$, we can find a certificate in time $LP(n, d)$. Theorem 5 presents a Clarkson type, output sensitive algorithm with running time $\mathcal{O}(d \cdot (n + d) \cdot s^{d-1} \cdot LP(s, d) + d \cdot s^d \cdot LP(n, d))$, that for a given LP outputs the set S' , where $s = |S'|$. Typically s and d are much smaller than n .

5.1 General Redundancy Detection

```

Redundancy Detection Algorithm( $D, g, f$ );
begin
   $R := \emptyset, S := \emptyset$ ;
  while  $R \cup S \neq E$  do
    Pick any  $r \notin R \cup S$  and test if  $r$  is redundant w.r.t.  $S$ ;
    if  $r$  redundant w.r.t.  $S$  then
       $R = R \cup \{r\}$ ;
    else /*  $r$  nonredundant w.r.t.  $S$  */
      test if  $r$  is redundant w.r.t.  $E \setminus R$ ;
      if  $r$  is nonredundant w.r.t.  $E \setminus R$  then
         $S = S \cup \{r\}$ ;
      else /*  $r$  redundant w.r.t.  $E \setminus R$  */
        Find some sets  $S^F \subseteq S'$  and  $R^F \subseteq R'$  such that  $S^F \not\subseteq S$ ;
         $R = R \cup R^F, S = S \cup S^F$ ;
      endif;
    endif;
  endwhile;
   $S^* := S$ ;
  output  $S^*$ ;
end.

```

Since in every round at least one variable is added to S or R , the algorithm terminates. The correctness of the output can easily be verified: If in the outer loop r is added to R , r is redundant w.r.t. S and hence redundant w.r.t. $S^* \supseteq S$. If in the inner loop r is added to S , r is nonredundant w.r.t. $E \setminus R$ and hence nonredundant w.r.t. $S^* \subseteq E \setminus R$.

The main issue is how to find the sets S^F and R^F efficiently in the last step. This will be discussed in (the proof of) Lemma 6.

A technical problem is that we cannot test for redundancy in the dictionary oracle when S does not contain a nonbasis. Therefore as long as this is the case, we fix an arbitrary

nonbasis N and execute the redundancy detection algorithm on $S \cup N$ instead of S . Since this does not change correctness or the order of the running time, we will omit this detail in the further discussion.

► **Theorem 5.** *The redundancy detection algorithm outputs S' , the set of nonredundant constraints in time*

$$R(n, d, s) = \mathcal{O} \left(\sum_{i=0}^{d-1} ((n+d) \cdot s^i \cdot LP(s, d-i) + s^{i+1} \cdot LP(n, d-i)) \right)$$

and consequently in time

$$R(n, d, s) = \mathcal{O} (d \cdot (n+d) \cdot s^{d-1} \cdot LP(s, d) + d \cdot s^d \cdot LP(n, d)).$$

The following Lemma implies Theorem 5.

► **Lemma 6.** *Let $R(n, d, s)$ be the running time of the redundancy detection algorithm in n basic variables, d nonbasic variables and s the number of nonredundant variables. Then in the last step of the inner loop some sets $S^F \subseteq S'$ and $R^F \subseteq R'$, with $S^F \not\subseteq S$, can be found in time $\mathcal{O}(R(n, d-1, s) + LP(n, d))$.*

Proof of Theorem 5. Termination and correctness of the algorithm are discussed above. The iteration of the outer loop of the algorithm takes time $\mathcal{O}(LP(s, d))$ and is executed at most $n+d$ times. By Lemma 6, the running time of the inner loop is $\mathcal{O}(R(n, d-1, s) + LP(n, d))$ and since in each round at least one variable is added to S , it is executed at most s times. Therefore the total running time is given recursively by

$$R(n, d, s) = \mathcal{O} ((n+d) \cdot LP(s, d) + s \cdot (R(n, d-1, s) + LP(n, d))).$$

The claim follows by solving the recursion and noting that $R(n, 0, s)$ can be set to $\mathcal{O}(n)$. ◀

It remains to prove Lemma 6, for which we first prove some basic results below, using the dictionary oracle setting.

► **Lemma 7.** *Let $D = D(B)$ be a feasible dictionary of an LP of form (3) and assume $F := \{i \in B \mid b_i = 0\} \neq \emptyset$. We consider the subproblem of the LP denoted LP^F (with dictionary D^F .) that only contains the rows of D indexed by F . Then $r \in F \cup N$ is nonredundant in LP if and only if it is nonredundant in LP^F .*

Proof. We only need to show the "if" part. Let $r \in F \cup N$ be nonredundant in LP^F with certificate \bar{D}^F . Then there exists a sequence of pivot steps from D^F to \bar{D}^F . Using the same ones on D and obtaining dictionary \bar{D} , this is a nonredundancy certificate for r , since $\bar{d}_{ig} = d_{ig} > 0$ for all $i \in B \setminus F$ by the definition of F . ◀

► **Lemma 8.** *Let $D = [b, -A]$ be the dictionary of an LP of form (3). Then a variable $r \in E$ is nonredundant in the LP given by D if and only if it is nonredundant in the LP given by $D^0 = [0, b, -A]$.*

Proof. If $D(B)$ is a redundancy certificate for r for some basis B , then $D^0(B)$ is a redundancy certificate for r as well.

For the converse, let $D = D(B)$ be a nonredundancy certificate for r for some basis B . For simplicity assume that $B = \{1, 2, \dots, n\}$. For now assume that $b_i > 0$ for all $i \in B$ and let D^i the dictionary obtained from D^0 by pivoting on b_i , $i = 1, 2, \dots, n$. We will show that at least one of the D^i , $i \in \{0, 1, \dots, n\}$ is a nonredundancy certificate for r . Since after any pivot the first column of D^i stays zero, D^i is a nonredundancy certificate if and only if $D^i_{\cdot r} \leq 0$. Let $R^i = (r_1^i, r_2^i, \dots, r_n^i)^T := D^i_{\cdot r}$ for $i \geq 1$ and $R = (r_1, r_2, \dots, r_n)^T := D^0_{\cdot r}$.

Claim: Assume that $r_i^i < 0$ for any fixed i and there are at least $i - 1$ additional nonpositive entries (w.l.o.g. we assume them to be $r_1^i, r_2^i, \dots, r_{i-1}^i$). If R^i has a positive entry (which w.l.o.g. we assume to be r_{i+1}^i), then $r_{i+1}^{i+1} < 0$ and $r_1^{i+1}, r_2^{i+1}, \dots, r_i^{i+1}$ are nonpositive.

If D^0 is not a certificate for r , then w.l.o.g. $r_1 > 0$ and hence $r_1^1 = -\frac{r_1}{b_1} < 0$. Therefore by induction the lemma follows from the claim.

Assume that $r_1^i, r_2^i, \dots, r_{i-1}^i \leq 0$, $r_i^i < 0$ and $r_{i+1}^i > 0$. Then we have $r_i > 0$ and

$$r_{i+1}^i = r_{i+1} - \frac{r_i b_{i+1}}{b_i} > 0 \Leftrightarrow r_i b_{i+1} < r_{i+1} b_i \Rightarrow r_{i+1} > 0, \quad (10)$$

$$\forall j < i : r_j^i = r_j - \frac{r_i b_j}{b_i} \leq 0 \Leftrightarrow r_j b_i \leq r_i b_j. \quad (11)$$

The following calculations show the claim.

$$r_{i+1}^{i+1} = -\frac{r_{i+1}}{b_{i+1}} < 0 \Leftrightarrow r_{i+1} > 0 \text{ which holds by (10).}$$

$$r_i^{i+1} = r_i - \frac{r_{i+1} b_i}{b_{i+1}} \leq 0 \Leftrightarrow r_i b_{i+1} \leq r_{i+1} b_i \text{ which holds by (10).}$$

$$\forall j < i : r_j^{i+1} = r_j - \frac{r_{i+1} b_j}{b_{j+1}} \leq 0 \Leftrightarrow r_j b_{i+1} \leq r_{i+1} b_j,$$

$$\text{and by (10) and (11), } r_j b_{i+1} = (r_j b_i)(r_i b_{i+1}) \cdot \frac{1}{r_i b_i} \leq r_{i+1} b_j.$$

Now suppose that $b_i = 0$ for some i . Then by the nonredundancy certificate $r_i \leq 0$, and it is easy to see that $r_i^j = r_i \leq 0$ for all admissible pivots on b_j . Hence we can use the above construction on the nonzero entries of b . ◀

Proof of Lemma 6. Suppose that during the execution of the algorithm, r is nonredundant w.r.t. the current set S , and redundant w.r.t. $E \setminus R$, with *feasible* redundancy certificate $D = [b, -A]$, which exists by Corollary 3. If $b > 0$, then all nonbasic indices in N are nonredundant by Theorem 4. Choose $S^F = N$, $R^F = \emptyset$. It holds that $S^F \not\subseteq S$, since otherwise r would be redundant w.r.t. S . The running time of the inner loop in this case is $LP(n, d)$.

Now if there exists $i \in B$ such that $b_i = 0$, define $F = \{i \in B | b_i = 0\}$, LP^F and D^F as in Lemma 7. We now recursively find all redundant and nonredundant constraints in the LP^F using Lemma 8 as follows. From LP^F we construct another LP, denoted LP^- with one less nonbasic variable, by deleting D_g^F (the column of all zeros), choosing any element $t \in N$ and setting $t = g$. Finding all redundancies and nonredundancies in LP^- takes time $R(|F|, d - 1, s)$. By Lemma 8 redundancies and nonredundancies are preserved for LP^F .

Therefore finding them in LP^F takes time $R(|F|, d - 1, s) + LP(n, d) \leq R(n, d - 1, s) + LP(n, d)$, where the $LP(n, d)$ term is needed to check separately whether t is redundant. Choose S^F as the set of nonredundant indices of LP^F and R^F as the set of redundant ones. By Lemma 7 $S^F \subseteq S'$ and $R^F \subseteq R'$. Since by Lemma 7 r is redundant in LP^F , $S^F \not\subseteq S$, since otherwise r would be redundant w.r.t. S . ◀

5.2 Strong Redundancy Detection

In this section we show how under certain assumptions the running time of the redundancy algorithm can be improved. If we allow the output to also contain some *weakly redundant* constraints (see definition below), it is basically the same as the running time of Clarkson's method.

A redundant variable r is called *strongly redundant* if for any basic feasible solution \bar{x} , $\bar{x}_r > 0$. In particular for any basic feasible solution, $r \in B$. If r is redundant but not strongly redundant r is called *weakly redundant*.

As before let s be the number of nonredundant constraints and let R_s , (with $|R_s| = r_s$) and R_w , (with $|R_w| = r_w$), be the set of strongly and weakly redundant constraints respectively.

► **Theorem 9.** *It is possible to find a set $S^* \supseteq S'$, $S^* \cap R_s = \emptyset$ in time $\mathcal{O}((n+d) \cdot LP(s+r_w, d) + (s+r_w) \cdot LP(n, d))$.*

The following corollary follows immediately.

► **Corollary 10.** *If there are no weakly redundant constraints, the set S' of nonredundant constraints can be found in time $\mathcal{O}((n+d) \cdot LP(s, d) + s \cdot LP(n, d))$.*

The theorem is proven using the following two lemmas, which can be verified with straight forward variable substitutions.

► **Lemma 11.** *[3, Part 1 Section 3] Let (LP) of form (3), where (LP) is not necessarily full dimensional. W.l.o.g. $B = \{1, 2, \dots, n\}$. For each $i \in \{1, 2, \dots, n\}$ replace the nonnegativity constraint $x_i \geq 0$ by $x_i \geq -\epsilon^i$, for $\epsilon > 0$ sufficiently small. Denote the resulting LP by (LP^ϵ) . Let D^σ be the output of the dictionary oracle for an arbitrary dictionary D of (LP) . Then (LP^ϵ) is full dimensional. Furthermore in $D^{\sigma, \epsilon}$, the corresponding output for the ϵ -perturbed version, all signs can be determined by D^σ , and $D_{.g}^{\sigma, \epsilon}$ has no zero entries.*

► **Lemma 12.** *[3, Part 1 Section 3] Let (LP) and (LP^ϵ) be as in Lemma 11. Then any nonredundant constraint in (LP) is nonredundant in (LP^ϵ) and any strongly redundant constraint in (LP) is strongly redundant in (LP^ϵ) .*

Proof of Theorem 9. Replace the given LP by its ϵ -perturbed version as in Lemma 11 and run the redundancy removal algorithm, which is possible by the same lemma. By Lemma 12, $S^* \supseteq S'$ and $S^* \cap R_s = \emptyset$. Since by Lemma 11, the entries of the g -column of any dictionary $D^{\sigma, \epsilon}$ are strictly positive the algorithm never runs the recursive step and the running time follows. ◀

► **Remark.** The ϵ -perturbation makes every feasible LP full dimensional, therefore the full dimensionality assumption can be dropped for Theorem 9.

5.3 Discussion

In this paper, we presented new combinatorial characterizations of redundancy and nonredundancy in linear inequality systems. We also presented a combinatorial algorithm for redundancy removal.

In contrast to the Clarkson algorithm our redundancy detection algorithm does not need the whole LP but only the combinatorial information of the dictionaries. Although in general the running time is worse, assuming that we have no weak redundancies, our redundancy removal algorithm basically has the same running time as the Clarkson algorithm. Still, a natural goal is to improve the runtime of our algorithm in the general case and get it closer to that of Clarkson's method. We do have a first output-sensitive algorithm for combinatorial redundancy detection, but the exponential dependence on the dimension d is prohibitive already for moderate d .

Our algorithm works in a more general setting of oriented matroids. This means one can remove redundancies from oriented pseudo hyperplane arrangements efficiently. Furthermore,

the algorithm can be run in parallel. Yet, analyzing the performance may not be easy because checking redundancy of two distinct variables simultaneously may lead to the discovery of the same (non)redundant constraint. This is an interesting subject of future research.

References

- 1 A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. Ziegler. *Oriented Matroids*. Cambridge University Press, 1993.
- 2 T. M. Chan. Output-sensitive results on convex hulls, extreme points, and related problems. *Discrete & Computational Geometry*, 16(4):369–387, 1996.
- 3 V. Chvatal. *Linear Programming*. W. H. Freeman, 1983.
- 4 K. L. Clarkson. More output-sensitive geometric algorithms. In *Proc. 35th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 695–702, 1994.
- 5 G. B. Dantzig. *Linear Programming and Extensions*. Princeton University Press, Princeton, NJ, 1963.
- 6 J. H. Dulá, R. V. Helgason, and N. Venugopal. An algorithm for identifying the frame of a pointed finite conical hull. *INFORMS J. Comput.*, 10(3):323–330, 1998.
- 7 K. Fukuda. Introduction to optimization. http://www.ifor.math.ethz.ch/teaching/Courses/Fall_2011/intro_fall_11, 2011.
- 8 K. Fukuda. Walking on the arrangement, not on the feasible region. Efficiency of the Simplex Method: Quo vadis Hirsch conjecture?, IPAM, UCLA, 2011. presentation slides available as http://helper.ipam.ucla.edu/publications/sm2011/sm2011_9630.pdf.
- 9 K. Fukuda. Lecture: Polyhedral computation. <http://www-oldurls.inf.ethz.ch/personal/fukudak/lect/pclect/notes2015/>, 2015.
- 10 K. Fukuda, B. Gärtner, and M. Szedlák. Combinatorial redundancy removal. *Preprint: arXiv:1412.1241*, 2014.
- 11 K. Fukuda and T. Terlaky. Criss-cross methods: A fresh view on pivot algorithms. *Mathematical Programming*, 79:369–395, 1997.
- 12 V. Klee and G. J. Minty. How good is the simplex algorithm? In O. Shisha, editor, *Inequalities III*, pages 159–175. Academic Press, 1972.
- 13 J. Matoušek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16:498–516, 1996.
- 14 Th. Ottmann, S. Schuierer, and S. Soundaralakshmi. Enumerating extreme points in higher dimensions. In E. W. Mayer and C. Puech, editors, *STACS 95: 12th Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 900, pages 562–570. Springer-Verlag, 1995.
- 15 C. Roos. An exponential example for Terlaky’s pivoting rule for the criss-cross simplex method. *Mathematical Programming*, 46:79–84, 1990.
- 16 T. Terlaky. A finite criss-cross method for the oriented matroids. *Journal of Combinatorial Theory Series B*, 42:319–327, 1987.
- 17 Z. Wang. A finite conormal-elimination free algorithm over oriented matroid programming. *Chinese Annals of Math.*, 8B:120–125, 1987.

Effectiveness of Local Search for Geometric Optimization

Vincent Cohen-Addad and Claire Mathieu*

Département d'Informatique, UMR CNRS 8548
École Normale Supérieure, Paris, France
{vcohen, cmathieu}@di.ens.fr

Abstract

What is the effectiveness of local search algorithms for geometric problems in the plane? We prove that local search with neighborhoods of magnitude $1/\epsilon^c$ is an approximation scheme for the following problems in the Euclidean plane: TSP with random inputs, Steiner tree with random inputs, uniform facility location (with worst case inputs), and bicriteria k -median (also with worst case inputs). The randomness assumption is necessary for TSP.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Local Search, PTAS, Facility Location, k -Median, TSP, Steiner Tree

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.329

1 Introduction

Local search. Local search techniques are popular heuristics for hard combinatorial optimization problems. Given a feasible solution, the algorithm repeatedly performs operations from the given class, each improving the cost of the current solution, until a solution is reached for which no operation yields an improvement (a locally optimal solution). Alternatively, we can view this as a neighborhood search process, where each solution has an associated neighborhood of adjacent solutions, i.e., those that can be reached with a single operation, and one moves to a better neighbor until none. Such techniques are easy to implement, easy to parallelize, and fast and give good results. One advantageous feature of local search algorithms is their flexibility; they can be applied to arbitrary cost functions, even in the presence of additional constraints. However, there has long been a gap between worst-case guarantees and real-world experience. Thus, it is interesting to analyze such algorithms rigorously and, even in settings where alternative, theoretically optimal polynomial-time algorithms are known.

Problems studied. We focus on Euclidean problems in the plane (the results extend to small dimensions), and study clustering and network connectivity type problems: the traveling salesman problem (TSP), Steiner tree, facility location, and k -median. The *traveling salesman* problem is to connect n input points with a tour of minimum total length. The *Steiner tree* problem, given n terminal points, is to choose additional *Steiner* points so as to minimize the length of the minimum tree spanning terminal and Steiner points. The *facility location* problem, given n client points and a facility opening cost f , chooses how many facilities to open and where to open them to minimize the combination of the cost of opening facilities

* Partially supported by ANR RDAM.



and of the total distance from each client to the nearest open facility. The k -median problem, given n points and an integer k , chooses where to open k facilities so as to minimize the total distance from each client to the nearest open facility.

Algorithms. Our goal is to prove, under minimal assumptions, that local search finds solutions whose cost is within a $(1 + \epsilon)$ factor of optimal. For that goal, local search must do a little more: instead of modifying the current solution by swapping a single point, edge or edge pair (depending on the problem) in and out of the solution, our version of local search swaps up to $1/\epsilon^c$ points, edges or edge pairs. This is a standard variation of local search (particularly for the traveling salesman tour), whereby each iteration is slowed down due to an increase in the size of the neighborhood, but the local optimum tends to be reached after fewer iterations and is of higher quality. Moreover, most implementations of local search do not continue iterating all the way to a local optimum, but stop once the gain obtained by each additional iteration is essentially negligible. Our algorithm thus has a stopping condition, when no local exchange could improve the cost by more than a factor of $1 - 1/n$. Then, the runtime is polynomial, at most $n^{1/\epsilon^{O(1)}}$.

Results. Our results are as follows.

1. For TSP, we assume that the input points are random uniform in $[0, 1]^2$. Here local search swaps $O(1/\epsilon^c)$ edges in the tour. Then local search finds a solution with cost $(1 + \mathcal{O}(\epsilon))OPT$. The proof is not difficult and serves as a warm-up to the later sections. The random input assumption is necessary: in the worst-case setting, we give an example where a locally optimal solution has cost more than $(2 - \epsilon)OPT$.
2. Similarly, for Steiner tree, assuming random uniform input, again local search finds a solution with cost $(1 + \epsilon)OPT$.
3. For facility location, we prove the following: consider the version of local search where local moves consist of adding, deleting or swapping $O(1/\epsilon^c)$ facilities. Then, even for worst case inputs, local search finds a solution with cost $(1 + \epsilon)OPT$. This is the core result of the paper. We transform the dissection technique from Kolliopoulos and Rao [14] into a tool for analyzing local search.
4. For k -median, our result is similar, except that local search uses $(1 + \epsilon)k$ medians instead of k , so that result is bicriteria. This is a technical, variant of the facility location result.

Related work

TSP and Steiner Tree. The TSP problem in the Euclidean plane has a long history, including work with local search [9, 17, 18]. Most relevant is the work of Karp [13] giving a simple construction of a near-optimal tour when points are drawn from a random distribution. That work has been subsumed by the approximation schemes of Arora [1] (and its improvements [2, 23]) and of Mitchell [21], using a hierarchical dissection technique. Arora noted the relation between that technique and local search, observing:

Local-exchange algorithms for the TSP work by identifying possible edge exchanges in the current tour that lower the cost [...]. Our dynamic programming algorithm can be restated as a slightly more inefficient backtracking [...]. Thus it resembles k -OPT for $k = O(c)$, except that cost-increasing exchanges have to be allowed in order to undo bad guesses. Maybe it is closer in spirit to more ad-hoc heuristics such as genetic algorithms, which do allow cost-increasing exchanges.

In fact, even with neighborhoods of size $f(\epsilon)$, even in the Euclidean plane, local search for TSP can get stuck in a local optimum whose value is far from the global optimum. However, in the case of random inputs the intuition is correct. Local search algorithms have been widely studied for TSP, but mostly for either a local neighborhood limited to size of 2 or 3 (the 2-OPT or 3-OPT algorithms), or for the general metric case. Those studies lead to proofs of constant factor approximations, see [6, 11, 20, 18, 25]. In particular, in [6], it is proved (by example) that for Euclidean TSP 2-OPT cannot be a constant-factor approximation in the worst case. For the metric Steiner Tree problem, the best approximation algorithm up to 2010 was a constant factor approximation due to Robins and Zelikovsky and was by local search [24].

Facility Location and k -Median. For clustering problems – facility location and k -median – there has also been much prior work. A proof of NP-Hardness of k -median even in the Euclidean setting is given in [19]. The first theoretical guarantees for local search algorithms for clustering problems are due to Korupolu et al. [15]. They show that the local search algorithm which allows swaps of size p is a constant factor approximation for the metric case of the k -Median and Facility Location problems. However, for k -Median the algorithm requires a constant-factor blowup in the parameter k . By further refining the analysis, Charikar et al. [7] improved the approximation ratio. More recently, Arya et al. showed in [3] that the local search algorithm which allows swaps of size p is a $3 + 2/p$ -approximation without any blowup in the number of medians. Nevertheless, no better results were known for the Euclidean case (See the survey paper [26]). Kolliopoulos and Rao define in [14] a recursive “adaptive” dissection of a square enclosing the input points. At each dissection step ¹, they cut the longer side of each rectangle produced by the previous step in such a way that each of the two parts has roughly the same surface area. Our analysis uses a new version of their dissection algorithm to analyze the local search algorithm.

Other related work. The question of the efficiency of local search for Euclidean problems was already posed by Mustafa and Ray and Chan and Har-Peled. They proved that local search (with local neighborhood enabling moves of size $\Theta(1/\epsilon)$) gives approximation schemes for hitting circular disks in two dimensions with the fewest points, for several other Euclidean hitting set problems [22], and for independent sets of pseudo-disks [5]. This led to further PTASs by local search for dominating set in disks graph [10] and for terrain guarding [16]. Those papers rely on the combinatorial properties of bipartite planar graphs. Our analysis technique is different since we rely on dissections.

One problem related to facility location is k -means. For k -means, Kanungo, Mount, Netanyahu and Piatko [12] proved that local search gives a constant factor approximation. Much remains to be understood.

We also note that there exists proofs of constant factor approximation by local search for the metric capacitated facility location [8].

Plan. The paper is organized as follows: in the next section, as a warm-up we prove the results on TSP and Steiner tree for random inputs. We then analyze local search for facility location, proposing a new recursive dissection. We suitably extend lemmas from [14]. The meat of that section is the proof of Proposition 4.2, which is our main technical contribution.

¹ There is also a “sub-rectangle” step not described here.

We end with the k -median result, that requires additional ideas to deal with the cardinality constraint.

2 Polynomial-Time Local Search Algorithms

Throughout this paper, we denote by $L \Delta L'$ the symmetric difference of the sets L and L' . We present the local search algorithm that is considered in this paper (see Algorithm 1 below).

Algorithm 1 Local Search (ε)

- 1: **Input:** A set \mathcal{C} of points in the Euclidean plane
 - 2: $S \leftarrow$ Arbitrary feasible solution (of cost at most $\mathcal{O}(2^n \text{OPT})$).
 - 3: **while** $\exists S'$ s.t. $\text{Condition}(S', \varepsilon)$ **and** $\text{cost}(S') \leq (1 - 1/n) \text{cost}(S)$
 - 4: **do**
 - 5: $S \leftarrow S'$
 - 6: **end while**
 - 7: **Output:** S
-

- Note that the type of S , Condition , $f(\varepsilon)$ and $\text{Cost}(S)$ are problem dependent. Namely,
- for Facility Location, S is a set of points, $\text{Condition}(S', \varepsilon)$ is $|S \Delta S'| = \mathcal{O}(1/\varepsilon^3)$ and $\text{Cost}(S) = |S| + \sum_{c \in \mathcal{C}} \min_{s \in S} d(c, s)$;
 - for k -Median, S is a set of points, $\text{Condition}(S', \varepsilon)$ is $|S \Delta S'| = \mathcal{O}(1/\varepsilon^9)$ and $|S'| \leq (1+3\varepsilon)k$ and $\text{Cost}(S) = \sum_{c \in \mathcal{C}} \min_{s \in S} d(c, s)$;
 - for TSP S is a set of edges, $\text{Condition}(S', \varepsilon)$ is $|S \Delta S'| = \mathcal{O}(1/\varepsilon^2)$ and “ S' is a tour and there is no two edges intersecting” (if the initial tour contains intersecting edges we start by modifying the tour so that no two edges intersect) and $\text{Cost}(S) = \sum_{s \in S} \text{length}(s)$;
 - for Steiner Tree, S is a set of points, $\text{Condition}(S', \varepsilon)$ is $|S \Delta S'| = \mathcal{O}(1/\varepsilon^2)$ and $|S'| \leq n$ (if the initial set of Steiner vertices is greater than n , we greedily remove Steiner vertices until the set has size n) and $\text{Cost}(S) = \text{MST}(S \cup \mathcal{C})$, where $\text{MST}(S \cup \mathcal{C})$ is the length of the minimum spanning tree of the points in $S \cup \mathcal{C}$.

We now focus on the guarantees on the execution time of the algorithms presented in this paper. The proof of the following Lemma is deferred to the Appendix.

► **Lemma 2.1.** *The number of iterations of Algorithm 1 is polynomial for the Facility Location, k -Median, Traveling Salesman and Steiner Tree Problems.*

► **Remark.** Up to discretizing the plane and replacing $(1 - 1/n)$ by $(1 - \Theta(1/n))$, finding S' takes time $\mathcal{O}(n^{\mathcal{O}(1/\varepsilon^c)} \varepsilon^{-1})$, for some constant c which depends on the algorithm.

3 Euclidean Traveling Salesman Problem and Steiner Tree

► **Theorem 3.1.** *Consider a set of points chosen independently and uniformly in $[0, 1]^2$. Algorithm 1 produces:*

- *In the case of the Traveling Salesman problem, a tour whose length is at most $(1 + \mathcal{O}(\varepsilon))T_{\text{OPT}}$, where T_{OPT} is the length of the optimal solution.*
- *In the case of the Steiner Tree problem, a tree whose length is at most $(1 + \mathcal{O}(\varepsilon))T_{\text{OPT}}$, where T_{OPT} is the length of the optimal solution.*

To prove Theorem 3.1, we first prove the following result.

- **Theorem 3.2.** *Consider an arbitrary set of points in $[0, 1]^2$. Algorithm 1 produces:*
 - *In the case of the Traveling Salesman problem, a tour whose length is at most $(1 + \mathcal{O}(\varepsilon^2))T_{OPT} + \mathcal{O}(\varepsilon\sqrt{n})$, where T_{OPT} is the length of the optimal solution.*
 - *In the case of the Steiner Tree problem, a tree whose length is at most $(1 + \mathcal{O}(\varepsilon^2))T_{OPT} + \mathcal{O}(\varepsilon\sqrt{n})$, where T_{OPT} is the length of the optimal solution.*

We model a random distribution of points in a region \mathcal{P} of the plane by a two-dimensional Poisson distribution $\Pi_n(\mathcal{P})$. The distribution $\Pi_n(\mathcal{P})$ is determined by the following assumptions:

1. the numbers of points occurring in two or more disjoint sub-regions are distributed independently of each other;
2. the expected number of points in a region A is $nv(A)$ where $v(A)$ is the area of A ; and
3. as $v(A)$ tends to zero, the probability of more than one point occurring in A tends to zero faster than $v(A)$.

From these assumptions it follows that $\Pr[A \text{ contains exactly } m \text{ points}] = e^{-\lambda} \lambda^m / m!$, where $\lambda = nv(A)$. The following result is known.

- **Theorem 3.3** ([4]). *Let \mathcal{P} be a set of n points distributed according to a two-dimensional Poisson distribution $\Pi_n(\mathcal{P})$ in $[0, 1]^2$ and let $T_n(\mathcal{P})$ be the random variable that denotes the length of the shortest tour through the points in \mathcal{P} . There exists a positive constant β (independent of \mathcal{P}) such that $T_n(\mathcal{P})/\sqrt{n} \rightarrow \beta$ with probability 1.*

Assuming Theorems 3.2 and 3.3, we can prove Theorem 3.1.

Proof of Theorem 3.1. We focus on the Traveling Salesman case. Let L be the tour produced by Algorithm 1 and T_{OPT} be the optimal tour. By Theorem 3.3, we have that $\text{Cost}(T_{OPT}) = \mathcal{O}(\sqrt{n})$ with probability 1. Hence, Theorem 3.2 implies

$$(1 - \varepsilon^2) \cdot \text{Cost}(L) \leq \text{Cost}(T_{OPT}) + \mathcal{O}(\varepsilon\sqrt{n}) = (1 + \mathcal{O}(\varepsilon)) \cdot \text{Cost}(T_{OPT}).$$

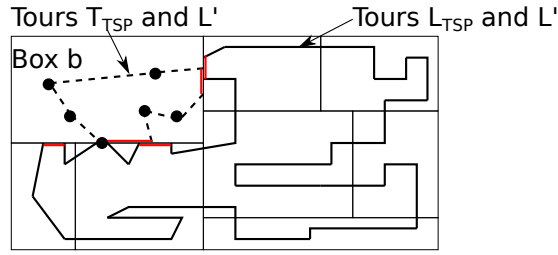
We now consider the random variable $ST_n(\mathcal{P})$ that denotes the length of the shortest Steiner Tree through the points in \mathcal{P} . Since the length of the optimal Steiner Tree is at least half the length of the optimal Traveling Salesman Tour, Theorem 3.3 implies that there exists a constant δ such that $ST_n(\mathcal{P})/\sqrt{n} \geq \delta$ with probability 1. Then, the exact same reasoning applies to prove the Steiner Tree case. ◀

The rest of the section is dedicated to the proof of Theorem 3.2. To this aim, we define a recursive dissection of the unit square according to a set of points \mathcal{P} . At each step we cut the longer side of each rectangle produced by the previous step in such a way that each of the two parts contains half the points of \mathcal{P} that lie in the rectangle. The process stops when each rectangle contains $\Theta(1/\varepsilon^2)$ points of \mathcal{P} . We now consider the final rectangles and we refer to them as *boxes*. Let \mathcal{B} be the set of boxes.

- **Lemma 3.4** ([13]). $\sum_{b \in \mathcal{B}} |\partial b| = \mathcal{O}(\varepsilon\sqrt{|\mathcal{P}|})$, where $|\partial b|$ is the perimeter of box b and $|\mathcal{P}|$ is the number of points in \mathcal{P} .

For any set of segments S and box b and for each segment s , let s_b be the part of s that lies inside b . We define $\text{In}(S, b) := \{s_b \mid s \in S \text{ and } s \text{ has at least one endpoint in } b\}$ and $\text{Cross}(S, b) := \{s_b \mid s \in S \text{ and } s \text{ has no endpoint in } b\}$. Moreover we define $\text{Out}(S, b) := \{s_{b'} \mid s \in S \text{ and } b \neq b'\}$. Additionally, let $S(b) = \sum_{s \in S} \text{length}(s_b)$.

We can now prove the two following structural Lemmas. See Fig. 1 for an illustration of the proof.



■ **Figure 1** The solid black segments form the tour L_{TSP} outside b . The dotted line segments are the tour T_{TSP} inside b . The red segments are the one needed to connect the two tours.

► **Lemma 3.5.** *Let L_{ST} be a locally optimal solution to the Steiner Tree problem and let T_{ST} be any Steiner Tree. Let \mathcal{B} be a set of boxes produced by a dissection of $\mathcal{P} \cup L_{ST} \cup T_{ST}$. Using the same notation for a set of segments and their total length, we then have for any box $b \in \mathcal{B}$*

$$(1 - \mathcal{O}(\varepsilon^2))L_{ST}(b) \leq \text{In}(T_{ST}, b) + |\partial b| + L_{ST}/n,$$

where $|\partial b|$ is the perimeter of b .

Proof. For each box b , the segments of $\text{Cross}(L_{ST}, b)$ can be distributed into 6 different classes according to which side of b they intersect.

We divide further. Since the segments of a class are pairwise disjoint, there is a natural ordering of the segments inside each class. For each class that contains more than $1/\varepsilon^2$ segments, we partition them into subsets that contain $\Theta(1/\varepsilon^2)$ consecutive segments (in the natural order of the class). We define a sub-box for each subset of each class as follows. Let s and s' be the two extreme segments of the set in the ordering of the class. The sides of the sub-box associated to this subset consists of s and s' and the two shortest paths p, p' along the sides of b that connects the endpoints of s and s' .

Remark that the sum of the lengths of the sides of all the sub-boxes is at most $|\partial b| + \mathcal{O}(\varepsilon^2 L_{ST}(b))$. For each sub-box b_0 , let L' be the set of vertices of L_{ST} that are outside b_0 , plus the set of vertices of T_{ST} that are inside b_0 , plus the set of the intersection points of the edges of L_{ST} and T_{ST} with the sides of b_0 . Thus, $L' \leq \text{Out}(L_{ST}, b_0) + \text{In}(T_{ST}, b_0) + |\partial b_0|$. Moreover, we have $|L_{ST} \triangle L'| = \mathcal{O}(1/\varepsilon^2)$ and the local near-optimality argument applies. Namely, we obtain that $(1 - 1/n)L_{ST} \leq L'$, and so

$$-1/n \cdot L_{ST} + \text{In}(L_{ST}, b_0) + \text{Cross}(L_{ST}, b_0) \leq \text{In}(T_{ST}, b_0) + |\partial b_0|.$$

We now sum over all sub-boxes of box b and we obtain

$$L_{ST}(b) = \text{In}(L_{ST}, b_0) + \text{Cross}(L_{ST}, b_0) \leq \text{In}(T_{ST}, b) + |\partial b| + \mathcal{O}(\varepsilon^2 L_{ST}(b)) + L_{ST}/n.$$

◀

► **Lemma 3.6.** *Let L_{TSP} be a locally optimal solution to the Traveling Salesman problem and let T_{TSP} be any tour. Let \mathcal{B} be a set of boxes produced by a dissection of \mathcal{P} . Using the same notation for a set of segments and their total length, we then have for any box $b \in \mathcal{B}$*

$$(1 - \mathcal{O}(\varepsilon^2))L_{TSP}(b) \leq \text{In}(T_{TSP}, b) + 3|\partial b|/2 + L_{TSP}/n,$$

where $|\partial b|$ is the perimeter of b .

Proof. We again further divide the boxes into sub-boxes as we did for Lemma 3.5. For each sub-box b_0 , we define a tour L' obtained by a traversal of the following Eulerian graph. The graph vertices are \mathcal{P} , plus the corners of ∂b_0 , plus all points of intersection of L_{TSP} and T_{TSP} with ∂b_0 . The edges are the segments of $\text{Out}(L_{\text{TSP}}, b_0)$, plus the segments of $\text{In}(T_{\text{TSP}}, b_0)$, plus ∂b_0 (so that the result is connected), plus a minimum length matching of the odd vertices of ∂b_0 (so that the result is Eulerian). Thus, $L' \leq \text{Out}(L_{\text{TSP}}, b_0) + \text{In}(T_{\text{TSP}}, b_0) + 3|\partial b_0|/2$.

Since the number of edges of L intersecting b_0 is $\mathcal{O}(1/\varepsilon^2)$ and the number of edges in $\text{In}(T_{\text{TSP}}, b_0)$ is $\mathcal{O}(1/\varepsilon^2)$, we have $|L_{\text{TSP}} \Delta L'| = \mathcal{O}(1/\varepsilon^2)$ and the local near-optimality argument applies. Namely, we obtain $(1 - 1/n)L_{\text{TSP}} \leq L'$, and so

$$-1/n \cdot L_{\text{TSP}} + \text{In}(L_{\text{TSP}}, b_0) + \text{Cross}(L_{\text{TSP}}, b_0) \leq \text{In}(T_{\text{TSP}}, b_0) + 3|\partial b_0|/2.$$

We now sum over all sub-boxes of box b and we obtain

$$L_{\text{TSP}}(b) = \text{In}(L_{\text{TSP}}, b) + \text{Cross}(L_{\text{TSP}}, b) \leq \text{In}(T_{\text{TSP}}, b) + 3|\partial b|/2 + \mathcal{O}(\varepsilon^2 L_{\text{TSP}}(b)) + L_{\text{TSP}}/n.$$

◀

We can now prove Theorem 3.2.

Proof of Theorem 3.2. We first consider the Traveling Salesman case. Let L_{TSP} be a tour produced by Algorithm 1 and T_{TSP} be any tour. Lemma 3.6 implies that for any box b , we have

$$(1 - \mathcal{O}(\varepsilon^2))L_{\text{TSP}}(b) \leq \text{In}(T_{\text{TSP}}, b) + 3|\partial b|/2 + L_{\text{TSP}}/n.$$

Since there are $\mathcal{O}(\varepsilon^2 n)$ boxes in total, by summing over all boxes, we obtain

$$-\mathcal{O}(\varepsilon^2 L_{\text{TSP}}) + \sum_{b \in \mathcal{B}} L_{\text{TSP}}(b) = (1 - \mathcal{O}(\varepsilon^2))L_{\text{TSP}} \leq \sum_{b \in \mathcal{B}} (\text{In}(T_{\text{TSP}}, b) + 3|\partial b|/2) \leq T_{\text{TSP}} + \frac{3}{2} \sum_{b \in \mathcal{B}} |\partial b|.$$

By Lemma 3.4, $\sum_{b \in \mathcal{B}} |\partial b| = \mathcal{O}(\varepsilon\sqrt{n})$ and so,

$$(1 - \mathcal{O}(\varepsilon^2)) \cdot L_{\text{TSP}} \leq T_{\text{TSP}} + \mathcal{O}(\varepsilon\sqrt{n}).$$

To prove the Steiner Tree case, it is sufficient to notice that the total number of vertices in $\mathcal{P} \cup L_{\text{ST}} \cup T_{\text{ST}}$ is at most $3n$. It follows that the total number of boxes is $\mathcal{O}(\varepsilon^2 n)$ and by Lemma 3.4, $\sum_{b \in \mathcal{B}} |\partial b| = \mathcal{O}(\varepsilon\sqrt{n})$. We apply a reasoning similar to the one for the TSP case to conclude the proof. ◀

Notice that we do not assume that the points are randomly distributed in the $[0, 1]^2$ for the proofs of Lemmas 3.5 and 3.6 and Theorem 3.2, thus they hold in the worst-case.

► **Remark.** One can ask whether it is possible to prove that the local search for TSP is a PTAS without the random input assumption. However, there exists a set of points such that there is a local optimum whose length is at least $(2 - o(\varepsilon))\text{Cost}(\text{OPT})$.

4 Clustering Problems

We now tackle the analysis of the local search algorithm for some Clustering problems. Recall that L and G denote the local and global optima respectively. In the following, for each facility l of L (resp. G), we denote by $V_L(l)$ (resp. $V_G(l)$) the Voronoi cell of l in the Voronoi diagram induced by L (resp. G). We extend this notation to any subset F of L , namely, $V_L(F)$ denotes the union of the Voronoi cells of the facilities of F induced by L . We define

a recursive randomized decomposition (Algorithm 2) based on L and G (and the Voronoi cells induced by L). This decomposition produces a tree encoded by the function $\text{Children}()$, where each node is associated to a region of the Euclidean plane. In the first step of the dissection, B is the smallest square that contains all the facilities of $L \cup G$. At every recursive call of the procedure for (B_r, L_r, G_r) , the algorithm maintains the following invariants:

- B_r is a rectangle of bounded aspect ratio;
- L_r consists of all the facilities of L that are contained in B_r ;
- G_r consists of all the facilities of G that are contained in B_r , plus some facilities of G that belong to $V_L(L_r)$.

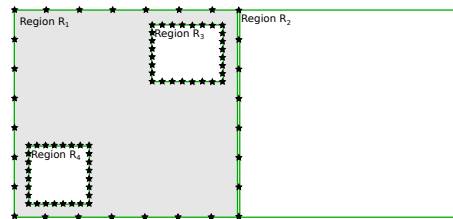
Algorithm 2 Recursive Adaptive Dissection Algorithm

```

1: procedure ADAPTIVE_DISSECTION( $B, L, G, V_L$ )
2:   if  $|L| + |G| \geq 1/2\varepsilon^2$  then
3:     if  $|L| > 1/2\varepsilon$  then
4:       Sub-Rectangle Process:
5:        $B' \leftarrow$  minimal rectangle containing all facilities of  $L$  in  $B$ 
6:        $b' \leftarrow$  maximum side-length of  $B'$ 
7:        $B'_+ \leftarrow$  Rectangle centered on  $B'$  and extended by  $b'/3$  in all four directions.
8:        $B'' \leftarrow B'_+ \cap B$ 
9:       Cut-Rectangle Process:
10:       $s'' \leftarrow$  maximum side-length of  $B''$ 
11:       $\ell \leftarrow$  line segment that is orthogonal to the side of length  $s''$  and intersects it in
      a random position in the middle  $s''/3$ .
12:      Cut  $B''$  into two rectangles  $B_1$  and  $B_2$  with  $\ell$ .
13:
14:       $\text{Children}(B) \leftarrow \{B_1, B_2\}$ 
15:       $L_1 \leftarrow L \cap B_1$ 
16:       $L_2 \leftarrow L \cap B_2$ 
17:       $G_1 \leftarrow G \cap \{g \mid g \in V_L(L_1) \text{ and } g \notin B_2\}$ 
18:       $G_2 \leftarrow G \setminus G_1$ 
19:      DISSECTION( $B_1, L_1, G_1, V_L$ )
20:      DISSECTION( $B_2, L_2, G_2, V_L$ )
21:     else
22:       Partition Process:
23:        $\text{Children}(B) \leftarrow$  Arbitrary partition of the facilities of  $L \cup G$  in parts of size in
        $[1/2\varepsilon^2, 1/\varepsilon^2]$ 
24:     end if
25:   end if
26: end procedure

```

Regions. We now introduce the crucial definition of *regions* of a dissection tree \mathcal{T} of solutions L and G . For any node N of the dissection produced by the Partition Process, we consider that the associated rectangle is the bounding box of the facilities of $L_N \cup G_N$. We assign labels to the nodes of the tree. The label of a leaf B is $|L_B| + |G_B|$. Then we proceed bottom-up, for each node of the tree, the labels of a node is equal to the sum of the labels of its two children. Once a node has a label greater than $1/2\varepsilon^2$, we say that this node is a *region node* of the tree and set its label to 0. We define the regions according to the region



■ **Figure 2** Details of the regions and portals associated to a dissection. The star-shaped points are the portals associated to Region R_1 . Regions R_2, R_3, R_4 are the only regions sharing portals with region R_1 . All the regions are disjoint.

nodes. For each region node R , the associated region is the rectangle defined by the node minus the regions of its descendants, namely minus the rectangles of nodes of label 0 that are descendants of R . See Fig. 2 for an illustration of the regions. In the following, we denote by \mathcal{R} the set of regions.

Portals. Let \mathcal{D} be a dissection produced by Algorithm 2. For any region R of \mathcal{D} not produced by the Partition Process, we place p equally-spaced *portals* along each boundary of R . We refer to the dissection \mathcal{D} along with the associated portals as \mathcal{D}_p . See Fig. 2 for more details on the regions and portals.

Definitions and Notations. For any clustering problem, we denote by \mathcal{C} the sets of the input points. We refer to an input point as a *client*. A solution to a clustering problem is a set of facilities $S \subset \mathbb{R}^2$.

For any solution S and any client c , we denote by c_S the distance from client c to the closest facility of S : $c_S = \min_{s \in S} d(c, s)$. The service cost of a solution S to a clustering problem is $\sum_{c \in \mathcal{C}} c_S$. Additionally, for any solution S and client c , we define $c(S)$ as the facility of S that serves c in solution S , namely $c(S) := \operatorname{argmin}_{s \in S} d(c, s)$

Let B be the smallest rectangle that contains all the clients. Let L and G be two sets of facilities. We now give the definition of an assignment which is crucial for the main proposition.

► **Definition 4.1.** We define an *assignment* as a function that maps the clients to the facility of $L \cup G$.

Let E_0 be the assignment that maps each client c to the facility of $\{c(L), c(G)\}$ that is the farther, namely, $\forall c \in \mathcal{C}, E_0(c) = \operatorname{argmax}(dist(c, c(G)), dist(c, c(L)))$.

We show the following proposition which is the technical center of the proof.

► **Proposition 4.2.** Let $1/\varepsilon^2 > 0$ be an integer, G and L be two sets of facilities. Let $\mathcal{D}_{1/\varepsilon^2}$ be a dissection tree with portals. There exists an assignment E that satisfies the following properties. Let R be a region not produced by the Partition Process. If a client c is such that $c(L) \in R$ and $c(G) \notin R$ then $E(c)$ is either a portal of R or a facility of $L \setminus R$. Moreover,

$$\mathbb{E} \left[\sum_{c \in \mathcal{C}} |dist(c, E(c)) - dist(c, E_0(c))| \right] = \sum_{c \in \mathcal{C}} \mathcal{O}(\varepsilon^2 \log(1/\varepsilon^2) \cdot (c_G + c_L)).$$

We start by proving some properties of Algorithm 2². The proofs of the following Lemmas are deferred to the Appendix.

► **Definition 4.3** (Aspect Ratio). We define the aspect ratio of a rectangle R that has sides of lengths r and r' as $\max(\frac{r}{r'}, \frac{r'}{r})$.

► **Lemma 4.4.** Let R be a rectangle produced by either the Sub-Rectangle or the Cut-Rectangle process of Algorithm 2. The aspect ratio of R is at most 5.

► **Lemma 4.5** ([14]). Let $l \in L$ be a facility and $v \in \mathbb{R}^2$ be any point. Let d be the distance between v and l . If a cutting line segment s produced by the Sub-Rectangle process during Algorithm 2 separates v and l for the first time, then $\text{length}(s) \leq 5d$.

► **Lemma 4.6.** Let L be a set of facilities. Let $v \in \mathbb{R}^2$, $l \in L$, $d_0 = \text{dist}(v, l)$. Suppose that, in Algorithm 2, v and l are first separated by a line s that is vertical and that l is to the right of s . Let d_1 be the distance from v to the closest open facility located to its left. Then, the length of s is either: (i) larger than $d_1/4$ or (ii) smaller than $12d_0$.

► **Lemma 4.7** ([14]). Let $\text{Event}_0(d, s)$ denote the event that an edge e of length d is separated by a cutting line of side-length s that is produced by Cut-Rectangle. Then, $\Pr[\text{Event}_0(d, s)] \leq 3d/s$.

We now show the proof of the Structure Theorem.

Proof of Proposition 4.2. Let $p := 1/\varepsilon^2$. By linearity of expectation, we only need to show this on a per-client basis.

Let c be a client and R a region containing $l := c(L)$ but not $g := c(G)$. Let B be the first box of the dissection, in top-down order, that contains l but not g , and let s be the side of B that is crossed by $[l, g]$. We have: $\text{dist}(g, l) \leq \text{dist}(g, c) + \text{dist}(c, l) = c_G + c_L$. Up to a rotation of center g , l is to the north-west of g . Let u, w be the closest facilities of L respectively to the south and to the east of g .

To construct E , we start with $E := E_0$, and modify E one client at a time so that each client satisfies the first property, and we bound the corresponding expected cost increase. The initial cost of E is $\sum_{c \in \mathcal{C}} \max(c_G, c_L)$. We modify $E(c)$ depending on whether s is vertical or horizontal and according to the length of s . We first provide an upper bound on the expected cost increase induced by $E(c)$ for the case where s is vertical. It is easy to see that, when s is horizontal, applying the same reasoning on w instead of u leads to an identical cost increase and thus, the total cost increase is at most twice the cost increase computed for the case where s is vertical.

By Lemma 4.6, the following cases cover all possibilities for the case where s is vertical.

- s is vertical and s was produced by Sub-Rectangle. Then we define $E(c)$ as the portal on s that is closest to $[g, l]$. By Lemma 4.5, the cost increase is at most $\mathcal{O}((c_G + c_L)/p)$.
- s is vertical and s was produced by Cut-Rectangle and its length is at most $12(c_L + c_G)$. Then again we define $E(c)$ as the portal on s that is closest to $[g, l]$. By assumption, again the cost increase is at most $\mathcal{O}((c_G + c_L)/p)$.
- s is vertical and s was produced by Cut-Rectangle and its length is greater than $12(c_L + c_G)$. Lemma 4.6 implies that s has length greater than $d_u/4$. If the length of s is in $[d_u/4, pd_u]$. Then again we define $E(c)$ as the portal on s that is closest to $[g, l]$. Let \mathcal{E}_0 be the event

² Lemma 4.5 is essentially Lemma 4 from [14] but a careful writing of the details of the calculation reveals slightly different constants.

that $d_u/4 \leq |s| \leq p \cdot d_u$ and s is vertical. The expected cost increase in this case is, by Lemma 4.7, at most

$$\sum_{\substack{d_u/4 \leq i \leq p \cdot d_u \\ \text{s.t } i/d_u \text{ is power of 2}}} \text{pr}[|s| = i \text{ and } \mathcal{E}_0] \cdot (i/p) \leq \mathcal{O}(\log(p)/p \cdot (c_G + c_L)).$$

- We now turn to the last case. Namely, s was produced by Cut-Rectangle and its length is greater than or equal to $p \cdot d_u$. We define $E(c)$ depending on whether u is in R or not. This leads to two different sub-cases.

1. $u \notin R$. Then we define $E(c) := u$. The cost is bounded by the cost to go to g ($\max(c_G, c_L)$) plus the cost to go from g to u , which is d_u . Let \mathcal{E}_1 be the event that $u \notin R$ and $p \cdot d_u < |s|$ and s is vertical. The cost increase is, by Lemma 4.7, at most,

$$\sum_{\substack{i > p \cdot d_u \\ \text{s.t } i/d_u \text{ is power of 2}}} \text{pr}[|s| = i \text{ and } \mathcal{E}_1] \cdot (d_u) \leq \mathcal{O}((c_G + c_L)/p).$$

2. $u \in R$. Let d denotes the first line that separates u from g . Since u is to the right of g , d is different from s and has size at least d_u . We have two sub-cases.

First, if d was produced before s in the dissection, then we also have $|d| > |s|$. Let \mathcal{E}_2 be the event $|d| > |s| > p \cdot d_u$ and s is vertical. We now fix d . We assign $E(c)$ to be the closest portal on R , the expected cost increase conditioned upon d is then at most:

$$\sum_{\substack{p \cdot d_u < i \leq |d| \\ \text{s.t } i/d_u \text{ is power of 2}}} \text{pr}[|s| = i \text{ and } \mathcal{E}_2] \cdot (i/p) \leq \mathcal{O}(\log(\frac{|d|}{p \cdot d_u}) \cdot (c_G + c_L)/p).$$

We then remove the conditioning on d . If d was produced by the Sub-Rectangle process, then $p \cdot d_u < |d| \leq 5d_u$ by Lemma 4.5 and the expected cost increase is at most $\mathcal{O}((c_G + c_L)/p)$. Otherwise, d was produced by the Cut-Rectangle process, and then the expected cost increase is at most

$$\sum_{\substack{i > p \cdot d_u \\ \text{s.t } i/d_u \text{ is power of 2}}} \text{pr}[|d| = i \text{ and } \mathcal{E}_2] \cdot \mathcal{O}(\log(\frac{i}{p \cdot d_u}) \cdot (c_G + c_L)/p) \leq \mathcal{O}((c_G + c_L)/p).$$

Second, if d was produced after s in the dissection, namely $|s| > |d|$. Let \mathcal{E}_3 denote the event that $|s| > |d|$ and $|s| > p \cdot d_u$ and s is vertical. We assign c to the closest portal located on d , which is at distance at most $d_u + |d|/p$ from g (and so at distance at most $c_G + d_u + |d|/p$ from c). We start by fixing s . The expected cost conditioned upon s is then (no matter how d was produced), at most

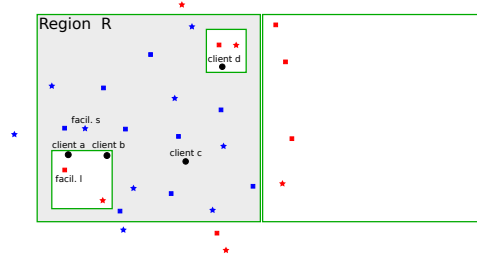
$$\sum_{\substack{d_u < i < |s| \\ \text{s.t } i/d_u \text{ is power of 2}}} \text{pr}[|d| = i \text{ and } \mathcal{E}_3] \cdot (d_u + i/p)$$

We then remove the conditioning on s , which leads to an expected cost of at most

$$\sum_{\substack{j > p \cdot d_u \\ \text{s.t } i/d_u \text{ is power of 2}}} \text{pr}[|s| = j \text{ and } \mathcal{E}_3] \sum_{d_u < i < j} 3(d_u/i) \cdot (d_u + i/p) \leq \mathcal{O}((c_G + c_L)/p)$$

Thus, the total expected cost increase for E is at most $\mathcal{O}((\log(p)/p) \cdot (c_G + c_L))$.





■ **Figure 3** Details of the partitioning of the client. The star-shaped points are the facilities of G and the square-shaped one are the facilities of L . The blue star-shaped and square-shaped belong to respectively G_R and L_R . Since client a is closer to facility l than to facility s , it belongs to the set C_L . Moreover, it is served in L by a facility that does not belong to $V_L(L_R)$, and so, it is not included in set C_R . Client b is closer to facility s than to facility l and so, it is included in set C_R albeit it is served by a facility located on another region in L . Client c is served by a facilities that belongs to $V_L(L_R)$ (in L and G) and so, it belongs to C_R . Finally, client d does not belong to $V_G(G_R)$ and so, is no included in set C_R .

Partitioning the Clients and the Facilities. Before going further, we need to define a partition of the clients and the facilities according to the dissection produced by Algorithm 2.

We partition the clients into two sets C_G and C_L . C_G contains the clients that are closer to a facility of G than to a facility of L and C_L contains the rest of the clients, namely $C_G := \{c \mid c_L = \max(c_L, c_G)\}$ and $C_L := \{c \mid c_G \neq \min(c_L, c_G)\}$. Let \mathcal{D} be a dissection produced by Algorithm 2 and the set of its associated regions \mathcal{R} . For any region R , we denote $C_G(R)$ the set of clients that are served by G_R in G and that do not lay on a region not in P . Furthermore, we define $C_L(R)$ as the set of clients that are served by L_R in L and let $C_R := V_G(G_R) \setminus (C_L \cap (V_L(L \setminus L_R)))$ ³. This set contains the clients served by G_R in G except those that belong to C_L and that are served by $L \setminus L_R$ in L . See Fig. 3 for an illustration. Additionally, we define $\Delta_R := V_L(L_R) \setminus V_G(G_R)$.

4.1 Facility Location

We now prove the approximation ratio of Algorithm 1 for facility location.

► **Theorem 4.8.** *For Facility Location, Algorithm 1 produces a solution L of cost at most $(1 + \mathcal{O}(\varepsilon)) \cdot \text{Cost}(\text{OPT})$.*

Proof. Let OPT be a globally optimum solution and L be a locally optimum solution. By Proposition 4.2, for any $p > 0$ there exists an assignment E for each random dissection \mathcal{D}_p with portals of $L \cup \text{OPT}$, such that for any client c and region R , if $c(L) \in R$ and $c(G) \notin R$ then c is served by a portal of R or a facility of $L \setminus R$ in E and the expected cost of E is at most $\mathbb{E} = \sum_{c \in \mathcal{C}} \max(c_L, c_G) + \mathcal{O}(\log(p)/p \cdot (\sum_{c \in \mathcal{C}} (c_G + c_L)))$. This implies that there exists a dissection \mathcal{D}_p for which E has value at most \mathbb{E} .

Throughout the proof, we consider this dissection \mathcal{D}_p and fix $\varepsilon := \log(p)/p$. Let \mathcal{R} be the set of regions associated to \mathcal{D}_p . We start by constructing a solution G based on OPT and we compare the cost of L to the cost of G . The solution G contains all the facilities of OPT plus some extra facilities. First, it has one facility at each portal of \mathcal{D}_p . Moreover, for each region R that is produced by the Partition Process, we open the facilities of L_R . Recall that for

³ This can be rewritten as $C_R := V_G(G_R) \cap (C_G \cup V_L(L_R))$.

each of these regions, $|L_R| \leq 1/\varepsilon$. We keep the same assignment for the clients. Since there are $\mathcal{O}(\varepsilon^2(|G| + |L|))$ regions and that for each region G uses at most $1/\varepsilon$ extra facilities, the cost of G is at most $\text{Cost}(\text{OPT}) + \mathcal{O}(\varepsilon(|\text{OPT}| + |L|)f)$. We now prove that the cost of L is at most $(1 + \mathcal{O}(\varepsilon))/(1 - \mathcal{O}(\varepsilon))$ times the cost of G , namely

$$|L| \cdot f + \sum_{c \in \mathcal{C}} c_L \leq \left(\frac{1 + \mathcal{O}(\varepsilon)}{1 - \mathcal{O}(\varepsilon)}\right) (|G| \cdot f + \sum_{c \in \mathcal{C}} c_G).$$

We focus on the cost of a region R . We show that, by local optimality, for each region R , replacing solution L by solution G does not lead to a much better cost. We serve the clients of C_R optimally (namely by the facilities that serve them in G) and the clients of $L_R \setminus G_R$ by the facilities located on the portals of R or by the facilities of $L \setminus L_R$, depending on whether they belong to C_L or C_G and according to the assignment E . Since $|L_R \setminus G_R| + |G_R \setminus L_R| = \mathcal{O}(\varepsilon^{-3})$, the locality argument applies. Namely, we have

$$(|G_R| - |L_R|)f + \sum_{c \notin C_R \cup \Delta_R} c_L + \sum_{c \in C_R} c_G + \sum_{c \in \Delta_R} c_E \geq (1 - 1/n)(|L|f + \sum_c c_L).$$

The rest of the proof is mainly computational and can be found in the appendix. ◀

4.2 k-Median

Let L and OPT be respectively local and global optimal solutions to the k -Median problem. We start with a technical Lemma which allows us to find “clusters” of regions of the plane that have roughly the same number of facilities of L and G . See Fig. 4 for an illustration. The proof of the Lemma is deferred to the Appendix.

► **Lemma 4.9** (Balanced Clustering). *Let $\mathcal{R} = \{r_1, \dots, r_p\}$ be a collection of disjoint sets. Each set contains elements of type either L or G and has size at least $1/2\varepsilon^2$ and at most $1/\varepsilon^2$. The total number of elements of type L is $(1 + 3\varepsilon)$ times higher than the number of elements of type G .*

There exists a clustering of $\{r_1, \dots, r_p\}$ in clusters satisfying the following two properties. For any cluster C ,

- *C contains at most $\mathcal{O}(1/\varepsilon^5)$ elements of \mathcal{R} , namely $|C| = \mathcal{O}(1/\varepsilon^5)$;*
- *the difference between the number of elements of L in the sets contained in C and the number of elements of G in the sets contained in C is at least $|C|/\varepsilon$:*

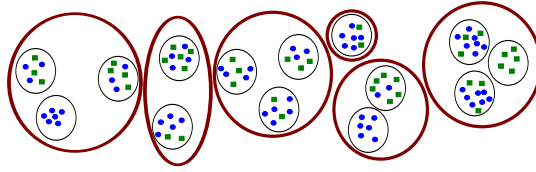
$$\sum_{r_i \in C} |r_i \cap L| - \sum_{r_i \in C} |r_i \cap G| \geq |C|/\varepsilon,$$

for any $1/\varepsilon \in \mathbb{N}$.

► **Theorem 4.10.** *For k -Median, Algorithm 1 for k -Median produces a solution L of cost at most $(1 + \mathcal{O}(\varepsilon))\text{Cost}(\text{OPT})$ using at most $1 + \mathcal{O}(\varepsilon)$ k Medians.*

Proof. Remark first that solution L uses $(1 + \mathcal{O}(\varepsilon))k$ facilities. We now show that the cost of solution L is at most $1 + \mathcal{O}(\varepsilon)$ times higher than the cost of the optimal solution.

Recall that by Proposition 4.2, for any $p > 0$ there exists an assignment E for each random dissection \mathcal{D}_p of $L \cup \text{OPT}$ with portals, such that for any client c and region R , if $c(L) \in R$ and $c(\text{OPT}) \notin R$ then c is served by a portal of R or a facility of $L \setminus R$ in E and the expected cost of E is at most $\mathbb{E} = \sum_{c \in \mathcal{C}} \max(c_L, c_{\text{OPT}}) + \mathcal{O}(\log(p)/p \cdot (\sum_{c \in \mathcal{C}} (c_{\text{OPT}} + c_L)))$.



■ **Figure 4** The circle-shaped points are the elements of type L and the square-shaped ones the elements of type G . The black circles mark the sets $\{r_1, \dots, r_p\}$ and the red ones show a clustering of those sets that satisfy the property of Lemma 4.9.

This implies that there exists a dissection \mathcal{D}_p for which E has value at most \mathbb{E} . Throughout the proof, we consider such a dissection \mathcal{D}_p and fix $\varepsilon := \log(p)/p$. Let \mathcal{R} be the set of regions associated to \mathcal{D}_p . We prove that the cost of L is at most $(1 + \mathcal{O}(\varepsilon))/(1 - \mathcal{O}(\varepsilon))$ times the cost of S , namely

$$\sum_{c \in \mathcal{C}} c_L \leq \frac{1 + \mathcal{O}(\varepsilon)}{1 - \mathcal{O}(\varepsilon)} \sum_{c \in \mathcal{C}} c_{\text{OPT}}.$$

Let \mathcal{P} be a clustering of the regions satisfying the properties of Lemma 4.9 (depending on L and OPT). We start by constructing a solution G based on OPT and we compare the cost of L to the cost of G . We construct G in a similar way to in the proof of Theorem 4.8. Namely, the solution G contains all the facilities of OPT plus some extra facilities: one facility at each portal of \mathcal{D}_p and for each region R that is produced by the Partition Process, we open the facilities of L_R . Recall that for each of these regions, $|L_R| \leq 1/\varepsilon$. We keep the same assignment for the clients. We now compare the costs of L and G . To do so, we consider all the regions of each cluster of the clustering \mathcal{P} at the same time. Namely for each cluster R , L uses at least as many facilities as G . Therefore $|S_P \setminus L| + |L \setminus S_P| = \mathcal{O}(1/\varepsilon^9)$ and the locality argument applies. The rest of the proof is similar to the proof of 4.8 and is mainly computational and can be found in the Appendix. ◀

Higher Dimensions. Previous results generalize to any dimension d . It leads to algorithms that have exponential dependency in d . More precisely, for any dimension d , more portals are needed to maintain the expected cost increase for the assignment E provided by the Structure Theorem. Each of the $2d$ faces of each region has to count p^{d-1} portals. Proposition 4.2 generalizes to any dimension d with $\mathcal{O}(dp^{d-1})$ portals instead of p . For Facility Location, $\text{Condition}(S', \varepsilon)$ has to be adapted to $|S' \setminus S| + |S \setminus S'| = \mathcal{O}(d/\varepsilon^{d+1})$. Thus, Theorem 4.8 still applies to show that the adapted Algorithm provides a $(1 + \mathcal{O}(\varepsilon))$ approximation. For the k -Median problem, $\text{Condition}(S', \varepsilon)$ has to be adapted to $|S'| \leq (1 + 3\varepsilon)k$ and $|S' \setminus S| + |S \setminus S'| = \mathcal{O}(d/\varepsilon^{7+d})$. Theorem 4.10 still applies to prove the approximation ratio of the adapted Algorithm.

References

- 1 S. Arora. Polynomial time approximation schemes for euclidean TSP and other geometric problems. In *Symp. on Foundations of Computer Science, FOCS'96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 2–11, 1996.
- 2 S. Arora. Nearly linear time approximation schemes for euclidean TSP and other geometric problems. In *Symp. on Foundations of Computer Science, FOCS'97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 554–563, 1997.

- 3 V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM J. Comput.*, 33(3):544–562, 2004.
- 4 J. Beardwood, J. H. Halton, and J. M. Hammersley. The shortest path through many points. *Mathematical Proc. of the Cambridge Philosophical Society*, 55:299–327, 1959.
- 5 T. M. Chan and S. Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. In *Proc. of the Symp. on Computational Geometry, SCG'09*, pages 333–340. ACM, 2009.
- 6 B. Chandra, H. J. Karloff, and C. A. Tovey. New results on the old k-opt algorithm for the TSP. In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia.*, pages 150–159, 1994.
- 7 M. Charikar and S. Guha. Improved combinatorial algorithms for facility location problems. *SIAM J. Comput.*, 34(4):803–824, 2005.
- 8 F. A. Chudak and D. P. Williamson. Improved approximation algorithms for capacitated facility location problems. *Math. Program.*, 102(2):207–222, March 2005.
- 9 G. A. Croes. A method for solving traveling salesman problems. *Operations Research*, 6(6):791–812, 1958.
- 10 M. Gibson and I. A. Pirwani. Algorithms for dominating set in disk graphs: Breaking the $\log n$ barrier – (extended abstract). In *Algorithms – ESA 2010, European Symp., Liverpool, UK, September 6-8, 2010. Proc., Part I*, pages 243–254, 2010.
- 11 D. S Johnson and L. A McGeoch. The traveling salesman problem : A case study in local optimization. *Local Search in Combinatorial Optimization*, 1:215–310, 1997.
- 12 T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Comput. Geom.*, 28(2-3):89–112, 2004.
- 13 R. M. Karp. Probabilistic analysis of partitioning algorithms for the traveling-salesman problem in the plane. *Mathematics of Operations Research*, 2(3):209–224, 1977.
- 14 S. G. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the euclidean k-median problem. *SIAM J. Comput.*, 37(3):757–782, 2007.
- 15 M. R. Korupolu, C. G. Plaxton, and R. Rajaraman. Analysis of a local search heuristic for facility location problems. *J. Algorithms*, 37(1):146–188, 2000.
- 16 E. Krohn, M. Gibson, G. Kanade, and K. R. Varadarajan. Guarding terrains via local search. *JoCG*, 5(1):168–178, 2014.
- 17 S. Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal, The*, 44(10):2245–2269, 1965.
- 18 S. Lin and B. W Kernighan. An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, 21(2):498–516, 1973.
- 19 N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM J. Comput.*, 13(1):182–196, 1984.
- 20 O. Mersmann, B. Bischl, J. Bossek, H. Trautmann, M. Wagner, and F. Neumann. Local search and the traveling salesman problem: A feature-based characterization of problem hardness. In *Learning and Intelligent Optimization – 6th International Conference, LION 6, Paris, France, January 16-20, 2012, Revised Selected Papers*, pages 115–129, 2012.
- 21 J. S. B. Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric tsp, k-mst, and related problems. *SIAM J. Comput.*, 28(4):1298–1309, 1999.
- 22 N. H. Mustafa and S. Ray. PTAS for geometric hitting set problems via local search. In *Proc. of the ACM Symp. on Computational Geometry, Aarhus, Denmark*, pages 17–22, 2009.

- 23 S. Rao and W. D. Smith. Approximating geometrical graphs via “spanners” and “banyans”. In *Proc. of the ACM Symp. on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 540–550, 1998.
- 24 G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *Proc. of the ACM-SIAM Symp. on Discrete Algorithms, SODA*, pages 770–779. SIAM, 2000.
- 25 D. J. Rosenkrantz, R. Edwin Stearns, and P. M. Lewis II. An analysis of several heuristics for the traveling salesman problem. *SIAM J. Comput.*, 6(3):563–581, 1977.
- 26 J. Vygen. Approximation algorithms for facility location problems. Technical Report 05950, Research Institute for Discrete Mathematics, University of Bonn, 2005.

On the Shadow Simplex Method for Curved Polyhedra

Daniel Dadush¹ and Nicolai Hähnle²

- 1 Centrum Wiskunde & Informatica, The Netherlands
dadush@cwi.nl
- 2 Universität Bonn, Germany
haehnle@or.uni-bonn.de

Abstract

We study the simplex method over polyhedra satisfying certain “discrete curvature” lower bounds, which enforce that the boundary always meets vertices at sharp angles. Motivated by linear programs with totally unimodular constraint matrices, recent results of Bonifas et al. (SOCG 2012), Brunsch and Röglin (ICALP 2013), and Eisenbrand and Vempala (2014) have improved our understanding of such polyhedra.

We develop a new type of *dual* analysis of the shadow simplex method which provides a clean and powerful tool for improving all previously mentioned results. Our methods are inspired by the recent work of Bonifas and the first named author [8], who analyzed a remarkably similar process as part of an algorithm for the Closest Vector Problem with Preprocessing.

For our first result, we obtain a constructive diameter bound of $O(\frac{n^2}{\delta} \ln \frac{n}{\delta})$ for n -dimensional polyhedra with curvature parameter $\delta \in (0, 1]$. For the class of polyhedra arising from totally unimodular constraint matrices, this implies a bound of $O(n^3 \ln n)$. For linear optimization, given an initial feasible vertex, we show that an optimal vertex can be found using an expected $O(\frac{n^3}{\delta} \ln \frac{n}{\delta})$ simplex pivots, each requiring $O(mn)$ time to compute. An initial feasible solution can be found using $O(\frac{mn^3}{\delta} \ln \frac{n}{\delta})$ pivot steps.

1998 ACM Subject Classification G.1.6 Optimization

Keywords and phrases Optimization, Linear Programming, Simplex Method, Diameter of Polyhedra

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.345

1 Introduction

The *simplex method* is one of the most important methods for solving linear programs (LPs), that is, optimization problems of the form $\max \{\langle \mathbf{c}, \mathbf{x} \rangle : \mathbf{x} \in P\}$ where P is a polyhedron defined by linear constraints. Starting from an initial vertex \mathbf{v} , a simplex algorithm provides a rule for moving from vertex to vertex along edges of the graph or 1-skeleton of P until an optimal vertex \mathbf{w} (or an unbounded ray) is found.

A long standing open question is whether there exists a polynomial-time simplex algorithm for LPs. The first obstacle in proving the existence (or non-existence) of such a method is the following fundamental question:

► **Question 1.** *Given any two vertices \mathbf{v}, \mathbf{w} of a polyhedron P , what is the best possible bound on the length of the shortest path between them, as a function of the dimension n and the number of constraints m ?*



© Daniel Dadush and Nicolai Hähnle;
licensed under Creative Commons License CC-BY
31st International Symposium on Computational Geometry (SoCG'15).
Editors: Lars Arge and János Pach; pp. 345–359



Leibniz International Proceedings in Informatics
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The polynomial Hirsch conjecture posits that the diameter of the graph of a polyhedron is bounded by a polynomial in m and n . The best known general upper bounds are however much larger. Barnette [3] and Larman [15] proved a bound of $O(2^n m)$, and Todd [20] recently proved a bound of $(m - n)^{\log n}$, slightly improving an earlier bound of Kalai and Kleitman [13, 14]. The original Hirsch conjecture, which posited a bound of $m - n$, was recently disproved for polytopes (i.e. bounded polyhedra) by Santos [18, 16], who gave a lower bound of $(1 + \varepsilon)m$ (only slightly violating the conjectured bound).

Given the difficulty of the general question, much research has been aimed at bounding the diameter of special classes of polyhedra. For example, polynomial bounds have been given for 0/1 polytopes [17], transportation polytopes [2, 6, 10], and flag polytopes [1].

Another important class, which has recently received much attention and is directly related to this work, are polyhedra whose constraint matrices are “well-conditioned”. Dyer and Frieze [11] showed that the diameter of totally unimodular polyhedra – i.e. having integer constraint matrices with all subdeterminants in $\{0, \pm 1\}$ – is bounded by $O(n^{16} m^3 (\log nm)^3)$. Their work also contains a polynomial time randomized simplex algorithm that solves linear programs over totally unimodular polyhedra.

The diameter bound of Dyer and Frieze was both generalized and improved in the work of Bonifas et al. [4]. They showed that polyhedra with integer constraint matrices and all subdeterminants bounded by Δ have diameter $O(\Delta^2 n^4 \log(n\Delta))$ if they are unbounded and $O(\Delta^2 n^{3.5} \log(n\Delta))$ if they are bounded. Their proof used certain expansion properties of the polyhedral graph and was non-constructive.

In an attempt to make the bound of [4] constructive, Brunsch and Röglin [7] showed that given any two vertices \mathbf{v}, \mathbf{w} on such a polyhedron P , a path between them of length $O(m\Delta^4 n^4)$ (note the dependence on m) can be constructed using the *shadow simplex method*. In fact, they give a more general bound based on the so-called δ -distance property of the constraint matrix, which measures how “well spread” the rows of the constraint matrix are¹. Using this parameter they give a bound of $O(mn^2/\delta^2)$ on the length of the constructed path, and recover the previous bound by the relationship $\delta \geq 1/(n\Delta^2)$.

Most recently, Eisenbrand and Vempala [12] provided a different approach to making the Bonifas et al. [4] result constructive, which more closely resembles the random walk approach of Dyer and Frieze and also extends to optimization. When the constraint matrix satisfies the δ -distance property, they show that given an initial vertex and objective, an optimal vertex can be computed using $\text{poly}(n, 1/\delta)$ random walk steps (no dependence on m). Furthermore, an initial feasible vertex can be computed using m calls to their optimization algorithm over subsets of the original constraints.

2 Results

Building and improving upon the works of Bonifas et al. [4], Brunsch and Röglin [7], and Vempala and Eisenbrand [12], we give an improved (constructive) diameter bound and simplex algorithm for polyhedra satisfying the δ -distance and other related properties. We also make improvements in the treatment of unbounded polyhedra and degeneracy. All our results are based on a new variant and analysis of the *shadow simplex method*.

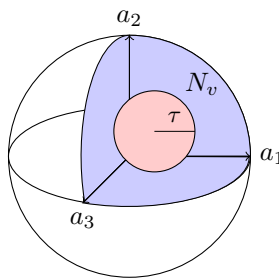
We now introduce the “discrete curvature measures” we use along with the corresponding results. We list these measures in order of increasing strength. In the next section, we

¹ We note that this measure is already implicit in [4] and that the diameter bound factors through it.

shall explain our variant of the shadow simplex method and compare it with previous implementations.

Let $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$, $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$ be a pointed polyhedron (A has full column rank $\Leftrightarrow P$ has vertices). For a vertex \mathbf{v} of P , the *normal cone* at \mathbf{v} is $N_{\mathbf{v}} = \{\sum_{i \in I_{\mathbf{v}}} \lambda_i \mathbf{a}_i : \lambda_i \geq 0, i \in I_{\mathbf{v}}\}$, where $I_{\mathbf{v}} = \{i \in [m] : \langle \mathbf{a}_i, \mathbf{v} \rangle = \mathbf{b}_i\}$ is the set of tight constraints. Equivalently, $N_{\mathbf{v}}$ is the set of all linear objective functions whose maximum over P is attained at \mathbf{v} . $N_{\mathbf{v}}$ is simplicial (non-degenerate) if it is generated by a basis of A , that is, if exactly n linearly independent constraints of P are tight at \mathbf{v} . The *normal fan* of P is the collection of all the vertex normal cones, and the *support of the normal fan* $N(P)$ is their union. A polyhedron is simple (or non-degenerate) if all its vertex normal cones are simplicial.

► **Definition 2** (τ -wide Polyhedra). We say that a cone C is τ -wide if it contains a Euclidean ball of radius τ centered on the unit sphere. We define a polyhedron P to have a τ -wide normal fan (or simply P to be τ -wide) if every vertex normal cone is τ -wide.



Roughly speaking, having a τ -wide normal fan enforces that facets always intersect at “sharp angles” (i.e. angle bounded away from π). In particular, for any vertex \mathbf{v} of P , the angle between any two rays emanating from \mathbf{v} and (non-trivially) passing through P is at most $\pi - 2\tau$. Hence one can interpret this condition as a discrete form of curvature for polyhedra. We now state our diameter bound for τ -wide polyhedra.

► **Theorem 3** (Diameter Bound, see Theorem 11). *Let $P \subseteq \mathbb{R}^n$ be an n -dimensional pointed polyhedron having a τ -wide normal fan. Then the graph of P has diameter bounded by $8n/\tau(1 + \ln(1/\tau))$. Furthermore, a path of this expected length can be constructed via the shadow simplex method.*

Restricting to n -dimensional polyhedra with subdeterminants bounded by Δ , using the relation $\tau \geq 1/(n\Delta)^2$ (for a proof, see the full version of the paper) we achieve a bound of $O(n^3 \Delta^2 \ln(n\Delta))$, improving on the existential bounds of Bonifas et al. [4]. When A is totally unimodular, the bound becomes $O(n^3 \ln n)$. In contrast to [4], we note that our bound (and proof) is the same for polytopes and unbounded polyhedra.

While our bound is constructive – we follow a shadow simplex path – it is in general only efficiently implementable when the polyhedron is simple. In the presence of degeneracy, we note that computing a single edge of the path is essentially as hard as solving linear programming. Furthermore, standard techniques for removing degeneracy, such as the perturbation or lexicographic method, may unfortunately introduce a large number of extra simplex pivots.

Interestingly, our diameter bound can take advantage of degeneracy in situations where it makes the normal cones wider. While degeneracy does not occur for “generic polyhedra”, it is very common for combinatorial polytopes. Furthermore, it can occur in ways that are

useful to our diameter bound. For example, we remark that using degeneracy one can prove that the normal fan of the perfect matching polytope is $\Omega(1/\sqrt{|E|})$ -wide [9].

To solve linear optimization problems via the shadow simplex method, we will need more than a wide normal fan. In fact, we will have different requirements for the two phases of the simplex algorithm: Phase 1, which finds an initial feasible vertex, will require more than Phase 2, which finds an optimal vertex with respect to the objective starting from a feasible vertex.

► **Definition 4** (δ -distance property). A set of linearly independent vectors $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbb{R}^n$ satisfy the δ -distance property if for every $i \in [k]$, the vector \mathbf{v}_i is at Euclidean distance at least $\delta \|\mathbf{v}_i\|$ from the span of $\{\mathbf{v}_j : j \in [k] \setminus \{i\}\}$.

For a polyhedron $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$, we define P to satisfy the *local δ -distance property* if every *feasible basis* of A , i.e. the rows of A defining a vertex of P , satisfies the δ -distance property.

We say that a set of vectors $\mathbf{v}_1, \dots, \mathbf{v}_m \in \mathbb{R}^n$ satisfy the *global δ -distance property* if every linearly independent subset satisfies the δ -distance property. We say that a matrix $A \in \mathbb{R}^{m \times n}$ satisfies the global δ -distance property if its row vectors do.

► **Lemma 5.** Let $\mathbf{v}_1, \dots, \mathbf{v}_n \in \mathbb{S}^{n-1}$ be a basis satisfying the δ -distance property. Then $\text{cone}(\mathbf{v}_1, \dots, \mathbf{v}_n)$ is δ/n -wide.

Proof. See full paper. ◀

The definitions differ in strength mainly based on the sets of bases to which they apply. The local δ -distance property is stronger than the τ -wide property for $\tau = \delta/n$, because it implies that *all triangulations* of the normal fan are τ -wide.² The global property is stronger than the local property since it applies also to infeasible bases, which allows one to control the geometry of polyhedra related to P , such as polyhedra obtained by removing a subset of constraints, which will be needed for Phase 1.

We now state our main result for Phase 2 simplex.

► **Theorem 6** (Optimization via Shadow Simplex, see Theorem 20). Let $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ be an n -dimensional polytope with m constraints satisfying the local δ -distance property. Then, given an objective $\mathbf{c} \in \mathbb{R}^n$ and a vertex \mathbf{v} of P , an optimal vertex can be computed using an expected $O((n^3/\delta) \ln(n/\delta))$ shadow simplex pivots, where each pivot requires $O(mn)$ arithmetic operations.

Our Phase 2 algorithm above is faster than the algorithms in [7, 12] and relies on a *weaker assumption* than [12]. The \mathbf{v}, \mathbf{w} path finding algorithm of Brunsch and Röglin [7] is in fact a special case of the above, since we can choose \mathbf{c} to be any objective maximized at \mathbf{w} . Comparing to the Phase 2 algorithm of Eisenbrand and Vempala [12], we require only the *local δ -distance property* instead of the global one. Whether one could rely only on the local property was left as open question in [12], which we resolve in the affirmative.

A small technical caveat is that as stated, the algorithm requires knowledge of δ . Since $\delta \leq 1$, we can always guess a number $\delta' \leq \delta \leq 2\delta'$ by trying $O(\ln 1/\delta)$ different values, incurring an $O(\ln 1/\delta)$ factor increase in running time (overestimating δ only affects correctness, not runtime). For simplicity, we shall henceforth assume that δ is known.

² However, the τ -wide property is weaker even when all normal cones are simplicial: a 2-dimensional cone of inner angle close to π is almost 1-wide, but satisfies δ -distance only for δ close to 0.

A more important caveat is that the above algorithm requires that P be a polytope (i.e. bounded). This restriction is due to the fact that we can only generate the randomness required for our bounds efficiently (that is, without solving a general LP) when the support of the normal fan equals \mathbb{R}^n .

The unbounded setting can be reduced to the bounded setting, in the standard way, by adding one or more constraints to make P bounded while not cutting off any of its vertices.

► **Definition 7.** Let $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ be a pointed polyhedron. Then a polytope $P' = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}, A'\mathbf{x} \leq \mathbf{b}'\}$ is *LP equivalent* to P if every vertex $\mathbf{v} \in P$ satisfies $\langle \mathbf{a}'_i, \mathbf{v} \rangle < \mathbf{b}'_i$ for all i ; in particular, \mathbf{v} is a vertex of P' .

Given an optimal vertex \mathbf{v} of P' as above, one can easily check whether \mathbf{v} is a vertex of P . If it is not, the original LP must be unbounded. In general, however, adding constraints to P happens at the expense of a degraded δ . In particular, the standard reduction of adding a large box constraint can degrade δ arbitrarily, hence the constraints must be added with care. We state the guarantees we can achieve below.

► **Lemma 8** (Removing Unboundedness, see full paper). Let $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ be an n -dimensional pointed polyhedron with m constraints. Let $\mathbf{a}_1, \dots, \mathbf{a}_m$ denote the rows of A and $b_{\max} = \max_{i \in [m]} \|\mathbf{b}_i\| / \|\mathbf{a}_i\|$.

1. Assume that P satisfies the local δ -distance property and that $I \subseteq [m]$, $|I| = n$, indexes the rows of a feasible basis. Letting $\mathbf{w} = -1/n \sum_{i \in I} \mathbf{a}_i / \|\mathbf{a}_i\|$, we have that

$$P' = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}, \quad \langle \mathbf{w}, \mathbf{x} \rangle \leq nb_{\max}/\delta\},$$

is a polytope that is LP equivalent to P and satisfies the local $\delta^2/(2n)$ -distance property.

2. Assume that A satisfies the global δ -distance property. Then

$$P' = \{\mathbf{x} \in \mathbb{R}^n : -n\|\mathbf{a}_i\|b_{\max}/\delta - 1 \leq \langle \mathbf{a}_i, \mathbf{x} \rangle \leq \mathbf{b}_i, \quad \forall i \in [m]\},$$

is a polytope that is LP equivalent to P and satisfies the global δ -distance property.

Finally, we use standard techniques for reducing feasibility to Phase 2 type optimization. As this generally requires pivoting over infeasible bases, we will require global instead of local properties here. Interestingly, for LPs with bounded subdeterminants, we get that the number of simplex pivots is completely independent of the number of constraints.

► **Theorem 9** (Feasibility via Shadow Simplex, see full paper). Let $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ be an n -dimensional polyhedron whose constraint matrix has full column rank and satisfies the global δ -distance property. Then a feasible solution to P can be computed using an expected $O((mn^3/\delta) \ln(n/\delta))$ shadow simplex pivots. Furthermore, if A is integral and has subdeterminants bounded by Δ , a feasible solution can be computed using an expected $O(n^5 \Delta^2 \ln(n\Delta))$ shadow simplex pivots.

Shadow Simplex Method

Our main technical contribution is a new analysis and variant of the *shadow simplex method*, which utilizes (rather unexpectedly) an approach developed in [8] for navigating over the Voronoi graph of a Euclidean lattice (see related work section).

The shadow simplex has been at the heart of many theoretical attempts to explain the surprising efficiency of the simplex method in practice. It has been shown to give polynomial bounds for the simplex method over random and smoothed linear programs [5, 19, 21]. As

mentioned above, Brunsch and Röglin [7] already showed that it yields short paths for the polyhedra we consider here.

At a high level, the shadow simplex over a polyhedron P works as follows. Given an initial objective function \mathbf{c} , a vertex \mathbf{v} of P which maximizes this objective, and a target objective function \mathbf{d} , the shadow simplex interpolates between the objective functions \mathbf{c} and \mathbf{d} and performs a pivot step whenever the optimal vertex changes (hence the alternative name *parametric* simplex method referring to the parameterization $\mathbf{c}(\lambda) = (1 - \lambda)\mathbf{c} + \lambda\mathbf{d}$ of the objective function, where λ grows from 0 to 1 over the course of the algorithm).

Traditionally, this method is understood and analyzed with a primal interpretation: The polyhedron P is orthogonally projected onto the 2-dimensional plane spanned by \mathbf{c} and \mathbf{d} (hence the term “shadow”), and the algorithm is understood in terms of the boundary of the projection P' . The optimal vertices for \mathbf{c} and \mathbf{d} project to the boundary of P' , and as long as \mathbf{c} and \mathbf{d} are in sufficiently general position, edges of P' lift to edges of P so that the boundary can be followed efficiently by an algorithm that performs simplex pivots in the original space. The number of pivot steps is then typically bounded in terms of the lengths of edges or in terms of angles between edges of P' .

Our analysis is substantially different and based on a *dual* perspective: The shadow simplex method follows the line segment $[\mathbf{c}, \mathbf{d}]$ through the normal fan of P , pivoting whenever the segment crosses into a different n -dimensional normal cone. We express the number of crossings, that is, the number of intersections between $[\mathbf{c}, \mathbf{d}]$ and the facets of the normal fan of P , in terms of certain surface area measures of translates of the normal fan. The bounds we obtain on the number of intersections are stated below.

► **Theorem 10** (Intersection bounds, see Lemmas 22 and 25). *Let $\mathcal{T} = (C_1, \dots, C_k)$ be a partition of a cone Σ into polyhedral τ -wide cones. Let $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ and let $X \in \mathbb{R}^n$ be exponentially distributed on Σ (see Section 3.1).*

1. *The expected number of facets hit by the shifted line segment $[\mathbf{c} + X, \mathbf{d} + X]$ satisfies*

$$\mathbb{E}[|\partial\mathcal{T} \cap [\mathbf{c} + X, \mathbf{d} + X]|] \leq \frac{\|\mathbf{d} - \mathbf{c}\|}{\tau} .$$

2. *Let $\alpha \in (0, 1)$. Then*

$$\mathbb{E}[|\partial\mathcal{T} \cap [\mathbf{c} + \alpha X, \mathbf{c} + X]|] \leq \frac{2n}{\tau} \ln \frac{1}{\alpha} .$$

To achieve the above bounds, the main idea is to relate the probability that the above random line segments pass through a normal cone to the probability that the associated perturbation vector lands in the cone (or some joint shift). Under the τ -wideness condition, we can in fact uniformly upper bound these proportionality factors. Since the jointly shifted normal cones are all disjoint, we can deduce the desired bounds from the fact that the sum of their measures is ≤ 1 .

We compose these bounds in a way that also departs from the classic template by using three consecutive shadow simplex paths instead of just one. For given vertices \mathbf{v} and \mathbf{w} we first pick objectives \mathbf{c} and \mathbf{d} that are “deep inside” the respective normal cones. From here, we sample an exponentially distributed perturbation vector X and traverse three paths through the normal fan in sequence:

$$\mathbf{c} \xrightarrow{(a)} \mathbf{c} + X \xrightarrow{(b)} \mathbf{d} + X \xrightarrow{(c)} \mathbf{d}$$

The perturbation X will be quite large and hence almost always large enough to push \mathbf{c} and \mathbf{d} away from their normal cones. Indeed, the high level intuition behind our path is that in

order to avoid unusually long paths from \mathbf{c} to \mathbf{d} , we first travel to a “random intermediate location”.

We note that in phases (a) and (c), randomness is only used to perturb one of the objectives. As far as we are aware, this paper provides the first successful analysis of the shadow simplex path in this setting. Furthermore, this extension is crucial to achieving our improved diameter bound. Previous algorithms were constrained to random perturbations that kept \mathbf{c} and \mathbf{d} inside their respective normal cones, making the amount of randomness they could take advantage of much smaller.

We now use the bounds from Theorem 10 to derive the diameter bound.

► **Theorem 11.** *Let $P \subseteq \mathbb{R}^n$ be a pointed full dimensional polyhedron with τ -wide normal cones. Then P has diameter bounded by $\frac{8n}{\tau}(1 + \ln 1/\tau)$.*

Proof. Let $\mathbf{v}_1, \mathbf{v}_2$ be vertices of P with normal cones $N_{\mathbf{v}_1}, N_{\mathbf{v}_2}$. Let $\mathbf{c}_1, \mathbf{c}_2 \in \mathbb{S}^{n-1}$ satisfy $\mathbf{c}_i + \tau\mathcal{B}_2^n \subseteq N_{\mathbf{v}_i}, i \in \{1, 2\}$. Let $\Sigma = N(P)$ denote the support of the normal fan of P , and let X be exponentially distributed over Σ .

We will construct a path from \mathbf{v}_1 to \mathbf{v}_2 by following the sequence of vertices optimizing the objectives in the segments $[s\mathbf{c}_1, s\mathbf{c}_1 + X], [s\mathbf{c}_1 + X, s\mathbf{c}_2 + X], [s\mathbf{c}_2 + X, s\mathbf{c}_2]$, where $s > 0$ is a scalar to be chosen later. We will condition on the event that $\|X\| \leq 2n$. Since $\mathbb{E}[\|X\|] = n$ (see Lemma 13), by Markov’s inequality this occurs with probability at least $1/2$. Under this event, by τ -wideness, we will not pivot in the segments $[s\mathbf{c}_1, s\mathbf{c}_1 + \frac{s\tau}{2n}X]$ and $[s\mathbf{c}_2 + \frac{s\tau}{2n}X, s\mathbf{c}_2]$. Using Theorem 10, the number of pivots along the segments $[s\mathbf{c}_1 + \frac{s\tau}{2n}X, s\mathbf{c}_1 + X], [s\mathbf{c}_1 + X, s\mathbf{c}_2 + X], [s\mathbf{c}_2 + X, s\mathbf{c}_2 + \frac{s\tau}{2n}X]$, is bounded by

$$\frac{\left(\frac{s\|\mathbf{c}_2 - \mathbf{c}_1\|}{\tau} + \frac{4n}{\tau} \ln\left(\frac{2n}{s\tau}\right)\right)}{\Pr[\|X\| \leq 2n]} \leq 2 \left(\frac{s\|\mathbf{c}_2 - \mathbf{c}_1\|}{\tau} + \frac{4n}{\tau} \ln\left(\frac{2n}{s\tau}\right)\right).$$

Setting $s = \frac{4n}{\|\mathbf{c}_2 - \mathbf{c}_1\|}$, the above bound becomes

$$\frac{8n}{\tau} \left(1 + \ln\left(\frac{\|\mathbf{c}_2 - \mathbf{c}_1\|}{2\tau}\right)\right) \leq \frac{8n}{\tau} \left(1 + \ln\frac{1}{\tau}\right), \quad \text{as needed.} \quad \blacktriangleleft$$

Related Work

In a surprising connection, we borrow techniques developed in a recent work of Bonifas and the first named author [8] for a totally different purpose, namely, for solving the Closest Vector Problem with Preprocessing on Euclidean lattices. In [8], a 3-step “perturbed” line path was analyzed to navigate over the Voronoi graph of the lattice, where lattice points are connected if their associated Voronoi cells touch in a facet.

In the current work, we show a strikingly close analogy between analyzing the number of intersections of a random straight line path with a Voronoi tiling of space and the intersections of a shadow simplex path with the normal fan of a polyhedron. This unexpected connection makes us hopeful that these ideas may have even broader applicability.

3 Notation and Definitions

For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we let $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i$ denote their inner product. We let $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ denote the Euclidean norm, $\mathcal{B}_2^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq 1\}$ the unit ball, and $\mathbb{S}^{n-1} = \partial\mathcal{B}_2^n$ the unit sphere. We denote the linear span of a set $A \subseteq \mathbb{R}^n$ by $\text{span}(A)$. We use the notation $\mathbb{I}[\mathbf{x} \in A]$ for the indicator of A , that is $\mathbb{I}[\mathbf{x} \in A]$ is 1 if $\mathbf{x} \in A$ and

0 otherwise. For a set of scalars $S \subseteq \mathbb{R}$, we write $SA = \{\mathbf{s}\mathbf{a} : \mathbf{s} \in S, \mathbf{a} \in A\}$. For two sets $A, B \subseteq \mathbb{R}^n$, we define their Minkowski sum $A + B = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A, \mathbf{b} \in B\}$. We let $d(A, B) = \inf \{\|\mathbf{x} - \mathbf{y}\| : \mathbf{x} \in A, \mathbf{y} \in B\}$, denote the Euclidean distance between A and B . For vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n$ we write $[\mathbf{a}, \mathbf{b}]$ for the closed line segment and $[\mathbf{a}, \mathbf{b})$ for the half-open line segment from \mathbf{a} to \mathbf{b} .

► **Definition 12** (Cone). A cone $\Sigma \subseteq \mathbb{R}^n$ satisfies the following three properties:

- $\mathbf{0} \in \Sigma$.
- $\mathbf{x} + \mathbf{y} \in \Sigma$ if \mathbf{x} and \mathbf{y} are in Σ .
- $\lambda\mathbf{x} \in \Sigma$ if $\mathbf{x} \in \Sigma$ and $\lambda \geq 0$.

For vectors $\mathbf{y}_1, \dots, \mathbf{y}_k \in \mathbb{R}^n$, we define the closed cone they generate as

$$\text{cone}(\mathbf{y}_1, \dots, \mathbf{y}_k) = \left\{ \sum_{i=1}^k \lambda_i \mathbf{y}_i : \lambda_i \geq 0, i \in [k] \right\}.$$

A cone is polyhedral if it can be generated by a finite number of vectors, and is simplicial if the generators are linearly independent. By convention, we let $\text{cone}(\emptyset) = \mathbf{0}$. A simplicial cone has the δ -distance property if its extreme rays satisfy the δ -distance property.³

The faces of a convex set $K \subseteq \mathbb{R}^n$ are its subsets of the form $F = \{\mathbf{x} \in K : \langle \mathbf{a}, \mathbf{x} \rangle = \beta\}$ where $\mathbf{a} \in \mathbb{R}^n$ and $\beta \in \mathbb{R}$ satisfy $\langle \mathbf{a}, \mathbf{x} \rangle \leq \beta$ for all $\mathbf{x} \in K$. Faces of co-dimension 1 are called facets. For a simplicial cone C , we note that its non-empty faces are exactly all the subcones generated by any subset of the generators of C .

A set of cones $\mathcal{T} = \{C_1, \dots, C_k\}$ is an n -dimensional *cone partition* if:

- Each $C_i \subseteq \mathbb{R}^n$, $i \in [k]$, is a closed n -dimensional cone.
- Any two cones C_i, C_j , $i \neq j$, meet in a shared face.
- The *support* of \mathcal{T} , $\text{sup}(\mathcal{T}) \stackrel{\text{def}}{=} \cup_{i \in [k]} C_i$, is a closed cone.

We say that F is a face of \mathcal{T} if it is a face of one of its contained cones. A cone partition \mathcal{T} is τ -wide if every C_i is τ -wide. It is simplicial if every C_i is simplicial. In this case, we also call \mathcal{T} a *cone triangulation*. A cone triangulation satisfies the local δ -distance property if every C_i satisfies it. We define the boundary of \mathcal{T} , $\partial\mathcal{T} = \cup_{i=1}^k \partial C_i$. We say that a cone triangulation \mathcal{T} *triangulates* a cone partition \mathcal{P} if \mathcal{T} and \mathcal{P} have the same support and every cone $C \in \mathcal{T}$ is generated by a subset of the extreme rays of some cone of \mathcal{P} . This means that \mathcal{T} partitions (“refines”) every cone of \mathcal{P} into simplicial cones.

3.1 Exponential distribution

We say that a random variable $X \in \mathbb{R}^n$ is exponentially distributed on a cone Σ if

$$\Pr[X \in S] = \int_S \zeta_\Sigma(\mathbf{x}) d\mathbf{x}$$

for every measurable $S \subseteq \mathbb{R}^n$, where $\zeta_\Sigma(\mathbf{x}) = c_\Sigma e^{-\|\mathbf{x}\|} \mathbf{I}[\mathbf{x} \in \Sigma]$. A standard computation, which we include for completeness, yields the normalizing constant and the expected norm.

► **Lemma 13.** *The normalizing constant is $c_\Sigma^{-1} = n! \text{vol}_n(\mathcal{B}_2^n \cap \Sigma)$. For X exponentially distributed on Σ , we have that $\mathbb{E}[\|X\|] = n$.*

Proof. See full paper. ◀

³ The δ -distance property is invariant under scaling, so the choice of generators of the extreme rays is irrelevant.

4 Optimization

While bounding the number of intersections of line segments $[\mathbf{c}, \mathbf{d}]$ with the facets of the normal fan of $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ is sufficient to obtain existential bounds on the diameter of P , we also need to be able to efficiently compute the corresponding pivots to obtain efficient algorithms. The following summarizes the required results, the technical details of which are found in the full version of the paper.

► **Theorem 14** (Shadow Simplex). *Let $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$ be pointed, $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$, and B an optimal basis for \mathbf{c} . If every intersection of $[\mathbf{c}, \mathbf{d}]$ with a facet F of a cone spanned by a feasible basis of P lies in the relative interior of F , the Shadow Simplex can be used to compute an optimal basis for \mathbf{d} in $O(mn^2 + Nmn)$ arithmetic operations, where N is the number of intersections of $[\mathbf{c}, \mathbf{d}]$ with some triangulation \mathcal{T} of the normal fan of P , where \mathcal{T} contains the cone spanned by the initial basis B .*

As explained in Section 2, we want to follow segments $[\mathbf{c}, \mathbf{c} + X]$, $[\mathbf{c} + X, \mathbf{d} + X]$, $[\mathbf{d} + X, \mathbf{d}]$ in the normal fan. Our intersection bounds from Theorem 10 are not quite sufficient to bound the number of steps on the first and last segments entirely. This is easily dealt with for the first segment, because we can control the initial objective function \mathbf{c} so that it lies deep in the initial normal cone.

For the final segment, we follow the approach of Eisenbrand and Vempala [12], who showed that if A satisfies the *global* δ -distance property, then an optimal facet for \mathbf{d} can be derived from a basis that is optimal for some $\tilde{\mathbf{d}}$ with $\|\mathbf{d} - \tilde{\mathbf{d}}\| \leq \frac{\delta}{n}$. Recursion can then be used on a problem of reduced dimension to move from $\tilde{\mathbf{d}}$ to \mathbf{d} . We strengthen their result (thereby answering a question left open by [12]) and show that the *local* δ -distance property is sufficient to get the same result as long as $\|\mathbf{d} - \tilde{\mathbf{d}}\| \leq \frac{\delta}{n^2}$.⁴

► **Lemma 15.** *Let $\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathbb{S}^{n-1}$ be a set of vectors. Then the following are equivalent:*

1. $\mathbf{x}_1, \dots, \mathbf{x}_m$ satisfy the δ -distance property.
2. $\forall I \subseteq [m]$ for which $\{\mathbf{x}_i : i \in I\}$ are linearly independent and $\forall (a_i \in \mathbb{R} : i \in I)$

$$\left\| \sum_{i \in I} a_i \mathbf{x}_i \right\| \geq \delta \max_{i \in I} |a_i| .$$

Proof. See full paper. ◀

► **Definition 16.** Let F be a face of a cone triangulation \mathcal{T} and let \mathbf{x} be a vector in the support of \mathcal{T} . Let $G = \text{cone}(\mathbf{x}_1, \dots, \mathbf{x}_k)$, $\|\mathbf{x}_i\| = 1$, be the minimal face of \mathcal{T} that contains \mathbf{x} and consider the unique conic combination $\mathbf{x} = \lambda_1 \mathbf{x}_1 + \dots + \lambda_k \mathbf{x}_k$. We define

$$\alpha_F(\mathbf{x}) := \sum_{i: \mathbf{x}_i \notin F} \lambda_i .$$

In particular, $\alpha_F(\mathbf{x}) \geq 1$ if \mathbf{x} is a unit vector and the minimal face containing it is disjoint from F , and $\alpha_F(\mathbf{x}) = 0$ if $\mathbf{x} \in F$.

► **Lemma 17.** *Let F be a cone of an n -dimensional cone triangulation \mathcal{T} satisfying the local δ -distance property. Let \mathbf{x} be a point in the support of \mathcal{T} . Then $d(\mathbf{x}, F) \geq \alpha_F(\mathbf{x}) \cdot \frac{\delta}{n}$.*

⁴ In the final bound, the loss of a factor n here disappears inside a logarithm.

Proof. Let $\mathbf{y} \in F$ be the (unique) point with $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{x}, F)$. Note that by convexity, the segment $[\mathbf{x}, \mathbf{y}]$ is contained in the support of \mathcal{T} . By considering the cones of \mathcal{T} that contain points on the segment $[\mathbf{x}, \mathbf{y}]$, we obtain a sequence of points

$$\mathbf{x} = \mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_r = \mathbf{y}$$

on the segment $[\mathbf{x}, \mathbf{y}]$ and (full-dimensional) cones G_1, \dots, G_r such that

$$G_i \cap [\mathbf{x}, \mathbf{y}] = [\mathbf{x}_{i-1}, \mathbf{x}_i].$$

Since $\alpha_F(\mathbf{y}) = 0$, the result of the lemma follows immediately from the claim that

$$d(\mathbf{x}_{i-1}, \mathbf{x}_i) \geq |\alpha_F(\mathbf{x}_{i-1}) - \alpha_F(\mathbf{x}_i)| \cdot \frac{\delta}{n},$$

which we will now prove.

Fix some $G_i = \text{cone}(\mathbf{y}_1, \dots, \mathbf{y}_n)$. By relabelling, we may assume that $\text{cone}(\mathbf{y}_1, \dots, \mathbf{y}_k) = G_i \cap F$ (since G_i and F are both faces of \mathcal{T}), for some $0 \leq k \leq n$ (if $k = 0$ then $G_i \cap F = \{\mathbf{0}\}$).

For every $\mathbf{z} \in G_i$, the minimal cone containing \mathbf{z} is a face of G_i . Therefore, using the unique conic combination $\mathbf{z} = \sum_{i=1}^n \lambda_i \mathbf{y}_i$, we have that $\alpha_F(\mathbf{z}) = \sum_{k < i \leq n} \lambda_i$.

Writing $\mathbf{x}_{i-1} = \sum_{i=1}^n a_i \mathbf{y}_i$ and $\mathbf{x}_i = \sum_{i=1}^n b_i \mathbf{y}_i$, by Lemma 15 we have that

$$\begin{aligned} d(\mathbf{x}_{i-1}, \mathbf{x}_i) &\geq \delta \max_{1 \leq i \leq n} |a_i - b_i| \geq \delta \max_{k < i \leq n} |a_i - b_i| \geq \frac{\delta}{n} \sum_{k < i \leq n} |a_i - b_i| \\ &\geq \frac{\delta}{n} |\alpha_F(\mathbf{x}_{i-1}) - \alpha_F(\mathbf{x}_i)|, \end{aligned}$$

which completes the proof of the claim. \blacktriangleleft

► Lemma 18. *Let F be a cone of a triangulation \mathcal{T} satisfying the local δ -distance property and let \mathbf{x} be a point in the support of \mathcal{T} with $d(\mathbf{x}, F) \leq \frac{\delta}{n^2}$. Let $G = \text{cone}(\mathbf{x}_1, \dots, \mathbf{x}_n)$, $\|\mathbf{x}_i\| = 1$, be a cone of \mathcal{T} containing \mathbf{x} and let*

$$\mathbf{x} = \lambda_1 \mathbf{x}_1 + \dots + \lambda_n \mathbf{x}_n$$

be the corresponding conic combination. Then for every $i \in [n]$ with $\lambda_i > \frac{1}{n}$ one has $\mathbf{x}_i \in F$.

Proof. Suppose there is some i with $\lambda_i > \frac{1}{n}$ and $\mathbf{x}_i \notin F$. Then $\alpha_F(\mathbf{x}) > \frac{1}{n}$ and by Lemma 17 we get $d(\mathbf{x}, F) > \frac{\delta}{n^2}$, which is a contradiction. \blacktriangleleft

For the recursion on a facet, we let $\pi_i(\mathbf{x}) := \mathbf{x} - \frac{\langle \mathbf{x}, \mathbf{a}_i \rangle}{\langle \mathbf{a}_i, \mathbf{a}_i \rangle} \mathbf{a}_i$ be the orthogonal projection onto the subspace orthogonal to \mathbf{a}_i and we let F_i be the facet of P defined by $\langle \mathbf{a}_i, \mathbf{x} \rangle = \mathbf{b}_i$.

► Lemma 19. *Let $\mathbf{v}_1, \dots, \mathbf{v}_k \in \mathbb{R}^n$ be linearly independent vectors that satisfy the δ -distance property and let π be the orthogonal projection onto the subspace orthogonal to \mathbf{v}_k . Then $\pi(\mathbf{v}_1), \dots, \pi(\mathbf{v}_{k-1})$ satisfy the δ -distance property.*

Proof. See full paper. \blacktriangleleft

This Lemma, which was already used by [12], implies that if P satisfies the local δ -distance property then so does F_i , where the definition of local δ -distance is understood relative to the affine hull of F_i ,⁵ because the normal vectors of F_i arise from orthogonal projections of the normal vectors of P .

⁵ Alternatively, one can apply a rotation and translation so that F_i lies in the subspace \mathbb{R}^{n-1} spanned by the first $n-1$ coordinates. The rotation does not affect the δ -distance property, and we can then treat F_i as a polytope in \mathbb{R}^{n-1} .

Input: polytope $P = \{\mathbf{x} \in \mathbb{R}^n : A\mathbf{x} \leq \mathbf{b}\}$, $\delta > 0$, feasible basis B , $\mathbf{d} \in \mathbb{R}^n$
Output: optimal basis $B \subset [m]$ for \mathbf{d}
 $\mathbf{c} \leftarrow \sum_{i \in B} \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|}$, $\mathbf{d} \leftarrow 2 \frac{\mathbf{d}}{\|\mathbf{d}\|}$
 Sample $X \in \mathbb{R}^n$ from the exponential distribution conditioned on $\|X\| \leq 2n$
 Follow segments $[\mathbf{c}, \mathbf{c} + X]$, $[\mathbf{c} + X, \mathbf{d} + X]$, $[\mathbf{d} + X, \mathbf{d} + \frac{\delta}{2n^3}X]$ using Shadow Simplex
 Find λ_i such that $\tilde{\mathbf{d}} := \mathbf{d} + \frac{\delta}{2n^3}X = \sum_{i \in B} \lambda_i \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|}$ where B is the current basis
 Choose i^* such that $\lambda_{i^*} > \frac{1}{n}$
 $B' \leftarrow$ optimal basis of F_{i^*} for $\pi_{i^*}(\mathbf{d})$, obtained by recursion starting at $B \setminus \{i^*\}$
return $B' \cup \{i^*\}$

Algorithm 1: Optimization

► **Theorem 20.** *If P satisfies the local δ -distance property, then Algorithm 1 correctly computes an optimal basis for \mathbf{d} using an expected $O((n^3/\delta) \ln(n/\delta))$ shadow simplex pivots.*

Proof. For correctness, let \mathcal{T} be some triangulation of the normal fan of P and let C be a cone in \mathcal{T} that contains \mathbf{d} . We have $\|\frac{\delta}{2n^3}X\| \leq \frac{\delta}{n^2}$ and therefore $d(\tilde{\mathbf{d}}, C) \leq d(\tilde{\mathbf{d}}, \mathbf{d}) \leq \frac{\delta}{n^2}$. Furthermore, $\|\tilde{\mathbf{d}}\| \geq \|\mathbf{d}\| - \frac{\delta}{n^2} > 1$ implies that $\sum_{i \in B} \lambda_i > 1$ so that there is some i with $\lambda_i > \frac{1}{n}$. Applying Lemma 18 yields that \mathbf{a}_{i^*} is a generator of C , which means that i^* is contained in some optimal basis for \mathbf{d} . This implies that recursion on F_{i^*} yields the correct result.

In order to bound the number of pivots, let C be the cone of the initial basis and observe that $\mathbf{c} + \delta B_2^n \subseteq C$ by the proof of Lemma 5. Hence the segment $[\mathbf{c} + \frac{\delta}{2n}X]$ does not cross a facet of the triangulation \mathcal{T}_1 of the normal fan that is implicitly used by the first leg of the shadow simplex path.

If X were exponentially distributed (without the conditioning on $\|X\| \leq 2n$), Theorem 10 together with Lemma 5 would bound the expected number of pivot steps along the three segments by

$$\mathbb{E}[N] \leq \frac{2n^2}{\delta} \ln \frac{2n}{\delta} + \frac{n\|\mathbf{d} - \mathbf{c}\|}{\delta} + \frac{2n^2}{\delta} \ln \frac{2n^3}{\delta} \leq O\left(\frac{n^2}{\delta} \ln\left(\frac{n}{\delta}\right)\right)$$

Since $\mathbb{E}[\|X\|] = n$ we have $\Pr[\|X\| \leq 2n] \geq \frac{1}{2}$ by Markov's inequality and therefore

$$\mathbb{E}[N \mid \|X\| \leq 2n] \leq 2\mathbb{E}[N] \leq O\left(\frac{n^2}{\delta} \ln\left(\frac{n}{\delta}\right)\right).$$

The bound on the total expected number of pivot steps follows from the depth n of recursion. ◀

5 Intersection Bounds and Diameter Bounds

► **Lemma 21.** *Let C be a polyhedral cone containing $\mathbf{u} + \tau B_2^n$, where $\|\mathbf{u}\| = 1$. Let $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ and let $X \in \mathbb{R}^n$ be exponentially distributed on a full dimensional cone $\Sigma \ni \mathbf{u}$. Then the expected number of times the shifted line segment $[\mathbf{c} + X, \mathbf{d} + X]$ hits the boundary of C is at most*

$$\mathbb{E}[|\partial C \cap [\mathbf{c} + X, \mathbf{d} + X]|] \leq \frac{\|\mathbf{d} - \mathbf{c}\|}{\tau} \int_0^1 \int_{(C - ((1-\lambda)\mathbf{c} + \lambda\mathbf{d})) \cap \Sigma} \zeta_\Sigma(\mathbf{x}) d\mathbf{x} d\lambda$$

Proof. Let F be a facet of C . Note that with probability 1, the line segment $[\mathbf{c} + X, \mathbf{d} + X]$ passes through F at most once. By linearity, we see that

$$\mathbb{E}[|\partial C \cap [\mathbf{c} + X, \mathbf{d} + X]|] = \sum_{F \text{ facet of } C} \Pr[(F \cap [\mathbf{c} + X, \mathbf{d} + X]) \neq \emptyset]. \quad (1)$$

We now bound the crossing probability for any facet F .

We first calculate the hitting probability as

$$\begin{aligned} \Pr[F \cap [\mathbf{c} + X, \mathbf{d} + X] \neq \emptyset] &= \Pr[X \in -[\mathbf{c}, \mathbf{d}] + F] \\ &= \int_{-[\mathbf{c}, \mathbf{d}] + F} \zeta_{\Sigma}(\mathbf{x}) d\mathbf{x} \\ &= |\langle \mathbf{n}, \mathbf{d} - \mathbf{c} \rangle| \int_0^1 \int_{F - ((1-\lambda)\mathbf{c} + \lambda\mathbf{d})} \zeta_{\Sigma}(\mathbf{x}) d\text{vol}_{n-1}(\mathbf{x}) d\lambda \\ &\leq \|\mathbf{d} - \mathbf{c}\| \int_0^1 \int_{(F - ((1-\lambda)\mathbf{c} + \lambda\mathbf{d})) \cap \Sigma} c_{\Sigma} e^{-\|\mathbf{x}\|} d\text{vol}_{n-1}(\mathbf{x}) d\lambda \quad (2) \end{aligned}$$

where $\mathbf{n} \in \mathbb{R}^n$ is a unit normal vector to F and we use $d\text{vol}_{n-1}(\mathbf{x})$ to indicate an integral with respect to the usual $(n-1)$ -dimensional measure on the affine hyperplane spanned by the integration domain. Bounding the hitting probability therefore boils down to bounding the measure of a shift of the facet F . Letting $h = |\langle \mathbf{n}, \mathbf{u} \rangle| \geq \tau$ (which holds by assumption on \mathbf{u}), for any shift $\mathbf{t} \in \mathbb{R}^n$ we have that

$$\begin{aligned} \int_{(F + \mathbf{t} + \text{cone}(\mathbf{u})) \cap \Sigma} e^{-\|\mathbf{x}\|} d\mathbf{x} &\geq \int_{((F + \mathbf{t}) \cap \Sigma) + \text{cone}(\mathbf{u})} e^{-\|\mathbf{x}\|} d\mathbf{x} \quad (\text{since } \mathbf{u} \in \Sigma) \\ &= \int_0^{\infty} \int_{((F + \mathbf{t}) \cap \Sigma) + \frac{r}{h}\mathbf{u}} e^{-\|\mathbf{x}\|} d\text{vol}_{n-1}(\mathbf{x}) dr \\ &= \int_0^{\infty} \int_{(F + \mathbf{t}) \cap \Sigma} e^{-\|\mathbf{x} + \frac{r}{h}\mathbf{u}\|} d\text{vol}_{n-1}(\mathbf{x}) dr \\ &\geq \int_0^{\infty} e^{-r/h} dr \int_{(F + \mathbf{t}) \cap \Sigma} e^{-\|\mathbf{x}\|} d\text{vol}_{n-1}(\mathbf{x}) \\ &\geq \tau \int_{(F + \mathbf{t}) \cap \Sigma} e^{-\|\mathbf{x}\|} d\text{vol}_{n-1}(\mathbf{x}) \quad (3) \end{aligned}$$

The lemma now follows by combining (1),(2),(3), using the fact that the $F + \text{cone}(\mathbf{u})$ partition the cone C up to sets of measure 0. \blacktriangleleft

► Lemma 22. Let $\mathcal{T} = (C_1, \dots, C_k)$ be a partition of a cone Σ into polyhedral τ -wide cones. Let $\mathbf{c}, \mathbf{d} \in \mathbb{R}^n$ and let $X \in \mathbb{R}^n$ be exponentially distributed on Σ . Then the expected number of facets hit by the shifted line segment $[\mathbf{c} + X, \mathbf{d} + X]$ satisfies

$$\mathbb{E}[|\partial \mathcal{T} \cap [\mathbf{c} + X, \mathbf{d} + X]|] \leq \frac{\|\mathbf{d} - \mathbf{c}\|}{\tau}.$$

Proof. Using Lemma 21, we bound

$$\begin{aligned} \mathbb{E}[|\partial\mathcal{T} \cap [\mathbf{c} + X, \mathbf{d} + X]|] &\leq \sum_{i=1}^k \mathbb{E}[|\partial C_i \cap [\mathbf{c} + X, \mathbf{d} + X]|] \\ &\leq \sum_{i=1}^k \frac{\|\mathbf{d} - \mathbf{c}\|}{\tau} \int_0^1 \int_{(C_i - ((1-\lambda)\mathbf{c} + \lambda\mathbf{d})) \cap \Sigma} \zeta_\Sigma(\mathbf{x}) d\mathbf{x} d\lambda \\ &\leq \frac{\|\mathbf{d} - \mathbf{c}\|}{\tau} \int_0^1 \int_\Sigma \zeta_\Sigma(\mathbf{x}) d\mathbf{x} d\lambda \\ &\leq \frac{\|\mathbf{d} - \mathbf{c}\|}{\tau}, \end{aligned}$$

as needed.

For the furthermore, note that each intersection is overcounted twice in the summation above, since each facet belongs to exactly two cones in the partition. ◀

We will need the following simple lemma about the exponential distribution.

► **Lemma 23.** *Let Y be exponentially distributed on \mathbb{R}_+ . Then for any $c \in \mathbb{R}$, $\mathbb{E}[|Y - c|] \geq |c|/2$.*

Proof. See full paper. ◀

While we could choose \mathbf{c} and \mathbf{d} such that $\mathbf{c} + X$ and $\mathbf{d} + X$ lie in the same cones as \mathbf{c} and \mathbf{d} with high probability, and hence no facets are hit by the segments $[\mathbf{c}, \mathbf{c} + X]$ and $[\mathbf{d}, \mathbf{d} + X]$, this would require us to choose $\|\mathbf{d} - \mathbf{c}\|$ quite large. We will show a better way to bound the number of facets that are hit by the segment $[\mathbf{c}, \mathbf{c} + X]$.

► **Lemma 24.** *Let $C \subseteq \mathbb{R}^n$ be a polyhedral cone containing $\mathbf{u} + \tau\mathcal{B}_2^n$, where $\|\mathbf{u}\| = 1$. Let $\mathbf{c} \in \mathbb{R}^n$ and let $X \in \mathbb{R}^n$ be exponentially distributed on a cone $\Sigma \ni \mathbf{u}$. Then for every $\alpha \in (0, 1)$ we have*

$$\mathbb{E}[|\partial C \cap [\mathbf{c} + \alpha X, \mathbf{c} + X]|] \leq \frac{2}{\tau} \int_1^{1/\alpha} \frac{1}{s} \int_{(C - s\mathbf{c}) \cap \Sigma} \|\mathbf{x}\| \zeta_\Sigma(\mathbf{x}) d\mathbf{x} ds$$

Proof. As in the proof of Lemma 21, we will decompose the expectation over the facets of C , where we have

$$\mathbb{E}[|\partial C \cap [\mathbf{c} + \alpha X, \mathbf{c} + X]|] = \sum_{F \text{ facet of } C} \Pr[F \cap [\mathbf{c} + \alpha X, \mathbf{c} + X] \neq \emptyset] \tag{4}$$

Take a facet F of C and let \mathbf{n} denote a unit normal to F pointing in the direction of the cone (i.e., $\langle \mathbf{n}, \mathbf{u} \rangle > 0$).

$$\begin{aligned} \Pr[F \cap [\mathbf{c} + \alpha X, \mathbf{c} + X] \neq \emptyset] &= \Pr[X \in [1, 1/\alpha](F - \mathbf{c})] \\ &= \int_1^{1/\alpha} \int_{(F - s\mathbf{c}) \cap \Sigma} |\langle \mathbf{n}, \mathbf{c} \rangle| \zeta_\Sigma(\mathbf{x}) d\text{vol}_{n-1}(\mathbf{x}) ds \\ &= \int_1^{1/\alpha} \frac{1}{s} \int_{(F - s\mathbf{c}) \cap \Sigma} |\langle \mathbf{n}, s\mathbf{c} \rangle| c_\Sigma e^{-\|\mathbf{x}\|} d\text{vol}_{n-1}(\mathbf{x}) ds. \end{aligned} \tag{5}$$

Again, we have to bound an integral over a shifted facet, similar to the proof of Lemma 21. Letting $h = |\langle \mathbf{n}, \mathbf{u} \rangle| \geq \tau$, we have that

$$\begin{aligned}
 \int_{(F+\mathbf{t}+\text{cone}(\mathbf{u}))\cap\Sigma} \|\mathbf{x}\|e^{-\|\mathbf{x}\|}d\mathbf{x} &\geq \int_{((F+\mathbf{t})\cap\Sigma)+\text{cone}(\mathbf{u})} \|\mathbf{x}\|e^{-\|\mathbf{x}\|}d\mathbf{x} \quad (\text{since } \mathbf{u} \in \Sigma) \\
 &= \int_0^\infty \int_{((F+\mathbf{t})\cap\Sigma)+\frac{r}{h}\mathbf{u}} \|\mathbf{x}\|e^{-\|\mathbf{x}\|}d\text{vol}_{n-1}(\mathbf{x})dr \\
 &= \int_0^\infty \int_{(F+\mathbf{t})\cap\Sigma} \|\mathbf{x} + \frac{r}{h}\mathbf{u}\|e^{-\|\mathbf{x}+\frac{r}{h}\mathbf{u}\|}d\text{vol}_{n-1}(\mathbf{x})dr \\
 &\geq \int_0^\infty \int_{(F+\mathbf{t})\cap\Sigma} |\langle \mathbf{n}, \mathbf{x} + \frac{r}{h}\mathbf{u} \rangle|e^{-r/h}e^{-\|\mathbf{x}\|}d\text{vol}_{n-1}(\mathbf{x})dr \\
 &= h^2 \int_0^\infty |\langle \mathbf{n}, \mathbf{t} \rangle/h + s|e^{-s}ds \int_{(F+\mathbf{t})\cap\Sigma} e^{-\|\mathbf{x}\|}d\text{vol}_{n-1}(\mathbf{x}) \\
 &\geq \frac{h}{2} \int_{(F+\mathbf{t})\cap\Sigma} |\langle \mathbf{n}, \mathbf{t} \rangle|e^{-\|\mathbf{x}\|}d\text{vol}_{n-1}(\mathbf{x}) \quad (\text{by Lemma 23}) \\
 &\geq \frac{\tau}{2} \int_{(F+\mathbf{t})\cap\Sigma} |\langle \mathbf{n}, \mathbf{t} \rangle|e^{-\|\mathbf{x}\|}d\text{vol}_{n-1}(\mathbf{x})
 \end{aligned} \tag{6}$$

The Lemma now follows by combining (4),(5),(6). ◀

► **Lemma 25.** *Let $\mathcal{T} = (C_1, \dots, C_k)$ be partition of a cone Σ into polyhedral τ -wide cones. Let $\mathbf{c} \in \mathbb{R}^n$ and $\alpha \in (0, 1)$ be fixed and let $X \in \mathbb{R}^n$ be exponentially distributed over Σ . Then*

$$\mathbb{E}[|\partial\mathcal{T} \cap [\mathbf{c} + \alpha X, \mathbf{c} + X]|] \leq \frac{2n}{\tau} \ln \frac{1}{\alpha} .$$

Proof. By Lemmas 13 and 24, we have that

$$\begin{aligned}
 \mathbb{E}[|\partial\mathcal{T} \cap [\mathbf{c} + \alpha X, \mathbf{c} + X]|] &\leq \sum_{i=1}^k \mathbb{E}[|\partial C_i \cap [\mathbf{c} + \alpha X, \mathbf{c} + X]|] \\
 &\leq \frac{2}{\tau} \sum_{i=1}^k \int_1^{1/\alpha} \frac{1}{s} \int_{(C_i-s\mathbf{c})\cap\Sigma} \|\mathbf{x}\|\zeta_\Sigma(\mathbf{x})d\mathbf{x}ds \\
 &\leq \frac{2}{\tau} \int_1^{1/\alpha} \frac{1}{s} \int_\Sigma \|\mathbf{x}\|\zeta_\Sigma(\mathbf{x})d\mathbf{x}ds \\
 &\leq \frac{2}{\tau} \int_1^{1/\alpha} \frac{1}{s} \mathbb{E}[|X|]ds = \frac{2n}{\tau} \ln \frac{1}{\alpha} \quad \blacktriangleleft
 \end{aligned}$$

Acknowledgements. We would like to thank Friedrich Eisenbrand and Santosh Vempala for useful discussions and an anonymous referee for valuable remarks.

References

- 1 Karim Alexander Adiprasito and Bruno Benedetti. The Hirsch conjecture holds for normal flag complexes. Arxiv Report 1303.3598, 2014.
- 2 M. L. Balinski. The Hirsch conjecture for dual transportation polyhedra. *Math. Oper. Res.*, 9(4):629–633, 1984.
- 3 David Barnette. An upper bound for the diameter of a polytope. *Discrete Math.*, 10:9–13, 1974.

- 4 Nicolas Bonifas, Marco Di Summa, Friedrich Eisenbrand, Nicolai Hähnle, and Martin Niemeier. On sub-determinants and the diameter of polyhedra. *Discrete Comput. Geom.*, 52(1):102–115, 2014. Preliminary version in SOCG 12.
- 5 Karl-Heinz Borgwardt. *The simplex method: A probabilistic analysis*, volume 1 of *Algorithms and Combinatorics: Study and Research Texts*. Springer-Verlag, Berlin, 1987.
- 6 Graham Brightwell, Jan van den Heuvel, and Leen Stougie. A linear bound on the diameter of the transportation polytope. *Combinatorica*, 26(2):133–139, 2006.
- 7 Tobias Brunsch and Heiko Röglin. Finding short paths on polytopes by the shadow vertex algorithm. In *Automata, languages, and programming. Part I*, volume 7965 of *Lecture Notes in Comput. Sci.*, pages 279–290. Springer, Heidelberg, 2013.
- 8 Daniel Dadush and Nicolas Bonifas. Short paths on the voronoi graph and closest vector problem with preprocessing. 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 295–314. SIAM, 2015.
- 9 Daniel Dadush and Nicolai Hähnle. On the shadow simplex method for curved polyhedra (draft of full paper). Arxiv Report 1412.6705, 2014.
- 10 Jesús A. De Loera, Edward D. Kim, Shmuel Onn, and Francisco Santos. Graphs of transportation polytopes. *J. Combin. Theory Ser. A*, 116(8):1306–1325, 2009.
- 11 Martin Dyer and Alan Frieze. Random walks, totally unimodular matrices, and a randomised dual simplex algorithm. *Math. Programming*, 64(1, Ser. A):1–16, 1994.
- 12 Friedrich Eisenbrand and Santosh Vempala. Geometric random edge. Arxiv Report 1404.1568, 2014.
- 13 Gil Kalai. The diameter of graphs of convex polytopes and f -vector theory. In *Applied geometry and discrete mathematics*, volume 4 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 387–411. Amer. Math. Soc., Providence, RI, 1991.
- 14 Gil Kalai and Daniel J. Kleitman. A quasi-polynomial bound for the diameter of graphs of polyhedra. *Bull. Amer. Math. Soc. (N.S.)*, 26(2):315–316, 1992.
- 15 D. G. Larman. Paths of polytopes. *Proc. London Math. Soc. (3)*, 20:161–178, 1970.
- 16 Benjamin Matschke, Francisco Santos, and Christophe Weibel. The width of 5-dimensional prismatoids. Arxiv Report 1202.4701, 2013.
- 17 Denis Naddef. The Hirsch conjecture is true for $(0, 1)$ -polytopes. *Math. Programming*, 45(1, Ser. B):109–110, 1989.
- 18 Francisco Santos. A counterexample to the Hirsch conjecture. *Ann. of Math. (2)*, 176(1):383–412, 2012.
- 19 Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. *J. ACM*, 51(3):385–463 (electronic), 2004.
- 20 Michael J. Todd. An improved Kalai-Kleitman bound for the diameter of a polyhedron. Arxiv Report 1402.3579, 2014.
- 21 Roman Vershynin. Beyond Hirsch conjecture: walks on random polytopes and smoothed complexity of the simplex method. *SIAM J. Comput.*, 39(2):646–678, 2009.

Pattern Overlap Implies Runaway Growth in Hierarchical Tile Systems

Ho-Lin Chen¹, David Doty², Ján Maňuch^{3,4}, Arash Rafiey^{4,5}, and Ladislav Stacho⁴

- 1 National Taiwan University
Taipei, Taiwan
holinc@gmail.com
- 2 California Institute of Technology
Pasadena, CA, USA
ddoty@caltech.edu
- 3 University of British Columbia
Vancouver, BC, Canada
jmanuch@cs.ubc.ca
- 4 Simon Fraser University
Burnaby, BC, Canada
{jmanuch, arashr, lstacho}@sfu.ca
- 5 Indiana State University, IN, USA

Abstract

We show that in the hierarchical tile assembly model, if there is a producible assembly that overlaps a nontrivial translation of itself consistently (i.e., the pattern of tile types in the overlap region is identical in both translations), then arbitrarily large assemblies are producible. The significance of this result is that tile systems intended to controllably produce finite structures must avoid pattern repetition in their producible assemblies that would lead to such overlap.

This answers an open question of Chen and Doty (*SODA 2012*), who showed that so-called “partial-order” systems producing a unique finite assembly **and** avoiding such overlaps must require time linear in the assembly diameter. An application of our main result is that any system producing a unique finite assembly is automatically guaranteed to avoid such overlaps, simplifying the hypothesis of Chen and Doty’s main theorem.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases self-assembly, hierarchical, pumping

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.360

1 Introduction

Winfree’s abstract Tile Assembly Model (aTAM) [23] is a model of crystal growth through cooperative binding of square-like monomers called *tiles*, implemented experimentally (for the current time) by DNA [2, 25]. It models the potentially algorithmic capabilities of tiles that are designed to bind if and only if the total strength of attachment (summed over all binding sites, called *glues* on the tile) is at least a threshold τ , sometimes called the *temperature*. When glue strengths are integers and $\tau = 2$, two strength 1 glues must cooperate to bind the tile to a growing assembly. Two assumptions are key: 1) growth starts from a single *seed* tile type, and 2) only individual tiles bind to an assembly. We refer to this model as the *seeded aTAM*.



© Ho-Lin Chen, David Doty, Ján Maňuch, Arash Rafiey, and Ladislav Stacho;
licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG’15).

Editors: Lars Arge and János Pach; pp. 360–373



Leibniz International Proceedings in Informatics

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

While violations of these assumptions are often viewed as errors in implementation of the seeded aTAM [20, 21], relaxing them results in a different model with its own programmable abilities. In the *hierarchical* (a.k.a. *multiple tile* [1], *polyomino* [15, 24], *two-handed* [3, 6, 9]) *aTAM*, there is no seed tile, and an assembly is considered producible so long as two producible assemblies are able to attach to each other with strength at least τ , with all individual tiles being considered as “base case” producible assemblies. In either model, an assembly is considered *terminal* if nothing can attach to it; viewing self-assembly as a computation, terminal assembly(ies) are often interpreted to be the output. See [7, 17] for an introduction to recent theoretical work using these models.

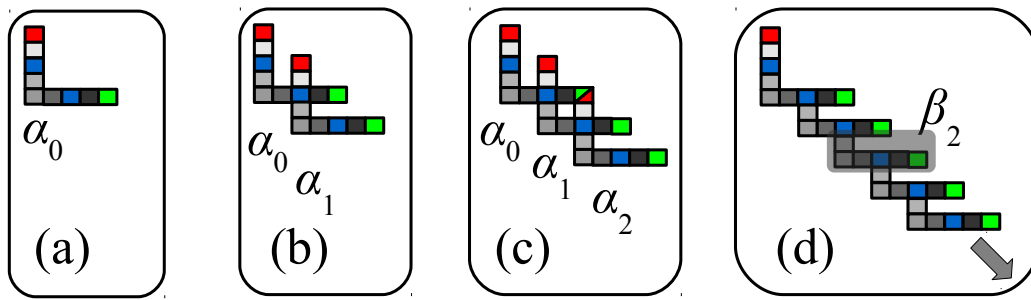
As with other models of computation, in general it is considerably more difficult to prove negative results (*limitations* on what a tile system can do) than to prove positive results. A common line of inquiry aimed at negative results in tile self-assembly concerns the notion of “pumping”: showing that a single repetition of a certain group of tiles implies that the same group can be repeated indefinitely to form an infinite periodic structure.

The *temperature-1* problem in the seeded model of tile assembly concerns the abilities of tile systems in which every positive-strength glue is sufficiently strong to bind two tiles. It may seem “obvious” that if two tile types repeat in an assembly, then a segment of tiles connecting them could be repeated indefinitely (“pumped”) to produce an infinite periodic path (since, at temperature 1, each tile along the segment has sufficient strength for the next tile in the segment to attach). However, this argument fails if the attempt to pump the segment “crashes” into an existing part of the assembly. It is conjectured [10] that only finite unions of periodic patterns (so-called *semilinear* sets) can be assembled at temperature 1 in the seeded model, but despite considerable investigation [11, 16, 18, 19], the question remains open. If true, temperature-1 *hierarchical* tile systems would suffer a similar limitation, due to a formal connection between producible assemblies in the seeded and hierarchical models [3, Lemma 4.1]. It *has* been established, using pumping arguments, that temperature-1 seeded tile systems are unable to simulate the dynamics of certain temperature-2 systems [11].

Moving to temperature 2, both models gain power to assemble much more complex structures; both are able to simulate Turing machines, for instance. In a certain sense, the hierarchical model is at least as powerful as the seeded model, since every seeded tile system can be simulated by a hierarchical tile system with a small “resolution loss”: each tile in the seeded system is represented by a 5×5 block of tiles in the hierarchical system [3, Theorem 4.2].

From this perspective, the main theorem of this paper, a negative result on hierarchical tile assembly that does not apply to seeded tile assembly, is somewhat surprising. We show that hierarchical systems, of *any* temperature, are forced to admit a sort of infinite “pumping” behavior if a special kind of “pattern repetition” occurs. More formally, suppose that a hierarchical tile system \mathcal{T} is able to produce an assembly α_0 such that, for some nonzero vector \vec{v} , the assembly $\alpha_1 = \alpha_0 + \vec{v}$ (meaning α_0 translated by \vec{v}) intersects α_0 , but the two translations agree on every tile type in the intersection (they are *consistent*). It is known that this implies that the union $\alpha_0 \cup \alpha_1$ is producible as well [8, Theorem 5.1]. Our main theorem, Theorem 11, shows that this condition implies that \mathcal{T} can produce arbitrarily large assemblies, answering the main open question of [8].

The assembly is not necessarily infinitely many translations of all of α_0 , since although α_0 and α_1 are consistent, which implies that α_1 must be consistent with $\alpha_2 = \alpha_0 + 2\vec{v}$, it may be that α_0 is not consistent with α_2 . However, our proof shows that a subassembly β_2 of α_2 can be assembled that is sufficient to grow another translated copy of β_2 , so that the infinite producible assembly consists of infinitely many translations of β_2 . See Figure 1 for an example illustration.



■ **Figure 1** Example of the main theorem of this paper. (a) A producible assembly α_0 . Gray tiles are all distinct types from each other, but red, green, and blue each represent one of three different tile types, so the two blue tiles are the same type. (b) By Theorem 9, $\alpha_0 \cup \alpha_1$ is producible, where $\alpha_1 = \alpha_0 + (2, -2)$, because they overlap in only one position, and they both have the blue tile type there. (c) α_0 and α_2 both have a tile at the same position, but the types are different (green in the case of α_0 and red in the case of α_2). (d) However, a subassembly β_i of each new α_i can grow, enough to allow the translated equivalent subassembly β_{i+1} of α_{i+1} to grow from β_i , so an infinite structure is producible.

An immediate application of this theorem is to strengthen a theorem of Chen and Doty [4]. They asked whether every hierarchical tile system obeying a technical condition known as the *partial order* property and producing a unique finite terminal assembly, also obeys the condition that no producible assembly is consistent with a translation of itself. The significance of the latter condition is that the main theorem of [4] shows that systems satisfying the condition obey a time lower bound for assembly: they assemble their final structure in time $\Omega(d)$, where d is the diameter of the final assembly. Our main theorem implies that every system *not* satisfying the condition must produce arbitrarily large assemblies and therefore cannot produce a unique finite terminal assembly. Hence *all* hierarchical partial order systems are constrained by this time lower bound, the same lower bound that applies to all seeded tile systems. Thus hierarchical partial order systems, despite the ability to assemble many sub-assemblies of the final assembly in parallel, provably cannot exploit this parallelism to obtain a speedup in assembly time compared to the seeded model.

It is worthwhile to note that our main theorem does not apply to the seeded model. For instance, it is routine to design a seeded tile system that assembles a unique terminal assembly shaped like a square, which uses the same tile type in the upper right and lower left corners of the square. Translating this assembly to overlap those two positions means that this tile system satisfies the hypothesis of our main theorem. Why does this not contradict the fact that this system, like all seeded systems, can be simulated by a hierarchical tile system at scale factor 5 [3, Theorem 4.2], which would apparently satisfy the same consistent overlap condition? The answer is that the hierarchical simulating system of [3] uses different 5×5 blocks to represent the same tile type from the seeded system, depending on the *sides* of the tile that are used to bind in the seeded system. Since the upper-right corner tile and lower-left corner tile in the seeded system must clearly bind using different sides, they are represented by different blocks in the simulating hierarchical system. Hence in the hierarchical system, the terminal assembly does *not* consistently overlap with itself.

Our argument proceeds by reducing the problem (via a simple argument) to a simple-to-state theorem in pure geometry. That theorem's proof contains almost all of the technical machinery required to prove our main theorem. Let S_0 be a discrete *shape*: a finite, connected subset of \mathbb{Z}^2 , and let $\vec{v} \in \mathbb{Z}^2$ be a nonzero vector. Let $S_1 = S_0 + \vec{v} (= \{ p + \vec{v} \mid p \in S_1 \})$,

and let $S_2 = S_1 + \vec{v}$. The theorem states that $S_2 \setminus S_1$ (possibly a disconnected set) contains a connected component that does not intersect S_0 . This is clear when \vec{v} is large enough that $S_0 \cap S_2 = \emptyset$, but for the general case, we encourage the reader to attempt to prove it before concluding that it is obvious. In Figure 1, $S_2 \setminus S_1$ (referring respectively to the shapes of assemblies α_2 and α_1) contains two connected components, one on top and the other on bottom. The top component intersects S_0 , but not the bottom.

This problem is in turn reduced to a more technical statement about simple curves (continuous, one-to-one functions $\varphi : [0, 1] \rightarrow \mathbb{R}^2$) whose intersection implies the shapes theorem. Although we need the curve theorem to hold only for polygonal curves on the integer grid \mathbb{Z}^2 , the result holds for general curves and may be useful in other contexts.

2 Informal definition of the hierarchical tile assembly model

We give an informal sketch of the hierarchical variant of the abstract Tile Assembly Model (aTAM). See Section A.1 for a formal definition.

Let \mathbb{R} , \mathbb{Z} , \mathbb{N} and \mathbb{Z}^+ denote the set of all real numbers, integers, non-negative integers and positive integers, respectively. Given a set $S \subseteq \mathbb{R}^2$ and a vector $\vec{v} \in \mathbb{R}^2$, let $S + \vec{v} = \{p + \vec{v} : p \in S\}$.

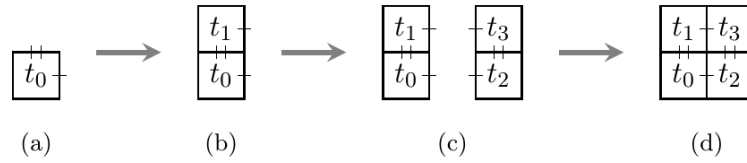
A *tile type* is a unit square with four sides, each consisting of a *glue label* (often represented as a finite string). Each glue type is assigned a nonnegative integer *strength*. We assume a finite set T of tile types, but an infinite number of copies of each tile type, each copy referred to as a *tile*. An *assembly* is a positioning of tiles on the integer lattice \mathbb{Z}^2 ; i.e., a partial function $\alpha : \mathbb{Z}^2 \dashrightarrow T$. Write $\alpha \sqsubseteq \beta$ to denote that α is a *subassembly* of β , which means that $\text{dom } \alpha \subseteq \text{dom } \beta$ and $\alpha(p) = \beta(p)$ for all points $p \in \text{dom } \alpha$. Given an assembly β and a set $D \subseteq \text{dom } \beta$, $\beta|_D$ is a subassembly of α with $\text{dom } (\beta|_D) = D$.

We abuse notation and take a tile type t to be equivalent to the single-tile assembly containing only t (at the origin if not otherwise specified). Two adjacent tiles in an assembly *interact* if the glue labels on their abutting sides are equal and have positive strength. Each assembly induces a *binding graph*, a grid graph whose vertices are tiles, with an edge between two tiles if they interact. The assembly is τ -*stable* if every cut of its binding graph has strength (the sum of the weights of the edges in the cut) at least τ , where the weight of an edge is the strength of the glue it represents.

A *hierarchical tile assembly system* (hierarchical TAS) is a pair $\mathcal{T} = (T, \tau)$, where T is a finite set of tile types and $\tau \in \mathbb{N}$ is the temperature. An assembly is *producible* if either it is a single tile from T , or it is the τ -stable result of translating two producible assemblies without overlap. The restriction on overlap is a model of a chemical phenomenon known as *steric hindrance* [22, Section 5.11] or, particularly when employed as a design tool for intentional prevention of unwanted binding in synthesized molecules, *steric protection* [13, 14, 12]. An assembly α is *terminal* if for every producible assembly β , α and β cannot be τ -stably attached. If α can grow into β by the attachment of zero or more assemblies, then we write $\alpha \rightarrow \beta$. Our definitions imply only finite assemblies are producible. Figure 2 shows an example of hierarchical attachment.

3 Main result

Section 3.1 proves a theorem about curves in \mathbb{R}^2 (Theorem 7) that contains most of the technical detail required for our main theorem. Theorem 7 states that if a finite set of simple curves $\varphi_1, \dots, \varphi_k$ do not intersect each other, and for some nonzero $\vec{v} \in \mathbb{R}^2$, for each



■ **Figure 2** Typical example of hierarchical assembly, at temperature $\tau = 2$. The segments between tiles represent the bonds, the number of segments encodes the strength of the bond (here, 1 or 2). In the seeded, single tile model with seed $\sigma = t_0$, the assembly at step (b) would be terminal.

i , there is $n \in \mathbb{Z}^+$ so that $\varphi_i(1) = \varphi_{i+1}(0) + n\vec{v}$ (where $\varphi_{k+1} = \varphi_1$, i.e., each curve ends a positive integer multiple of \vec{v} from the start of the next), then some curve φ_i intersects $\varphi_i + \vec{v}$. Section 3.2 uses Theorem 7 to prove a geometrical theorem about shapes in \mathbb{Z}^2 (Theorem 8), namely that for any shape S_0 , with $S_1 = S_0 + \vec{v}$ and $S_2 = S_0 + 2\vec{v}$, it holds that $S_2 \setminus S_1$ has a connected component that does not intersect S_0 . Section 3.3 uses Theorem 8 to prove our main theorem (Theorem 11), which is that if a tile system can produce an assembly that overlaps itself consistently, the arbitrarily large assemblies are producible.

The high-level intuition of the proofs of these results is as follows (described in reverse order). Theorem 11 intuitively holds by the following argument. If a producible assembly α_0 is consistent with its translation $\alpha_1 = \alpha + \vec{v}$ by some nonzero vector $\vec{v} \in \mathbb{Z}^2$, then Theorem 8 implies that some portion C of $\alpha_2 = \alpha_0 + 2\vec{v}$ does not intersect α_0 , and C is furthermore assemblable from α_1 (by Theorem 9). Therefore, it is assemblable from $\alpha_0 \cup \alpha_1$ (since α_2 is consistent with α_1 , and this part C of α_2 does not intersect α_0 , ruling out inconsistency due to C clashing with α_0). Thus $\alpha_1 \cup C$ is producible and overlaps consistently with its translation by \vec{v} . Since C is nonempty, $\alpha_1 \cup C$ is strictly larger than α_0 . Iterating this argument shows that arbitrarily large producible assemblies exist.

Why does Theorem 8 hold? If it did not, then every connected component C_i of $S_2 \setminus S_1$ would intersect S_0 at a point p_i . Since $p_i \in S_0$, $p_i + 2\vec{v} \in S_2$. Since $p_i \in S_2$, there is a path q_i from p_i to $p_i + 2\vec{v}$ lying entirely inside of S_2 . But Corollary 5 implies that q_i must intersect $q_i - \vec{v}$, which, being a path inside of S_1 , implies that $p_i + 2\vec{v}$ is in a different connected component C_{i+1} of $S_2 \setminus S_1$. But since C_{i+1} also intersects S_0 , there is a point p_{i+1} in this intersection, and there is a curve φ_i from $p_i + 2\vec{v}$ to p_{i+1} . Since every connected component of $S_2 \setminus S_1$ intersects S_0 , we can repeat this argument until we return to the original connected component C_i . But then the various curves φ_i defined within each component will satisfy the conditions of Theorem 7, a contradiction.

3.1 A theorem about curves

► **Definition 1.** Given a nonzero vector $\vec{v} \in \mathbb{R}^2$ and a point $p \in \mathbb{R}^2$ the \vec{v} -axis through p , denoted as $L_{\vec{v},p}$, is the line parallel to \vec{v} through p .

► **Definition 2.** Let $\varphi : [0, 1] \rightarrow \mathbb{R}^2$ be continuous one-to-one mapping. Then $\varphi([0, 1])$ is called a *simple* (non-self-intersecting) curve from $\varphi(0)$ to $\varphi(1)$. If $\varphi : [0, 1] \rightarrow \mathbb{R}^2$ is continuous with $\varphi(0) = \varphi(1)$ and one-to-one on $[0, 1)$, then $\varphi([0, 1])$ is called a *simple closed* curve.

Obviously, any curve $\varphi([0, 1])$ from $\varphi(0)$ to $\varphi(1)$ (being a subset of the plane) can be considered also as a curve from $\varphi(1)$ to $\varphi(0)$. Therefore, for the sake of brevity, we sometimes denote this curve simply by φ and say that φ connects points $\varphi(0)$, $\varphi(1)$. If $0 \leq t_1 \leq t_2 \leq 1$, then $\varphi([t_1, t_2])$ is a simple curve as well. If φ_1 and φ_2 are simple non-closed curves such that $\varphi_1 \cap \varphi_2 = \{\varphi_1(1)\} = \{\varphi_2(0)\}$ then their concatenation $\varphi_1 \oplus \varphi_2$, defined by

$(\varphi_1 \oplus \varphi_2)(t) = \varphi_1(2t)$ if $t \leq \frac{1}{2}$ and $(\varphi_1 \oplus \varphi_2)(t) = \varphi_2(2(t - \frac{1}{2}))$ otherwise, is also a simple curve (closed if $\varphi_2(1) = \varphi_1(0)$).

► **Definition 3.** Given a subset of a plane $A \subseteq \mathbb{R}^2$ and a vector $\vec{v} \in \mathbb{R}^2$, the *shift (or translation)* of A by \vec{v} , denoted by $A + \vec{v}$, is the set $A + \vec{v} = \{p + \vec{v} : p \in A\}$.

The following lemma, due to Demaine, Demaine, Fekete, Patitz, Schweller, Winslow, and Woods [5, Lemma 6.3], states that if a curve does not intersect a translation of itself, then it also does not intersect any integer multiples of the same translation. We state the lemma in terms of curves instead of shapes as in ref.[5], and for the sake of self-containment, we provide a proof stated in these terms.

► **Lemma 4 ([5]).** Consider points $p_1, p_2 \in \mathbb{R}^2$, nonzero vector $\vec{v} \in \mathbb{R}^2$ and a simple curve φ connecting p_1 and p_2 (φ may be closed if $p_1 = p_2$) such that $\varphi \cap (\varphi + \vec{v}) = \emptyset$. Let $\varphi^{\rightarrow k} = \varphi + k\vec{v}$, $k \in \mathbb{Z}$. Then all $\varphi^{\rightarrow k}$'s are mutually disjoint.

Proof. To every point of φ we can assign “relative distance” d from the line $L_{\vec{v}, p_1}$ —positive for points left to the line and negative for points right to the line (with respect to \vec{v}). Since the function $d \circ \varphi : [0, 1] \rightarrow \mathbb{R}$ is continuous, by the extreme value theorem it attains both its minimum d_{\min} and maximum d_{\max} .

If $d_{\min} = d_{\max}$ then φ is just a line segment on the line $L_{\vec{v}, p_1}$ with a length less than $|\vec{v}|$ and the statement of the lemma holds true.

If $d_{\min} < d_{\max}$, let $T_{\min} = \{t \in [0, 1] : d \circ \varphi(t) = d_{\min}\}$ and $T_{\max} = \{t \in [0, 1] : d \circ \varphi(t) = d_{\max}\}$. Since both T_{\min} and T_{\max} are closed and non-empty, we can take $t_{\min} \in T_{\min}$ and $t_{\max} \in T_{\max}$ such that $d_{\min} < d \circ \varphi(t) < d_{\max}$ for every $t \in (\min\{t_{\min}, t_{\max}\}, \max\{t_{\min}, t_{\max}\})$. Denote $p_{\min} = \varphi(t_{\min})$ and $p_{\max} = \varphi(t_{\max})$. All curves $\varphi^{\rightarrow k}$, $k \in \mathbb{Z}$, lie within the stripe between lines $L_{\vec{v}, p_{\min}}$ and $L_{\vec{v}, p_{\max}}$. Denote $\psi = \varphi([\min\{t_{\min}, t_{\max}\}, \max\{t_{\min}, t_{\max}\}])$ (a simple curve connecting p_{\min} and p_{\max}) and let $\psi^{\rightarrow k} = \psi + k\vec{v}$, $k \in \mathbb{Z}$, be the corresponding shifts of ψ .

Since $\psi^{\rightarrow k}$ meets neither $L_{\vec{v}, p_{\min}}$ nor $L_{\vec{v}, p_{\max}}$ at any point except its end-points, it splits the stripe into two disjoint regions—left and right (with respect to vector \vec{v})—let us denote the left region by L_k and the right one by R_k .

Since $\varphi \cap (\varphi + \vec{v}) = \emptyset$, we have for every $k \in \mathbb{Z}$, $\psi^{\rightarrow k} \cap \varphi^{\rightarrow k+1} \subseteq \varphi^{\rightarrow k} \cap \varphi^{\rightarrow k+1} = \emptyset$. Since the point $p_{\min} + (k + 1)\vec{v} \in \varphi^{\rightarrow k+1}$ lies in R_k and $\varphi^{\rightarrow k+1} \cap \psi^{\rightarrow k} = \emptyset$, the whole curve φ_{k+1} lies in R_k . Hence $\psi^{\rightarrow k+1} \subseteq R_k$ and similarly $\psi^{\rightarrow k-1} \subseteq L_k$. This yields $R_{k+1} \subseteq R_k$ and $L_{k-1} \subseteq L_k$ and consequently $R_\ell \subseteq R_k$ and $L_k \subseteq L_\ell$ for any $k \leq \ell$, $k, \ell \in \mathbb{Z}$.

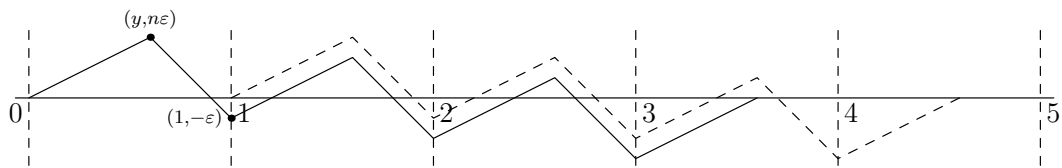
Consider now any $k < \ell$, $k, \ell \in \mathbb{Z}$. If $\ell = k + 1$ then $\varphi^{\rightarrow k} \cap \varphi^{\rightarrow \ell} = \emptyset$ by the assumption of the lemma. If $\ell > k + 1$ then $\varphi^{\rightarrow k} \subseteq L_{k+1}$ and $\varphi^{\rightarrow \ell} \subseteq R_{\ell-1} \subseteq R_{k+1}$, i.e., $\varphi^{\rightarrow k}$ and $\varphi^{\rightarrow \ell}$ are disjoint. ◀

The following corollary of Lemma 4 shows that if a curve is translated by a vector \vec{v} , and the vector between its start and end points is an integer multiple of \vec{v} , then the curve must intersect its translation by \vec{v} .

► **Corollary 5.** Consider an integer $n \geq 1$, a point $p \in \mathbb{R}^2$ and a nonzero vector $\vec{v} \in \mathbb{R}^2$. Let φ be a simple curve connecting p and $p + n\vec{v}$. Then φ intersects its translation by \vec{v} .

Proof. Assume for the sake of contradiction that φ and $\varphi + \vec{v}$ do not intersect. By Lemma 4 all curves $\varphi + n\vec{v}$, $n \in \mathbb{N}$, are mutually disjoint but $(p + n\vec{v}) \in \varphi \cap (\varphi + n\vec{v})$ —a contradiction. ◀

The assumption that the vector from the start point to the end point of the curve φ is an integer multiple of the vector \vec{v} is essential in Corollary 5. The following example



■ **Figure 3** An example of a curve φ from $(0, 0)$ to $(3.6, 0)$ (solid) that does not intersect its shift $\varphi + (1, 0)$ (dashed).

provides a general construction of a curve $\varphi \subseteq \mathbb{R}^2$ connecting points p and $p + x\vec{v}$ such that $\varphi \cap (\varphi + \vec{v}) = \emptyset$, where $\vec{0} \neq \vec{v} \in \mathbb{R}^2$ and $x \in \mathbb{R} \setminus \mathbb{Z}$, $|x| > 1$. Note that for $|x| < 1$ the line segment from p to $p + x\vec{v}$ does not intersect its shift by \vec{v} .

► **Example 6.** For simplicity assume that $p = (0, 0)$ and $\vec{v} = (1, 0)$. Let $n = \lfloor x \rfloor$, $y = x - n$ and choose any $\varepsilon > 0$. Let μ denote the line segment (simple curve) from $(0, 0)$ to $(y, n\varepsilon)$ and ν denote the line segment from $(y, n\varepsilon)$ to $(1, -\varepsilon)$. Denote $\mu_k = \mu + k(1, -\varepsilon)$ and $\nu_k = \nu + k(1, -\varepsilon)$ for $k \in \mathbb{Z}$. Then let $\varphi = \mu_0 \oplus \nu_0 \oplus \cdots \oplus \mu_{n-1} \oplus \nu_{n-1} \oplus \mu_n$ be the desired curve. Figure 3 shows an example of this construction for $x = 3.6$. Note that φ starts and ends on the x -axis and that $\varphi + \vec{v}$ does not intersect φ since for each stripe between $x = i$ and $x = i + 1$, $i = 1, \dots, n$, the part of $\varphi + \vec{v}$ in this stripe lies above the part of φ in the same stripe (shifted up by ε).

The following theorem is quite technical to state. Informally, it concerns a finite set of non-intersecting curves $\varphi_1, \dots, \varphi_k$ and a vector \vec{v} of the following form. Each curve connects two points in the plane, subject to the condition that the end point of φ_i is the start point of φ_{i+1} translated by a positive integer multiple of \vec{v} , with $\varphi_{k+1} = \varphi_1$. See Figure 4(a) for an example. An alternative way to think of these curves is as a single “mostly continuous” simple closed curve, with k discontinuities allowed, where each discontinuity is of the form “jump backwards by some positive integer multiple of \vec{v} .” The theorem states that this curve must intersect its translation by \vec{v} .

► **Theorem 7.** Let $k \in \mathbb{Z}^+$, let $p_1, \dots, p_k \in \mathbb{R}^2$ be points, let $n_1, \dots, n_k \in \mathbb{Z}^+$, and let $\vec{v} \in \mathbb{R}^2$ be a nonzero vector. Then there do not exist curves $\varphi_1, \dots, \varphi_k : [0, 1] \rightarrow \mathbb{R}^2$ satisfying the following conditions:

1. φ_i is a simple curve from p_i to $(p_{i+1} + n_{i+1}\vec{v})$, for every $1 \leq i \leq k$, where $p_{k+1} = p_1$ and $n_{k+1} = n_1$,
2. $\varphi_i \cap (\varphi_i + \vec{v}) = \emptyset$, for every $1 \leq i \leq k$,
3. $\varphi_i \cap \varphi_j = \emptyset$, for every $1 \leq i < j \leq k$.

Proof. By induction on k . The base case $k = 1$ immediately follows by Corollary 5.

Intuitively, the inductive case will show that if we suppose, for the sake of contradiction, that k curves exist satisfying the conditions, then we can find a common point of intersection between two of their integer translations by \vec{v} , and we can connect two subcurves of these translations to create a set of $k' < k$ curves also satisfying the hypothesis of the theorem, without introducing an intersection. Figure 4 shows an example of three curves being reduced to two. The new curves will simply be $k' - 1$ translations of some of the original k curves (which already satisfy the conditions by hypothesis), together with one new curve ψ , so our main task will be to show that ψ , in the presence of the other pre-existing curves, satisfies the three conditions.

More formally, let $k > 1$ and suppose the theorem holds for all integers $0 < k' < k$. Assume for the sake of contradiction that there are curves $\varphi_1, \dots, \varphi_k$ satisfying conditions 1, 2, and 3,

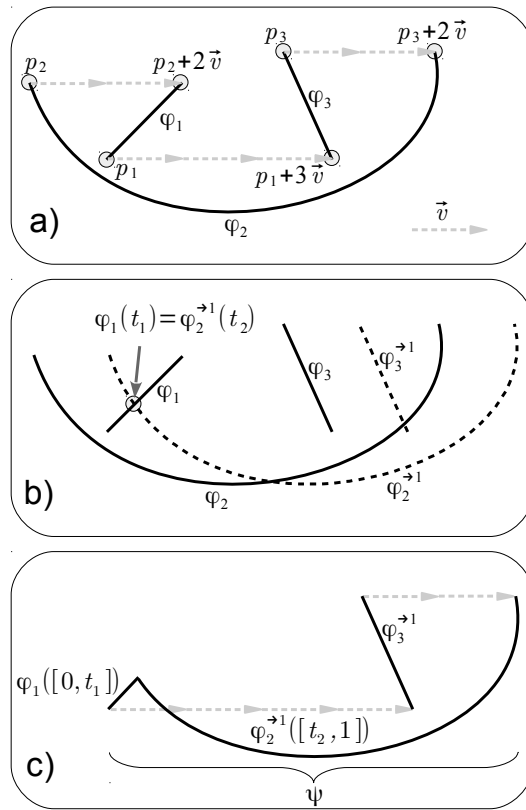


Figure 4 An example of the proof of Theorem 7 for $k = 3$ curves.
 (a) Three curves, φ_1 , φ_2 , and φ_3 , with start and end points obeying condition 1 and also condition 3 (the curves violate condition 2, however, as Theorem 7 dictates they must if obeying the other two conditions). In this case, $n_1 = 3$, $n_2 = 2$, and $n_3 = 2$.
 (b) Translations of curves φ_2 and φ_3 by \vec{v} , showing that φ_1 first intersects $\varphi_2^{\rightarrow 1}$, among all positive integer translations of φ_2 and φ_3 . So in this example, $M = 2$ and $L = 1$.
 (c) ψ defined as the concatenation of $\varphi_1([0, t_1])$ with $\varphi_2^{\rightarrow 1}([t_2, 1])$. ψ and $\varphi_3^{\rightarrow 1}$ and are the two curves produced by the proof for the inductive argument.

and define $\varphi_m^{\rightarrow \ell} = \varphi_m + \ell\vec{v}$ for all $m \in \{1, \dots, k\}$ and $\ell \in \mathbb{N}$. We find the first intersection of φ_1 with any of curves $\varphi_m^{\rightarrow \ell}$ for all $m \in \{2, \dots, k\}$ and $\ell \in \mathbb{N}$. Let

$$\begin{aligned}
 t_1 &= \min\{t \in [0, 1] : (\exists m \in \{2, \dots, k\})(\exists \ell \in \mathbb{N}) \varphi_1(t) \in \varphi_m^{\rightarrow \ell}\}, \\
 M &= \text{any } m \in \{2, \dots, k\} \text{ such that } (\exists \ell \in \mathbb{N}) \varphi_1(t_1) \in \varphi_m^{\rightarrow \ell}, \\
 L &= \text{the unique } \ell \in \mathbb{N} \text{ such that } \varphi_1(t_1) \in \varphi_M^{\rightarrow \ell}, \\
 t_2 &= \text{the unique } t \in [0, 1] \text{ such that } \varphi_1(t_1) = \varphi_M^{\rightarrow L}(t).
 \end{aligned}$$

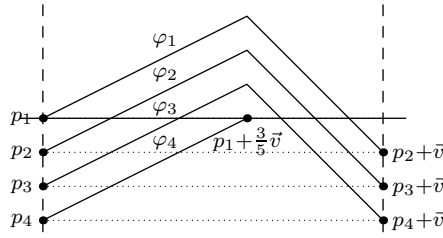
Since φ_1 intersects $\varphi_2^{\rightarrow n_2}$ at $p_2 + n_2\vec{v}$ by condition 1, t_1 , M , and L are well-defined. The uniqueness of L follows by Lemma 4. The uniqueness of t_2 follows from the fact that $\varphi_M^{\rightarrow L}$ is simple.

Now define the curve ψ as a concatenation

$$\psi = \varphi_1([0, t_1]) \oplus \varphi_M^{\rightarrow L}([t_2, 1])$$

and consider its shift

$$\psi + \vec{v} = \varphi_1^{\rightarrow 1}([0, t_1]) \oplus \varphi_M^{\rightarrow L+1}([t_2, 1]).$$



■ **Figure 5** An example of four curves $\varphi_1, \dots, \varphi_4$ that satisfy the conditions of Theorem 7, except that $n_1 = \frac{3}{5}$ is not an integer.

In what follows we will show that points $p_1, p_{M+1} + L\vec{v}, \dots, p_k + L\vec{v}$, integers $n_1 + L, n_{M+1}, \dots, n_k$ and curves $\psi, \varphi_{M+1}^{\rightarrow L}, \dots, \varphi_k^{\rightarrow L}$ form another instance satisfying conditions 1, 2, and 3.

Observe that ψ is a curve connecting the point p_1 to the point $p_{M+1} + (n_{M+1} + L)\vec{v}$. It consists of subcurves of two simple curves whose concatenation at the intersection point $\varphi_1(t_1) = \varphi_M^{\rightarrow L}(t_2)$, by the definition of t_1 , is the first point of intersection between φ_1 and $\varphi_M^{\rightarrow L}$. The curve $\varphi_M^{\rightarrow L}$ after that point (i.e., $\varphi_M^{\rightarrow L}([t_2, 1])$) therefore cannot intersect $\varphi_1([0, t_1])$, so ψ is simple. It follows that ψ satisfies condition 1 of the new instance.

We establish that ψ does not intersect its shift by vector \vec{v} by analyzing each of the two parts of ψ , $\varphi_1([0, t_1])$ and $\varphi_M^{\rightarrow L}([t_2, 1])$, and their translations by \vec{v} , separately:

- $\varphi_1([0, t_1]) \cap \varphi_1^{\rightarrow 1}([0, t_1]) = \emptyset$, since $\varphi_1 \cap \varphi_1^{\rightarrow 1} = \emptyset$ by condition 2.
- $\varphi_M^{\rightarrow L}([t_2, 1]) \cap \varphi_M^{\rightarrow L+1}([t_2, 1]) = \emptyset$, since it follows by condition 2 that $\varphi_M^{\rightarrow L} \cap \varphi_M^{\rightarrow L+1} = \emptyset$.
- $\varphi_1([0, t_1]) \cap \varphi_M^{\rightarrow L+1}([t_2, 1]) = \emptyset$, since by the definition of t_1 (in particular, the fact that it is the minimum element of the set defining it), $\varphi_1([0, t_1])$ does not intersect any $\varphi_m^{\rightarrow \ell}$, for any $m \geq 2, \ell \in \mathbb{N}$.
- $\varphi_M^{\rightarrow L}([t_2, 1]) \cap \varphi_1^{\rightarrow 1}([0, t_1]) = \emptyset$, since otherwise $\varphi_1([0, t_1])$ would intersect $\varphi_M^{\rightarrow L-1}$, violating the definition of t_1 similarly to the previous point.

This implies that ψ satisfies condition 2.

We have $\varphi_i^{\rightarrow L} \cap \psi = \emptyset$ for every $i > M$, since $\varphi_i^{\rightarrow L}$ cannot intersect $\varphi_1([0, t_1])$ (by definition of t_1) and $\varphi_i^{\rightarrow L} \cap \varphi_M^{\rightarrow L} = \emptyset$ by condition 3. This implies that ψ satisfies condition 3 of the new instance.

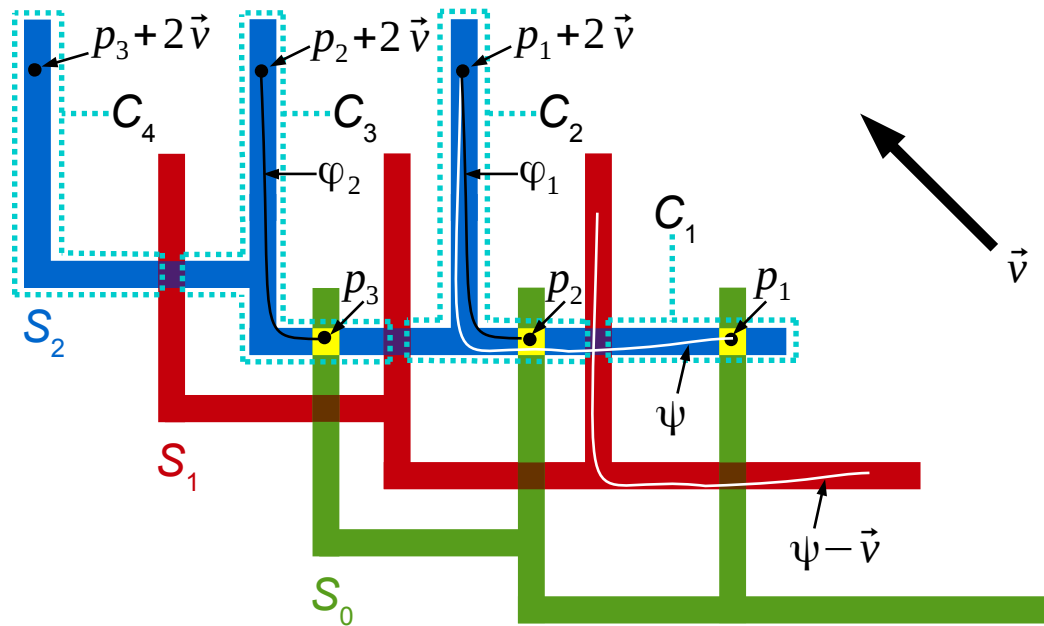
Thus, the new instance with points $p_1, p_{M+1} + L\vec{v}, \dots, p_k + L\vec{v}$, integers $n_1 + L, n_{M+1}, \dots, n_k$ and curves $\psi, \varphi_{M+1}^{\rightarrow L}, \dots, \varphi_k^{\rightarrow L}$ satisfy conditions 1, 2, and 3. In addition, it has a smaller number of curves ($k + 1 - M = k' < k$), and hence, using the induction hypothesis we have a contradiction. ◀

The example in Figure 5 shows that the theorem does not hold if we allow just one of the numbers n_1, \dots, n_k to be a non-integer.

3.2 A theorem about shapes

Theorem 7 gives rise to the following geometrical theorem about discrete shapes, which is the main technical tool to prove our main self-assembly result, Theorem 11. We define a *shape* to be a finite, connected subset of \mathbb{Z}^2 .

► **Theorem 8.** *Let $S_0 \subset \mathbb{Z}^2$ be a shape, and let $\vec{v} \in \mathbb{Z}^2$ be a nonzero vector. Let $S_1 = S_0 + \vec{v}$ and $S_2 = S_1 + \vec{v}$. Then there is a connected component of $S_2 \setminus S_1$ that does not intersect S_0 .*



■ **Figure 6** An example of a shape S_0 and its two translations. Starting at $p_1 \in (S_2 \cap S_0) \setminus S_1$, we repeat the following procedure: from point p_i in connected component C_i of $S_2 \setminus S_1$, jump to point $p_i + 2\vec{v}$, which is guaranteed to be in a different connected component C_{i+1} of $S_2 \setminus S_1$ from p_i (see proof of Theorem 8 to see why this is implied by Corollary 5). If C_{i+1} intersects S_0 at point p_{i+1} , then there is a curve φ_i in $S_2 \setminus S_1$ from $p_i + 2\vec{v}$ to p_{i+1} , and jumping to point $p_{i+1} + 2\vec{v}$ takes us to yet another connected component $C_{i+2} \neq C_{i+1}$. Repeating this must eventually result in a connected component (in this example, C_4) that does not intersect S_0 , or else the curves φ_i would contradict Theorem 7.

Proof. We first sketch an informal intuition of the proof, shown by example in Figure 6. The argument is constructive: it shows a way to iterate through some connected components of $S_2 \setminus S_1$ to actually find one that does not intersect S_0 .

Start with component C_1 , and suppose it intersects S_0 at point $p_1 \in C_1 \cap S_0$. Then $p_1 + 2\vec{v} \in S_2$ since $p_1 \in S_0$.¹ Let ψ be a path (simple curve) from p_1 to $p_1 + 2\vec{v}$ lying entirely within S_2 . Corollary 5 implies that ψ intersects $\psi - \vec{v}$, which is a curve lying entirely within S_1 . In other words, every path from p_1 to $p_1 + 2\vec{v}$ lying inside S_2 hits S_1 , i.e., $p_1 + 2\vec{v}$ and p_1 are in different connected components of $S_2 \setminus S_1$. We call $C_2 \neq C_1$ the connected component of $p_1 + 2\vec{v}$. Suppose C_2 also intersects S_0 ; then there is some curve φ_1 lying entirely within $S_2 \setminus S_1$ and going from $p_1 + 2\vec{v}$ to this new point $p_2 \in C_2 \cap S_0$. Repeating the previous argument, $p_2 + 2\vec{v}$ must be in a different connected component $C_3 \neq C_2$, and if C_3 also intersects S_0 , then there is another curve $\varphi_2 \subset C_3$ from $p_2 + 2\vec{v}$ to $p_3 \in C_3 \cap S_0$. In this example, we iterate this one more time and find that connected component $C_4 \subset S_2 \setminus S_1$ does not intersect S_0 .

For the sake of contradiction, suppose that we fail to find such a connected component, i.e., every one of the connected components C_1, \dots, C_k of $S_2 \setminus S_1$ intersects S_0 . Then eventually the above described procedure cycles back to a previously visited connected component, and

¹ In this example $p_1 + 2\vec{v} \notin S_1$; in the full argument we consider $p_1 + n\vec{v}$ for $n \in \mathbb{Z}^+$ large enough to ensure this.

the curves φ_j contained in $S_2 \setminus S_1$ satisfy condition 1 of Theorem 7. Since each $\varphi_i \in S_2 \setminus S_1$, we have $\varphi_i + \vec{v} \in S_3 \setminus S_2$, hence $\varphi_i \cap (\varphi_i + \vec{v}) = \emptyset$ for all $1 \leq i \leq k$, so they satisfy condition 2. Since each curve lies in a different connected component of $S_2 \setminus S_1$, they do not intersect each other, satisfying condition 3, a contradiction.

More formally, consider connected components of $S_2 \setminus S_1$, say C_1, \dots, C_k , for some $k \geq 1$. We say that C_i is *non-conflicting* if $C_i \cap S_0 = \emptyset$. We will show that there is a non-conflicting C_i . Assume for the sake of contradiction that for every $i = 1, \dots, k$, $C_i \cap S_0 \neq \emptyset$ and let $p_i \in C_i \cap S_0$. Note that $p_i + \vec{v} \in S_1$. Let n_i be the smallest positive integer such that $p_i + n_i \vec{v} \notin S_1$ (since S_1 is finite, such an n_i must exist). Since $p_i + (n_i - 1)\vec{v} \in S_1$, we have $p_i + n_i \vec{v} \in S_2 \setminus S_1$. Hence, $p_i + n_i \vec{v}$ belongs to some connected component of $S_2 \setminus S_1$. Both p_i and $p_i + n_i \vec{v}$ are in S_2 , but by Corollary 5, any path within S_2 connecting them must intersect its translation by $-\vec{v}$, which is a path in S_1 , so $p_i + n_i \vec{v}$ must be in a different connected component than C_i . We call this connected component C_{i+1} .²

Consider a simple curve (a self-avoiding path in the lattice) φ_i from p_i to $p_{i+1} + n_{i+1} \vec{v}$ in $C_i \subseteq S_2 \setminus S_1$. Since these paths lie in different connected components they do not intersect. Furthermore, since $\varphi_i + \vec{v} \subset S_3 \setminus S_2$, it does not intersect $\varphi_i \subset S_2$. But these curves contradict Theorem 7. \blacktriangleleft

3.3 Implication for self-assembly

In this section we use Theorem 8 to prove our main theorem, Theorem 11. We require the following theorem from [8]. We say that two overlapping assemblies α and β are *consistent* if $\alpha(p) = \beta(p)$ for every $p \in \text{dom } \alpha \cap \text{dom } \beta$. If α and β are consistent, define their *union* $\alpha \cup \beta$ to be the assembly with $\text{dom } (\alpha \cup \beta) = \text{dom } \alpha \cup \text{dom } \beta$ defined by $(\alpha \cup \beta)(p) = \alpha(p)$ if $p \in \text{dom } \alpha$ and $(\alpha \cup \beta)(p) = \beta(p)$ if $p \in \text{dom } \beta$. Let $\alpha \cup \beta$ be undefined if α and β are not consistent.

► **Theorem 9** ([8]). *If α and β are \mathcal{T} -producible assemblies that are consistent and overlapping, then $\alpha \cup \beta$ is \mathcal{T} -producible. Furthermore, it is possible to assemble first α and then assemble the missing portions of β , i.e., $\beta|_{C_1}, \dots, \beta|_{C_k}$, where C_1, \dots, C_k are connected components of $\text{dom } \beta \setminus \text{dom } \alpha$.*

► **Definition 10.** Let $\alpha + \vec{v}$ denote the translation of α by \vec{v} , i.e., an assembly β such that $\text{dom } \beta = \text{dom } \alpha + \vec{v}$ and $\beta(p) = \alpha(p - \vec{v})$ for all $p \in \text{dom } \beta$. We say that assembly α is *repetitious* if there exists a nonzero vector $\vec{v} \in \mathbb{Z}^2$ such that $\text{dom } \alpha \cap \text{dom } (\alpha + \vec{v}) \neq \emptyset$ and α and $\alpha + \vec{v}$ are consistent.

Note that Theorem 9 implies that if a producible assembly α is repetitious with translation vector \vec{v} , then $\alpha \cup (\alpha + \vec{v})$ is also producible. The following is the main theorem of this paper.

► **Theorem 11.** *Let \mathcal{T} be a hierarchical tile assembly system. If \mathcal{T} has a producible repetitious assembly, then arbitrarily large assemblies are producible in \mathcal{T} .*

Proof. It suffices to show that the existence of a producible repetitious assembly α implies the existence of a strictly larger producible repetitious assembly $\alpha' \sqsupset \alpha$. Let α be a producible

² Assuming we do this for every point p_i , at some point we must cycle back to a connected component already visited. It may not be that this cycle contains all connected components of $S_2 \setminus S_1$, but in this case we consider C_1, \dots, C_k to be not every connected component of $S_2 \setminus S_1$, but merely those encountered in the cycle, so that for the sake of notational convenience we can assume that C_1, \dots, C_k are all encountered, and indexed by the order in which they are encountered.

repetitious assembly, with $\vec{v} \in \mathbb{Z}^2$ a nonzero vector such that α and $\alpha + \vec{v}$ overlap and are consistent. For all $i \in \{0, 1, 2\}$, let $\alpha_i = \alpha + i\vec{v}$ and $S_i = \text{dom } \alpha_i$.

By Theorem 8, at least one connected component $C_2 \subseteq S_2 \setminus S_1$ does not intersect S_0 . Define $C_1 = C_2 - \vec{v}$. Note that $C_1 \subseteq S_1 \setminus S_0$, which implies, since $C_2 \subseteq S_2 \setminus S_1$, that $C_2 \cap C_1 = \emptyset$. Let $\bar{\alpha} = \alpha_1 \upharpoonright_{C_1}$. Define $\alpha' = \alpha \cup \bar{\alpha}$. By Theorem 9, α' is producible. Consider $\text{dom } \alpha' \cap \text{dom } (\alpha' + \vec{v})$; it suffices to show that α' and $\alpha' + \vec{v}$ are consistent on every tile type in this intersection. We have

$$\begin{aligned} \text{dom } \alpha' \cap \text{dom } (\alpha' + \vec{v}) &= (S_0 \cup C_1) \cap (S_1 \cup C_2) \\ &= (S_0 \cap S_1) \cup (S_0 \cap C_2) \cup (C_1 \cap S_1) \cup (C_1 \cap C_2) \\ &= (S_0 \cap S_1) \cup \emptyset \cup (C_1 \cap S_1) \cup \emptyset \\ &= (S_0 \cap S_1) \cup C_1. \end{aligned}$$

We handle the cases for $S_0 \cap S_1$ and C_1 separately:

$S_0 \cap S_1$: Since $C_1 \cap S_0 \cap S_1 = \emptyset$, the addition of $\bar{\alpha}$ to α_0 cannot introduce new tiles anywhere in $S_0 \cap S_1$, so only tiles from α_0 could appear here. By the hypothesis that α_0 is consistent with α_1 , α' and $\alpha' + \vec{v}$ are consistent on $S_0 \cap S_1$.

C_1 : Observe that $\alpha' \upharpoonright_{C_1 - \vec{v}} \sqsubset \alpha_0$ (this is the subassembly of α' that will overlap C_1 after being translated by \vec{v}) and $(\alpha' + \vec{v}) \upharpoonright_{C_1} \sqsubset \alpha_1$, so the fact that α_0 is consistent with α_1 implies that α' and $\alpha' + \vec{v}$ are consistent on C_1 as well.

Hence α' is repetitious. Since $C_1 \subseteq S_1 \setminus S_0$ and is nonempty, $|\text{dom } \alpha'| > |\text{dom } \alpha|$. ◀

Acknowledgements. The authors are extremely grateful to Jozef Haleš for the proof of Theorem 7. Although Jozef requested not to be a coauthor, that theorem is the keystone of the paper. The second author is also grateful to David Kirkpatrick, Pierre-Étienne Meunier, Damien Woods, Shinnosuke Seki, and Andrew Winslow for several insightful discussions. The third author would like to thank Sheung-Hung Poon for useful discussions.

References

- 1 Gagan Aggarwal, Qi Cheng, Michael H. Goldwasser, Ming-Yang Kao, Pablo Moisset de Espanés, and Robert T. Schweller. Complexities for generalized models of self-assembly. *SIAM Journal on Computing*, 34:1493–1515, 2005. Preliminary version appeared in SODA 2004.
- 2 Robert D. Barish, Rebecca Schulman, Paul W. K. Rothmund, and Erik Winfree. An information-bearing seed for nucleating algorithmic self-assembly. *Proceedings of the National Academy of Sciences*, 106(15):6054–6059, March 2009.
- 3 Sarah Cannon, Erik D. Demaine, Martin L. Demaine, Sarah Eisenstat, Matthew J. Patitz, Robert T. Schweller, Scott M. Summers, and Andrew Winslow. Two hands are better than one (up to constant factors). In *STACS 2013: Proceedings of the Thirtieth International Symposium on Theoretical Aspects of Computer Science*, pages 172–184, 2013.
- 4 Ho-Lin Chen and David Doty. Parallelism and time in hierarchical self-assembly. In *SODA 2012: Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1163–1182, 2012.
- 5 Erik D. Demaine, Martin L. Demaine, Sándor P. Fekete, Matthew J. Patitz, Robert T. Schweller, Andrew Winslow, and Damien Woods. One tile to rule them all: Simulating any Turing machine, tile assembly system, or tiling system with a single puzzle piece. In *ICALP 2014: Proceedings of the 41st International Colloquium on Automata, Languages, and Programming*, 2014.

- 6 Erik D. Demaine, Matthew J. Patitz, Trent Rogers, Robert T. Schweller, Scott M. Summers, and Damien Woods. The two-handed tile assembly model is not intrinsically universal. In *ICALP 2013: Proceedings of the 40th International Colloquium on Automata, Languages and Programming*, July 2013.
- 7 David Doty. Theory of algorithmic self-assembly. *Communications of the ACM*, 55(12):78–88, December 2012.
- 8 David Doty. Producibility in hierarchical self-assembly. In *UCNC 2014: Proceedings of 13th Unconventional Computation and Natural Computation*, 2014.
- 9 David Doty, Matthew J. Patitz, Dustin Reishus, Robert T. Schweller, and Scott M. Summers. Strong fault-tolerance for self-assembly with fuzzy temperature. In *FOCS 2010: Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 417–426. IEEE, 2010.
- 10 David Doty, Matthew J. Patitz, and Scott M. Summers. Limitations of self-assembly at temperature 1. *Theoretical Computer Science*, 412(1–2):145–158, January 2011. Preliminary version appeared in DNA 2009.
- 11 Pierre Étienne Meunier, Matthew J. Patitz, Scott M. Summers, Guillaume Theyssier, Andrew Winslow, and Damien Woods. Intrinsic universality in tile self-assembly requires cooperation. In *SODA 2014: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 752–771, 2014.
- 12 Kei Goto, Yoko Hinob, Takayuki Kawashima, Masahiro Kaminagab, Emiko Yanob, Gaku Yamamoto, Nozomi Takagic, and Shigeru Nagasec. Synthesis and crystal structure of a stable S-nitrosothiol bearing a novel steric protection group and of the corresponding S-nitrothiol. *Tetrahedron Letters*, 41(44):8479–8483, 2000.
- 13 Wilfried Heller and Thomas L. Pugh. “Steric protection” of hydrophobic colloidal particles by adsorption of flexible macromolecules. *Journal of Chemical Physics*, 22(10):1778, 1954.
- 14 Wilfried Heller and Thomas L. Pugh. “Steric” stabilization of colloidal solutions by adsorption of flexible macromolecules. *Journal of Polymer Science*, 47(149):203–217, 1960.
- 15 Chris Luhrs. Polyomino-safe DNA self-assembly via block replacement. *Natural Computing*, 9(1):97–109, March 2010. Preliminary version appeared in DNA 2008.
- 16 Ján Maňuch, Ladislav Stacho, and Christine Stoll. Two lower bounds for self-assemblies at temperature 1. *Journal of Computational Biology*, 17(6):841–852, 2010.
- 17 Matthew J. Patitz. An introduction to tile-based self-assembly. In *UCNC 2012: Proceedings of the 11th international conference on Unconventional Computation and Natural Computation*, pages 34–62, Berlin, Heidelberg, 2012. Springer-Verlag.
- 18 John H. Reif and Tianqi Song. Complexity and computability of temperature-1 tilings. Poster at DNA 2013: 19th International Meeting on DNA Computing and Molecular Programming, 2013.
- 19 Paul W. K. Rothmund and Erik Winfree. The program-size complexity of self-assembled squares (extended abstract). In *STOC 2000: Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pages 459–468, 2000.
- 20 Rebecca Schulman and Erik Winfree. Synthesis of crystals with a programmable kinetic barrier to nucleation. *Proceedings of the National Academy of Sciences*, 104(39):15236–15241, 2007.
- 21 Rebecca Schulman and Erik Winfree. Programmable control of nucleation for algorithmic self-assembly. *SIAM Journal on Computing*, 39(4):1581–1616, 2009. Preliminary version appeared in DNA 2004.
- 22 Leroy G. Wade. *Organic Chemistry*. Prentice Hall, 2nd edition, 1991.
- 23 Erik Winfree. Simulations of computing by self-assembly. Technical Report CaltechC-STR:1998.22, California Institute of Technology, 1998.

- 24 Erik Winfree. Self-healing tile sets. In Junghuei Chen, Natasa Jonoska, and Grzegorz Rozenberg, editors, *Nanotechnology: Science and Computation*, Natural Computing Series, pages 55–78. Springer, 2006.
- 25 Erik Winfree, Furong Liu, Lisa A. Wenzler, and Nadrian C. Seeman. Design and self-assembly of two-dimensional DNA crystals. *Nature*, 394(6693):539–44, 1998.

A Appendix

A.1 Formal definition of the hierarchical tile assembly model

We will consider the square lattice, i.e., the graph L_{\square} with the vertex set \mathbb{Z}^2 and the edge set $\{(u, v) : |u, v| = 1\}$. The directions $\mathcal{D} = \{N, E, S, W\}$ are used to indicate the natural directions in the lattice. Formally, they are functions from $\mathbb{Z} \times \mathbb{Z}$ to $\mathbb{Z} \times \mathbb{Z}$: $N(x, y) = (x, y + 1)$, $E(x, y) = (x + 1, y)$, $S(x, y) = (x, y - 1)$, and $W(x, y) = (x - 1, y)$. Note that $-E = W$ and $-N = S$.

Informally, a tile is a square with the north, east, south, and west edges labeled from some finite alphabet Σ of *glues*. Formally, a tile t is a 4-tuple $(g_N, g_E, g_S, g_W) \in \Sigma^4$, indicating the glues on the north, east, south, and west side, respectively. Each pair of glues g and g' is associated with a nonnegative integer $str(g, g')$ called the *interaction strength*.

An *assembly* on a set of tiles T is a partial map $\alpha : \mathbb{Z}^2 \dashrightarrow T$ such that the subgraph of L_{\square} induced by the domain of α , denoted by $L_{\square}[\text{dom } \alpha]$, is connected. The *weighted subgraph induced by α* , denoted by $L_{\square}[\alpha]$, is $L_{\square}[\text{dom } \alpha]$ in which every edge pq has weight equal to the interaction strength of the glues on the abutting sides of tiles at positions p and q , respectively, i.e., $str(\alpha(p)_d, \alpha(q)_{-d})$ where $d = q - p$. Given a positive integer $\tau \in \mathbb{Z}^+$, called a *temperature*, a set of edges of $L_{\square}[\alpha]$ of an assembly α is τ -*stable* if the sum of the weights of edges in this set is at least τ , and assembly α is τ -*stable* if every edge cut of $L_{\square}[\alpha]$ is τ -stable.

A *hierarchical tile assembly system* (hierarchical TAS) is a triple $\mathcal{T} = (T, \tau, str)$, where T is a finite set of tile types, $\tau \in \mathbb{Z}^+$ and $str : \Sigma \times \Sigma \rightarrow \mathbb{N}$ is the interaction strength function. Let $\alpha, \beta : \mathbb{Z}^2 \dashrightarrow T$ be two assemblies. We say that α and β are *nonoverlapping* if $\text{dom } \alpha \cap \text{dom } \beta = \emptyset$. Two assemblies α and β are *consistent* if $\alpha(p) = \beta(p)$ for all $p \in \text{dom } \alpha \cap \text{dom } \beta$. If α and β are consistent assemblies, define the assembly $\alpha \cup \beta$ in a natural way, i.e., $\text{dom } (\alpha \cup \beta) = \text{dom } \alpha \cup \text{dom } \beta$ and $(\alpha \cup \beta)(p) = \alpha(p)$ for $p \in \text{dom } \alpha$ and $(\alpha \cup \beta)(p) = \beta(p)$ for $p \in \text{dom } \beta$. If α and β are nonoverlapping, the *cut of the union $\alpha \cup \beta$* is the set of edges of L_{\square} with one end-point in $\text{dom } \alpha$ and the other end-point in $\text{dom } \beta$. An assembly γ is *singular* if $|\text{dom } \gamma| = 1$. We say that an assembly γ is \mathcal{T} -*producible* if either γ is singular or there exist \mathcal{T} -producible nonoverlapping assemblies α and β such that $\gamma = \alpha \cup \beta$ and the cut of $\alpha \cup \beta$ is τ -stable. In the latter case, we write $\alpha + \beta \xrightarrow{\mathcal{T}}_1 \gamma$. Note that every \mathcal{T} -producible assembly is τ -stable. A \mathcal{T} -producible assembly α is \mathcal{T} -*terminal* if there are no \mathcal{T} -producible assemblies β and γ such that $\alpha + \beta \xrightarrow{\mathcal{T}}_1 \gamma$. We say two assemblies α and β are *equivalent up to translation*, written $\alpha \simeq \beta$, if there is a vector $\vec{x} \in \mathbb{Z}^2$ such that $\text{dom } \alpha = \text{dom } \beta + \vec{x}$ and for all $p \in \text{dom } \beta$, $\alpha(p + \vec{x}) = \beta(p)$. We say that \mathcal{T} *uniquely produces* α if α is \mathcal{T} -terminal and for every \mathcal{T} -terminal assembly β , $\alpha \simeq \beta$.

A *restriction* of an assembly α to a set $D \subseteq \text{dom } \alpha$, denoted by $\alpha \upharpoonright_D$, is $\text{dom } \alpha \upharpoonright_D = D$ and for every $p \in D$, $\alpha \upharpoonright_D(p) = \alpha(p)$. If C is a subgraph of L_{\square} such that $V(C) \subseteq \text{dom } \alpha$, we define $\alpha \upharpoonright_C = \alpha \upharpoonright_{V(C)}$.

When \mathcal{T} is clear from context, we may omit \mathcal{T} from the notation above and instead write \rightarrow_1 , \rightarrow , *produces*, *producible*, and *terminal*.

Space Exploration via Proximity Search*

Sariel Har-Peled¹, Nirman Kumar², David M. Mount³, and Benjamin Raichel¹

- 1 Department of Computer Science, University of Illinois
201 N. Goodwin Avenue, Urbana, IL, 61801, USA
{sariel,raichel2}@illinois.edu
- 2 Department of Computer Science, University of California
2120B Harold Frank Hall, Santa Barbara, CA, 93106, USA
nirman@cs.ucsb.edu
- 3 Department of Computer Science, University of Maryland
College Park, MD, 20742, USA
mount@cs.umd.edu

Abstract

We investigate what computational tasks can be performed on a point set in \mathbb{R}^d , if we are only given black-box access to it via nearest-neighbor search. This is a reasonable assumption if the underlying point set is either provided implicitly, or it is stored in a data structure that can answer such queries. In particular, we show the following:

- (A) One can compute an approximate bi-criteria k -center clustering of the point set, and more generally compute a greedy permutation of the point set.
- (B) One can decide if a query point is (approximately) inside the convex-hull of the point set.

We also investigate the problem of clustering the given point set, such that meaningful proximity queries can be carried out on the centers of the clusters, instead of the whole point set.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, I.1.2 Algorithms, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Proximity search, implicit point set, probing

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.374

1 Introduction

Many problems in Computational Geometry involve sets of points in \mathbb{R}^d . Traditionally, such a point set is presented explicitly, say, as a list of coordinate vectors. There are, however, numerous applications in science and engineering where point sets are presented *implicitly*. This may arise for various reasons: (I) the point set (which might be infinite) is a physical structure that is represented in terms of a finite set of sensed measurements such as a point cloud, (II) the set is too large to be stored explicitly in memory, or (II) the set is procedurally generated from a highly compressed form. (A number of concrete examples are described below.)

Access to such an implicitly-represented point set P is performed through an *oracle* that is capable of answering queries of a particular type. We can think of this oracle as a black-box

* Work on this paper by S.H. and B.R. was partially supported by NSF AF awards CCF-1421231, and CCF-1217462. N. K. was partially supported by a NSF AF award CCF-1217462 while at UIUC, and by NSF grant CCF-1161495 and a grant from DARPA while at UCSB. D. M. was partially supported by NSF award CCF-1117259 and ONR award N00014-08-1-1015. The full paper is available online [12].



data structure, which is provided to us in lieu of an explicit representation. Various types of probes have been studied (such as finger probes, line probes, and X-ray probes [19]). Most of these assume that P is connected (e.g., a convex polygon) and cannot be applied when dealing with arbitrary point sets. In this paper, we consider *proximity probes* – a natural choice for probing general point sets based on computing nearest neighbors.

More formally, we assume that the point set P is a (not necessarily finite) compact subset of \mathbb{R}^d . The point set P is accessible only through a nearest-neighbor data structure, which given a query point q , returns the closest point of P to q . Some of our results assume that the data structure returns an exact nearest neighbor (NN) and others assume that the data structure returns a $(1 + \varepsilon)$ -approximate nearest-neighbor (ANN). (See Section 2 for definitions.) In any probing scenario, it is necessary to begin with a general notion of the set's spatial location. The point set P is contained within a given *domain*, which is a compact subset \mathcal{D} of \mathbb{R}^d .

The oracle is given as a black-box, and no deletions or insertions are allowed from the data structure. Furthermore, the number of data points in P is not necessarily known, nor is there any assumption on continuity or smoothness. Indeed, most of our results apply to infinite point sets, including volumes or surfaces.

Prior Work and Applications

Implicitly-represented point sets arise in various applications. One example is that of analyzing a geometric shape through probing. An example of this is Atomic Force Microscopy (AFM) [22]. This technology can reveal the undulations of a surface at the resolution of fractions of a nanometer. It relies on the principle that when an appropriately designed tip (the probe) is brought in the proximity of a surface to scan it, certain atomic forces minutely deflect the tip in the direction of the surface. Since the deflection of the tip is generally to the closest point on the surface, this mode of acquisition is an example of proximity probing. A sufficient number of such samples can be used to reconstruct the surface [2].

The topic of shape analysis through probing has been well studied within the field of computational geometry. The most commonly assumed probe is a *finger probe*, which determines the first point of contact of a ray and the set. Cole and Yap [6] pioneered this area by analyzing the minimum number of finger probes needed to reconstruct a convex polygon. Since then, various alternative probing methods have been considered. For good surveys of this area, see Skiena [19, 20].

More recently, Boissonnat *et al.* [4] presented an algorithm for learning a smooth unknown surface S bounding an object \mathcal{O} in \mathbb{R}^3 through the use of finger probes. Under some reasonable assumptions, their algorithm computes a triangulated surface \hat{S} that approximates S to a given level of accuracy. In contrast to our work, which applies to general point sets, all of these earlier results assume that the set in question is a connected shape or surface.

Implicitly-represented point sets also arise in geometric modeling. Complex geometric sets are often generated from much smaller representations. One example are fractals sets, which are often used to model natural phenomena such as plants, clouds, and terrains [21]. Fractals are often expressed as the limit of an iterative process [16]. Due to their regular, recursive structure it is often possible to answer proximity queries about such a set without generating the set itself.

Two other examples of infinite sets generated implicitly from finite models include (I) subdivision surfaces [1], where a smooth surface is generated by applying a recursive refinement process to a finite set of boundary points, and (II) metaballs [3], where a surface is defined by a blending function applied to a collection of geometric balls. In both cases, it

is possible to answer nearest neighbor queries for the underlying object to arbitrarily high precision without the need to generate its boundary.

Proximity queries have been applied before. Panahi *et al.* [18] use proximity probes on a convex polygon in the plane to reconstruct it exactly. Goel *et al.* [8], reduce the approximation versions of several problems like diameter, farthest neighbors, discrete center, metric facility location, bottleneck matching and minimum weight matching to nearest neighbor queries. They sometimes require other primitives for their algorithms, for example computation of the minimum enclosing ball or a dynamic version of the approximate nearest-neighbor oracle. Similarly, the computation of the minimum spanning tree [11] can be done using nearest-neighbor queries (but the data structure needs to support deletions). For more details, see the survey by Indyk [14].

Our contributions

In this paper we consider a number of problems on implicitly-represented point sets.

k-center clustering and the greedy permutation. Given a point set P , a *greedy permutation* (informally) is an ordering of the points of P : p_1, \dots, p_k, \dots , such that for any k , the set of points $\{p_1, \dots, p_k\}$ is a $O(1)$ -approximation to the optimal k -center clustering. This sequence arises in the k -center approximation of Gonzalez [9], and its properties were analyzed by Har-Peled and Mendel [13]. Specifically, if P can be covered by k balls of radius r_k , then the maximum distance of any point of P to its nearest neighbor in $\{p_1, \dots, p_k\}$ is $O(r_k)$.

In Section 3, we show that under reasonable assumptions, in constant dimension, one can compute a permutation that is a bi-criteria approximation to the optimal k center clustering. More formally, we can compute a sequence of points from P , p_1, p_2, \dots , such for any k , the radius of clustering using the centers in $\{p_1, \dots, p_{ck}\}$ is an $O(1)$ -approximation to the optimal k center clustering radius, where c is a constant depending only on the dimension. This result uses exact proximity queries, and only one query per sequence point generated. If the oracle answers $(1 + \varepsilon)$ -ANN queries only, then for any k , the permutation generated is competitive with the optimal k -center clustering, considering the first $O\left(k \log_{1/\varepsilon} \Phi\right)$ points in this permutation, where Φ is (roughly) the spread of the point set. The hidden constant factors grow exponentially in the dimension.

Approximate convex-hull membership. Given a point set P in \mathbb{R}^d , consider the problem of deciding whether a given query point $q \in \mathbb{R}^d$ is inside its convex-hull $\mathcal{C} = \mathcal{CH}(P)$. The answer for such a query is ε -approximately correct if the answer is correct whenever the query point's distance from the boundary of \mathcal{C} is at least $\varepsilon \cdot \text{diam}(\mathcal{C})$. In Section 4, we show that, given an oracle for $(1 + \varepsilon^2/c)$ -ANN queries, for some sufficiently large constant c , it is possible to answer approximate convex-hull membership queries using $O(1/\varepsilon^2)$ proximity queries. Remarkably, the number of queries is independent of the dimension of the data.

Our algorithm operates iteratively, by employing a gradient descent-like approach. It generates a sequence of points, all within the convex hull, that converges to the query point. Similar techniques have been used before, and are sometimes referred to as the Frank-Wolfe algorithm. Clarkson provides a survey and some new results of this type [5]. A recent algorithm of this type is the work by Kalantari [15]. Our main new contribution for the convex-hull membership problem is showing that the iterative algorithm can be applied to implicit point sets using nearest-neighbor queries.

Balanced proximity clustering. We study a problem that involves summarizing a point set in a way that preserves proximity information. Specifically, given a set P of n points in \mathbb{R}^d , and a parameter k , the objective is to select m centers from P , such that if we assign every point of P to its nearest center, no center has been selected by more than k points. This problem is related to topic of capacitated clustering from operations research [17].

In Section 5, we show that in the plane there exists such a clustering consisting of $O(n/k)$ such centers, and that in higher dimensions one can select $O((n/k) \log(n/k))$ centers (where the constant depends on the dimension). This result is not directly related to the other results in the paper.

Paper organization. In Section 2 we review some relevant work on k -center clustering. In Section 3 we provide our algorithm to compute an approximate k -center clustering. In Section 4 we show how we can decide approximately if a query point is within the convex hull of the given data points in a constant number of queries, where the constant depends on the degree of accuracy desired. Finally, in Section 5 we investigate balanced Voronoi partitions, which provides a density-based clustering of the data. Here we assume that all the data is known and the goal is to come up with a useful clustering that can help in proximity search queries.

2 Preliminaries

2.1 Background – k -center clustering and the greedy permutation

The following is taken from [10, Chap. 4], and is provided here for the sake of completeness.

In the k -center clustering problem, a set $P \subseteq \mathbb{R}^d$ of n points is provided together with a parameter k . The objective is to find a set of k points, $C \subseteq P$, such that the maximum distance of a point in P to its closest point in C is minimized. Formally, define $\text{price}(C, P) = \max_{p \in P} \min_{c \in C} \|p - c\|$. Let C_{opt} denote the set of centers achieving this minimum. The k -center problem can be interpreted as the problem of computing the minimum radius, called the *k -center clustering radius*, such that it is possible to cover the points of P using k balls of this radius, each centered at one of the data points. It is known that k -center clustering is NP-hard. Even in the plane, it is NP-hard to approximate to within a factor of $(1 + \sqrt{7})/2 \approx 1.82$ [7].

The greedy clustering algorithm. Gonzalez [9] provided a 2-approximation algorithm for k -center clustering. This algorithm, denoted by **GreedyKCenter**, repeatedly picks the point farthest away from the current set of centers and adds it to this set. Specifically, it starts by picking an arbitrary point, \bar{c}_1 , and setting $C_1 = \{\bar{c}_1\}$. For $i > 1$, in the i th iteration, the algorithm computes

$$r_{i-1} = \text{price}(C_{i-1}, P) = \max_{p \in P} d(p, C_{i-1}) \quad (2.1)$$

and the point \bar{c}_i that realizes it, where $d(p, C_{i-1}) = \min_{c \in C_{i-1}} \|p - c\|$. Next, the algorithm adds \bar{c}_i to C_{i-1} to form the new set C_i . This process is repeated until k points have been collected.

If we run **GreedyKCenter** till it exhausts all the points of P (i.e., $k = n$), then this algorithm generates a permutation of P ; that is, $\langle P \rangle = \langle \bar{c}_1, \dots, \bar{c}_n \rangle$. We will refer to $\langle P \rangle$ as the *greedy permutation* of P . There is also an associated sequence of radii $\langle r_1, \dots, r_n \rangle$, and the key property of the greedy permutation is that for each i with $1 \leq i \leq n$, all the

points of P are within a distance at most r_i from the points of $C_i = \langle \bar{c}_1, \dots, \bar{c}_i \rangle$. The greedy permutation has applications to packings, which we describe next.

► **Definition 1.** A set $S \subseteq P$ is an r -packing for P if the following two properties hold:

- (i) *Covering property:* All the points of P are within a distance at most r from the points of S .
- (ii) *Separation property:* For any pair of points $\mathbf{p}, \mathbf{x} \in S$, $\|\mathbf{p} - \mathbf{x}\| \geq r$.

(For most purposes, one can relax the separation property by requiring that the points of S be at distance $\Omega(r)$ from each other.)

Intuitively, an r -packing of a point set P is a compact representation of P at resolution r . Surprisingly, the greedy permutation of P provides us with such a representation for all resolutions.

► **Lemma 2** ([10]).

- (A) Let P be a set of n points in \mathbb{R}^d , and let its greedy permutation be $\langle \bar{c}_1, \dots, \bar{c}_n \rangle$ with the associated sequence of radii $\langle r_1, \dots, r_n \rangle$. For any i , $C_i = \langle \bar{c}_1, \dots, \bar{c}_i \rangle$ is an r_i -packing of P . Furthermore, r_i is a 2-approximation for the optimal i -center clustering radius of P .
- (B) For any k , let r_{opt}^k be the radius of the optimal k -center clustering of P . Then, for any constant c , $r_{\text{opt}}^{O(c^d k)} \leq r_{\text{opt}}^k / c$.
- (C) Computing the optimal k -center clustering of the first $O(k/\varepsilon^d)$ points of the greedy permutation, after appropriate rescaling, results in a $(1 + \varepsilon)$ -approximation to the optimal k -center clustering of P .

2.2 Setup

Our algorithms operate on a (not necessarily finite) point set P in \mathbb{R}^d . We assume that we are given a compact subset of \mathbb{R}^d , called the *domain* and denoted \mathcal{D} , such that $P \subseteq \mathcal{D}$. Throughout we assume that \mathcal{D} is the unit hypercube $[0, 1]^d$. The set P (not necessarily finite) is contained in \mathcal{D} .

Given a query point $\mathbf{q} \in [0, 1]^d$, let $\text{nn}(\mathbf{q}, P) = \arg \min_{\mathbf{p} \in P} \|\mathbf{q} - \mathbf{p}\|$ denote the nearest neighbor (NN) of \mathbf{q} . We say a point \mathbf{x} is a $(1 + \varepsilon)$ -approximate nearest-neighbor (ANN) for \mathbf{q} if $\|\mathbf{q} - \mathbf{x}\| \leq (1 + \varepsilon) \|\mathbf{q} - \text{nn}(\mathbf{q}, P)\|$. We assume that the sole access to P is through “black-box” data structures T_{nn} and T_{ann} , which given a query point \mathbf{q} , return the NN and ANN, respectively, to \mathbf{q} in P .

3 Using proximity search to compute k -center clustering

The problem. Our purpose is to compute (or approximately compute) a k -center clustering of P through the ANN black box we have, where k is a given parameter between 1 and n .

3.1 Greedy permutation via NN queries: GreedyPermutNN

Let \mathbf{q}_0 be an arbitrary point in \mathcal{D} . Let ν_0 be its nearest-neighbor in P computed using the provided NN data structure T_{nn} . Let $b_0 = \text{ball}(\mathbf{q}_0, \|\mathbf{q}_0 - \nu_0\|)$ be the open ball of radius $\|\mathbf{q}_0 - \nu_0\|$ centered at \mathbf{q}_0 . Finally, let $G_0 = \{\nu_0\}$, and let $\mathcal{D}_0 = \mathcal{D} \setminus b_0$.

In the i th iteration, for $i > 0$, let \mathbf{q}_i be the point in \mathcal{D}_{i-1} farthest away from G_{i-1} . Formally, this is the point in \mathcal{D}_{i-1} that maximizes $d(\mathbf{q}_i, G_{i-1})$, where $d(\mathbf{q}, X) = \min_{\mathbf{c} \in X} \|\mathbf{c} - \mathbf{q}\|$. Let $\nu_i = \text{nn}(\mathbf{q}_i, P)$ denote the nearest-neighbor ν_i to \mathbf{q}_i in P , computed using T_{nn} . Let

$$r_i = d(\mathbf{q}_i, G_{i-1}), \quad b_i = \text{ball}(\mathbf{q}_i, r_i), \quad G_i = G_{i-1} \cup \{\nu_i\}, \quad \text{and} \quad \mathcal{D}_i = \mathcal{D}_{i-1} \setminus b_i.$$

Left to its own devices, this algorithm computes a sequence of not necessarily distinct points ν_0, ν_1, \dots of P . If P is not finite then this sequence may also have infinitely many distinct points. Furthermore, $\mathcal{D}_0 \supseteq \mathcal{D}_1 \supseteq \dots$ is a sequence of outer approximations to P .

The execution of this algorithm is illustrated in Figure 1.

3.2 Analysis

Let $\mathcal{O} = \{o_1, \dots, o_k\}$ be an optimal set of k centers of P . Formally, it is a set of k points in P that minimizes the quantity $r_{\text{opt}}^k = \max_{q \in P} d(q, \mathcal{O})$. Specifically, r_{opt}^k is the smallest possible radius such that k closed balls of that radius centered at points in P , cover P . Our claim is that after $O(k)$ iterations of the algorithm **GreedyPermutNN**, the sequence of points provides a similar quality clustering of P .

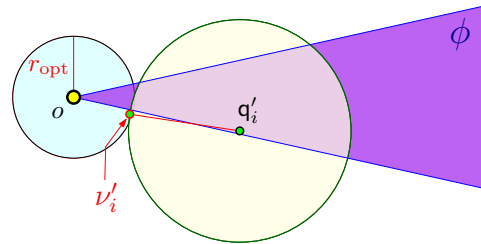
For any given point $p \in \mathbb{R}^d$ we can cover the sphere of directions centered at p by narrow cones of angular diameter at most $\pi/12$. We fix such a covering, denoting the set of cones by \mathcal{C}_p , and observe that the number of such cones is a constant c_d that depends on the dimension. Moreover, by simple translation we can transfer such a covering to be centered at any point $p' \in \mathbb{R}^d$.

► **Lemma 3.** *After $\mu = kc_d$ iterations, for any optimal center $o_i \in \mathcal{O}$, we have $d(o_i, G_\mu) \leq 3r_{\text{opt}}$, where $r_{\text{opt}} = r_{\text{opt}}^k$.*

Proof. If for any $j \leq \mu$, we have $r_j \leq 3r_{\text{opt}}$ then all the points of $\mathcal{D}_{j-1} \supseteq P$ are in distance at most $3r_{\text{opt}}$ from G_j , and the claim trivially holds as $\mathcal{O} \subseteq P$.

Let o be an optimal center and let P_o be the set of points of P that are closest to o among all the centers of \mathcal{O} , i.e., P_o is the cluster of o in the optimal clustering. Fix a cone ϕ from \mathcal{C}_o (ϕ 's apex is at o). Consider the output sequence ν_0, ν_1, \dots , and the corresponding query sequence q_0, q_1, \dots computed by the algorithm. In the following, we use the property of the algorithm that $r_1 \geq r_2 \geq \dots$, where $r_i = d(q_i, G_{i-1})$. A point q_j is *admissible* if (i) $\nu_j \in P_o$, and (ii) $q_j \in \phi$ (in particular, ν_j is not necessarily in ϕ).

We proceed to show that there are at most $O(1)$ admissible points for a fixed cone, which by a packing argument will imply the claim as every q_j is admissible for exactly one cone. Consider the induced subsequence of the output sequence restricted to the admissible points of ϕ : ν'_1, ν'_2, \dots , and let q'_1, q'_2, \dots be the corresponding query points used by the algorithm.

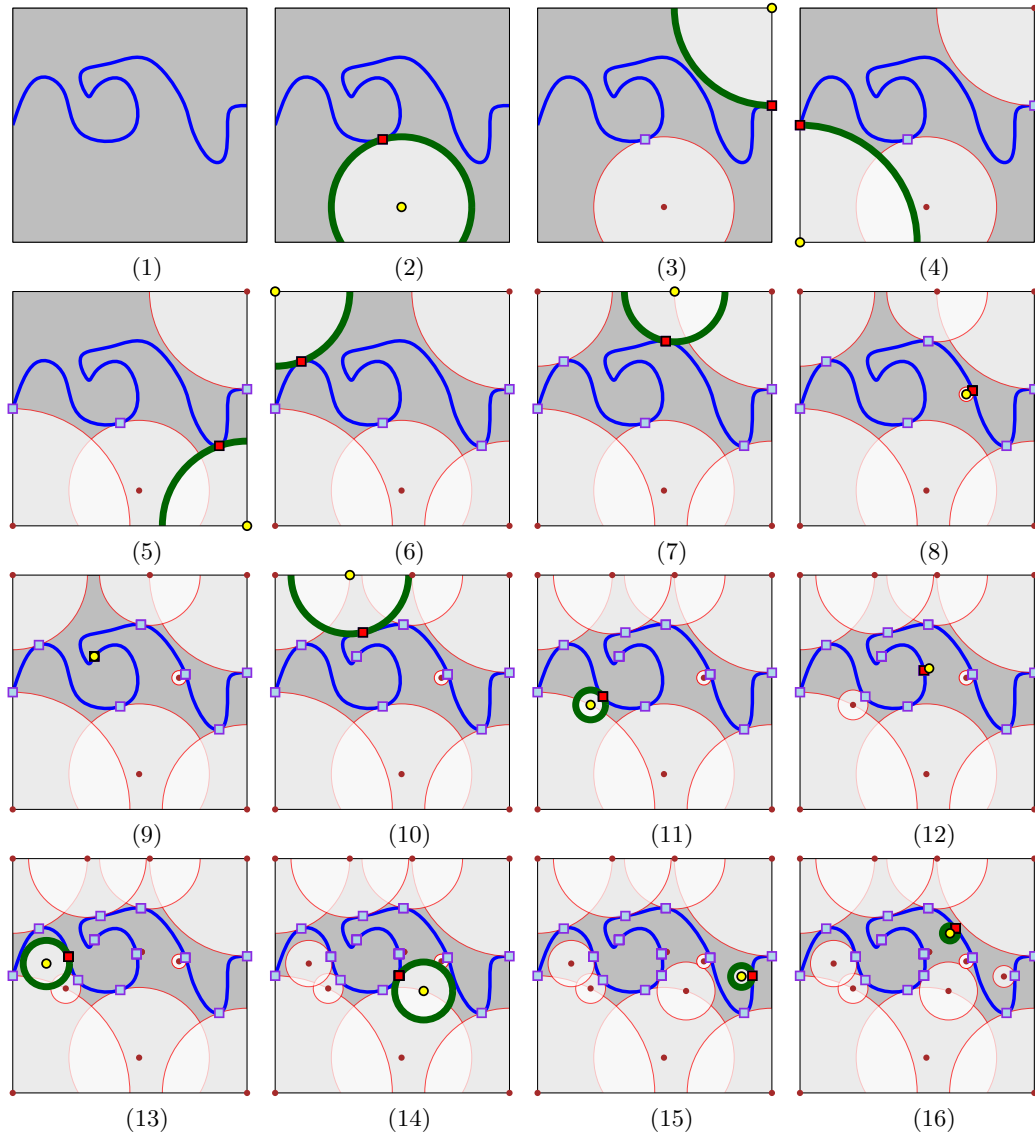


Formally, for a point ν'_i in this sequence, let $\text{iter}(i)$ be the iteration of the algorithm it was created. Thus, for all i , we have $q'_i = q_{\text{iter}(i)}$ and $\nu'_i = \nu_{\text{iter}(i)}$.

Observe that $P_o \subseteq P \cap \text{ball}(o, r_{\text{opt}})$. This implies that $\|\nu'_j - o\| \leq r_{\text{opt}}$, for all j .

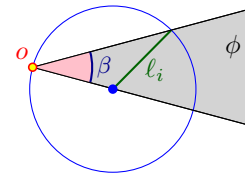
Let $\ell'_i = \|q'_i - \nu'_i\|$ and $r'_i = d(q'_i, G_{\text{iter}(i)-1})$. Observe that for $i > 1$, we have $\ell'_i \leq r'_i \leq \ell'_i + 2r_{\text{opt}}$, as $\nu'_{i-1} \in P_o$. Hence, if $\ell'_i \leq r_{\text{opt}}$, then $r'_i \leq 3r_{\text{opt}}$, and we are done. This implies that for any i, j , such that $1 < i < j$, it must be that $\|q'_i - q'_j\| \geq \ell'_i > r_{\text{opt}}$, as the algorithm carves out a ball of radius ℓ'_i around q'_i , and q'_j must be outside this ball.

By a standard packing argument, there can be only $O(1)$ points in the sequence q'_2, q'_3, \dots that are within distance at most $10r_{\text{opt}}$ from o . If there are no points beyond this distance, we are done. Otherwise, let $i > 1$ be the minimum index, such that q'_i is at distance larger than $10r_{\text{opt}}$ from o . We now prove that the points of $\phi \setminus \text{ball}(q'_i, \ell'_i)$ are of two types – those contained within $\text{ball}(o, 3r_{\text{opt}})$ and those that lie at distance greater than $(4/3)\ell'_i$ from o .



■ **Figure 1** An example of the execution of the algorithm **GreedyPermutNN** of Section 3.1.

To see this, observe that since the angle of the cone was chosen to be sufficiently small, $\text{ball}(\mathbf{q}'_i, \ell'_i)$ splits ϕ into two components, where all the points in the component containing o are distance $< 3r_{\text{opt}}$ from o . The minimum distance to o (from a point in the component not containing o) is realized when \mathbf{q}'_i is on the boundary of ϕ and o is on the boundary of $\text{ball}(\mathbf{q}'_i, \ell'_i)$. Then the distance of any point of $\phi \setminus \text{ball}(\mathbf{q}'_i, \ell'_i)$ from o is at least $2\ell'_i \cos(\beta) \geq 2\ell'_i \sqrt{3}/4 \geq 1.73\ell_i$, as the opening angle of the cone is at most $\pi/12$. (See the figure on the right.) The general case is somewhat more complicated as o might be in distance at most r_{opt} from the boundary of $\text{ball}(\mathbf{q}'_i, \ell'_i)$, but as $\ell_i \geq 10r_{\text{opt}}$, the claim still holds – we omit the tedious but straightforward calculations.



In particular, this implies that any later point \mathbf{q}'_k in the sequence (i.e., $k > i$) is either one of the $O(1)$ close points, or it must be far away, but then it is easy to argue that r'_k must be larger than r'_i , which is a contradiction as $r_2 \geq r_3 \geq \dots$ (as r'_i appears before r'_k in this sequence). ◀

The above lemma readily implies the following.

► **Theorem 4.** *Let $P \subseteq \mathcal{D}$ be a given set of points in \mathbb{R}^d (not necessarily finite), where \mathcal{D} is a bounded set in \mathbb{R}^d . Furthermore, assume that P can be accessed only via a data structure T_{nn} that answers exact nearest-neighbor (NN) queries on P . The algorithm **GreedyPermutNN**, described in Section 3.1, computes a permutation $\langle \nu_0, \dots \rangle$ of P , such that, for any $k > 0$, $P \subseteq \bigcup_{i=1}^{ck} \text{ball}(\nu_i, r_{\text{opt}}^k)$, where c is a constant (independent of k), and r_{opt}^k is the minimum radius of k balls (of the same radius) needed to cover P .*

The algorithm can be implemented, such that running it for i iterations, takes polynomial time in i and involves i calls to T_{nn} .

Proof. Using Lemma 2b in Lemma 3 implies the result. As for the running time, naively one needs to maintain the arrangement of balls inside the domain, and this can be done in polynomial time in the number of balls. ◀

► **Observation 5.** *If P is finite of size n , the above theorem implies that after $i \geq cn$ iterations, one can recover the entire point set P (as $r_{\text{opt}}^n = 0$). Therefore cn is an upper bound on the number of queries for any problem. Note however that in general our goal is to demonstrate when problems can be solved using a significantly smaller amount of NN queries.*

The above also implies an algorithm for approximating the diameter.

► **Lemma 6.** *Consider the setting of Theorem 4 using an exact nearest-neighbor oracle. Suppose that the algorithm is run for $m = c_d + 1$ iterations, and let ν_1, \dots, ν_m be the set of output centers and r_1, \dots, r_m be the corresponding distances. Then, $\text{diam}(P)/3 \leq \max(\text{diam}(\nu_1, \dots, \nu_m), r_m) \leq 3 \cdot \text{diam}(P)$.*

Proof. Since the discrete one-center clustering radius lies in the interval $[\text{diam}(P)/2, \text{diam}(P)]$, the proof of Lemma 3 implies that $r_m \leq 3r_{\text{opt}} \leq 3 \cdot \text{diam}(P)$. Moreover, each ν_i is in P , and so $\text{diam}(\nu_1, \dots, \nu_m) \leq \text{diam}(P)$. Thus the upper bound follows.

For the lower bound, observe that if $\text{diam}(\nu_1, \dots, \nu_m) < \text{diam}(P)/3$, as well as $r_m < \text{diam}(P)/3$, then it must be true that $P \subseteq \mathcal{D}_{m-1} \subseteq \bigcup_{j=1}^l \text{ball}(\nu_j, r_m)$ has diameter less than $\text{diam}(P)$, a contradiction. ◀

3.3 Using approximate nearest-neighbor search

If we are using an ANN black box T_{ann} to implement the algorithm, one can no longer scoop away the ball $b_i = \text{ball}(\mathbf{q}_i, \|\mathbf{q}_i - \nu_i\|)$ at the i th iteration, as it might contain some of the points of P . Instead, one has to be more conservative, and use the ball $b'_i = \text{ball}(\mathbf{q}_i, (1 - \varepsilon)\|\mathbf{q}_i - \nu_i\|)$. Now, we might need to perform several queries till the volume being scooped away is equivalent to a single exact query.

Specifically, let P be a finite set, and consider its associated *spread*: $\Phi = \frac{\text{diam}(\mathcal{D}_0)}{\min_{\mathbf{p}, \mathbf{x} \in P} \|\mathbf{p} - \mathbf{x}\|}$. We can no longer claim, as in Lemma 3, that each cone would be visited only one time (or constant number of times). Instead, it is easy to verify that each query point in the cone, shrinks the diameter of the domain restricted to the cone by a factor of roughly ε . As such, at most $O(\log_{1/\varepsilon} \Phi)$ query points would be associated with each cone.

► **Corollary 7.** *Consider the setting of Theorem 4, with the modification that we use a $(1 + \varepsilon)$ -ANN data structure T_{ann} to access P . Then, for any k , $P \subseteq \bigcup_{i=1}^{f(k)} \text{ball}(\nu_i, r_{\text{opt}}^k)$, where $f(k) = O(k \log_{1/\varepsilon} \Phi)$.*

3.4 Discussion

Outer approximation. As implied by the algorithm description, one can think about the algorithm providing an outer approximation to the set: $\mathcal{D}_1 \supseteq \mathcal{D}_2 \supseteq \dots \supseteq P$. As demonstrated in Figure 1, the sequence of points computed by the algorithm seems to be a reasonable greedy permutation of the underlying set. However, the generated outer approximation seems to be inferior. If the purpose is to obtain a better outer approximation, a better strategy may be to pick the i th query point \mathbf{q}_i as the point inside \mathcal{D}_i farthest away from $\partial\mathcal{D}_{i-1} \cup G_{i-1}$

Implementation details. We have not spent any effort to describe in detail the algorithm of Theorem 4, mainly because an implementation of the exact version seems quite challenging in practice. A more practical approach would be to describe the uncovered domain \mathcal{D}_i approximately, by approximating from the inside, every ball b_i by an $O(1/\varepsilon^d)$ grid of cubes, and maintaining these cubes using a (compressed) quadtree. This provides an explicit representation of the complement of the union of the approximate balls. Next, one would need to maintain for every free leaf of this quadtree, a list of points of G_i that might serve as its nearest neighbors – in the spirit of approximate Voronoi diagrams [10].

4 Convex-hull membership queries via proximity queries

Let P be a set of n points in \mathbb{R}^d , let Δ denote P 's diameter, and let $\varepsilon > 0$ be a prespecified parameter. We assume that the value of Δ is known, although a constant approximation to this value is sufficient for our purposes. (See Lemma 6 on how to compute this under reasonable assumptions.)

Let $\mathcal{C} = \mathcal{CH}(P)$ denote P 's convex hull. Given a query point $\mathbf{q} \in \mathbb{R}^d$, the task at hand is to determine whether \mathbf{q} is in \mathcal{C} . As before, we assume that our only access to P is via an ANN data structure. There are two possible outputs:

- (A) IN: if $\mathbf{q} \in \mathcal{C}$, and
- (B) OUT: if \mathbf{q} is at distance greater than $\varepsilon\Delta$ from \mathcal{C} ,

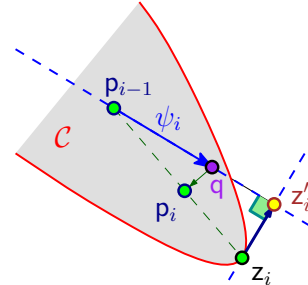
Either answer is acceptable if \mathbf{q} lies within distance $\varepsilon\Delta$ of $\partial\mathcal{C}$.

4.1 Convex hull membership queries using exact extremal queries

We first solve the problem using exact extremal queries and then later show these queries can be answered approximately with ANN queries.

4.1.1 The algorithm

We construct a sequence of points p_0, p_1, \dots each guaranteed to be in the convex hull \mathcal{C} of P and use them to determine whether $q \in \mathcal{C}$. The algorithm is as follows. Let p_0 be an arbitrary point of P . For $i > 0$, in the i th iteration, the algorithm checks whether $\|p_{i-1} - q\| \leq \varepsilon\Delta$, and if so the algorithm outputs IN and stops.



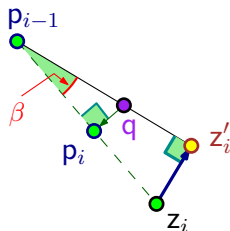
Otherwise, consider the ray ψ_i emanating from p_{i-1} in the direction of q . The algorithm computes the point $z_i \in P$ that is extremal in the direction of this ray. If the projection z'_i of z_i on the line supporting ψ_i is between p_{i-1} and q , then q is outside the convex-hull \mathcal{C} , and the algorithm stops and returns OUT. Otherwise, the algorithm sets p_i to be the projection of q on the line segment $p_{i-1}z_i$, and continues to the next iteration. (See the figure on the right and Figure 2.)

For a suitable constant c (see Lemma 9), if the algorithm does not terminate after c/ε^2 iterations, it stops and returns OUT.

4.1.2 Analysis

► **Lemma 8.** *If the algorithm runs for more than i iterations, then $d_i < \left(1 - \frac{\varepsilon^2}{2}\right)d_{i-1}$, where $d_i = \|q - p_i\|$.*

Proof. By construction, p_i, p_{i-1} , and q form a right angle triangle. The proof now follows by a direct trigonometric argument. Consider Figure 2. We have the following properties:



- (A) The triangles $\triangle p_{i-1}z'_iz_i$ and $\triangle p_{i-1}p_iq$ are similar.
- (B) Because the algorithm has not terminated in the i th iteration, $\|p_{i-1} - q\| > \varepsilon\Delta$.
- (C) The point q must be between p_{i-1} and z'_i , as otherwise the algorithm would have. Thus, $\|p_{i-1} - z'_i\| \geq \|p_{i-1} - q\| > \varepsilon\Delta$.
- (D) We have $\|p_{i-1} - z_i\| \leq \Delta$, since both points are in \mathcal{C} .

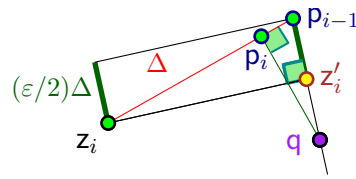
■ **Figure 2**

We conclude that $\cos \beta = \frac{\|p_{i-1} - z'_i\|}{\|p_{i-1} - z_i\|} > \frac{\varepsilon\Delta}{\Delta} = \varepsilon$. Now, we have

$$\begin{aligned} \|q - p_i\| &= \|q - p_{i-1}\| \sin \beta = \|q - p_{i-1}\| \sqrt{1 - \cos^2 \beta} < \sqrt{1 - \varepsilon^2} \|q - p_{i-1}\| \\ &< \left(1 - \frac{\varepsilon^2}{2}\right) \|q - p_{i-1}\|, \end{aligned}$$

since $(1 - \varepsilon^2/2)^2 > 1 - \varepsilon^2$. ◀

► **Lemma 9.** *Either the algorithm stops within $O(1/\varepsilon^2)$ iterations with a correct answer, or the query point lies at distance more than $\varepsilon\Delta$ from the convex hull \mathcal{C} ; in the latter case, since the algorithm says OUT its output is correct.*



■ **Figure 3** Worse case if extremal queries are approximate.

Proof. If the algorithm stops before it completes the maximum number of iterations, it can be verified that the output is correct as there is an easy certificate for this in each of the possible cases.

Otherwise, suppose that the query point is within $\epsilon\Delta$ of \mathcal{C} . We argue that this leads to a contradiction; thus the query point must be more than $\epsilon\Delta$ far from \mathcal{C} and the output of the algorithm is correct. Observe that d_i is a monotone decreasing quantity that starts at values $\leq \Delta$ (i.e, $d_0 \leq \Delta$), since otherwise the algorithm terminates after the first iteration, as z'_1 would be between q and p_0 on ψ_1 .

Consider the j th epoch to be block of iterations of the algorithm, where $2^{-j}\Delta < d_i \leq 2^{-j+1}\Delta$. Following the proof of Lemma 8, one observes that during the j th epoch one can set $\epsilon_j = 1/2^j$ in place of ϵ , and using the argument it is easy to show that the j th epoch lasts $O(1/\epsilon_j^2)$ iterations. By assumption, since the algorithm continued for the maximum number of iterations we have $d_i > \epsilon\Delta$, and so the maximum number of epochs is $\lceil \lg(1/\epsilon) \rceil$. As such, the total number of iterations is $\sum_{j=1}^{\lceil \lg(1/\epsilon) \rceil} O(1/\epsilon_j^2) = O(1/\epsilon^2)$. Since the algorithm did not stop, this is a contradiction. ◀

4.1.3 Approximate extremal queries

For our purposes, approximate extremal queries on P are sufficient.

► **Definition 10.** A data structure provides ϵ -approximate extremal queries for P , if for any query unit vector v , it returns a point p , such that

$$\forall x \in P, \quad \langle v, x \rangle \leq \langle v, p \rangle + \epsilon \cdot \text{diam}(P),$$

where $\langle v, x \rangle$ denotes the dot-product of v with x .

One can now modify the algorithm of Section 4.1.1 to use, say, $\epsilon/4$ -approximate extremal queries on P . Indeed, one modifies the algorithm so it stops only if z_i is on the segment $p_{i-1}q$, and it is in distance more than $\epsilon\Delta/4$ away from q . Otherwise the algorithm continues. It is straightforward but tedious to prove that the same algorithm performs asymptotically the same number of iterations (intuitively, all that happens is that the constants get slightly worse). The worse case as far progress in a single iteration is depicted in Figure 3.

► **Lemma 11.** The algorithm of Section 4.1.1 can be modified to use $\epsilon/4$ -approximate extremal queries and output a correct answer after performing $O(1/\epsilon^2)$ iterations.

4.2 Convex-hull membership via ANN queries

4.2.1 Approximate extremal queries via ANN queries

The basic idea is to replace the extremal empty half-space query, by an ANN query. Specifically, a $(1 + \delta)$ -ANN query performed at q returns us a point p , such that

$$\forall x \in P, \quad \|q - p\| \leq (1 + \delta) \|q - x\|.$$

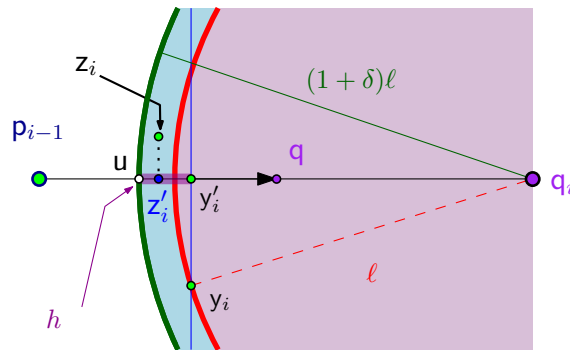


Figure 4 Illustration of the proof of Lemma 12.

Namely, $\text{ball}\left(\mathbf{q}, \frac{\|\mathbf{q}-\mathbf{p}\|}{1+\delta}\right)$ does not contain any points of P . Locally, a ball looks like a halfspace, and so by taking the query point to be sufficiently far and the approximation parameter to be sufficiently small, the resulting empty ball and its associated ANN can be used as the answer to an extremal direction query.

4.2.2 The modified algorithm

Assume the algorithm is given a data structure T_{ann} that can answer $(1 + \delta)$ -ANN queries on P . Also assume that it is provided with an initial point $\mathbf{p}_0 \in P$, and a value Δ' that is, say, a 2-approximation to $\Delta = \text{diam}(P)$, that is $\Delta \leq \Delta' \leq 2\Delta$.

In the i th iteration, the algorithm considers (again) the ray ψ_i starting from \mathbf{p}_i , in the direction of \mathbf{q} . Let \mathbf{q}_i be the point within distance, say,

$$\tau = c\Delta'/\varepsilon \tag{4.1}$$

from \mathbf{p}_{i-1} along ψ_i , where c is an appropriate constant to be determined shortly. Next, let \mathbf{z}_i be the $(1 + \delta)$ -ANN returned by T_{ann} for the query point \mathbf{q}_i , where the value of δ would be specified shortly. The algorithm now continues as before, by setting \mathbf{p}_i to be the nearest point on $\mathbf{p}_{i-1}\mathbf{z}_i$ to \mathbf{q} . Naturally, if $\|\mathbf{q} - \mathbf{p}_i\|$ falls below $\varepsilon\Delta'/2$, the algorithm stops, and returns IN, and otherwise the algorithm continues to the next iteration. As before, for a suitable constant c , if the algorithm does not terminate after c/ε^2 iterations, it stops and returns OUT.

4.2.3 Analysis

► **Lemma 12.** *Let $0 < \varepsilon \leq 1$ be a prespecified parameter, and let $\delta = \varepsilon^2/(32 - \varepsilon)^2 = O(\varepsilon^2)$. Then, a $(1 + \delta)$ -ANN query done using \mathbf{q}_i (as defined in Section 4.2.2), returns a point \mathbf{z}_i which is a valid ε -approximate extremal query on P , in the direction of ψ_i .*

Proof. Consider the extreme point $\mathbf{y}_i \in P$ in the direction of ψ_i . Let \mathbf{y}'_i be the projection of \mathbf{y}_i to the segment $\mathbf{p}_{i-1}\mathbf{q}_i$, and let $\ell = \|\mathbf{q}_i - \mathbf{y}_i\|$. See Figure 4.

The $(1 + \delta)$ -ANN to \mathbf{q}_i (i.e., the point \mathbf{z}_i), must be inside the ball $b = \text{ball}(\mathbf{q}_i, (1 + \delta)\ell)$, and let \mathbf{z}'_i be its projection to the segment $\mathbf{p}_{i-1}\mathbf{q}_i$.

Now, if we interpret \mathbf{z}_i as the returned answer for the approximate extremal query, then the error is the distance $\|\mathbf{z}'_i - \mathbf{y}'_i\|$, which is maximized if \mathbf{z}'_i is as close to \mathbf{p}_{i-1} as possible. In

particular, let \mathbf{u} be the point in distance $(1 + \delta)\ell$ from \mathbf{q}_i along the segment $\mathbf{p}_{i-1}\mathbf{q}_i$. We then have that $\|\mathbf{z}'_i - \mathbf{y}'_i\| \leq h = \|\mathbf{u} - \mathbf{y}'_i\|$. Now, since $\|\mathbf{y}'_i - \mathbf{y}_i\| \leq \|\mathbf{p}_{i-1} - \mathbf{y}_i\| \leq \Delta'$, we have

$$\begin{aligned} h &= \|\mathbf{u} - \mathbf{y}'_i\| \leq (1 + \delta)\ell - \|\mathbf{y}'_i - \mathbf{q}_i\| = (1 + \delta)\ell - \sqrt{\ell^2 - \|\mathbf{y}'_i - \mathbf{y}_i\|^2} \\ &\leq (1 + \delta)\ell - \sqrt{\ell^2 - (\Delta')^2} = \frac{(1 + \delta)^2\ell^2 - \ell^2 + (\Delta')^2}{(1 + \delta)\ell + \sqrt{\ell^2 - (\Delta')^2}} \leq \frac{(2\delta + \delta^2)\ell^2 + (\sqrt{\delta}\ell)^2}{\ell} \\ &\leq \frac{4\delta\ell^2}{\ell} = 4\delta\ell, \end{aligned}$$

since $\delta \leq 1$, and assuming that $\Delta' \leq \sqrt{\delta}\ell$. For our purposes, we need that $4\delta\ell \leq \varepsilon\Delta$. Both of these constraints translate to the inequalities, $\left(\frac{\Delta'}{\ell}\right)^2 \leq \delta \leq \frac{\varepsilon\Delta}{4\ell}$. Observe that, by the triangle inequality, it follows that

$$\ell = \|\mathbf{q}_i - \mathbf{y}_i\| \leq \|\mathbf{q}_i - \mathbf{p}_{i-1}\| + \|\mathbf{p}_{i-1} - \mathbf{y}_i\| \leq \tau + \Delta.$$

A similar argument implies that $\ell \geq \tau - \Delta$. In particular, it is enough to satisfy the constraint $\left(\frac{\Delta'}{\tau - \Delta}\right)^2 \leq \delta \leq \frac{\varepsilon\Delta}{4(\tau + \Delta)}$, which is satisfied if $\left(\frac{\Delta'}{\tau - \Delta}\right)^2 \leq \delta \leq \frac{\varepsilon\Delta'/2}{4(\tau + \Delta')}$, as $\Delta \leq \Delta' \leq 2\Delta$. Substituting the value of $\tau = c\Delta'/\varepsilon$, see Eq. (4.1), this is equivalent to $\left(\frac{1}{c/\varepsilon - 1}\right)^2 \leq \delta \leq \frac{\varepsilon/2}{4(c/\varepsilon + 1)}$, which holds for $c = 32$, as can be easily verified, and setting $\delta = \varepsilon^2/(32 - \varepsilon)^2 = O(\varepsilon^2)$. ◀

► **Theorem 13.** *Given a set P of n points in \mathbb{R}^d , let $\varepsilon \in (0, 1]$ be a parameter, and let Δ' be a constant approximation to the diameter of P . Assume that you are given a data structure that can answer $(1 + \delta)$ -ANN queries on P , for $\delta = O(\varepsilon^2)$. Then, given a query point \mathbf{q} , one can decide, by performing $O(1/\varepsilon^2)$ $(1 + \delta)$ -ANN queries whether \mathbf{q} is inside the convex-hull $\mathcal{C} = \mathcal{CH}(P)$. Specifically, the algorithm returns*

- IN: if $\mathbf{q} \in \mathcal{C}$, and
- OUT: if \mathbf{q} is more than $\varepsilon\Delta$ away from \mathcal{C} , where $\Delta = \text{diam}(P)$.

The algorithm is allowed to return either answer if $\mathbf{q} \notin \mathcal{C}$, but the distance of \mathbf{q} from \mathcal{C} is at most $\varepsilon\Delta$.

5 Density clustering

5.1 Definition

Given a set P of n points in \mathbb{R}^d , and a parameter μ , with $1 \leq \mu \leq n$, we are interested in computing a set $C \subseteq P$ of “centers”, such that each center is assigned at most μ points, and the number of centers is (roughly) n/μ . In addition, we require that:

- (A) A point of P is assigned to its nearest neighbor in C (i.e., C induces a *Voronoi partition* of P).
- (B) The centers come from the original point set.

Intuitively, this clustering tries to capture the local density – in areas where the density is low, the clusters can be quite large (in the volume they occupy), but in regions with high density the clusters have to be tight and relatively “small”.

Formally, given a set of centers C , and a center $c \in C$, its *cluster* is

$$P_c = \left\{ p \in P \mid \|c - p\| < d(p, C \setminus \{c\}) \right\},$$

where $d(p, X) = \min_{x \in X} \|p - x\|$ (and assuming for the sake of simplicity of exposition that all distances are distinct). The resulting *clustering* is $\Pi(P, C) = \{P_c \mid c \in C\}$. A set of points P , and a set of centers $C \subseteq P$ is a μ -*density clustering* of P if for any $c \in C$, we have $|P_c| \leq \mu$. As mentioned, we want to compute a balanced partitioning, i.e., one where the number of centers is roughly n/μ . We show below that this is not always possible in high enough dimensions.

5.1.1 A counterexample in high dimension

► **Lemma 14** (For proof see [12]). *For any integer $n > 0$, there exists a set P of n points in \mathbb{R}^n , such that for any $\mu < n$, a μ -density clustering of P must use at least $n - \mu + 1$ centers.*

5.2 Algorithms

5.2.1 Density clustering via nets

► **Lemma 15.** *For any set of n points P in \mathbb{R}^d , and a parameter $\mu < n$, there exists a μ -density clustering with $O\left(\frac{n}{\mu} \log \frac{n}{\mu}\right)$ centers (the O notation hides constants that depend on d).*

Proof. Consider the hypercube $[-1, 1]^d$. Cover its outer faces (which are $(d-1)$ -dimensional hypercubes) by a grid of side length $1/3\sqrt{d}$. Consider a cell C in this grid – it has diameter $\leq 1/3$, and it is easy to verify that the cone $\phi = \{tp \mid p \in C, t \geq 0\}$ formed by the origin and C has angular diameter $< \pi/3$. This results in a set \mathcal{C} of $N = O(d^d)$ cones covering \mathbb{R}^d .

Fix a cone $\phi \in \mathcal{C}$. For a point $p \in \mathbb{R}^d$, let ϕ_p denote the translation of ϕ such that p is its apex. Note that ϕ is formed by the intersection of $2(d-1)$ halfspaces. As such, the range space consisting of all ranges ϕ_p , such that $p \in \mathbb{R}^d$, has VC dimension at most $d' = O(d^2 \log d)$ [10, Theorem 5.22]. For a radius r and point p , let a ϕ -*slice* be the set $s_\phi(p, r) = \phi_p \cap \text{ball}(p, r)$, i.e. the set formed by intersecting ϕ_p with a ball centered at p and of radius r . The range space of all ϕ -slices, $S_\phi = \{s_\phi(p, r) \mid p \in \mathbb{R}^d, r \geq 0\}$, has VC dimension $d'' = O(d + 2 + d') = O(d^2 \log d)$, since the VC dimension of balls in \mathbb{R}^d is $d + 2$, and one can combine range spaces as done above, see the book [10] for background on this.

Now, for $\varepsilon = (\mu/N)/n = \mu/(nN)$, consider an ε -net R of the point set P for ϕ -slices. The size of such a net is $|R| = O((d''/\varepsilon) \log \varepsilon^{-1}) = O\left(\frac{nNd^2 \log d}{\mu} \log \frac{nN}{\mu}\right) = O\left(d^{O(d)} \frac{n}{\mu} \log \frac{n}{\mu}\right) = O\left(\frac{n}{\mu} \log \frac{n}{\mu}\right)$, by the ε -net theorem.

Consider a point $p \in P$ that is in R . Let ν_ϕ be the nearest point to p in the set $\{R \setminus \{p\}\} \cap \phi_p$. The key observation is that any point in $P \cap \phi_p$ that is farther away from p than ν_ϕ , is closer to ν_ϕ than to p ; that is, only points closer to p than ν_ϕ might be assigned to p in the Voronoi clustering. Since R is an ε -net for ϕ -slices, $s_\phi(p, \|p - \nu_\phi\|) = \phi_p \cap \text{ball}(p, \|p - \nu_\phi\|)$, contains at most $\varepsilon n = \mu/N$ points of P . It follows that at most μ/N points of $P \cap \phi_p$ are assigned to the cluster associated with p . By summing over all N cones, at most $(\mu/N)N = \mu$ points are assigned to p , as desired. ◀

5.2.2 The planar case

► **Lemma 16** (For proof see [12]). *For any set of n points P in \mathbb{R}^2 , and a parameter μ with $1 \leq \mu \leq n$, there exists a μ -density clustering with $O(n/\mu)$ centers.*

Acknowledgments. N. K. would like to thank Anil Gannepalli for telling him about Atomic Force Microscopy.

References

- 1 L.-E. Andersson and N. F. Stewart. *Introduction to the Mathematics of Subdivision Surfaces*. SIAM, 2010.
- 2 G. Binnig, C. F. Quate, and Ch. Gerber. Atomic force microscope. *Phys. Rev. Lett.*, 56:930–933, Mar 1986.
- 3 J. F. Blinn. A generalization of algebraic surface drawing. *ACM Trans. Graphics*, 1:235–256, 1982.
- 4 J.-D. Boissonnat, L. J. Guibas, and S. Oudot. Learning smooth shapes by probing. *Comput. Geom. Theory Appl.*, 37(1):38–58, 2007.
- 5 K. L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algo.*, 6(4), 2010.
- 6 R. Cole and C. K. Yap. Shape from probing. *J. Algorithms*, 8(1):19–38, 1987.
- 7 T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. In *Proc. 20th Annu. ACM Sympos. Theory Comput.* (STOC), pages 434–444, 1988.
- 8 A. Goel, P. Indyk, and K. R. Varadarajan. Reductions among high dimensional proximity problems. In *Proc. 12th ACM-SIAM Sympos. Discrete Algs.* (SODA), pages 769–778, 2001.
- 9 T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.
- 10 S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Mathematical Surveys and Monographs*. Amer. Math. Soc., 2011.
- 11 S. Har-Peled, P. Indyk, and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Theory Comput.*, 8:321–350, 2012. Special issue in honor of Rajeev Motwani.
- 12 S. Har-Peled, N. Kumar, D. Mount, and B. Raichel. Space exploration via proximity search. *CoRR*, abs/1412.1398, 2014.
- 13 S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. *SIAM J. Comput.*, 35(5):1148–1184, 2006.
- 14 P. Indyk. Nearest neighbors in high-dimensional spaces. In J. E. Goodman and J. O’Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 39, pages 877–892. CRC Press LLC, 2nd edition, 2004.
- 15 B. Kalantari. A characterization theorem and an algorithm for A convex hull problem. *CoRR*, abs/1204.1873, 2012.
- 16 B. B. Mandelbrot. *The fractal geometry of nature*. Macmillan, 1983.
- 17 J. M. Mulvey and M. P. Beck. Solving capacitated clustering problems. *Euro. J. Oper. Res.*, 18:339–348, 1984.
- 18 F. Panahi, A. Adler, A. F. van der Stappen, and K. Goldberg. An efficient proximity probing algorithm for metrology. In *Proc. IEEE Int. Conf. Autom. Sci. Engin. (CASE)*, pages 342–349, 2013.
- 19 S. S. Skiena. Problems in geometric probing. *Algorithmica*, 4:599–605, 1989.
- 20 S. S. Skiena. Geometric reconstruction problems. In J. E. Goodman and J. O’Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 26, pages 481–490. CRC Press LLC, Boca Raton, FL, 1997.

- 21 R. M. Smelik, K. J. De Kraker, S. A. Groenewegen, T. Tutenel, and R. Bidarra. A survey of procedural methods for terrain modelling. In *Proc. of the CASA Work. 3D Adv. Media Gaming Simul.*, 2009.
- 22 Wikipedia. Atomic force microscopy – wikipedia, the free encyclopedia, 2014.

Star Unfolding from a Geodesic Curve

Stephen Kiazzyk and Anna Lubiw

Cheriton School of Computer Science
University of Waterloo, Waterloo, ON, Canada
{skiazk,alubiw}@uwaterloo.ca

Abstract

There are two known ways to unfold a convex polyhedron without overlap: the star unfolding and the source unfolding, both of which use shortest paths from vertices to a source point on the surface of the polyhedron. Non-overlap of the source unfolding is straightforward; non-overlap of the star unfolding was proved by Aronov and O’Rourke in 1992. Our first contribution is a much simpler proof of non-overlap of the star unfolding.

Both the source and star unfolding can be generalized to use a simple geodesic curve instead of a source point. The *star unfolding from a geodesic curve* cuts the geodesic curve and a shortest path from each vertex to the geodesic curve. Demaine and Lubiw conjectured that the star unfolding from a geodesic curve does not overlap. We prove a special case of the conjecture. Our special case includes the previously known case of unfolding from a geodesic loop. For the general case we prove that the star unfolding from a geodesic curve can be separated into at most two non-overlapping pieces.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases unfolding, convex polyhedra, geodesic curve

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.390

1 Introduction

An *unfolding* of a polyhedron \mathcal{P} is obtained by cutting the surface of \mathcal{P} in such a way that it can be flattened into the plane, forming a single polygon. A main goal is to find unfoldings that are simple, that is, do not self-overlap. If we have an unfolding that does not overlap, we can make a model of the polyhedron from a sheet of paper by cutting the outline of the polygon and gluing appropriate pairs of edges together.

Unfoldings have fascinated people since the time of Dürer’s beautiful examples [6]. A long-standing open question is whether every convex polyhedron has a non-overlapping *edge unfolding*, where only edges of the polyhedron are cut. However, if we allow cuts that cross faces—which is the model used in this paper—then there are several known methods to unfold convex polyhedra without overlap.

Unfoldings of polyhedra have applications in product manufacturing, for constructing a 3-dimensional object from a sheet of metal or plastic, and also in graphics for applying texture mapping, where 2-dimensional image coordinates must be assigned to points on a 3-dimensional model. Unfolding is also applied as a theoretical tool for the study of shortest paths on the surface of a polyhedron.

There are two main methods known to unfold convex polyhedra without overlap, both defined in terms of shortest paths on the polyhedron surface to a “source” point x . A fundamental property of shortest paths on the surface of a convex polyhedron is that they unfold to straight-line segments. More generally, any *geodesic* (or *locally shortest*) path on the surface of a polyhedron unfolds to a straight-line segment.



© Stephen Kiazzyk and Anna Lubiw;

licensed under Creative Commons License CC-BY

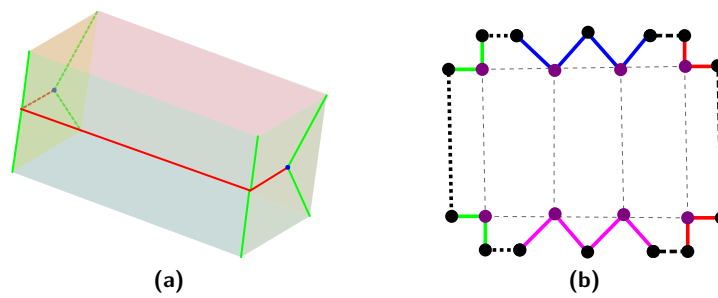
31st International Symposium on Computational Geometry (SoCG’15).

Editors: Lars Arge and János Pach; pp. 390–404



Leibniz International Proceedings in Informatics

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



■ **Figure 1** An example star unfolding from a geodesic curve (in red) on a rectangular box. Faint dashed lines inside the unfolding indicate some of the original edges.

The *star unfolding* is obtained by cutting a shortest path from every vertex of the polyhedron to the point x . The cuts form a star at x , hence the name. The *source unfolding* is obtained by cutting the *ridge tree* (also known as the *cut locus*), the locus of points that have more than one shortest path to x . It is easy to see that the source unfolding does not overlap, because all the shortest paths from x to points on the surface unfold into straight lines radiating from x . The star and source unfoldings are dual in the sense that the pieces of the surface delimited by the ridge tree and the vertex-to-source shortest path cuts are joined at “opposite ends”: in the star unfolding the pieces are joined at the ridge tree; and in the source unfolding the pieces are joined at the source point x .

Alexandrov thought the star unfolding might overlap (see [5]); the surprising result that it does not was proved by Aronov and O’Rourke [2]. Their proof is by induction and involves combining two vertices into one in a kind of “Alexandrov surgery.” The proof is long, and Demaine and O’Rourke, in their book, “Geometric Folding Algorithms” [5] call the proof “not straightforward” and omit it.

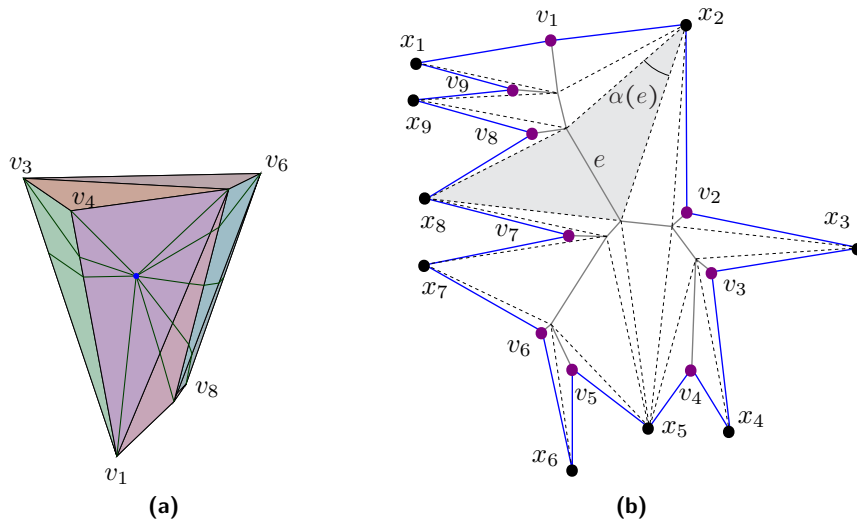
Our first main result is a new proof of non-overlap of the star unfolding that is more straightforward. We do not modify the polyhedron or appeal to Alexandrov’s gluing theorem.

The star and source unfoldings can be generalized in a natural way by using a simple geodesic curve λ instead of a source point x . For the source unfolding, we cut the ridge tree (the locus of points that have more than one shortest path to the curve λ), and for the star unfolding we cut the curve λ itself and a shortest path from every vertex of the polyhedron to λ . See Figure 1. Such generalizations were first introduced by Itoh et al. [9] who proved non-overlap results for the case of closed curves (see also [11, 8]). Demaine and Lubiw [4, Lemma 1] proved non-overlap of the source unfolding for (open) geodesic curves, and conjectured the same for the star unfolding.

Our second main result is a special case of this conjecture: we prove that the star unfolding from a geodesic curve unfolds without overlap if the curve is “balanced” (as defined in Section 3). The balance condition automatically holds for the point star unfolding.

We give two implications of our second result. The first is that the star unfolding does not overlap if the curve is a *geodesic loop*, meaning the curve endpoints a and b are (almost) coincident. In the limit when $a = b$ the unfolding consists of two pieces joined at a point. This gives an alternative to the result of Itoh et al. [9] that the star unfoldings of the inside and the outside of a geodesic loop do not overlap, and that the two pieces may be joined into one non-overlapping piece. Their proof that the outside unfolds without overlap had a flaw; our result repairs it. We can extend this result (and our other results) from geodesics to *quasigeodesics*, to be defined below.

The second consequence of our result is that every star unfolding from a geodesic curve



■ **Figure 2** The star unfolding from a point. **(a)** The polyhedron, the source point, and the shortest paths from vertices to the source point. **(b)** The corresponding star unfolding. The ridge tree is shown in grey. The kites are shown with dashed lines. The kite on ridge tree edge e is shaded, and its source angle $\alpha(e)$ is indicated. Note that the unfolding shows the inside surface of the polyhedron.

can be cut into two pieces such that each piece is non-overlapping. The extra cuts consist of shortest paths from a point on the ridge tree to the curve.

To conclude this section we mention a few reasons to explore new unfoldings of convex polyhedra. One is to find “nicer” unfoldings. As the number of vertices of a polyhedron increases, the star unfolding from a point becomes very spiky with many sharp angles, for example see [1, Figure 7]. By contrast, in the star unfolding from a geodesic curve many (or in some cases, all) vertices may have shortest paths to interior points of the curve, resulting in many 90° angles and fewer sharp angles. New unfolding methods for convex polyhedra might also shed light on the case of non-convex polyhedra. Having a larger repertoire of unfoldings also opens the door to optimization, e.g., minimize the area of a bounding box of the unfolding, or minimize the total cut length, or maximize the minimum angle.

Geodesic star unfoldings may also have implications for the conjecture [3] that every convex polyhedron has a *general zipper unfolding*, a non-overlapping unfolding formed by cutting a single path on the polyhedron surface. If quasigeodesic star unfoldings do not overlap, then it would suffice to find a quasigeodesic curve that goes through all the vertices.

1.1 Preliminaries and Definitions

► **Definition 1.** Let \mathcal{P} be a convex polyhedron, and let x be a point on \mathcal{P} . The *star unfolding*, S_x , is a 2-dimensional polygon formed by cutting \mathcal{P} along a shortest path from every vertex of \mathcal{P} to x , and flattening the result into the plane. See Figure 2. Note that there is a choice of cuts if a vertex has more than one shortest path to x .

If \mathcal{P} has n vertices, then the polygon S_x will have $2n$ vertices in general, and $2(n - 1)$ vertices if x is located at a vertex of \mathcal{P} . The vertices of S_x alternate around the boundary between *vertex images*, denoted v_i , that correspond to the vertices of \mathcal{P} , and n ‘copies’ of x , called *source images* and denoted by x_i . See Figure 2.

The edges of S_x correspond to the shortest path cuts made from each vertex to x . Therefore, the two edges incident to any vertex image v_i are always the same length.

The *ridge tree* (or “cut locus”), T_x , is the closure of the set of all points on the surface of \mathcal{P} that have more than one shortest path to x . It is known that the ridge tree is in fact a tree [12], and that its edges are shortest paths [1] and thus correspond to straight-line segments in S_x . See Figure 2(b). As a corollary to their proof of non-overlap [2, Theorem 10.2], Aronov and O’Rourke proved that the ridge tree is a subset of the Voronoi diagram of the images of x .

Readers interested only in the point case may proceed directly to Section 2. In the remainder of this section we give definitions for the star unfolding from a geodesic curve.

Given a polyhedron \mathcal{P} , a *geodesic curve* λ on \mathcal{P} is a curve on the surface of \mathcal{P} , such that at every interior point p of λ , the surface angle to either side of p is *exactly* π . If the surface angle to either side of p is at most π then the curve is called a *quasigeodesic*. We will only consider [quasi]geodesic curves that are *simple*, meaning there is no point of self-intersection between any two interior (i.e., non-endpoint) points of the curve.

► **Definition 2.** Let \mathcal{P} be a convex polyhedron, and let λ be a simple geodesic curve on the surface of \mathcal{P} . The *geodesic star unfolding* S_λ is a 2-dimensional polygon formed by cutting λ , and cutting a shortest path along the surface of \mathcal{P} from every vertex v_i of \mathcal{P} to λ , and flattening the result into the plane. See Figure 5.

The endpoints of λ are labelled a and b and the two sides of λ are labelled s and t , with the convention that the clockwise order around λ on the outside surface of \mathcal{P} is a, s, b, t . We distinguish shortest paths to λ that arrive at (or “report to”) a or b , and shortest paths that arrive at interior points of λ on side s or t . Any shortest path that arrives at an interior point of λ forms a right angle with λ , as shown by the following lemma, adapted from Ieiri et al. [7, Corollary 1].

► **Lemma 3.** Let λ be a geodesic curve on a convex polyhedron \mathcal{P} , with points x_0 on λ and x not on λ such that xx_0 is a shortest path from x to λ . The angles formed between xx_0 and λ are at least $\frac{\pi}{2}$ to each side.

In the unfolding S_λ , a *source image* is either a copy of an endpoint (a or b), called a *point image*, or a sub-segment of λ corresponding to side s or t , called a *segment image*. (Note that a segment image might include one of the endpoints of λ .) Around a clockwise traversal of the boundary of S_λ we encounter source images in order a, s, b, t —this is because our convention is to draw the *inside* of the polyhedron’s surface facing up. Each consecutive pair of source images is separated by two edges (of equal length) joined at a vertex image, such that the edges correspond to the shortest path cut at the vertex.

The *ridge tree* of a geodesic λ , denoted T_λ , is the closure of the set of all points on the surface of \mathcal{P} that have more than one shortest path to λ . That the ridge tree is an actual tree was established by Demaine and Lubiw [4, Lemma 4]. (In fact, their lemma applies to the outside of any closed convex curve on the surface of \mathcal{P} —in the present situation, the closed convex curve is the set of points at some fixed distance ϵ from λ .) See Figure 5.

A key difference from the point case is that the edges of the geodesic ridge tree are not necessarily straight-line segments. Every edge of the ridge tree is the locus of points that are equidistant from two source images. Thus, when the two sources to either side of an edge are a point image and a segment image respectively, a parabolic ridge tree edge will result. The ridge tree edges between pairs of point images or pairs of segment images will still be straight.

2 New Proof for the Point Star Unfolding

In this section we give a new proof of non-overlap for the point star unfolding that is much simpler than the original proof of Aronov and O’Rourke [2].

The most important idea of our proof is to partition the star unfolding into pairs of congruent triangles. Each vertex of the ridge tree has three or more shortest paths to x on the surface of P . Add all these shortest paths as line segments in the star unfolding. Notice that each edge of the ridge tree will now have a triangle to each side. Because all shortest paths from a ridge tree vertex to the nearest source image have the same length, the two triangles to either side of a ridge tree edge have corresponding sides of equal length, and therefore are congruent. We call each such pair of triangles a *kite*. The kite associated with ridge tree edge e is denoted $kite(e)$. The two images of x to each side of e are called the *apices* of the kite. Observe that the kites form a partition of S_x . See Figure 2(b).

We define the *source angle* of e , $\alpha(e)$, to be the interior angle at either apex of $kite(e)$. See Figure 2(b). We extend this definition to paths: For a path σ of ridge tree edges, where $\sigma = e_0, \dots, e_t$, the *source angle* of σ is $\alpha(\sigma) = \sum_{i=0}^t \alpha(e_i)$. Observe that $\alpha(\sigma) \leq \pi$ because $2\alpha(\sigma)$ measures the total source angle at both apices of every kite on the path, so this is bounded by the total surface angle at the source point x , which is bounded by 2π .

► **Theorem 4** (Theorem 9.1 in [2]). *The star unfolding S_x does not overlap.*

Proof. We will show that no two kites overlap. Consider two kites, and the path in the ridge tree between them. Let e_1, \dots, e_t be the edges of the path in the ridge tree, and let $k_i = kite(e_i)$. We will show that k_1 and k_t have disjoint interiors. We will define a sequence of regions W_1, \dots, W_{t-1} , called *W-wedges*, so that W_i includes k_1, \dots, k_i and excludes k_{i+1} . Then W_{t-1} includes k_1 and excludes k_t , which will complete the proof.

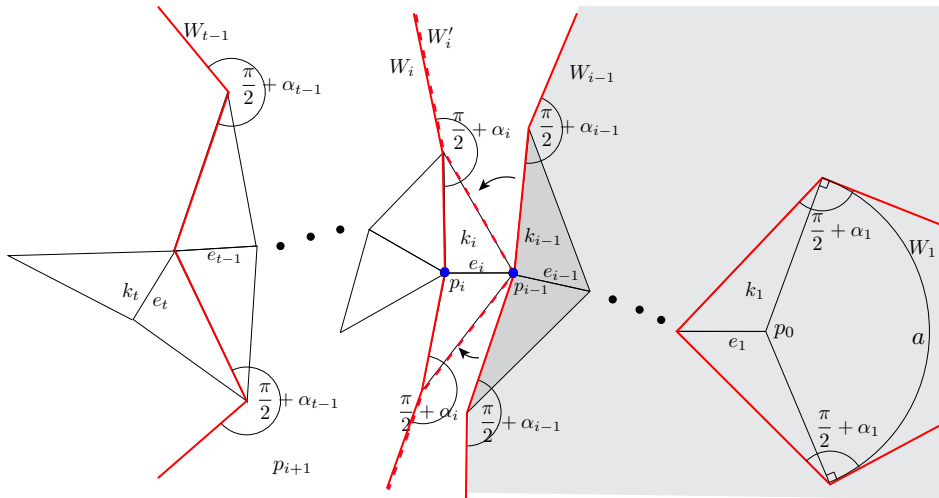
The boundary of the *W-wedge* W_i is shaped like a ‘W’ and defined as follows. The *center point* of W_i is the point p_i , where e_i and e_{i+1} intersect; the *inner legs* are the edges of k_i that are incident to p_i ; and the *outer legs* form an angle with the inner legs (on the side of k_i) of $\frac{\pi}{2} + \alpha_i$, where $\alpha_i = \sum_{j=1}^i \alpha(e_j)$. The outer legs extend either to their point of intersection, or as infinite rays if they do not intersect. This boundary divides the plane into two regions and we define W_i to be the region containing k_i in a neighbourhood of p_i . See Figure 3. Note that α_i is in the range $[0, \pi]$ as observed above.

We will prove by induction that k_i is outside W_{i-1} and that W_i contains $k_i \cup W_{i-1}$. At each step, including the base case, we will need the following:

► **Lemma 5.** *Let p be an endpoint of ridge tree edge e , and let W be a *W-wedge* centered at p such that the inner legs of W are the edges of $kite(e)$, and the outer legs are rotated out (on the side of $kite(e)$) by an angle in the range $[\frac{\pi}{2} + \alpha(e), \frac{3\pi}{2}]$. Then $kite(e) \subseteq W$.*

Proof. Consider the two circular sectors centered at the endpoints p and q of e and bounded by the two incident kite edges as radii (see for example the circular sector marked a centered at $q = p_0$ on kite k_1 in Figure 3). At each apex of $kite(e)$ the two angles between the outer leg of W and the two sides of $kite(e)$ are at least $\frac{\pi}{2}$. Thus the circular sector at q is inside W , and the circular sector at p is outside W . This implies that $kite(e) \subseteq W$. ◀

We are now ready to prove by induction that k_i is outside W_{i-1} and that W_i contains $k_i \cup W_{i-1}$. For the base case $i = 1$, there is no W_0 , and W_1 contains k_1 by the lemma above. Suppose by induction that W_{i-1} contains $k_{i-1} \cup W_{i-2}$. We will show how to transform W_{i-1} into W_i in a way that makes it clear that k_i is outside W_{i-1} and W_i contains $k_i \cup W_{i-1}$.



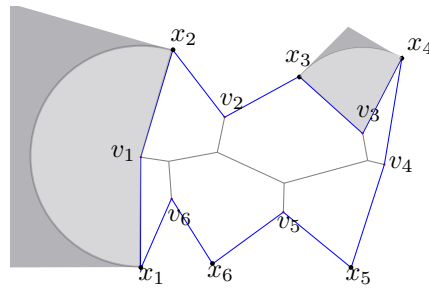
■ **Figure 3** Kite k_{i-1} (shaded) and the corresponding W-wedge W_{i-1} (lightly shaded). W_1 contains kite k_1 because it contains the circular sector a . To prove by induction that W_i contains all previous kites, we expand W_{i-1} , first rotating the legs about p_{i-1} to the W-wedge W'_i (dashed line), and then moving the center point to p_i to obtain W_i . Note that although the figure shows kites k_{i-1} and k_i sharing only a vertex, in non-degenerate situations ridge tree vertices have degree 3, and two consecutive kites will share an edge as well.

Note that the unfolding does not overlap in the neighbourhood of point p_i —this is true whether p_i is a point with 2π surface area, or a vertex, which will be incident to a single cut. Thus the kites k_i and k_{i-1} , which are both incident to p_i , do not overlap. Rotate the two inner legs of W_{i-1} about point p_{i-1} , away from k_{i-1} to the edges of k_i . Keep the angle between inner and outer legs fixed throughout the rotation. Observe that all the kite edges incident to p_{i-1} have the same length, so we really perform a rigid transformation on each half of the W. Call the resulting W-wedge W'_i (shown as a dashed poly-line in Figure 3). Notice that W'_i contains W_{i-1} , because the angle $\alpha_{i-1} + \frac{\pi}{2}$ is in the range $[\frac{\pi}{2}, \frac{3\pi}{2}]$ so the outer legs remain outside the rotation sector of the inner legs. (Here it is crucial that we added the extra $\frac{\pi}{2}$ to the initial angle.) That k_i is outside W'_i follows from applying Lemma 5 to the outside of W'_i , noting that $\alpha_{i-1} + \alpha(e_i) \leq \pi$ so the angle $\alpha_{i-1} + \frac{\pi}{2}$ is actually in the range $[\frac{\pi}{2}, \frac{3\pi}{2} - \alpha(e_i)]$ and therefore the complementary angle is in the range $[\alpha(e_i) + \frac{\pi}{2}, \frac{3\pi}{2}]$.

The second step of the transformation is to move the center point of W'_i from p_{i-1} to p_i , while keeping the outer legs fixed. The W-wedge increases until it contains k_i . The angle between inner and outer legs increases by $\alpha(e_i)$, to α_i . Thus the result is precisely W_i , and therefore W_i contains $k_i \cup W_{i-1}$. ◀

Our proof, like Aronov and O’Rourke’s, shows a stronger result that certain regions outside the star unfolding are empty. Aronov and O’Rourke [2] showed that at any vertex v_i adjacent to source images x_i and x_{i+1} in the unfolding, no part of the unfolding enters the circular sector centered at v_i exterior to the unfolding near v_i and bounded by the radii $v_i x_i$ and $v_i x_{i+1}$. See Figure 4. Our proof shows that larger regions are empty:

▶ **Lemma 6.** *Let v_i be a vertex adjacent to source images x_i and x_{i+1} in a star unfolding S_x , and let W be the region bounded by $v_i x_i$, $v_i x_{i+1}$, and the rays extending from x_i and x_{i+1} at right angles from these segments on the exterior side of v_i . Then no part of the unfolding intersects the interior of W .*



■ **Figure 4** Old sectors of emptiness (light grey) established by Aronov and O'Rourke and new empty regions (light grey+dark grey) established by Lemma 6.

Proof outline. In non-degenerate situations, v_i is a leaf of the ridge tree, incident to ridge tree edge e , say. We apply the argument in the proof above to the ridge tree path from e to any other edge. The initial W-wedge always contains W , so no other kite enters W . More generally, v_i may be an internal vertex of the ridge tree and we repeat the argument for every path in the ridge tree starting at v_i , taking the intersection of the initial wedges in the path argument to show that W is empty. ◀

3 Geodesic Star Unfolding

In this section we will consider the star unfolding from a geodesic curve λ . By generalizing the proof for the point case, we will establish some situations in which the geodesic star unfolding does not overlap. See Figure 5.

As for the point-source case, we will partition the unfolding by adding, for every ridge tree vertex p , the line segments that correspond to shortest paths from p to the curve. We include as a vertex of the ridge tree any point of the ridge tree that has a shortest path to an endpoint of λ appearing in a segment image. The added line segments partition the unfolding into regions called *wings*. See Figure 5(b). The two wings on either side of a ridge tree edge form a *wing-pair*. A wing with an endpoint source image is a *point wing* and a wing with a segment source image is a *segment wing*. A wing-pair may involve two point wings (these are the kites of the previous section), or two segments wings, or one of each, in which case we call it a *hybrid wing-pair*. The ridge tree edge of a hybrid wing-pair is a parabolic segment; all others are straight-line segments.

The *source angle* of a point wing is the angle at its source image point; the *source angle* of a segment wing is 0. The *source angle* of a wing-pair is the sum of the source angles of the two wings. If e is an edge of the ridge tree, and A designates one side of this edge, then $\alpha^A(e)$ denotes the source angle of the wing on that side. If σ is a path in the ridge tree, with its two sides (arbitrarily) labelled A and B , then the source angle of σ on side A is $\alpha^A(\sigma) = \sum_{i=0}^t \alpha^A(e_i)$ (and similarly for B). The path σ is *balanced* if $\alpha^A(\sigma) \leq \pi$ and $\alpha^B(\sigma) \leq \pi$. We say that the ridge tree [or the geodesic curve] is *balanced* if every path in the ridge tree is balanced. There are examples where the ridge tree is not balanced, and in fact it is possible to have all 2π of source angle to one side of a ridge tree path, see [10].

Our main result in this section is that wing-pairs along a balanced path do not overlap.

► **Lemma 7.** *Let \mathcal{P} be a convex polyhedron with a geodesic curve λ on its surface. Suppose that u and v are distinct edges of the ridge tree such that the path in the ridge tree from u to v is balanced. Then the wing-pairs of u and v do not overlap.*

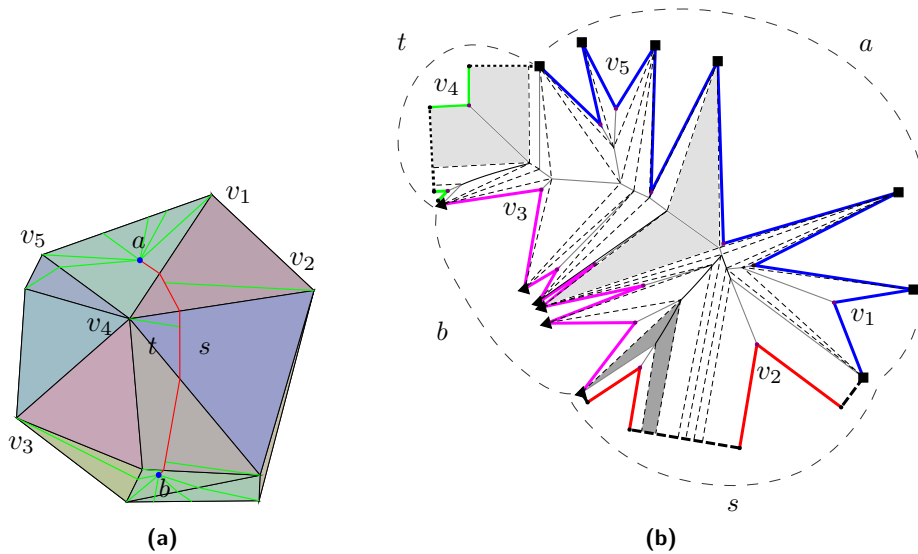


Figure 5 The star unfolding from a geodesic curve. **(a)** The polyhedron, the geodesic curve with endpoints a and b and sides s and t , and shortest paths from vertices to the curve. **(b)** The corresponding star unfolding with the source images, a, b, s or t , indicated. Images of a [b] are drawn as squares [triangles]; segment images of s [t] are drawn as heavy dashed [dotted] lines. Shortest path cuts are coloured according to their destination type. Wings are indicated by dashed lines. Three wing-pairs are shaded; the darkly shaded one is a hybrid wing-pair with a parabolic ridge tree edge. Note that the unfolding shows the inside surface of the polyhedron.

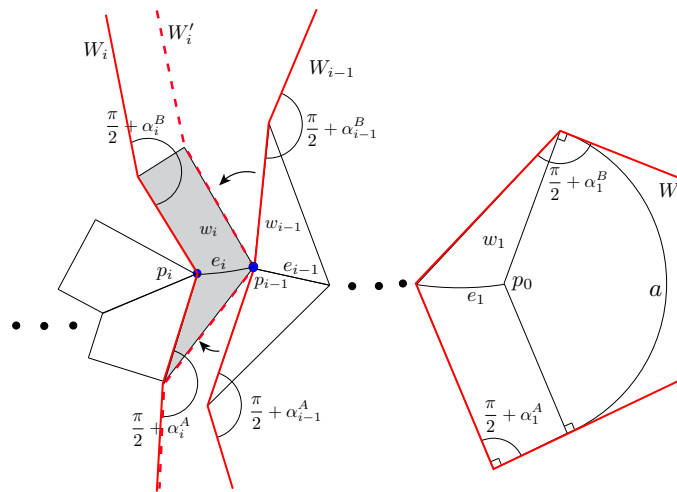
Before proceeding with the proof, we note the consequence that the star unfolding from a balanced geodesic curve does not overlap:

► **Corollary 8.** *If \mathcal{P} is a convex polyhedron with a balanced geodesic curve λ then the geodesic star unfolding from λ does not overlap.*

Proof of Lemma. We follow the same plan as in the proof of Theorem 4, that is, we will prove that no two wing-pairs in the unfolding overlap, by examining W -wedges along the ridge tree path between the two wing-pairs. A quick summary is that there are only two differences in the argument: (1) as we move from W -wedge W_{i-1} to W_i we may increase the angle between inner and outer legs differently on its two sides; (2) when the W -wedge moves past a segment wing, the angle between inner and outer legs does not change, and the inner+outer pair translates. See Figure 6. We now give the details.

Let σ be the path from u to v in the ridge tree, with edges $u = e_1, \dots, e_t = v$. Let w_i be the wing-pair of edge e_i . Label the two sides of σ by A and B . Let $\alpha_i^A = \sum_{j=1}^i \alpha^A(e_j)$ and let $\alpha_i^B = \sum_{j=1}^i \alpha^B(e_j)$. Note that α_i^A and α_i^B are in the range $[0, \pi]$ by assumption.

We will show that w_1 and w_t have disjoint interiors by defining W -wedges W_i so that W_i includes w_1, \dots, w_i and excludes w_{i+1} . Then W_{t-1} includes w_1 and excludes w_t , which will complete the proof. Define W_i , for $i = 1, \dots, t$, to be the W -wedge with center point at p_i where e_i and e_{i+1} intersect, with inner legs along the two incident edges of w_i , and outer legs rotated out (on the side of w_i) by $\alpha_i^A + \frac{\pi}{2}$ on side A , and by $\alpha_i^B + \frac{\pi}{2}$ on side B . The outer legs extend either to their point of intersection, or as infinite rays if they do not intersect. This boundary divides the plane into two regions and we define W_i to be the region containing w_i . See Figure 6.



■ **Figure 6** Wing-pair w_i (shaded) is a hybrid wing-pair. W_1 contains w_1 because it contains the circular sector a . To prove by induction that W_i contains all previous wing-pairs, we expand W_{i-1} to W_i , first rotating the legs about p_{i-1} to the W-wedge W'_i (dashed line), and then expanding to include w_i . Note that in this example $\alpha_{i-1}^B = \alpha_i^B$ since the wing to that side is a segment-wing.

We will prove by induction that w_i is outside W_{i-1} and that W_i contains $w_i \cup W_{i-1}$. At each step, including the base case, we will need the following:

► **Lemma 9.** *Let p be an endpoint of ridge tree edge e , and let W be a W-wedge centered at p such that the inner legs of W are the edges of the incident wings of e , and the outer leg on the A side is rotated out by an angle in the range $[\frac{\pi}{2} + \alpha^A(e), \frac{3\pi}{2}]$ and the outer leg on the B side is rotated out by an angle in the range $[\frac{\pi}{2} + \alpha^B(e), \frac{3\pi}{2}]$. Then the wing-pair of e is contained in W .*

Proof. Consider the two circular sectors centered at the endpoints p and q of e and bounded by the two incident wing edges as radii (see for example the circular sector marked a centered at $q = p_0$ on wing-pair w_1 in Figure 6). On both the A side and the B side, the angles between the outer leg of W and the sides of the wing of e are at least $\frac{\pi}{2}$. Thus the circular sector at q is inside W , and the circular sector at p is outside W . This implies that the wing-pair of e is contained in W . ◀

We are now ready to prove by induction that w_i is outside W_{i-1} and that W_i contains $w_i \cup W_{i-1}$. For the base case $i = 1$, there is no W_0 , and W_1 contains w_1 by the lemma above. Suppose by induction that W_{i-1} contains $w_{i-1} \cup W_{i-2}$. We will show how to transform W_{i-1} into W_i in a way that makes it clear that w_i is outside W_{i-1} and W_i contains $w_i \cup W_{i-1}$.

Since there is at most 2π surface angle at any point on the surface, the unfolding does not overlap in the neighbourhood of any point. Thus w_{i-1} and w_i are disjoint. Rotate the two inner legs of W_{i-1} about point p_{i-1} , away from w_{i-1} to the edges of w_i . Keep the angle between inner and outer legs fixed throughout the rotation. Observe that all the wing edges incident to p_{i-1} have the same length, so we really perform a rigid transformation on each half of the W . Call the resulting W-wedge W'_i (shown as a dashed poly-line in Figure 6). Notice that W'_i contains W_{i-1} , because the angles $\alpha_{i-1}^A + \frac{\pi}{2}$, and $\alpha_{i-1}^B + \frac{\pi}{2}$ are in the range $[\frac{\pi}{2}, \frac{3\pi}{2}]$ so the outer legs remain outside the rotation sector of the inner legs. We prove that w_i is outside W'_i by applying Lemma 9 to the outside of W'_i . To show that the angle on

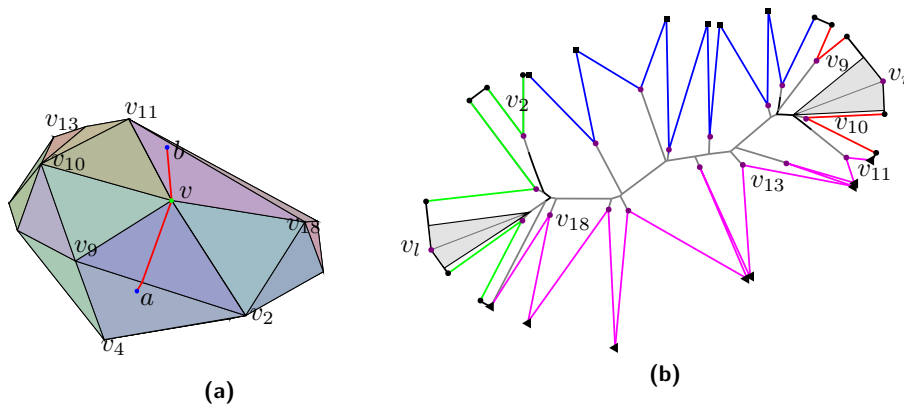


Figure 7 The star unfolding from a quasi-geodesic curve. **(a)** The polyhedron and the quasi-geodesic curve passing through vertex v . **(b)** The corresponding star unfolding with the two images of v and their wing-pairs indicated.

the A side is in the required range, observe that $\alpha_{i-1}^A + \alpha^A(e_i) \leq \pi$ so the angle $\alpha_{i-1}^A + \frac{\pi}{2}$ is actually in the range $[\frac{\pi}{2}, \frac{3\pi}{2} - \alpha^A(e_i)]$ and therefore the complementary angle is in the range $[\alpha^A(e_i) + \frac{\pi}{2}, \frac{3\pi}{2}]$. A similar argument applies on the B side.

The second step of the transformation is to move W'_i past w_i to W_i . We do this separately on the A side and the B side. To go past a point wing, we move the inner leg from p_{i-1} to p_i , while keeping the outer leg fixed. To go past a segment wing, we translate the inner+outer legs so that their common point moves along the segment image; since the segment image is perpendicular to the inner leg, each leg is parallel to its initial version. The resulting W -wedge contains w_i . On the A side, the angle between inner and outer legs increases by $\alpha^A(e_i)$, to α_i^A , and similarly on the B side. Thus the result is precisely W_i , and therefore W_i contains $w_i \cup W_{i-1}$. ◀

3.1 Extension to Quasigeodesic Curves

We now show that our geodesic star unfolding result (Lemma 7) carries over to quasigeodesic curves. Recall that a quasigeodesic curve on the surface of a polyhedron \mathcal{P} is a curve such that at each interior point along the curve the surface angle to each side is $\leq \pi$. (The angle can only be $< \pi$ at a vertex of \mathcal{P} .)

Consider a quasigeodesic curve λ on the surface of \mathcal{P} . We define the *quasigeodesic star unfolding* in the same way as the geodesic star unfolding, specifically, we cut the curve and a shortest path from every vertex to the curve. See Figure 7.

To argue about the ridge tree of λ , we consider the closed curve consisting of the points at some small fixed distance ϵ from λ . This curve is composed of circular arcs and geodesic segments joined at angles $\leq \pi$ (on the side opposite λ). Thus by [4, Lemma 4], its ridge tree is a tree. The ridge tree of λ itself (when $\epsilon = 0$) has the peculiarity that it has a cycle if λ has an interior point with surface angle $< \pi$ on both sides of the curve (e.g., vertex v in Figure 7). However, we split any such vertex into two copies, corresponding to the s and t sides of the curve, which breaks the cycle in the unfolded ridge tree.

Suppose p is an interior point of the quasigeodesic curve λ where the surface angle to one side, say the s side, is β , where $\beta < \pi$. Necessarily, p is a vertex of the polyhedron, otherwise the surface angle on the other side of the curve would be greater than π . We do not introduce a cut for this vertex in the unfolding, since it already lies on λ .

Using Lemma 3, we claim that no shortest path cut from any vertex v will report to point p on side s , since one of the two surface angles formed would be $< \frac{\pi}{2}$. Thus the quasigeodesic star unfolding from λ will have a vertex image with an angle β corresponding to the s side of p . Let p_s denote this vertex image in the unfolding.

Observe that p_s is a leaf of the unfolded ridge tree and that the incident ridge tree edge e is a straight segment forming angles $\frac{\beta}{2}$ with the segment images to either side of p_s . We can consider e to have two segment wings, albeit degenerate ones, with one side (between p_s and λ) of length 0. We call this a *quasi-wing-pair* (see the shaded examples in Figure 7).

With these observations in hand, we can extend the result of the previous section to quasigeodesics. The main idea is to show that quasi-wing pairs can only occur as the first or last wing-pair along a path of the unfolded ridge tree, and thus that the induction proof of the previous section carries over.

► **Lemma 10.** *Let \mathcal{P} be a convex polyhedron with a quasigeodesic curve λ on its surface. Suppose that u and v are distinct edges of the ridge tree such that the path in the unfolded ridge tree from u to v is balanced. Then the (quasi-)wing-pairs of u and v do not overlap.*

3.2 Quasigeodesic Loops

In this section, we prove non-overlap of the star unfolding from geodesic (and quasigeodesic) loops. When the two endpoints a and b of a geodesic or quasigeodesic curve λ coincide at point o , we call this a *(quasi)geodesic loop* with *loop point* o . A (quasi)geodesic loop cuts the surface of the polyhedron into two pieces. One piece must have a surface angle at o that is $\leq \pi$, and we call this the *inside* of the loop and identify it with the s side of the curve. The other piece is called the *outside* and will be identified with the t side of the curve.

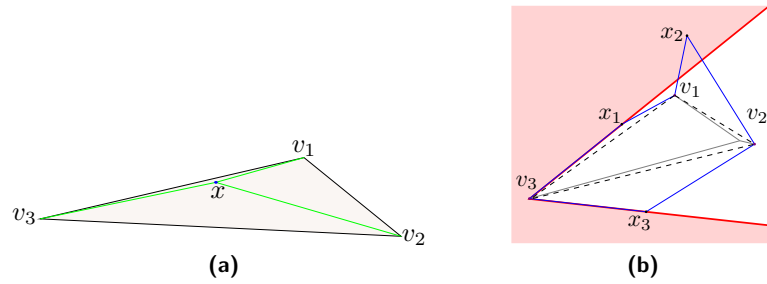
Itoh, O'Rourke, and Vilcu [9] proved that for any quasigeodesic loop: (1) the inside unfolds without overlap; (2) the outside unfolds without overlap; and (3) the two unfolded pieces can be reattached (without overlap) along a common segment of the cut curve. Their proof of (2) relies on a Lemma [9, Lemma 7] about the star unfolding from a point, which they say will be proved in a companion paper, but unfortunately, they discovered¹ that the Lemma is false. The Lemma claims that for any star unfolding from a point x and for any polyhedron vertex v , the exterior angle at v in the unfolding yields an empty wedge. More precisely, if the exterior angle at v is (counterclockwise) x_i, v, x_{i+1} then the claim is that the counterclockwise wedge formed by the rays vx_i and vx_{i+1} does not contain any part of the unfolding. An example where this fails is shown in Figure 8.

In this section we examine the star unfolding from a quasigeodesic curve where the two endpoints, a and b , of the curve are arbitrarily close together. In the limit when $a = b$ the unfolding consists of two pieces joined at the point $a = b$ with the angular bisectors at the point $a = b$ aligned in the unfolding. We call this the *conjoined star unfolding* from a quasigeodesic loop. See Figure 9. Our main result of this section is that the conjoined star unfolding from a quasigeodesic loop does not overlap. This implies that the outside piece unfolds without overlap, which repairs the missing step of Itoh, O'Rourke and Vilcu's result.

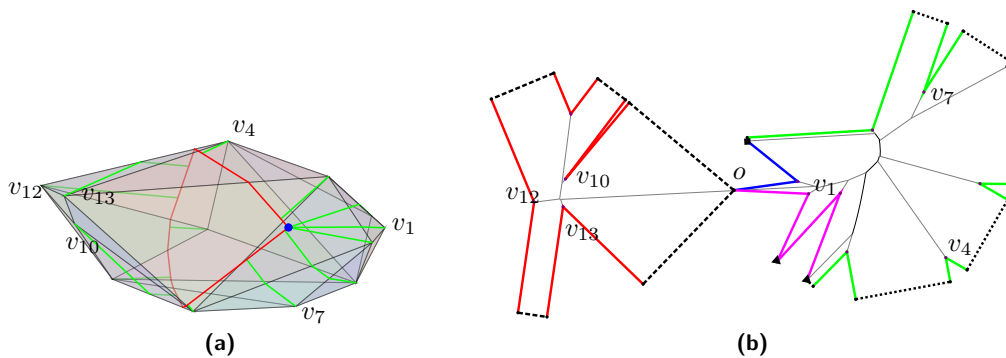
► **Theorem 11.** *The conjoined star unfolding from a quasigeodesic loop does not overlap.*

We prove Theorem 11 by showing that every path through the ridge tree for a geodesic loop is balanced. Then by Lemma 7, the unfolding does not overlap. Furthermore, by Lemma 10, the result extends to quasigeodesics.

¹ Private communication from J. O'Rourke



■ **Figure 8** A counterexample to [9, Lemma 7]. **(a)** A doubly covered triangle with source point x . **(b)** The star unfolding from x showing a wedge formed by the exterior angle x_3, v_3, x_1 that intersects the unfolding. Dashed edges show the back face; grey edges show the ridge tree.



■ **Figure 9** Conjoined star unfolding from a geodesic loop. **(a)** Geodesic loop on the surface of a polyhedron (partially transparent to view the full loop). **(b)** Unfolding.

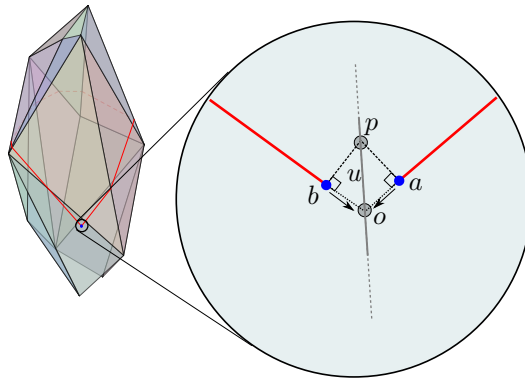
► **Lemma 12.** *Every path through the unfolded ridge tree of the conjoined star unfolding from a quasigeodesic loop is balanced.*

Proof. Recall our assumption that the s -side of the curve is inside the loop. By Lemma 3, no vertex inside the loop can report to the loop point o , since the surface angle to the interior of the loop is $< \pi$ (or if it is equal to π , we can assume the vertex reports to b instead).

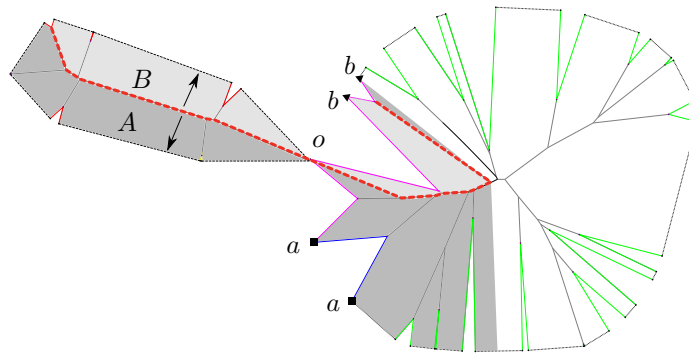
Consider the segment of the ridge tree (call it u) that lies between a and b on the inside of the loop and touches loop point o . Observe what happens in the limit as a and b approach point o . Consider the kite-shaped region of the surface delimited by a, o, b , and p , where p is the intersection between the rays perpendicular to the geodesic at a and b respectively (see Figure 10). Call this region R . When a and b are close enough to o there are no vertices or other ridge tree edges inside the region R , and therefore some sub-segment of u will extend from p to o (i.e., perpendicularly bisect \overline{ab}). Therefore the source angle of each wing of u is at least $\frac{\pi}{2}$, and the source angle of u 's wing-pair is at least π . Furthermore, this is the only edge of the ridge tree on the inside of the loop that has point-wings reporting to a or b .

Consider any path σ through the unfolded ridge tree. Let A and B be the sides of σ . We must show that $\alpha^A(\sigma) \leq \pi$ and $\alpha^B(\sigma) \leq \pi$. There are three possible cases for the path:

1. The path σ does not include u , and remains entirely on the inside of the loop. Because u is the only ridge tree edge inside the loop whose wings have non-zero source angle, therefore $\alpha^A(\sigma) = \alpha^B(\sigma) = 0$.



■ **Figure 10** Zoomed-in view of the surface as a and b approach o . Assuming no vertices are inside the region $aobp$, there is at least $\frac{\pi}{2}$ source angle to either side of ridge tree edge u .



■ **Figure 11** An illustration of the third case in the proof of Lemma 12, where the A side of the path σ (in heavy dashed red) has point wings reporting to a and to b .

2. The path σ does not include u , and remains entirely on the outside of the loop. As noted above, the source angle of u 's wing-pair is at least π . Therefore, the remaining source angle of all wings along every other possible path (i.e., not including u) must be $\leq \pi$. Thus $\alpha^A(\sigma) \leq \pi$ and $\alpha^B(\sigma) \leq \pi$.
3. The path σ includes u . We must show that $\alpha^A(\sigma) \leq \pi$ and $\alpha^B(\sigma) \leq \pi$. Any edge of σ (apart from u) that lies inside the geodesic loop only has segment wings to either side and these contribute 0 to the source angle of the path. Thus it suffices to look at the portion of σ starting with u and then following ridge tree edges that lie outside the geodesic loop. Call this subpath σ' . Ridge tree edge u has a point wing to either side, one reporting to a and one reporting to b . Suppose that the A side of the path has the point wing reporting to a . Suppose that $\alpha^A(\sigma') \geq \alpha^B(\sigma')$. If side A only has point wings that report to a , then its source angle is at most π . So suppose that side A has a point wing that reports to b . See Figure 11. As we walk around the path σ' , in counterclockwise order starting with u on the A side, the wings report in order to a , then t , then b . Thus all the wings on the B side must be point wings that report to b , which implies that the sum of their source angles is at most π , i.e., $\alpha^B(\sigma') \leq \pi$. Since every wing on the B side is a point-wing reporting to b , every point wing on the A side must be paired with a point wing on the B side, and each such pair has equal source angles. Therefore $\alpha^A(\sigma') \leq \alpha^B(\sigma')$, and we just showed this is $\leq \pi$. ◀

3.3 The Geodesic Star Unfolding as Two Non-overlapping Pieces

Although we have not proved that the geodesic star unfolding never overlaps, in this section we show that it can always be cut into two pieces each of which is non-overlapping. The extra cuts consist of two shortest paths from a point on the ridge tree to the geodesic curve.

► **Lemma 13.** *Let \mathcal{P} be a convex polyhedron and λ be a geodesic curve on \mathcal{P} . Then there is a point p of the ridge tree of λ such that cutting two shortest paths on \mathcal{P} from p to the geodesic curve λ separates the geodesic star unfolding S_λ into two pieces each of which is non-overlapping.*

Proof. We will split the ridge tree into two subtrees, either at an internal point of an edge or at a ridge vertex v , in which case we split the ridge tree edges incident to v into two subsets, each consecutive in the cyclic order of edges around v . Call such subtrees *proper*. The geodesic star unfolding S_λ can then be cut into two pieces as follows: if the two subtrees are joined at an internal point p of a ridge tree edge, then we cut the two shortest paths from p to the geodesic curve λ ; and if the two subtrees are joined at vertex v then we cut the two shortest paths from v to λ that separate the incident ridge tree edges as specified. This ensures that if two wing-pairs are in the same piece of the unfolding, then the ridge tree path between them lies in the same subtree. So long as each subtree is balanced, Lemma 7 ensures that no two wing-pairs from the same piece overlap, i.e., that each piece is non-overlapping.

In the remainder of the proof we show how to partition the ridge tree into two proper balanced subtrees. Each edge e of the ridge tree has an associated source angle of its wing pair, $\sigma(e)$, and the sum of these weights over the whole ridge tree is 2π . We remark that a weaker form of the lemma with three non-overlapping pieces can be obtained from the result that any edge-weighted tree can be separated at a single vertex into three subtrees of weight at most one half the total weight. To prove the lemma we will argue about the source angles on each side of each ridge tree edge.

Among all proper subtrees of the ridge tree, let T be a maximal subtree that is balanced. Let \bar{T} be the complement. We claim that \bar{T} is balanced.

If the source angle of T is at least π , then \bar{T} has source angle $\leq \pi$ so it must be balanced as well. Otherwise the source angle of T is $< \pi$. Note that T cannot be rooted at an interior point of an edge otherwise we could move the point further along the edge to increase the source angle of T a small amount without exceeding π . Therefore T must be rooted at a vertex v of the ridge tree. Among the edges incident to v in clockwise order, let e and f be the first and last edges outside T . Note that $e \neq f$ (i.e., there is more than one edge incident to v in \bar{T}) otherwise we could move the root of T along e to increase the source angle by a small amount. Adding e and its subtree to T gives an unbalanced subtree, so there must be an unbalanced path μ_e that contains e and a subpath in T . Similarly, there must be an unbalanced path μ_f that contains f and a subpath in T .

Note that any two unbalanced path-sides must have a wing in common, otherwise we would have two disjoint sets of wings each with source angle greater than π . Thus the unbalanced sides of μ_e and μ_f must both lie on the clockwise side of e and f or both on the counterclockwise side of e and f (relative to the cyclic ordering of edges at v). Suppose the former, without loss of generality.

Suppose that \bar{T} has an unbalanced path μ . The unbalanced side must share wings with the unbalanced side of μ_e and with the unbalanced side of μ_f , and therefore must include the clockwise sides of both e and of f , which is impossible. Therefore \bar{T} is balanced, and we can separate the geodesic star unfolding S_λ into two pieces each of which is non-overlapping. ◀

4 Conclusions

We have given a simple proof that the star unfolding from a point does not overlap, and extended it to some cases of the star unfolding from a geodesic curve. We leave open the main conjecture that the geodesic star unfolding does not overlap. All we can say about the general case is that the unfolding can be partitioned into two non-overlapping pieces.

The first author’s thesis [10] contains further results on geodesics that have been “fully extended” until the endpoints a and b intersect the curve. When the endpoints reach opposite sides of the curve (“S-shaped”) the unfolding does not overlap because it is balanced. When the endpoints reach the same side of the curve (“C-shaped”) the unfolding need not be balanced, though some special cases can still be proved to avoid overlap.

The figures in this paper were generated with a custom program written using CGAL, OpenGL, and Cairo. For more information, see the first author’s thesis [10].

Acknowledgements. We thank Timothy Chan for suggesting Lemma 13. We thank Joseph O’Rourke, Costin Vilcu, and anonymous referees for helpful comments.

References

- 1 Pankaj K. Agarwal, Boris Aronov, and Catherine A. Schevon. Star unfolding of a polytope with applications. *SIAM Journal on Computing*, 26:1689–1713, 1997.
- 2 Boris Aronov and Joseph O’Rourke. Nonoverlap of the star unfolding. *Discrete & Computational Geometry*, 8(3):219–250, 1992.
- 3 Erik D. Demaine, Martin L. Demaine, Anna Lubiw, Arlo Shallit, and Jonah Shallit. Zipper unfoldings of polyhedral complexes. In *Proceedings of the 22nd Annual Canadian Conference on Computational Geometry (CCCG)*, pages 219–222, August 2010.
- 4 Erik D. Demaine and Anna Lubiw. A generalization of the source unfolding of convex polyhedra. In *Revised Papers from the 14th Spanish Meeting on Computational Geometry (EGC 2011)*, volume 7579 of *Lecture Notes in Computer Science*, pages 185–199, Alcalá de Henares, Spain, June 27–30 2012.
- 5 Erik D. Demaine and Joseph O’Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, New York, NY, USA, 2007.
- 6 Albrecht Dürer. *The Painter’s Manual: A Manual of Measurement of Lines, Areas, and Solids by Means of Compass and Ruler Assembled by Albrecht Dürer for the Use of All Lovers of Art with Appropriate Illustrations Arranged to be Printed in the Year MDXXV*. The literary remains of Albrecht Dürer. Abaris Books, 1977.
- 7 Kouki Ieiri, Jin-ichi Itoh, and Costin Vilcu. Quasigeodesics and farthest points on convex surfaces. *Advances in Geometry*, 11(4):571–584, 2011.
- 8 Jin-Ichi Itoh, Joseph O’Rourke, and Costin Vilcu. Source unfoldings of convex polyhedra with respect to certain closed polygonal curves. In *Proceedings of the 25th European Workshop on Computational Geometry (EuroCG)*, pages 61–64, 2009.
- 9 Jin-ichi Itoh, Joseph O’Rourke, and Costin Vilcu. Star unfolding convex polyhedra via quasigeodesic loops. *Discrete & Computational Geometry*, 44(1):35–54, 2010.
- 10 Stephen Kiazzyk. The star unfolding from a geodesic curve. Master’s thesis, Cheriton School of Computer Science, University of Waterloo, 2014.
- 11 Joseph O’Rourke and Costin Vilcu. Development of curves on polyhedra via conical existence. *Computational Geometry*, 47(2, Part A):149–163, 2014.
- 12 Micha Sharir and Amir Schorr. On shortest paths in polyhedral spaces. *SIAM Journal on Computing*, 15(1):193–215, 1986.

The Dirac-Motzkin Problem on Ordinary Lines and the Orchard Problem*

Ben J. Green

Mathematical Institute, University of Oxford
Oxford, UK
ben.green@maths.ox.ac.uk

Abstract

Suppose you have n points in the plane, not all on a line. A famous theorem of Sylvester-Gallai asserts that there is at least one *ordinary line*, that is to say a line passing through precisely two of the n points. But how many ordinary lines must there be? It turns out that the answer is at least $n/2$ (if n is even) and roughly $3n/4$ (if n is odd), provided that n is sufficiently large. This resolves a conjecture of Dirac and Motzkin from the 1950s. We will also discuss the classical orchard problem, which asks how to arrange n trees so that there are as many triples of colinear trees as possible, but no four in a line. This is joint work with Terence Tao and reports on the results of [1].

1998 ACM Subject Classification G.2 Discrete Mathematics

Keywords and phrases combinatorial geometry, incidences

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.405

Category Invited Talk

References

- 1 B. J. Green and T. C. Tao, *On sets with few ordinary lines*, Discrete and Computational Geometry **50** (2013), no. 2, 409–468.

* This work was partially supported by ERC Starting Grant number 279438, *Approximate algebraic structure and applications*.



On the Beer Index of Convexity and Its Variants*

Martin Balko¹, Vít Jelínek², Pavel Valtr¹, and Bartosz Walczak^{3,4}

1 Department of Applied Mathematics, Faculty of Mathematics and Physics,
Charles University, Prague, Czech Republic

balko@kam.mff.cuni.cz, valtr@kam.mff.cuni.cz

2 Institute for Theoretical Computer Science, Faculty of Mathematics and
Physics, Charles University, Prague, Czech Republic

jelinek@iuuk.mff.cuni.cz

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

walczak@tcs.uj.edu.pl

4 School of Mathematics, Georgia Institute of Technology, Atlanta, GA, USA

Abstract

Let S be a subset of \mathbb{R}^d with finite positive Lebesgue measure. The *Beer index of convexity* $b(S)$ of S is the probability that two points of S chosen uniformly independently at random see each other in S . The *convexity ratio* $c(S)$ of S is the Lebesgue measure of the largest convex subset of S divided by the Lebesgue measure of S . We investigate the relationship between these two natural measures of convexity of S .

We show that every set $S \subseteq \mathbb{R}^2$ with simply connected components satisfies $b(S) \leq \alpha c(S)$ for an absolute constant α , provided $b(S)$ is defined. This implies an affirmative answer to the conjecture of Cabello et al. asserting that this estimate holds for simple polygons.

We also consider higher-order generalizations of $b(S)$. For $1 \leq k \leq d$, the *k-index of convexity* $b_k(S)$ of $S \subseteq \mathbb{R}^d$ is the probability that the convex hull of a $(k+1)$ -tuple of points chosen uniformly independently at random from S is contained in S . We show that for every $d \geq 2$ there is a constant $\beta(d) > 0$ such that every set $S \subseteq \mathbb{R}^d$ satisfies $b_d(S) \leq \beta c(S)$, provided $b_d(S)$ exists. We provide an almost matching lower bound by showing that there is a constant $\gamma(d) > 0$ such that for every $\varepsilon \in (0, 1]$ there is a set $S \subseteq \mathbb{R}^d$ of Lebesgue measure one satisfying $c(S) \leq \varepsilon$ and $b_d(S) \geq \gamma \frac{\varepsilon}{\log_2 1/\varepsilon} \geq \gamma \frac{c(S)}{\log_2 1/c(S)}$.

1998 ACM Subject Classification F.2.2 Geometrical problems and computations

Keywords and phrases Beer index of convexity, convexity ratio, convexity measure, visibility

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.406

1 Introduction

For positive integers k and d and a Lebesgue measurable set $S \subseteq \mathbb{R}^d$, we use $\lambda_k(S)$ to denote the k -dimensional Lebesgue measure of S . We omit the subscript k when it is clear from the context. We also write ‘measure’ instead of ‘Lebesgue measure’, as we do not use any other measure in the paper.

For a set $S \subseteq \mathbb{R}^d$, let $\text{smc}(S)$ denote the supremum of the measures of convex subsets of S . Since all convex subsets of \mathbb{R}^d are measurable [12], the value of $\text{smc}(S)$ is well defined.

* The first three authors were supported by the grant GAČR 14-14179S. The first author acknowledges the support of the Grant Agency of the Charles University, GAUK 690214 and the project SVV-2014-260103 (Discrete Models and Algorithms). The last author was supported by the Ministry of Science and Higher Education of Poland *Mobility Plus* grant 911/MOB/2012/0.



Moreover, Goodman's result [9] implies that the supremum is achieved on compact sets S , hence it can be replaced by maximum in this case. When S has finite positive measure, let $c(S)$ be defined as $\text{smc}(S)/\lambda_d(S)$. We call the parameter $c(S)$ the *convexity ratio* of S .

For two points $A, B \in \mathbb{R}^d$, let \overline{AB} denote the closed line segment with endpoints A and B . Let S be a subset of \mathbb{R}^d . We say that points $A, B \in S$ are *visible* one from the other or *see* each other in S if the line segment \overline{AB} is contained in S . For a point $A \in S$, we use $\text{Vis}(A, S)$ to denote the set of points that are visible from A in S . More generally, for a subset T of S , we use $\text{Vis}(T, S)$ to denote the set of points that are visible in S from T . That is, $\text{Vis}(T, S)$ is the set of points $A \in S$ for which there is a point $B \in T$ such that $\overline{AB} \subseteq S$.

Let $\text{Seg}(S)$ denote the set $\{(A, B) \in S \times S : \overline{AB} \subseteq S\} \subseteq (\mathbb{R}^d)^2$, which we call the *segment set* of S . For a set $S \subseteq \mathbb{R}^d$ with finite positive measure and with measurable $\text{Seg}(S)$, we define the parameter $b(S) \in [0, 1]$ by

$$b(S) := \frac{\lambda_{2d}(\text{Seg}(S))}{\lambda_d(S)^2}.$$

If S is not measurable, or if its measure is not positive and finite, or if $\text{Seg}(S)$ is not measurable, we leave $b(S)$ undefined. Note that if $b(S)$ is defined for a set S , then $c(S)$ is defined as well.

We call $b(S)$ the *Beer index of convexity* (or just *Beer index*) of S . It can be interpreted as the probability that two points A and B of S chosen uniformly independently at random see each other in S .

1.1 Previous results

The Beer index was introduced in the 1970s by Beer [2, 3, 4], who called it ‘the index of convexity’. Beer was motivated by studying the continuity properties of $\lambda(\text{Vis}(A, S))$ as a function of A . For polygonal regions, an equivalent parameter was later independently defined by Stern [19], who called it ‘the degree of convexity’. Stern was motivated by the problem of finding a computationally tractable way to quantify how close a given set is to being convex. He showed that the Beer index of a polygon P can be approximated by a Monte Carlo estimation. Later, Rote [17] showed that for a polygonal region P with n edges the Beer index can be evaluated in polynomial time as a sum of $O(n^9)$ closed-form expressions.

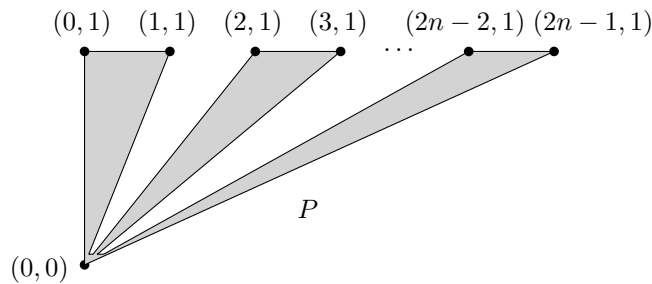
Cabello et al. [7] have studied the relationship between the Beer index and the convexity ratio, and applied their results in the analysis of their near-linear-time approximation algorithm for finding the largest convex subset of a polygon. We describe some of their results in more detail in Subsection 1.3.

1.2 Terminology and notation

We assume familiarity with basic topological notions such as path-connectedness, simple connectedness, Jordan curve, etc. The reader can find these definitions, for example, in Prasolov's book [16].

Let ∂S , S° , and \overline{S} denote the boundary, the interior, and the closure of a set S , respectively. For a point $A \in \mathbb{R}^2$ and $\varepsilon > 0$, let $\mathcal{N}_\varepsilon(A)$ denote the open disc centered at A with radius ε . For a set $X \subseteq \mathbb{R}^2$ and $\varepsilon > 0$, let $\mathcal{N}_\varepsilon(X) = \bigcup_{A \in X} \mathcal{N}_\varepsilon(A)$. A *neighborhood* of a point $A \in \mathbb{R}^2$ or a set $X \subseteq \mathbb{R}^2$ is a set of the form $\mathcal{N}_\varepsilon(A)$ or $\mathcal{N}_\varepsilon(X)$, respectively, for some $\varepsilon > 0$.

A closed interval with endpoints a and b is denoted by $[a, b]$. Intervals $[a, b]$ with $a > b$ are considered empty. For a point $A \in \mathbb{R}^2$, we use $x(A)$ and $y(A)$ to denote the x -coordinate and the y -coordinate of A , respectively.



■ **Figure 1** A star-shaped polygon P with $b(P) \geq \frac{1}{n} - \varepsilon$ and $c(P) \leq \frac{1}{n}$. The polygon P with $4n - 1$ vertices is a union of n triangles $(0,0)(2i,1)(2i + 1,1)$, $i = 0, \dots, n - 1$, and of a triangle $(0,0)(0,\delta)((2n - 1)\delta, \delta)$, where δ is very small.

A *polygonal curve* Γ in \mathbb{R}^d is a curve specified by a sequence (A_1, \dots, A_n) of points of \mathbb{R}^d such that Γ consists of the line segments connecting the points A_i and A_{i+1} for $i = 1, \dots, n - 1$. If $A_1 = A_n$, then the polygonal curve Γ is *closed*. A polygonal curve that is not closed is called a *polygonal line*.

A set $X \subseteq \mathbb{R}^2$ is *polygonally connected*, or *p-connected* for short, if any two points of X can be connected by a polygonal line in X , or equivalently, by a self-avoiding polygonal line in X . For a set X , the relation “ A and B can be connected by a polygonal line in X ” is an equivalence relation on X , and its equivalence classes are the *p-components* of X . A set S is *p-componentwise simply connected* if every p-component of S is simply connected.

A *line segment* in \mathbb{R}^d is a bounded convex subset of a line. A *closed line segment* includes both endpoints, while an *open line segment* excludes both endpoints. For two points A and B in \mathbb{R}^d , we use AB to denote the open line segment with endpoints A and B . A closed line segment with endpoints A and B is denoted by \overline{AB} .

We say that a set $S \subseteq \mathbb{R}^d$ is *star-shaped* if there is a point $C \in S$ such that $\text{Vis}(C, S) = S$. That is, a star-shaped set S contains a point which sees the entire S . Similarly, we say that a set S is *weakly star-shaped* if S contains a line segment ℓ such that $\text{Vis}(\ell, S) = S$.

1.3 Results

We start with a few simple observations. Let S be a subset of \mathbb{R}^2 such that $\text{Seg}(S)$ is measurable. For every $\varepsilon > 0$, S contains a convex subset K of measure at least $(c(S) - \varepsilon)\lambda_2(S)$. Two random points of S both belong to K with probability at least $(c(S) - \varepsilon)^2$, hence $b(S) \geq (c(S) - \varepsilon)^2$. This yields $b(S) \geq c(S)^2$. This simple lower bound on $b(S)$ is tight, as shown by a set S which is a disjoint union of a single large convex component and a large number of small components of negligible size.

It is more challenging to find an upper bound on $b(S)$ in terms of $c(S)$, possibly under additional assumptions on the set S . This is the general problem addressed in this paper.

As a motivating example, observe that a set S consisting of n disjoint convex components of the same size satisfies $b(S) = c(S) = \frac{1}{n}$. It is easy to modify this example to obtain, for any $\varepsilon > 0$, a simple star-shaped polygon P with $b(P) \geq \frac{1}{n} - \varepsilon$ and $c(P) \leq \frac{1}{n}$, see Figure 1. This shows that $b(S)$ cannot be bounded from above by a sublinear function of $c(S)$, even for simple polygons S .

For weakly star-shaped polygons, Cabello et al. [7] showed that the above example is essentially optimal, providing the following linear upper bound on $b(S)$.

► **Theorem 1** ([7, Theorem 5]). *For every weakly star-shaped simple polygon P , we have $b(P) \leq 18c(P)$.*

For polygons that are not weakly star-shaped, Cabello et al. [7] gave a superlinear bound.

► **Theorem 2** ([7, Theorem 6]). *Every simple polygon P satisfies*

$$b(P) \leq 12c(P) \left(1 + \log_2 \frac{1}{c(P)}\right).$$

Moreover, Cabello et al. [7] conjectured that even for a general simple polygon P , $b(P)$ can be bounded from above by a linear function of $c(P)$. The next theorem, which is the first main result of this paper, confirms this conjecture. Recall that $b(S)$ is defined for a set S if and only if S has finite positive measure and $\text{Seg}(S)$ is measurable. Recall also that a set is p -componentwise simply connected if its polygonally-connected components are simply connected. In particular, every simply connected set is p -componentwise simply connected.

► **Theorem 3.** *Every p -componentwise simply connected set $S \subseteq \mathbb{R}^2$ whose $b(S)$ is defined satisfies $b(S) \leq 180c(S)$.*

It is clear that every simple polygon satisfies the assumptions of Theorem 3, hence we directly obtain the following, which confirms the conjecture of Cabello et al. [7].

► **Corollary 4.** *Every simple polygon $P \subseteq \mathbb{R}^2$ satisfies $b(P) \leq 180c(P)$.*

The main restriction in Theorem 3 is the assumption that S is p -componentwise simply connected. This assumption cannot be omitted, as shown by the set $S = [0, 1]^2 \setminus \mathbb{Q}^2$, where it is easy to verify that $c(S) = 0$ and $b(S) = 1$.

A related construction shows that Theorem 3 fails in higher dimensions. To see this, consider again the set $S = [0, 1]^2 \setminus \mathbb{Q}^2$, and define a set $S' \subseteq \mathbb{R}^3$ by

$$S' := \{(tx, ty, t) : t \in [0, 1] \text{ and } (x, y) \in S\}.$$

Again, it is easy to verify that $c(S') = 0$ and $b(S') = 1$, although S' is simply connected, even star-shaped.

Despite these examples, we will show that meaningful analogues of Theorem 3 for higher dimensions and for sets that are not p -componentwise simply connected are possible. The key is to use higher-order generalizations of the Beer index, which we introduce now.

For a set $S \subseteq \mathbb{R}^d$, we define the set $\text{Simp}_k(S) \subseteq (\mathbb{R}^d)^{k+1}$ by

$$\text{Simp}_k(S) := \{(A_0, \dots, A_k) \in S^{k+1} : \text{Conv}(\{A_0, \dots, A_k\}) \subseteq S\},$$

where the operator Conv denotes the convex hull of a set of points. We call $\text{Simp}_k(S)$ the k -simplex set of S . Note that $\text{Simp}_1(S) = \text{Seg}(S)$.

For an integer $k \in \{1, 2, \dots, d\}$ and a set $S \subseteq \mathbb{R}^d$ with finite positive measure and with measurable $\text{Simp}_k(S)$, we define $b_k(S)$ by

$$b_k(S) := \frac{\lambda_{(k+1)d}(\text{Simp}_k(S))}{\lambda_d(S)^{k+1}}.$$

Note that $b_1(S) = b(S)$. We call $b_k(S)$ the k -index of convexity of S . We again leave $b_k(S)$ undefined if S or $\text{Simp}_k(S)$ is non-measurable, or if the measure of S is not finite and positive.

We can view $b_k(S)$ as the probability that the convex hull of $k + 1$ points chosen from S uniformly independently at random is contained in S . For any $S \subseteq \mathbb{R}^d$, we have $b_1(S) \geq b_2(S) \geq \dots \geq b_d(S)$, provided all the $b_k(S)$ are defined.

We remark that the set $S = [0, 1]^d \setminus \mathbb{Q}^d$ satisfies $c(S) = 0$ and $b_1(S) = b_2(S) = \dots = b_{d-1}(S) = 1$. Thus, for a general set $S \subseteq \mathbb{R}^d$, only the d -index of convexity can conceivably admit a nontrivial upper bound in terms of $c(S)$. Our next result shows that such an upper bound on $b_d(S)$ exists and is linear in $c(S)$.

► **Theorem 5.** *For every $d \geq 2$, there is a constant $\beta = \beta(d) > 0$ such that every set $S \subseteq \mathbb{R}^d$ with defined $b_d(S)$ satisfies $b_d(S) \leq \beta c(S)$.*

We do not know if the linear upper bound in Theorem 5 is best possible. We can, however, construct examples showing that the bound is optimal up to a logarithmic factor. This is our last main result.

► **Theorem 6.** *For every $d \geq 2$, there is a constant $\gamma = \gamma(d) > 0$ such that for every $\varepsilon \in (0, 1]$, there is a set $S \subseteq \mathbb{R}^d$ satisfying $c(S) \leq \varepsilon$ and $b_d(S) \geq \gamma \frac{\varepsilon}{\log_2 1/\varepsilon}$, and in particular, we have $b_d(S) \geq \gamma \frac{c(S)}{\log_2 1/c(S)}$.*

In this extended abstract, some proofs have been omitted due to space constraints. The omitted proofs can be found in the full version of this paper [1].

2 Bounding the mutual visibility in the plane

The goal of this section is to prove Theorem 3. Since the proof is rather long and complicated, let us first present a high-level overview of its main ideas.

We first show that it is sufficient to prove the estimate from Theorem 3 for bounded open simply connected sets. This is formalized by the next lemma, whose proof is omitted.

► **Lemma 7.** *Let $\alpha > 0$ be a constant such that every open bounded simply connected set $T \subseteq \mathbb{R}^2$ satisfies $b(T) \leq \alpha c(T)$. It follows that every p -componentwise simply connected set $S \subseteq \mathbb{R}^2$ with defined $b(S)$ satisfies $b(S) \leq \alpha c(S)$.*

Suppose now that S is a bounded open simply connected set. We seek a bound of the form $b(S) = O(c(S))$. This is equivalent to a bound of the form $\lambda_4(\text{Seg}(S)) = O(\text{smc}(S)\lambda_2(S))$. We therefore need a suitable upper bound on $\lambda_4(\text{Seg}(S))$.

We first choose in S a *diagonal* ℓ (i.e., an inclusion-maximal line segment in S), and show that the set $S \setminus \ell$ is a union of two open simply connected sets S_1 and S_2 (Lemma 10). It is not hard to show that the segments in S that cross the diagonal ℓ contribute to $\lambda_4(\text{Seg}(S))$ by at most $O(\text{smc}(S)\lambda_2(S))$ (Lemma 14). Our main task is to bound the measure of $\text{Seg}(S_i \cup \ell)$ for $i = 1, 2$. The two sets $S_i \cup \ell$ are what we call *rooted sets*. Informally, a rooted set is a union of a simply connected open set S' and an open segment $r \subseteq \partial S'$, called the root.

To bound $\lambda_4(\text{Seg}(R))$ for a rooted set R with root r , we partition R into *levels* L_1, L_2, \dots , where L_k contains the points of R that can be connected to r by a polygonal line with k segments, but not by a polygonal line with $k - 1$ segments. Each segment in R is contained in a union $L_i \cup L_{i+1}$ for some $i \geq 1$. Thus, a bound of the form $\lambda_4(\text{Seg}(L_i \cup L_{i+1})) = O(\text{smc}(R)\lambda_2(L_i \cup L_{i+1}))$ implies the required bound for $\lambda_4(\text{Seg}(R))$.

We will show that each p -component of $L_i \cup L_{i+1}$ is a rooted set, with the extra property that all its points are reachable from its root by a polygonal line with at most two segments (Lemma 11). To handle such sets, we will generalize the techniques that Cabello et al. [7] have used to handle weakly star-shaped sets in their proof of Theorem 1. We will assign to every point $A \in R$ a set $\mathfrak{T}(A)$ of measure $O(\text{smc}(R))$, such that for every $(A, B) \in \text{Seg}(R)$, we have either $B \in \mathfrak{T}(A)$ or $A \in \mathfrak{T}(B)$ (Lemma 13). From this, Theorem 3 will follow easily.

To proceed with the proof of Theorem 3 for bounded open simply connected sets, we need a few auxiliary lemmas.

► **Lemma 8.** *For every positive integer d , if S is an open subset of \mathbb{R}^d , then the set $\text{Seg}(S)$ is open and the set $\text{Vis}(A, S)$ is open for every point $A \in S$.*

Proof. Choose a pair of points $(A, B) \in \text{Seg}(S)$. Since S is open and \overline{AB} is compact, there is $\varepsilon > 0$ such that $\mathcal{N}_\varepsilon(\overline{AB}) \subseteq S$. Consequently, for any $A' \in \mathcal{N}_\varepsilon(A)$ and $B' \in \mathcal{N}_\varepsilon(B)$, we have $\overline{A'B'} \subseteq S$, that is, $(A', B') \in \text{Seg}(S)$. This shows that the set $\text{Seg}(S)$ is open. If we fix $A' = A$, then it follows that the set $\text{Vis}(A, S)$ is open. ◀

► **Lemma 9.** *Let S be a simply connected subset of \mathbb{R}^2 and let ℓ and ℓ' be line segments in S . It follows that the set $\text{Vis}(\ell', S) \cap \ell$ is a (possibly empty) subsegment of ℓ .*

Proof. The statement is trivially true if ℓ and ℓ' intersect or have the same supporting line, or if $\text{Vis}(\ell', S) \cap \ell$ is empty. Suppose that these situations do not occur. Let $A, B \in \ell$ and $A', B' \in \ell'$ be such that $\overline{AA'}, \overline{BB'} \subseteq S$. The points A, A', B', B form a (possibly self-intersecting) tetragon Q whose boundary is contained in S . Since S is simply connected, the interior of Q is contained in S . If Q is not self-intersecting, then clearly $\overline{AB} \subseteq \text{Vis}(\ell', S)$. Otherwise, $\overline{AA'}$ and $\overline{BB'}$ have a point D in common, and every point $C \in AB$ is visible in R from the point $C' \in A'B'$ such that $D \in \overline{CC'}$. This shows that $\text{Vis}(\ell', S) \cap \ell$ is a convex subset and hence a subsegment of ℓ . ◀

Now, we define rooted sets and their tree-structured decomposition, and we explain how they arise in the proof of Theorem 3.

A set $S \subseteq \mathbb{R}^2$ is *half-open* if every point $A \in S$ has a neighborhood $\mathcal{N}_\varepsilon(A)$ that satisfies one of the following two conditions:

1. $\mathcal{N}_\varepsilon(A) \subseteq S$,
2. $\mathcal{N}_\varepsilon(A) \cap \partial S$ is a diameter of $\mathcal{N}_\varepsilon(A)$ splitting it into two subsets, one of which (including the diameter) is $\mathcal{N}_\varepsilon(A) \cap S$ and the other (excluding the diameter) is $\mathcal{N}_\varepsilon(A) \setminus S$.

The condition 1 holds for points $A \in S^\circ$, while the condition 2 holds for points $A \in \partial S$. A set $R \subseteq \mathbb{R}^2$ is a *rooted set* if the following conditions are satisfied:

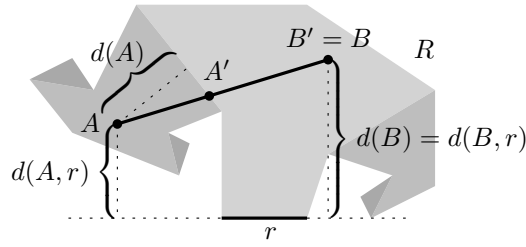
1. R is bounded,
2. R is p-connected and simply connected,
3. R is half-open,
4. $R \cap \partial R$ is an open line segment.

The open line segment $R \cap \partial R$ is called the *root* of R . Every rooted set, as the union of a non-empty open set and an open line segment, is measurable and has positive measure.

A *diagonal* of a set $S \subseteq \mathbb{R}^2$ is a line segment contained in S that is not a proper subset of any other line segment contained in S . Clearly, if S is open, then every diagonal of S is an open line segment. It is easy to see that the root of a rooted set is a diagonal. The following lemma allows us to use a diagonal to split a bounded open simply connected subset of \mathbb{R}^2 into two rooted sets. It is intuitively clear, and its formal proof is omitted.

► **Lemma 10.** *Let S be a bounded open simply connected subset of \mathbb{R}^2 , and let ℓ be a diagonal of S . It follows that the set $S \setminus \ell$ has two p-components S_1 and S_2 . Moreover, $S_1 \cup \ell$ and $S_2 \cup \ell$ are rooted sets, and ℓ is their common root.*

Let R be a rooted set. For a positive integer k , the *kth level* L_k of R is the set of points of R that can be connected to the root of R by a polygonal line in R consisting of k segments but cannot be connected to the root of R by a polygonal line in R consisting of fewer than k segments. We consider a degenerate one-vertex polygonal line as consisting of one degenerate segment, so the root of R is part of L_1 . Thus $L_1 = \text{Vis}(r, R)$, where r denotes the root of R . A *k-body* of R is a p-component of L_k . A *body* of R is a k -body of R for some k . See Figure 2 for an example of a rooted set and its partitioning into levels and bodies.



■ **Figure 2** Example of a rooted set R partitioned into six bodies. The three levels of R are distinguished with three shades of gray. The segment $A'B'$ is the base segment of \overline{AB} .

We say that a rooted set P is *attached* to a set $Q \subseteq \mathbb{R}^2 \setminus P$ if the root of P is subset of the interior of $P \cup Q$. The following lemma explains the structure of levels and bodies. Although it is intuitively clear, its formal proof requires quite a lot of work and is omitted.

- **Lemma 11.** *Let R be a rooted set and $(L_k)_{k \geq 1}$ be its partition into levels. It follows that*
1. $R = \bigcup_{k \geq 1} L_k$; consequently, R is the union of all its bodies;
 2. every body P of R is a rooted set such that $P = \text{Vis}(r, P)$, where r denotes the root of P ;
 3. L_1 is the unique 1-body of R , and the root of L_1 is the root of R ;
 4. every j -body P of R with $j \geq 2$ is attached to a unique $(j - 1)$ -body of R .

Lemma 11 yields a tree structure on the bodies of R . The root of this tree is the unique 1-body L_1 of R , called the *root body* of R . For a k -body P of R with $k \geq 2$, the parent of P in the tree is the unique $(k - 1)$ -body of R that P is attached to, called the *parent body* of P .

- **Lemma 12.** *Let R be a rooted set, $(L_k)_{k \geq 1}$ be the partition of R into levels, ℓ be a closed line segment in R , and $k \geq 1$ be minimum such that $\ell \cap L_k \neq \emptyset$. It follows that $\ell \subseteq L_k \cup L_{k+1}$, $\ell \cap L_k$ is a subsegment of ℓ contained in a single k -body P of R , and $\ell \cap L_{k+1}$ consists of at most two subsegments of ℓ each contained in a single $(k + 1)$ -body whose parent body is P .*

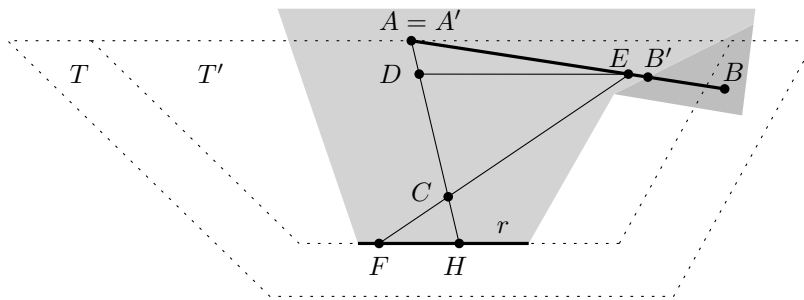
Proof. The definition of the levels directly yields $\ell \subseteq L_k \cup L_{k+1}$. The segment ℓ splits into subsegments each contained in a single k -body or $(k + 1)$ -body of R . By Lemma 11, the bodies of any two consecutive of these subsegments are in the parent-child relation of the body tree. This implies that $\ell \cap L_k$ lies within a single k -body P . By Lemma 9, $\ell \cap L_k$ is a subsegment of ℓ . Consequently, $\ell \cap L_{k+1}$ consists of at most two subsegments. ◀

In the setting of Lemma 12, we call the subsegment $\ell \cap L_k$ of ℓ the *base segment* of ℓ , and we call the body P that contains $\ell \cap L_k$ the *base body* of ℓ . See Figure 2 for an example.

The following lemma is the crucial part of the proof of Theorem 3.

- **Lemma 13.** *If R is a rooted set, then every point $A \in R$ can be assigned a measurable set $\mathfrak{T}(A) \subseteq \mathbb{R}^2$ so that the following is satisfied:*
1. $\lambda_2(\mathfrak{T}(A)) < 87 \text{smc}(R)$;
 2. for every line segment \overline{BC} in R , we have either $B \in \mathfrak{T}(C)$ or $C \in \mathfrak{T}(B)$;
 3. the set $\{(A, B) : A \in R \text{ and } B \in \mathfrak{T}(A)\}$ is measurable.

Proof. Let P be a body of R with the root r . First, we show that P is entirely contained in one closed half-plane defined by the supporting line of r . Let h^- and h^+ be the two open half-planes defined by the supporting line of r . According to the definition of a rooted set, the sets $\{D \in r : \exists \varepsilon > 0 : \mathcal{N}_\varepsilon(D) \cap h^- = \mathcal{N}_\varepsilon(D) \cap (P \setminus r)\}$ and $\{D \in r : \exists \varepsilon > 0 : \mathcal{N}_\varepsilon(D) \cap h^+ = \mathcal{N}_\varepsilon(D) \cap (P \setminus r)\}$ are open and partition the entire r , hence one of them must be empty. This



■ **Figure 3** Illustration for the proof of Claim 1 in the proof of Lemma 13.

implies that the segments connecting r to $P \setminus r$ lie all in h^- or all in h^+ . Since $P = \text{Vis}(r, P)$, we conclude that $P \subseteq h^-$ or $P \subseteq h^+$.

According to the above, we can rotate and translate the set R so that r lies on the x -axis and P lies in the half-plane $\{B \in \mathbb{R}^2 : y(B) \geq 0\}$. For a point $A \in R$, we use $d(A, r)$ to denote the y -coordinate of A after such a rotation and translation of R . We use $d(A)$ to denote $d(A, r)$ where r is the root of the body of A . It follows that $d(A) \geq 0$ for every $A \in R$.

Let $\gamma \in (0, 1)$ be a fixed constant whose value will be specified at the end of the proof. For a point $A \in R$, we define the sets

$$\begin{aligned} \mathfrak{V}_1(A) &:= \{B \in \text{Vis}(A, S) : |A'B'| \geq \gamma|AB|, A \in \text{Vis}(r'', R), d(A', r'') \geq d(B', r'')\}, \\ \mathfrak{V}_2(A) &:= \{B \in \text{Vis}(A, S) : |A'B'| \geq \gamma|AB|, A \notin \text{Vis}(r'', R), d(A', r'') \geq d(B', r'')\}, \\ \mathfrak{V}_3(A) &:= \{B \in \text{Vis}(A, S) : |A'B'| < \gamma|AB|, |AA'| \geq |BB'| \}, \end{aligned}$$

where r'' denotes the root of the base body of \overline{AB} and A' and B' denote the endpoints of the base segment of \overline{AB} such that $|AA'| < |AB'|$. These sets are pairwise disjoint, and we have $A \in \bigcup_{i=1}^3 \mathfrak{V}_i(B)$ or $B \in \bigcup_{i=1}^3 \mathfrak{V}_i(A)$ for every line segment \overline{AB} in R . If for some $B \in \bigcup_{i=1}^3 \mathfrak{V}_i(A)$ the point A lies on r'' , then we have $B \in \mathfrak{V}_1(A)$ and $\mathfrak{V}_1(A) \subseteq r''$.

For the rest of the proof, we fix a point $A \in R$. We show that the union $\bigcup_{i=1}^3 \mathfrak{V}_i(A)$ is contained in a measurable set $\mathfrak{T}(A) \subseteq \mathbb{R}^2$ with $\lambda_2(\mathfrak{T}(A)) < 87 \text{smc}(R)$ that is the union of three trapezoids. We let P be the body of A and r be the root of P . If P is a k -body with $k \geq 2$, then we use r' to denote the root of the parent body of P .

► **Claim 1.** $\mathfrak{V}_1(A)$ is contained in a trapezoid $\mathfrak{T}_1(A)$ with area $6\gamma^{-2} \text{smc}(R)$.

Let H be a point of r such that $\overline{AH} \subseteq R$. Let T' be the r -parallel trapezoid of height $d(A)$ with bases of length $\frac{8 \text{smc}(R)}{d(A)}$ and $\frac{4 \text{smc}(R)}{d(A)}$ such that A is the center of the larger base and H is the center of the smaller base. The homothety with center A and ratio γ^{-1} transforms T' into the trapezoid $T := A + \gamma^{-1}(T' - A)$. Since the area of T' is $6 \text{smc}(R)$, the area of T is $6\gamma^{-2} \text{smc}(R)$. We show that $\mathfrak{V}_1(A) \subseteq T$. See Figure 3 for an illustration.

Let B be a point in $\mathfrak{V}_1(A)$. Using similar techniques to the ones used by Cabello et al. [7] in the proof of Theorem 1, we show that $B \in T$. Let $A'B'$ be the base segment of \overline{AB} such that $|AA'| < |AB'|$. Since $B \in \mathfrak{V}_1(A)$, we have $|A'B'| \geq \gamma|AB|$, $A \in \text{Vis}(r'', R)$, and $d(B, r'') \leq d(A, r'')$, where r'' denotes the root of the base level of \overline{AB} . Since A is visible from r'' in R , the base body of \overline{AB} is the body of A and thus $A = A'$ and $r = r''$. As we have observed, every point $C \in \{A\} \cup AB'$ satisfies $d(C, r) = d(C) \geq 0$.

Let $\varepsilon > 0$. There is a point $E \in AB'$ such that $|B'E| < \varepsilon$. Since E lies on the base segment of \overline{AB} , there is $F \in r$ such that $\overline{EF} \subseteq R$. It is possible to choose F so that \overline{AH} and \overline{EF} have a point C in common where $C \neq F, H$. Let D be a point of \overline{AH} with $d(D) = d(E)$. The point D exists, as $d(H) = 0 \leq d(E) \leq d(A)$. The points A, E, F, H

form a self-intersecting tetragon Q whose boundary is contained in R . Since R is simply connected, the interior of Q is contained in R and the triangles ACE and CFH have area at most $\text{smc}(R)$.

The triangle ACE is partitioned into triangles ADE and CDE with areas $\frac{1}{2}(d(A) - d(D))|DE|$ and $\frac{1}{2}(d(D) - d(C))|DE|$, respectively. Therefore, we have $\frac{1}{2}(d(A) - d(C))|DE| = \lambda_2(ACE) \leq \text{smc}(R)$. This implies

$$|DE| \leq \frac{2 \text{smc}(R)}{d(A) - d(C)}.$$

For the triangle CFH , we have $\frac{1}{2}d(C)|FH| = \lambda_2(CFH) \leq \text{smc}(R)$. By the similarity of the triangles CFH and CDE , we have $|FH| = |DE|d(C)/(d(E) - d(C))$ and therefore

$$|DE| \leq \frac{2 \text{smc}(R)}{d(C)^2}(d(E) - d(C)).$$

Since the first upper bound on $|DE|$ is increasing in $d(C)$ and the second is decreasing in $d(C)$, the minimum of the two is maximized when they are equal, that is, when $d(C) = d(A)d(E)/(d(A) + d(E))$. Then we obtain $|DE| \leq \frac{2 \text{smc}(R)}{d(A)^2}(d(A) + d(E))$. This and $0 \leq d(E) \leq d(A)$ imply $E \in T'$. Since ε can be made arbitrarily small and T' is compact, we have $B' \in T'$. Since $|AB'| \geq \gamma|AB|$, we conclude that $B \in T$. This completes the proof of Claim 1.

► **Claim 2.** $\mathfrak{V}_2(A)$ is contained in a trapezoid $\mathfrak{T}_2(A)$ with area $3(1 - \gamma)^{-2}\gamma^{-2} \text{smc}(R)$.

We assume the point A is not contained in the first level of R , as otherwise $\mathfrak{V}_2(A)$ is empty. Let p be the r' -parallel line that contains the point A and let q be the supporting line of r . Let p^+ and q^+ denote the closed half-planes defined by p and q , respectively, such that $r' \subseteq p^+$ and $A \notin q^+$. Let O be the intersection point of p and q .

Let $T' \subseteq p^+ \cap q^+$ be the trapezoid of height $d(A, r')$ with one base of length $\frac{4 \text{smc}(R)}{(1-\gamma)^2 d(A, r')}$ on p , the other base of length $\frac{2 \text{smc}(R)}{(1-\gamma)^2 d(A, r')}$ on the supporting line of r' , and one lateral side on q . The homothety with center O and ratio γ^{-1} transforms T' into the trapezoid $T := O + \gamma^{-1}(T' - O)$. Since the area of T' is $3(1 - \gamma)^{-2} \text{smc}(R)$, the area of T is $3(1 - \gamma)^{-2}\gamma^{-2} \text{smc}(R)$. We show that $\mathfrak{V}_2(A) \subseteq T$. See Figure 3 for an illustration.

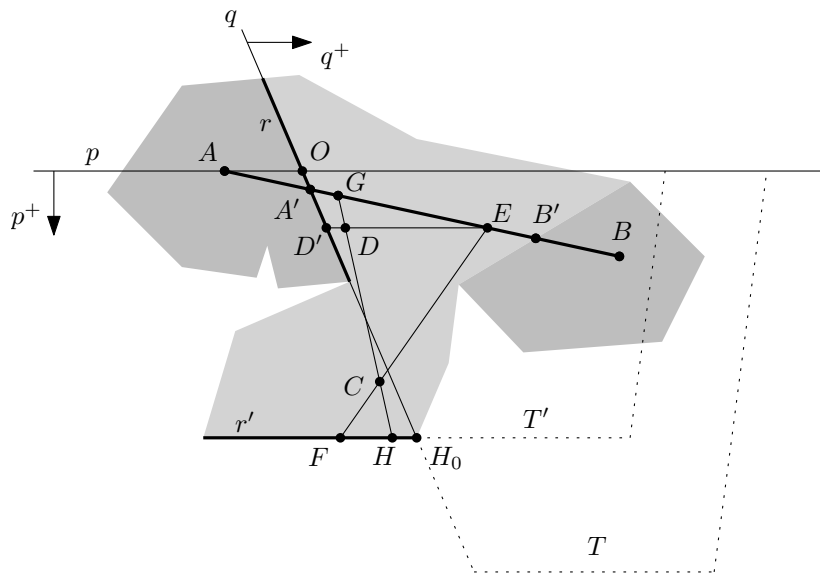
Let B be a point of $\mathfrak{V}_2(A)$. We use $A'B'$ to denote the base segment of \overline{AB} such that $|AA'| < |AB'|$. By the definition of $\mathfrak{V}_2(A)$, we have $|A'B'| \geq \gamma|AB|$, $A \notin \text{Vis}(r'', R)$, and $d(B, r'') \leq d(A, r'')$, where r'' denotes the root of the base body of \overline{AB} . By Lemma 12 and the fact that $A \notin \text{Vis}(r'', R)$, we have $r' = r''$. The bound $d(A, r') \geq d(B, r')$ thus implies $A' \in r \cap p^+$ and $B \in q^+$. We have $d(C, r') = d(C) \geq 0$ for every $C \in A'B'$.

Observe that $(1 - \gamma)d(A, r') \leq d(A', r') \leq d(A, r')$. The upper bound is trivial, as $d(B, r') \leq d(A, r')$ and A' lies on \overline{AB} . For the lower bound, we use the expression $A' = tA + (1 - t)B'$ for some $t \in [0, 1]$. This gives us $d(A', r') = td(A, r') + (1 - t)d(B', r')$. By the estimate $|A'B'| \geq \gamma|AB|$, we have

$$|AA'| + |BB'| \leq (1 - \gamma)|AB| = (1 - \gamma)(|AB'| + |BB'|).$$

This can be rewritten as $|AA'| \leq (1 - \gamma)|AB'| - \gamma|BB'|$. Consequently, $|BB'| \geq 0$ and $\gamma > 0$ imply $|AA'| \leq (1 - \gamma)|AB'|$. This implies $t \geq 1 - \gamma$. Applying the bound $d(B', r') \geq 0$, we conclude that $d(A', r') \geq (1 - \gamma)d(A, r')$.

Let $(G_n)_{n \in \mathbb{N}}$ be a sequence of points from $A'B'$ that converges to A' . For every $n \in \mathbb{N}$, there is a point $H_n \in r'$ such that $\overline{G_n H_n} \subseteq R$. Since $\overline{r'}$ is compact, there is a subsequence of $(H_n)_{n \in \mathbb{N}}$ that converges to a point $H_0 \in \overline{r'}$. We claim that $H_0 \in q$. Suppose otherwise, and



■ **Figure 4** Illustration for the proof of Claim 2 in the proof of Lemma 13.

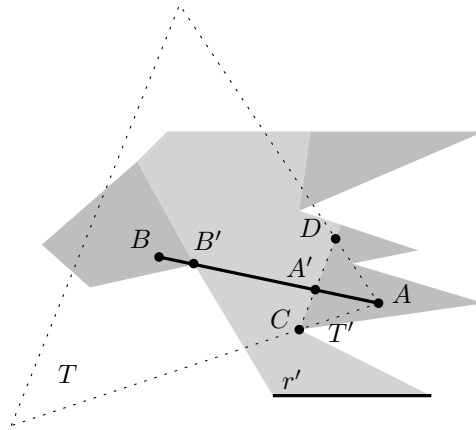
let $q' \neq q$ be the supporting line of $\overline{A'H_0}$. Let $\varepsilon > 0$ be small enough so that $\mathcal{N}_\varepsilon(A') \subseteq R$. For n large enough, $\overline{G_n H_n}$ is contained in an arbitrarily small neighborhood of q' . Consequently, for n large enough, the supporting line of $\overline{G_n H_n}$ intersects q at a point K_n such that $\overline{G_n K_n} \subseteq \mathcal{N}_\varepsilon(A')$, which implies $K_n \in r \cap \text{Vis}(r', R)$, a contradiction.

Again, let $\varepsilon > 0$. There is a point $E \in A'B'$ such that $|B'E| < \varepsilon$. Let D' be a point of q with $d(D', r') = d(E)$. Let $\delta > 0$. There are points $G \in A'B'$ and $H \in r'$ such that $G \in \mathcal{N}_\delta(A')$ and $\overline{GH} \subseteq R \cap \mathcal{N}_\delta(q)$. If δ is small enough, then $d(E) \leq d(A', r') - \delta \leq d(G) \leq d(A', r')$. Let D be the point of \overline{GH} with $d(D) = d(E)$. The point E lies on $A'B'$ and thus it is visible from a point $F \in r'$. Again, we can choose F so that the line segments \overline{EF} and \overline{GH} have a point C in common where $C \neq F, H$. The points E, F, H, G form a self-intersecting tetragon Q whose boundary is in R . The interior of Q is contained in R , as R is simply connected. Therefore, the area of the triangles CEG and CFH is at most $\text{smc}(R)$. The argument used in the proof of Claim 1 yields $|DE| \leq \frac{2 \text{smc}(R)}{d(G)^2} (d(G) + d(E)) \leq \frac{2 \text{smc}(R)}{(d(A', r') - \delta)^2} (d(A', r') + d(E))$. This and the fact that δ (and consequently $|D'D|$) can be made arbitrarily small yield $|D'E| \leq \frac{2 \text{smc}(R)}{d(A', r')^2} (d(A', r') + d(E))$. This together with $d(A', r') \geq (1 - \gamma)d(A, r')$ yield $|D'E| \leq \frac{2 \text{smc}(R)}{(1 - \gamma)^2 d(A, r')^2} (d(A, r') + d(E))$. This and $0 \leq d(E) \leq d(A, r')$ imply $E \in T'$. Since ε can be made arbitrarily small and T' is compact, we have $B' \in T'$. Since $|A'B'| \geq \gamma|AB| \geq \gamma|A'B|$, we conclude that $B \in T$. This completes the proof of Claim 2.

► **Claim 3.** $\mathfrak{V}_3(A)$ is contained in a trapezoid $\mathfrak{T}_3(A)$ with area $(4(1 - \gamma)^{-2} - 1) \text{smc}(R)$.

By Lemma 9, the points of r that are visible from A in R form a subsegment CD of r . The homothety with center A and ratio $2(1 - \gamma)^{-1}$ transforms the triangle $T' := ACD$ into the triangle $T'' := A + 2(1 - \gamma)^{-1}(T' - A)$. See Figure 5 for an illustration. We claim that $\mathfrak{V}_3(A)$ is a subset of the trapezoid $T := T'' \setminus T'$.

Let B be an arbitrary point of $\mathfrak{V}_3(A)$. Consider the segment \overline{AB} with the base segment $A'B'$ such that $|AA'| < |AB'|$. Since $B \in \mathfrak{V}_3(A)$, we have $|A'B'| < \gamma|AB|$ and $|AA'| \geq |BB'|$. This implies $|AA'| \geq \frac{1 - \gamma}{2}|AB| > 0$ and hence $A \neq A'$ and $B \notin P$. From the definition of C and D , we have $A' \in \overline{CD}$. Since $|AA'| \geq \frac{1 - \gamma}{2}|AB|$ and $B \notin P$, we have $B \in T$.



■ **Figure 5** Illustration for the proof of Claim 3 in the proof of Lemma 13.

The area of T is $(4(1 - \gamma)^{-2} - 1)\lambda_2(T')$. The interior of T' is contained in R , as all points of the open segment CD are visible from A in R . The area of T' is at most $\text{smc}(R)$, as its interior is a convex subset of R . Consequently, the area of T is at most $(4(1 - \gamma)^{-2} - 1)\text{smc}(R)$. This completes the proof of Claim 3.

To put everything together, we set $\mathfrak{T}(A) := \bigcup_{i=1}^3 \mathfrak{T}_i(A)$. It follows that $\bigcup_{i=1}^3 \mathfrak{B}_i(A) \subseteq \mathfrak{T}(A)$ for every $A \in R$. Clearly, the set $\mathfrak{T}(A)$ is measurable. Summing the three estimates on areas of the trapezoids, we obtain

$$\lambda_2(\mathfrak{T}(A)) \leq (6\gamma^{-2} + 3(1 - \gamma)^{-2}\gamma^{-2} + 4(1 - \gamma)^{-2} - 1)\text{smc}(R)$$

for every point $A \in R$. We choose $\gamma \in (0, 1)$ so that the value of the coefficient is minimized. For $x \in (0, 1)$, the function $x \mapsto 6x^{-2} + 3(1 - x)^{-2}x^{-2} + 4(1 - x)^{-2} - 1$ attains its minimum $86.7027 < 87$ at $x \approx 0.5186$. Altogether, we have $\lambda_2(\mathfrak{T}(A)) < 87\text{smc}(R)$ for every $A \in R$.

It remains to show that the set $\{(A, B) : A \in R \text{ and } B \in \mathfrak{T}(A)\}$ is measurable. For every body P of R and for $i \in \{1, 2, 3\}$, the definition of the trapezoid $\mathfrak{T}_i(A)$ in Claim i implies that the set $\{(A, B) : A \in P \text{ and } B \in \mathfrak{T}_i(A)\}$ is the intersection of $P \times \mathbb{R}^2$ with a semialgebraic (hence measurable) subset of $(\mathbb{R}^2)^2$ and hence is measurable. There are countably many bodies of R , as each of them has positive measure. Therefore, $\{(A, B) : A \in R \text{ and } B \in \mathfrak{T}(A)\}$ is a countable union of measurable sets and hence is measurable. ◀

Let S be a bounded open subset of the plane, and let ℓ be a diagonal of S that lies on the x -axis. For a point $A \in S$, we define the set

$$\mathfrak{S}(A, \ell) := \{B \in \text{Vis}(A, S) : AB \cap \ell \neq \emptyset \text{ and } |y(A)| \geq |y(B)|\}.$$

The following lemma is a slightly more general version of a result of Cabello et al. [7].

► **Lemma 14.** *Let S be a bounded open simply connected subset of \mathbb{R}^2 , and let ℓ be its diagonal that lies on the x -axis. It follows that $\lambda_2(\mathfrak{S}(A, \ell)) \leq 3\text{smc}(S)$ for every $A \in S$.*

Proof. Using an argument similar to the proof of Lemma 8, we can show that the set $\{B \in \text{Vis}(A, S) : AB \cap \ell \neq \emptyset\}$ is open. Therefore, $\mathfrak{S}(A, \ell)$ is the intersection of an open set and the closed half-plane $\{(x, y) \in \mathbb{R}^2 : y \leq -y(A)\}$ or $\{(x, y) \in \mathbb{R}^2 : y \geq -y(A)\}$, whichever contains A . Consequently, the set $\mathfrak{S}(A, \ell)$ is measurable for every point $A \in S$.

We clearly have $\lambda_2(\mathfrak{S}(A, \ell)) = 0$ for points $A \in S \setminus \text{Vis}(\ell, S)$. By Lemma 9, the set $\text{Vis}(A, S) \cap \ell$ is an open subsegment CD of ℓ . The interior T° of the triangle $T := ACD$ is

contained in S . Since T° is a convex subset of S , we have $\lambda_2(T^\circ) = \frac{1}{2}|CD| \cdot |y(A)| \leq \text{smc}(S)$. Therefore, every point $B \in \mathfrak{S}(A, \ell)$ is contained in a trapezoid of height $|y(A)|$ with bases of length $|CD|$ and $2|CD|$. The area of this trapezoid is $\frac{3}{2}|CD| \cdot |y(A)| \leq 3 \text{smc}(S)$. Hence we have $\lambda_2(\mathfrak{S}(A, \ell)) \leq 3 \text{smc}(S)$ for every point $A \in S$. \blacktriangleleft

Proof of Theorem 3. In view of Lemma 7, we can assume without loss of generality that S is an open bounded simply connected set. Let ℓ be a diagonal of S . We can assume without loss of generality that ℓ lies on the x -axis. According to Lemma 10, the set $S \setminus \ell$ has exactly two p-components S_1 and S_2 , the sets $S_1 \cup \ell$ and $S_2 \cup \ell$ are rooted sets, and ℓ is their common root. By Lemma 13, for $i \in \{1, 2\}$, every point $A \in S_i \cup \ell$ can be assigned a measurable set $\mathfrak{T}_i(A)$ so that $\lambda_2(\mathfrak{T}_i(A)) < 87 \text{smc}(S_i \cup \ell) \leq 87 \text{smc}(S)$, every line segment \overline{BC} in $S_i \cup \ell$ satisfies $B \in \mathfrak{T}_i(C)$ or $C \in \mathfrak{T}_i(B)$, and the set $\{(A, B) : A \in S_i \cup \ell \text{ and } B \in \mathfrak{T}_i(A)\}$ is measurable. We set $\mathfrak{S}(A) := \mathfrak{T}_i(A) \cup \mathfrak{S}(A, \ell)$ for every point $A \in S_i$ with $i \in \{1, 2\}$. We set $\mathfrak{S}(A) := \mathfrak{T}_1(A) \cup \mathfrak{T}_2(A)$ for every point $A \in \ell = S \setminus (S_1 \cup S_2)$. Let

$$\mathfrak{S} := \{(A, B) : A \in S \text{ and } B \in \mathfrak{S}(A)\} \cup \{(B, A) : A \in S \text{ and } B \in \mathfrak{S}(A)\} \subseteq (\mathbb{R}^2)^2.$$

It follows that the set \mathfrak{S} is measurable.

Let \overline{AB} be a line segment in S , and suppose $|y(A)| \geq |y(B)|$. Then either A and B are in distinct p-components of $S \setminus \ell$ or they both lie in the same component S_i with $i \in \{1, 2\}$. In the first case, we have $B \in \mathfrak{S}(A)$, since AB intersects ℓ and $\mathfrak{S}(A, \ell) \subseteq \mathfrak{S}(A)$. In the second case, we have $B \in \mathfrak{T}_i(A) \subseteq \mathfrak{S}(A)$ or $A \in \mathfrak{T}_i(B) \subseteq \mathfrak{S}(B)$. Therefore, we have $\text{Seg}(S) \subseteq \mathfrak{S}$. Since both $\text{Seg}(S)$ and \mathfrak{S} are measurable, we have

$$\lambda_4(\text{Seg}(S)) \leq \lambda_4(\mathfrak{S}) \leq 2 \int_{A \in S} \lambda_2(\mathfrak{S}(A)),$$

where the second inequality is implied by Fubini's Theorem. Using the bound $\lambda_2(\mathfrak{S}(A)) \leq 90 \text{smc}(S)$, we obtain

$$\lambda_4(\text{Seg}(S)) \leq 2 \int_S 90 \text{smc}(S) = 180 \text{smc}(S) \lambda_2(S).$$

Finally, this bound can be rewritten as $b(S) = \lambda_4(\text{Seg}(S)) \lambda_2(S)^{-2} \leq 180 c(S)$. \blacktriangleleft

3 General dimension

In this section, we sketch the proofs of Theorem 5 and Theorem 6. The detailed proofs can be found in the full version of this paper [1]. In both proofs, we use the operator Aff to denote the affine hull of a set of points.

Sketch of the proof of Theorem 5. Let $T = (B_0, B_1, \dots, B_d)$ be a $(d + 1)$ -tuple of distinct affinely independent points of S , ordered in such a way that the following two conditions hold:

1. the segment $\overline{B_0 B_1}$ is the diameter of T , and
2. for $i = 2, \dots, d - 1$, the point B_i has the maximum distance to $\text{Aff}(\{B_0, \dots, B_{i-1}\})$ among the points B_i, B_{i+1}, \dots, B_d .

For $i = 1, \dots, d - 1$, we define $\text{Box}_i(T)$ inductively as follows:

1. $\text{Box}_1(T) := \overline{B_0 B_1}$,
2. for $i = 2, \dots, d - 1$, $\text{Box}_i(T)$ is the box containing all the points $P \in \text{Aff}(\{B_0, B_1, \dots, B_i\})$ with the following two properties:

- a. the orthogonal projection of P to $\text{Aff}(\{B_0, B_1, \dots, B_{i-1}\})$ lies in $\text{Box}_{i-1}(T)$, and
 - b. the distance of P to $\text{Aff}(\{B_0, B_1, \dots, B_{i-1}\})$ does not exceed the distance of B_i to $\text{Aff}(\{B_0, B_1, \dots, B_{i-1}\})$,
3. $\text{Box}_d(T)$ is the box containing all the points $P \in \mathbb{R}^d$ such that the orthogonal projection of P to $\text{Aff}(\{B_0, B_1, \dots, B_{d-1}\})$ lies in $\text{Box}_{d-1}(T)$ and $\lambda_d(\text{Conv}(\{B_0, B_1, \dots, B_{d-1}, P\})) \leq \lambda_d(S) c(S)$.

It can be verified that if $T \in \text{Simp}_d(S)$, then $\text{Box}_d(T)$ contains the point B_d . Also, it can be shown that the λ_d -measure of $\text{Box}_d(T)$ is equal to $z := 2^{d-2}d! \text{smc}(S)$, which is independent of T . From this, we can deduce that the measure of $\text{Simp}_d(S)$ is at most $(d + 1)\lambda_d(S)^d z$, and hence $b_d(S)$ is at most $(d + 1)z/\lambda_d(S)$, which is of order $c(S)$. ◀

Sketch of the proof of Theorem 6. To obtain a set S with arbitrarily small convexity ratio $c(S)$ and with the d -index of convexity $b_d(S)$ of order $c(S)/\log_2(1/c(S))$, we let S be the open d -dimensional box $(0, 1)^d$ with n points removed. We show that no matter which n -tuple of points we remove from the box, the d -index of convexity $b_d(S)$ is still of order $\Omega(\frac{1}{n})$. Moreover, we show that for some constant $\alpha = \alpha(d) > 0$ it is possible to remove $n = \alpha \frac{1}{\varepsilon} \log_2 \frac{1}{\varepsilon}$ points from the box such that every convex subset of $(0, 1)^d$ with measure at least ε contains a removed point. That is, we obtain $c(S) \leq \varepsilon$ and $b_d(S) \geq \gamma\varepsilon/\log_2(1/\varepsilon)$ for some constant $\gamma = \gamma(d) > 0$. Such an n -tuple of points to be removed is called an ε -net for convex subsets of $(0, 1)^d$. To find it, we first use John’s Lemma [11] to reduce the problem to finding, for a suitably smaller ε' , an ε' -net for ellipsoids restricted to $(0, 1)^d$. Then, we apply a continuous version of the well-known Epsilon Net Theorem for families with bounded Vapnik-Chervonenkis dimension due to Haussler and Welzl [10] (see also [14]). ◀

It is a natural question whether the bound for $b_d(S)$ in Theorem 6 can be improved to $b_d(S) = \Omega(c(S))$. In the plane, this is related to the famous problem of Danzer and Rogers (see [6, 15] and Problem E14 in [8]) which asks whether for given $\varepsilon > 0$ there is a set $N' \subseteq (0, 1)^2$ of size $O(\frac{1}{\varepsilon})$ with the property that every convex set of area ε within the unit square contains at least one point from N' .

If this problem was to be answered affirmatively, then we could use such a set N' to stab $(0, 1)^2$ in our proof of Theorem 6 which would yield the desired bound for $b_2(S)$. However it is generally believed that the answer is likely to be nonlinear in $\frac{1}{\varepsilon}$.

4 Other variants and open problems

We have seen in Theorem 3 that a p -componentwise simply connected set $S \subseteq \mathbb{R}^2$ whose $b(S)$ is defined satisfies $b(S) \leq \alpha c(S)$, for an absolute constant $\alpha \leq 180$. Equivalently, such a set S satisfies $\text{smc}(S) \geq b(S)\lambda_2(S)/180$.

By a result of Blaschke [5] (see also Sas [18]), every convex set $K \subseteq \mathbb{R}^2$ contains a triangle of measure at least $\frac{3\sqrt{3}}{4\pi}\lambda_2(K)$. In view of this, Theorem 3 yields the following consequence.

► **Corollary 15.** *There is a constant $\alpha > 0$ such that every p -componentwise simply connected set $S \subseteq \mathbb{R}^2$ whose $b(S)$ is defined contains a triangle $T \subseteq S$ of measure at least $\alpha b(S)\lambda_2(S)$.*

A similar argument works in higher dimensions as well. For every $d \geq 2$, there is a constant $\beta = \beta(d)$ such that every convex set $K \subseteq \mathbb{R}^d$ contains a simplex of measure at least $\beta\lambda_d(K)$ (see e.g. Lassak [13]). Therefore, Theorem 5 can be rephrased in the following equivalent form.

► **Corollary 16.** *For every $d \geq 2$, there is a constant $\alpha = \alpha(d) > 0$ such that every set $S \subseteq \mathbb{R}^d$ whose $b_d(S)$ is defined contains a simplex T of measure at least $\alpha b_d(S)\lambda_d(S)$.*

What can we say about sets $S \subseteq \mathbb{R}^2$ that are not p -componentwise simply connected? First of all, we can consider a weaker form of simple connectivity: we call a set S *p -componentwise simply Δ -connected* if for every triangle T such that $\partial T \subseteq S$ we have $T \subseteq S$. We conjecture that Theorem 3 can be extended to p -componentwise simply Δ -connected sets.

► **Conjecture 17.** *There is an absolute constant $\alpha > 0$ such that every p -componentwise simply Δ -connected set $S \subseteq \mathbb{R}^2$ whose $b(S)$ is defined satisfies $b(S) \leq \alpha c(S)$.*

What does the value of $b(S)$ say about a planar set S that does not satisfy even a weak form of simple connectivity? Such a set may not contain any convex subset of positive measure, even when $b(S)$ is equal to 1. However, we conjecture that a large $b(S)$ implies the existence of a large convex set whose boundary belongs to S .

► **Conjecture 18.** *For every $\varepsilon > 0$, there is a $\delta > 0$ such that if $S \subseteq \mathbb{R}^2$ is a set with $b(S) \geq \varepsilon$, then there is a bounded convex set $C \subseteq \mathbb{R}^2$ with $\lambda(C) \geq \delta\lambda(S)$ and $\partial C \subseteq S$.*

Theorem 3 shows that Conjecture 18 holds for p -componentwise simply connected sets, with δ being a constant multiple of ε . It is possible that even in the general setting of Conjecture 18, δ can be taken as a constant multiple of ε .

Motivated by Corollary 15, we propose a stronger version of Conjecture 18, where the convex set C is required to be a triangle.

► **Conjecture 19.** *For every $\varepsilon > 0$, there is a $\delta > 0$ such that if $S \subseteq \mathbb{R}^2$ is a set with $b(S) \geq \varepsilon$, then there is a triangle $T \subseteq \mathbb{R}^2$ with $\lambda(T) \geq \delta\lambda(S)$ and $\partial T \subseteq S$.*

Note that Conjecture 19 holds when restricted to p -componentwise simply connected sets, as implied by Corollary 15.

We can generalise Conjecture 19 to higher dimensions and to higher-order indices of convexity. To state the general conjecture, we introduce the following notation: for a set $X \subseteq \mathbb{R}^d$, let $\binom{X}{k}$ be the set of k -element subsets of X , and let the set $\text{Skel}_k(X)$ be defined by

$$\text{Skel}_k(X) := \bigcup_{Y \in \binom{X}{k+1}} \text{Conv}(Y).$$

If X is the vertex set of a d -dimensional simplex $T = \text{Conv}(X)$, then $\text{Skel}_k(X)$ is often called the *k -dimensional skeleton* of T . Our general conjecture states, roughly speaking, that sets with large k -index of convexity should contain the k -dimensional skeleton of a large simplex. Here is the precise statement.

► **Conjecture 20.** *For every $k, d \in \mathbb{N}$ such that $1 \leq k \leq d$ and every $\varepsilon > 0$, there is a $\delta > 0$ such that if $S \subseteq \mathbb{R}^d$ is a set with $b_k(S) \geq \varepsilon$, then there is a simplex T with vertex set X such that $\lambda_d(T) \geq \delta\lambda_d(S)$ and $\text{Skel}_k(X) \subseteq S$.*

Corollary 16 asserts that this conjecture holds in the special case of $k = d \geq 2$, since $\text{Skel}_d(X) = \text{Conv}(X) = T$. Corollary 15 shows that the conjecture holds for $k = 1$ and $d = 2$ if S is further assumed to be p -componentwise simply connected. In all these cases, δ can be taken as a constant multiple of ε , with the constant depending on k and d .

Finally, we can ask whether there is a way to generalize Theorem 3 to higher dimensions, by replacing simple connectivity with another topological property. Here is an example of one such possible generalization.

► **Conjecture 21.** For every $d \geq 2$, there is a constant $\alpha = \alpha(d) > 0$ such that if $S \subseteq \mathbb{R}^d$ is a set with defined $b_{d-1}(S)$ whose every p -component is contractible, then $b_{d-1}(S) \leq \alpha c(S)$.

A modification of the proof of Theorem 5 implies that Conjecture 21 is true for star-shaped sets S .

Acknowledgment. The authors would like to thank to Marek Eliáš for interesting discussions about the problem and participation in our meetings during the early stages of the research.

References

- 1 M. Balko, V. Jelínek, P. Valtr, and B. Walczak. On the Beer index of convexity and its variants. full version, arXiv:1412.1769.
- 2 G. Beer. Continuity properties of the visibility function. *Michigan Math. J.*, 20:297–302, 1973.
- 3 G. Beer. The index of convexity and the visibility function. *Pacific J. Math.*, 44(1):59–67, 1973.
- 4 G. Beer. The index of convexity and parallel bodies. *Pacific J. Math.*, 53(2):337–345, 1974.
- 5 W. Blaschke. Über affine Geometrie III: Eine Minimumeigenschaft der Ellipse. *Ber. Verh. Kön. Sächs. Ges. Wiss. Leipzig Math.-Phys. Kl.*, 69:3–12, 1917.
- 6 P. G. Bradford and V. Capovleas. Weak ε -nets for points on a hypersphere. *Discrete Comput. Geom.*, 18(1):83–91, 1997.
- 7 S. Cabello, J. Cibulka, J. Kynčl, M. Saumell, and P. Valtr. Peeling potatoes near-optimally in near-linear time. In *Proceedings of the 30th Annual Symposium on Computational Geometry*, pages 224–231, 2014.
- 8 H. T. Croft, K. J. Falconer, and R. K. Guy. *Unsolved Problems in Geometry*. Unsolved Problems in Intuitive Mathematics. Springer New York, 2nd edition, 1991.
- 9 J. E. Goodman. On the largest convex polygon contained in a non-convex n -gon, or how to peel a potato. *Geom. Dedicata*, 11(1):99–106, 1981.
- 10 D. Haussler and E. Welzl. Epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2(2):127–151, 1987.
- 11 F. John. Extremum problems with inequalities as subsidiary conditions. In *Studies and Essays, presented to R. Courant on his 60th birthday, January 8, 1948*, pages 187–204, 1948.
- 12 R. Lang. A note on the measurability of convex sets. *Arch. Math. (Basel)*, 47:90–92, 1986.
- 13 M. Lassak. Approximation of convex bodies by inscribed simplices of maximum volume. *Beitr. Algebra Geom.*, 52(2):389–394, 2011.
- 14 J. Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Graduate Texts in Mathematics*. Springer New York, 2002.
- 15 J. Pach and G. Tardos. Piercing quasi-rectangles—on a problem of Danzer and Rogers. *J. Combin. Theory Ser. A*, 119(7):1391–1397, 2012.
- 16 V. V. Prasolov. *Elements of combinatorial and differential topology*, volume 74 of *Graduate Studies in Mathematics*. American Mathematical Society, 2006.
- 17 G. Rote. The degree of convexity. In *Abstracts of the 29th European Workshop on Computational Geometry*, pages 69–72, 2013.
- 18 E. Sas. Über eine Extremumeigenschaft der Ellipsen. *Compositio Math.*, 6:468–470, 1939.
- 19 H. I. Stern. Polygonal entropy: a convexity measure for polygons. *Pattern Recogn. Lett.*, 10(4):229–235, 1989.

Tight Bounds for Conflict-Free Chromatic Guarding of Orthogonal Art Galleries

Frank Hoffmann¹, Klaus Kriegel¹, Subhash Suri², Kevin Verbeek³,
and Max Willert¹

1 Freie Universität Berlin, Institut für Informatik, 14195 Berlin, Germany
{hoffmann,kriegel,willerma}@mi.fu-berlin.de

2 Dept. of Computer Science, University of California, Santa Barbara, USA
suri@cs.ucsb.edu

3 Dept. of Mathematics and Computer Science, TU Eindhoven, The Netherlands
k.a.b.verbeek@tue.nl

Abstract

The chromatic art gallery problem asks for the minimum number of “colors” t so that a collection of point guards, each assigned one of the t colors, can see the entire polygon subject to some conditions on the colors visible to each point. In this paper, we explore this problem for orthogonal polygons using *orthogonal visibility*—two points p and q are mutually visible if the smallest axis-aligned rectangle containing them lies within the polygon. Our main result establishes that for a *conflict-free* guarding of an orthogonal n -gon, in which at least one of the colors seen by every point is unique, the number of colors is $\Theta(\log \log n)$. By contrast, the best upper bound for orthogonal polygons under standard (non-orthogonal) visibility is $O(\log n)$ colors. We also show that the number of colors needed for *strong* guarding of simple orthogonal polygons, where all the colors visible to a point are unique, is $\Theta(\log n)$. Finally, our techniques also help us establish the first non-trivial lower bound of $\Omega(\log \log n / \log \log \log n)$ for conflict-free guarding under standard visibility. To this end we introduce and utilize a novel discrete combinatorial structure called *multicolor tableau*.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms, G.2.2 Graph Theory

Keywords and phrases Orthogonal polygons, art gallery problem, hypergraph coloring

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.421

1 Introduction

The classic Art Gallery Problem (AGP) posed by Klee in 1973 asks for the minimum number of guards sufficient to watch an art gallery modelled by an n -sided simple polygon P . A guard sees a point in P if the connecting line segment is contained in P . Therefore, a guard watches a star polygon contained in P and the question is to cover P by a collection of stars with smallest possible cardinality. The answer is $\lfloor \frac{n}{3} \rfloor$ as shown by Chvátal [3]. This result was the starting point for a rich body of research about algorithms, complexity and combinatorial aspects for many variants of the original question. Surveys can be found in the seminal monograph by O’Rourke [10], in Shermer [12], and Urrutia [15].

Graph coloring arguments have been frequently used for proving worst case combinatorial bounds for art gallery type questions starting with Fisk’s proof [5]. Somehow surprisingly, chromatic versions of the AGP have been proposed and studied only recently. There are two chromatic variants: strong chromatic guarding and conflict-free guarding of a polygon P . In both versions we look for a guard set G and give each guard one of t colors. The chromatic



© Frank Hoffmann, Klaus Kriegel, Subhash Suri, Kevin Verbeek, and Max Willert;
licensed under Creative Commons License CC-BY

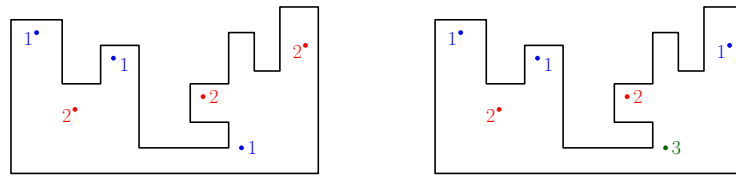
31st International Symposium on Computational Geometry (SoCG’15).

Editors: Lars Arge and János Pach; pp. 421–435



Leibniz International Proceedings in Informatics

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



■ **Figure 1** Example of conflict-free (left) and strong chromatic (right) r-guarding.

guarding is said to be strong if for each point $p \in P$ all guards $G(p)$ that see p have pairwise different colors [4]. It is conflict-free if in each $G(p)$ there is at least one guard with a unique color, see [1]. The goal is to determine guard sets such that the number of colors sufficient for these purposes is minimal. Observe, in both versions minimizing the number of guards is not part of the objective function. Figure 1 shows a simple orthogonal polygon with both conflict-free and strong chromatic guardings in the orthogonal visibility model.

To grasp the nature of the problem, observe that it has two conflicting aspects. We have to guard the polygon but at the same time we want the guards to hide from each other, since then we can give them the same color. For example, in the strong version we want a guard set that can be partitioned into a minimal number of subsets and in each subset the pairwise orthogonal link distance is at least 3. Moreover, we will see a strong dependence of the results on the underlying visibility model, standard vs. orthogonal. We refer to standard and orthogonal visibility as l-visibility (line visibility) and r-visibility, respectively. We use superscripts l and r in the bounds to indicate the model.

Let $\chi_{st}^l(n)$ and $\chi_{cf}^l(n)$ denote the minimal number of colors sufficient for any simple polygon on n vertices in the strong chromatic and in the conflict-free version if based on line visibility.

Here is a short summary of known bounds. For simple orthogonal polygons on n vertices $\chi_{cf}^l(n) \in O(\log n)$, as shown in [1]. The same bound applies to simple general polygons, see [2]. Both proofs are based on subdividing the polygon into weak visibility subpolygons that are in a certain sense independent with respect to conflict-free chromatic guarding. For the strong chromatic version we have $\chi_{st}^l(n) \in \Theta(n)$ for simple polygons and $\chi_{st}^l(n) \in \Omega(\sqrt{n})$ even for the monotone orthogonal case, see [4]. NP-hardness is discussed in [6]. In [4], simple $O(1)$ upper bounds are shown for special polygon classes like spiral polygons and orthogonal staircase polygons combined with line visibility.

Next we state our main contributions for simple orthogonal polygons:

1. For the strong chromatic version we show $\chi_{st}^r(n) \in \Theta(\log n)$.
2. For the conflict-free chromatic version we show $\chi_{cf}^r(n) \in \Theta(\log \log n)$.
3. For line visibility guards we have: $\chi_{cf}^l(n) \in \Omega(\log \log n / \log \log \log n)$.

This is the first super-constant lower bound also for general simple polygons.

The chromatic AGP versions can be easily interpreted as coloring questions for concrete geometric hypergraphs. Smorodinsky ([14]) gives a nice survey of both practical and theoretical aspects of hypergraph coloring. A special role play hypergraphs that arise in geometry. For example, given a set of points P in the plane and a set of regions \mathcal{R} (e.g. rectangles, disks), we can define the hypergraph $H_{\mathcal{R}}(P) = (P, \{P \cap S \mid S \in \mathcal{R}\})$. The discrete interval hypergraph $H_{\mathcal{I}}$ is a concrete example of such a hypergraph: We take n points on a line and all possible intervals as regions. It is not difficult to see that $\chi_{cf}(H_{\mathcal{I}}) \in \Theta(\log n)$. As to our AGP versions, we can associate with a given polygon and a guard set a geometric hypergraph. Its vertices are the guards and a hyperedge is defined by a set of guards for which there exists a point that can see exactly these guards. Then one wants to color this hypergraph in

a conflict-free or in a strong manner. Another example is the following rectangle hypergraph. The vertex set is a set of n axis-aligned rectangles and each maximal subset of rectangles with a common intersection forms a hyperedge. Here the order for the conflict-free chromatic number is $\Omega(\log n)$ and $O(\log^2 n)$ as shown in [11, 14].

Looking at our results, it is not a big surprise that the combination of orthogonal polygons with r -visibility yields the strongest bounds. This is simply due to additional structural properties and this phenomenon has already been observed for the original AGP. For example, the $\lfloor \frac{n}{4} \rfloor$ tight worst case bound for covering simple orthogonal polygons with general stars can also be proven for r -stars (see [10]) and it holds even for orthogonal polygons with holes, see [7]. Further, while minimizing the number of guards is NP-hard both for simple general and orthogonal polygons if based on line visibility, it becomes polynomially solvable for r -visibility in the simple orthogonal case, see [9, 17]. The latter result is based on the solution of the strong perfect graph conjecture.

The paper is organized as follows. We give necessary basic definitions in the next section. Then we prove upper bounds in Section 3 using techniques developed in [1, 2]. That means we also subdivide a simple orthogonal polygon into histograms which are independent with respect to chromatic guarding. To deal with a single histogram we introduce the notion of its *spine tree*. The spine tree provides an elegant and efficient way to describe r -visibility properties of the histogram. Our main contributions are the lower bound proofs in Section 4. Especially, we introduce a novel combinatorial structure called multicolor tableau. This structure enables us to show a first super-constant lower bound for chromatic conflict-free guarding based on the line visibility model.

2 Preliminaries

We study simple orthogonal polygons, i.e., polygons consisting of alternating vertical and horizontal edges only. By $|P|$ we denote the number of vertices, by ∂P the boundary and by $\text{int}P = P \setminus \partial P$ the interior of the polygon. Vertices can be reflex or convex. A *reflex* vertex has an interior angle $3\pi/2$ while *convex* vertices have an interior angle of $\pi/2$. We do not make any general position assumption for the simple orthogonal polygons P . Points $p, q \in P$ are *r -visible* to each other if the closed axis-parallel rectangle $r[p, q]$ with diagonal pq is contained in P . In the following, unless stated otherwise, visible always means r -visible. The *visibility polygon* of p , the set of all points visible from p , is formally defined as $V(p) = \{q \in P \mid r[p, q] \subseteq P\}$. A polygon that is fully visible from one of its points is called a *star*. For $P' \subset P$ we define its visibility polygon by $V(P') = \cup_{p \in P'} V(p)$. The *windows* of a subpolygon P' in P are those parts of $\partial P'$ that do not belong to ∂P .

For an orthogonal polygon P we construct its induced *r -visibility arrangement* $\mathcal{A}^r(P)$: For each reflex vertex of P we extend both incident boundary edges into $\text{int}P$ until they meet the boundary again, therefore defining a subdivision of the polygon. The 2-dimensional faces of this arrangement are rectangles. Clearly, points from the interior of the same rectangle (subsequently called *cell*) have the same visibility polygon.

Finally, we define special classes of orthogonal polygons. A weak visibility polygon, also known as *histogram*, has a boundary *base edge* e connecting two convex vertices such that $V(e) = P$. A histogram that is a star is called a *pyramid*.

Conflict-free and strong chromatic guarding

A set G of points is a *guard set* for an orthogonal polygon P if their visibility polygons jointly cover the whole polygon. If in addition each guard $g \in G$ is assigned one color $\gamma(g)$ from a

fixed finite set of colors $[t] = \{1, 2, \dots, t\}$ we have a *chromatic guarding* (G, γ) . Next we give the central definitions. Since these definitions are independent of the visibility model, we drop the superscripts l and r in the following.

A chromatic guard set (G, γ) for P is *strong* if each point in P sees only differently colored guards. (G, γ) is a *conflict-free* guarding if for any point $p \in P$ there is at least one guard in the guard set $G(p) = V(p) \cap G$ whose color is unique among all guards visible from p .

Figure 1 illustrates both concepts. We denote by $\chi_{cf}(P)$ the minimal t such that there is a conflict-free chromatic guarding set for P using t colors. Maximizing this value over all polygons with n vertices from a specified polygon class is denoted by $\chi_{cf}(n)$.

Consequently, we denote by $\chi_{st}(P)$ the minimal t such that there is strong chromatic guarding set using t colors. Maximizing this value for all polygons with n vertices from a specified polygon class defines the value $\chi_{st}(n)$. Observe that minimizing the guard number is not part of the objective function. However, in our upper bound proofs we use at most a linear number of guards, which is asymptotically optimal in worst case.

3 Upper Bounds

We show two upper bounds for simple orthogonal polygons of size n in the r -visibility model: $\chi_{st}^r(n) \in O(\log n)$ and $\chi_{cf}^r(n) \in O(\log \log n)$. These bounds are even realized by guards placed in the interior of visibility cells. This restriction simplifies the arguments and does not affect the asymptotic bounds. Furthermore we use the simple fact that a polygon is guarded iff its interior is guarded. The upper bound proof is inspired by ideas developed in [1, 2] for conflict-free guarding of simple polygons based on line visibility.

3.1 Reduction to histograms

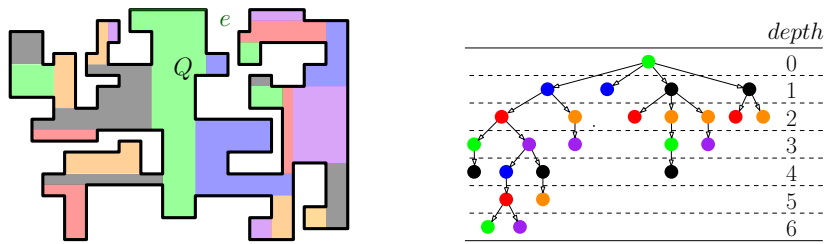
We reuse the central concept of independence introduced in [1, 2] for line visibility. Independence means that one can use the same color sets for coloring guards in independent subpolygons. The following definition suffices for our purposes and covers both the strong and the conflict-free variant:

Let P be a simple orthogonal polygon and P_1 and P_2 subpolygons of P . We call P_1 and P_2 *independent* if no point in P can simultaneously see points from $\text{int}P_1$ and $\text{int}P_2$.

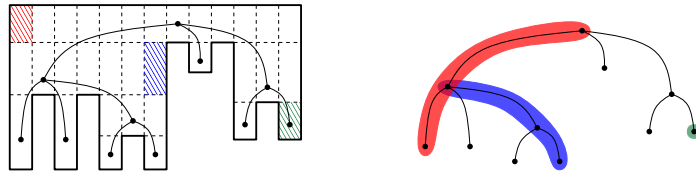
Next, we hierarchically subdivide an orthogonal polygon P into a linear number of histograms by a standard window partitioning process, see [1]. For the sake of simplicity we make the (weak) assumption that the obtained histograms have no degenerate edges.

The subdivision is represented by a partition tree $\mathcal{H} = \mathcal{H}_P(e)$ with histograms as node set. Let e be a highest horizontal boundary edge. The visibility polygon of e is a histogram Q . This is the root vertex of \mathcal{H} . Now Q splits P into parts and defines a finite set (possibly empty if $Q = P$) of vertical windows w_1, \dots, w_k . Then we recurse, see Figure 2, with the windows being the new base edges. Each window corresponds to a last left or right turn of a shortest orthogonal path from e to the histogram defined by the window. So we can accordingly label the histograms to be *left* or *right* histograms. We define the root Q to be a left histogram.

Let $H_d, d = 0, 1, 2$, be the family of all histograms corresponding to nodes in \mathcal{H} with depth congruent $d \pmod 3$. We further partition H_d into H_d^L and H_d^R depending on whether the histograms are left or right histograms, respectively. In Figure 2 the six families of histograms are color-coded for illustration. For example, the dark gray histograms are right children with depth congruent $1 \pmod 3$.



■ **Figure 2** The partition into histograms and the corresponding partition tree.



■ **Figure 3** Spine tree and the bijection between open cells and monotone paths.

► **Lemma 1.** *Let P be a polygon and $H_d^L, d = 0, 1, 2$ the family of histograms corresponding to left nodes in \mathcal{H} with depth congruent $d \pmod 3$. Then the interior of histograms in each H_d^L have pairwise orthogonal link distance at least three, analogously for H_d^R , so they are independent.*

3.2 Guarding a histogram

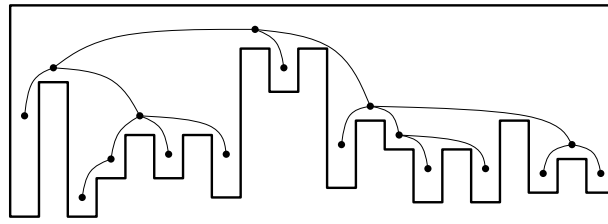
Consider a histogram $H \subseteq P$ with a top horizontal base edge. We associate with H a tree T as follows. Consider the set of cells in the r-visibility arrangement \mathcal{A}^r . If several cells have the same visibility polygon we choose the leftmost cell as representative of this equivalence class. Let R be the set of all representatives and $B \subseteq R$ the subset of cells incident to bottom horizontal histogram edges. We define a partial order for B : We say $b' \leq b$ iff the horizontal polygon edge of b' is not above that of b and there is an $r \in R$ that sees both b and b' . The Hasse diagram of this poset is a tree T which we call the *spine tree* of H . A *monotone* path π in T is a directed subpath of a root-to-leaf path. It corresponds to a pair (b, b') with $b' \leq b$.

► **Lemma 2.** *There is a bijective mapping Φ between cells of R and monotone paths in T such that two cells are visible from each other iff the corresponding monotone paths in T share a node.*

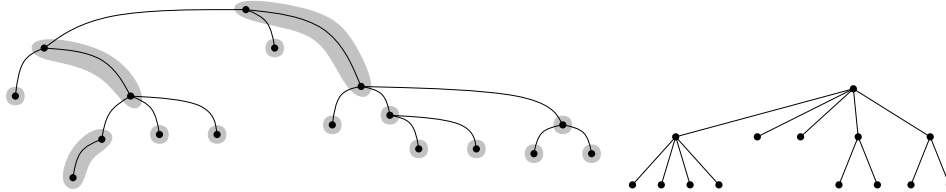
Proof. Let r be a cell in R . Then $V(r) \cap T$ is some monotone path π in T and we set $\Phi(r) = \pi$. For the inverse function let π be a monotone path in T from vertex b down to b' . We associate with π the unique cell $\Phi^{-1}(\pi) = r \in R$ that is vertically aligned with b' and horizontally with b .

We observe that Φ is well-defined by the choice of the leftmost representative for visibility equivalent cells and it is clearly a bijection. Especially, for $\pi = (b, b)$ we have $\Phi^{-1}(\pi) = b$.

For the second assertion consider two cells r, r' visible from each other and the smallest rectangle that includes both. By extending this rectangle downwards it hits a horizontal boundary edge. The vertex of T corresponding to that edge is in both $\Phi(r)$ and $\Phi(r')$. For the other direction consider a cell r'' corresponding to a vertex in $\Phi(r) \cap \Phi(r')$. It has a bottom horizontal edge. We form a rectangle in H above this edge of maximal width and maximal height. All cells visible from r'' are in this rectangle, therefore r sees r' . Figure 3 illustrates the bijection. ◀



■ **Figure 4** Example histogram with spine tree.



■ **Figure 5** Monotone paths covering the spine tree and the corresponding compressed spine tree.

Now we translate the geometric concepts of strong and conflict-free guardings of H into equivalent combinatorial questions for the spine tree T . First of all, a colored guard set for H defines a set of colored cells in R and this defines, using Φ , a covering of T with colored monotone *guarding paths* and vice versa. The condition for strong guarding now reads: No monotone path in T can intersect two guarding paths of the same color.

For conflict-free guardings we have:

► **Lemma 3.** *Colored guards g_1, \dots, g_r define a conflict-free guarding for H iff for each monotone path π in T there exists a color and exactly one guarding path $\Phi(g_i)$ with that color such that $\pi \cap \Phi(g_i) \neq \emptyset$.*

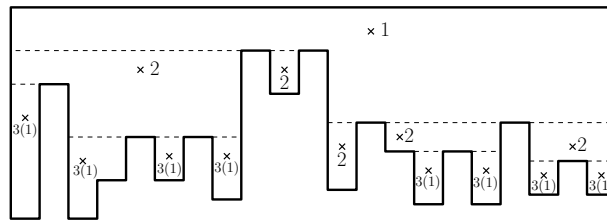
Proof. Consider the cell $\Phi^{-1}(\pi)$. It is seen by a guard g with a unique color c . Therefore $\Phi(g) \cap \pi \neq \emptyset$. Assume, some other c -colored guarding path $\Phi(g')$ intersects π . Then g' sees $\Phi^{-1}(\pi)$, a contradiction. The other direction is analogous. ◀

Path compression: We use a bottom-up path compression to define a covering (in fact, it is a partition) of T by monotone paths. To this end we form, in parallel, for all leaves l the maximal length monotone paths $\pi(l)$ that end in l and do not use nodes of outdegree bigger than one. We cut off all $\pi(l)$ from T . Iterating this procedure yields a unique tree T^* . Its nodes represent monotone paths in T . T^* has depth $O(\log |H|)$ since in each iteration the number of leaves is reduced by at least half. Figure 4 shows an example histogram with its spine tree T . The derived compressed spine tree is depicted in Figure 5.

The above path compression technique is similar but not equivalent to that of heavy path decompositions [13]. In fact, the same bounds can be achieved using the (heavy) path tree of heavy path decompositions as T^* .

► **Proposition 4.** *Let H be a histogram with n vertices. In the r -visibility model there is a strong chromatic guarding with $O(\log n)$ colors and a conflict-free chromatic guarding with $O(\log \log n)$ colors.*

Proof. We construct the spine tree T and the compressed tree T^* with depth $O(\log n)$. To get a strong guarding we color the nodes of T^* , i.e. the guarding paths in T , by their depth in T^* .



■ **Figure 6** Chromatic guarding positions for the example histogram: with colors $\{1,2,3\}$ strong, with colors $\{1,2\}$ conflict-free (in brackets) guarding.

For a conflict-free guarding, consider the color set $[t] = \{1, 2, \dots, t\}$ and the following recursively defined set of words: $s_1 = 1$ and $s_i = s_{i-1} \circ i \circ s_{i-1}$. Clearly, a prefix of s_t with length k has no more than $\lceil \log(k + 1) \rceil$ different colors and each connected subword contains a unique color [14]. Now we color the nodes of T^* from the root to the leaves with the sequence s_t of length at most the height of the tree, that is $O(\log n)$. A color alphabet of size $O(\log \log n)$ suffices. ◀

We illustrate the construction in Figure 6. Observe that we use the same guard positions for both strong and conflict-free guarding. The compressed spine tree has depth 2. For the strong guarding we use colors 1, 2 and 3 while for the conflict-free version we use the color sequence 1-2-1. The guard positions are in the open cells corresponding to the monotone paths via bijection Φ . Moreover, each guard covers a pyramid as indicated in the figure.

► **Theorem 5.** *Let P be an orthogonal polygon with $|P| = n$. We have $\chi_{st}^r(n) \in O(\log n)$ and $\chi_{cf}^r(n) \in O(\log \log n)$.*

Proof. We decompose P into 6 families $H_d^L, H_d^R, d = 0, 1, 2$. Each of the families consists of pairwise independent histograms each of size at most n . Then we apply Proposition 4. ◀

4 Lower Bounds

This section contains three lower bound proofs, two tight lower bounds for strong and for conflict-free r-visibility guards, and a first non-trivial lower bound for conflict-free l-visibility guards in simple polygons. All use the same underlying orthogonal histogram but they completely differ with respect to proof techniques. Both r-visibility cases rely on the spine tree concept from Section 3.2. For line visibility guards we introduce a new combinatorial method which we call multicolor tableau.

4.1 Lower bounds for r-visibility

All lower bounds established in this paper are based on a simple, recursively defined family of so called *spike polygons* S_m , where S_1 is a simple square and S_{m+1} is formed by two copies of S_m separated by a vertical spike, but joined by an additional horizontal layer. Figure 7 illustrates this construction together with the subdivision of S_2 into visibility cells. Obviously, the spine tree T of S_m is a balanced binary tree of height $m - 1$ with vertices corresponding one-to-one to bottom cells in the r-visibility arrangement. Recall, a colored r-visibility guard set for S_m corresponds to a covering of T with colored monotone guarding paths and vice versa.

► **Theorem 6.** *For simple orthogonal polygons $\chi_{st}^r(n) \in \Omega(\log n)$.*

Proof. We show that any strong guarding of S_m requires m different colors. Consider in the spine tree T a guarding path π covering the root with unique color c . Then c does not occur in the left or in the right subtree of the root. By induction we need $m - 1$ more colors for the subtree missed by π . Since S_m has $n = 2^{m+1}$ vertices, the claim follows. ◀

Next we consider a lower bound for the conflict-free version of the problem. To that end, we analyze the special case that a root-to-leaf path π in T is covered by short guarding paths only. Later we show the existence of such a path.

► **Lemma 7.** *Let $\mathcal{P} = \{\pi_1, \dots, \pi_r\}$ be conflict-free guarding paths for a path π with m nodes such that $|\pi_i| = O(m^\epsilon)$ for $1 \leq i \leq r$ and some $0 < \epsilon < 1$. Then this guarding uses at least $\Omega(\log m)$ colors.*

Proof. Let $K = \max\{|\pi_i|, 1 \leq i \leq r\}$. We subdivide π into $k = \frac{m}{K} \in \Omega(m^{1-\epsilon})$ buckets of size K each. This way every π_i can overlap with at most two buckets. Since \mathcal{P} is induced by a conflict-free guarding, there is a color c_1 such that exactly one π_i (responsible for π) is colored with c_1 . Hence there is a subpath of π consisting of at least $\frac{k-2}{2}$ buckets that does not intersect any c_1 -colored path. Applying this argument recursively we obtain the following recursive relation for the number of colors needed for k buckets: $T(k) \geq T(\frac{k-2}{2}) + 1$. This recursive relation easily solves to $T(k) \in \Omega(\log k) \subseteq \Omega(\log m^{1-\epsilon}) = \Omega(\log m)$. ◀

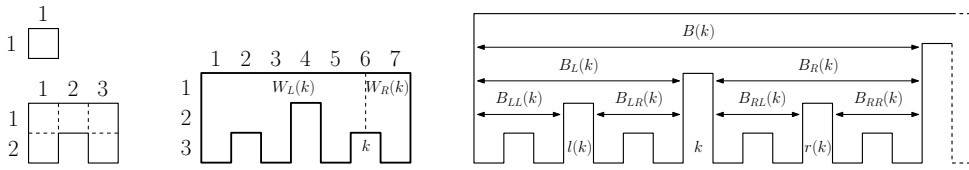
► **Theorem 8.** *For simple orthogonal polygons $\chi_{cf}^r(n) \in \Omega(\log \log n)$.*

Proof. We start with a conflict-free guarding of $S_m, n = 2^{m+1}$ that uses a minimum number of t colors. By Theorem 5 we have $t \in O(\log \log n) = O(\log m)$. Consider the corresponding family \mathcal{F} of guarding paths in T . We denote by $\mathcal{U}(v_0)$ the set of all guarding paths from \mathcal{F} covering the root v_0 of T with a unique color. Since $|\mathcal{U}(v_0)| \leq t$ there must be a vertex v_1 at depth $\lfloor \log t \rfloor + 1$ that is not part of any path from $\mathcal{U}(v_0)$. Now we iterate starting from v_1 . We take all guarding paths covering v_1 with a unique color and we determine a node v_2 at depth $2\lfloor \log t \rfloor + 2$ missed by these paths, and so on. We call the v_i 's *checkpoints*. The checkpoints define a root-to-leaf path π with length $m = \log n - 1$. Consider $\mathcal{F}_\pi = \{\pi \cap \pi_i | \pi_i \in \mathcal{F}\}$. Now form a new family \mathcal{U}_π that consists of all maximal subpaths σ of members $\pi_i \cap \pi \in \mathcal{F}_\pi$ such that σ does not intersect any other member of the same color in \mathcal{F}_π . Let $\pi' \subset \pi$ and assume $\pi_i \in \mathcal{F}$ gives a unique color to π' . Then $\pi' \cap \pi_i$ is part of some path in \mathcal{U}_π . Thus \mathcal{U}_π is a conflict-free guarding path family for π . By construction, paths in \mathcal{U}_π have length at most $2\lfloor \log t \rfloor + 1$. Now we can apply Lemma 7. Since $2\log t + 1 \in O(\log \log m) \subseteq O(m^{0.5})$ we get $t \in \Omega(\log m) = \Omega(\log \log n)$. ◀

4.2 Blocks, stretched spike polygons, and multicolor tableaux

We now turn our attention to conflict-free chromatic guarding in the line visibility model. The concepts needed in our lower bound proof are explained in this section.

Columns of S_m are numbered left to right by indices $k \in [2^m - 1]$, and cells in column k top down by an additional index $i \in [d_m(k)]$ where $d_m(k)$ is the depth of column k in S_m . Formally, we have $d_m(k) = m - \pi_2(k)$, where $\pi_2(k)$ is the multiplicity of factor 2 in the prime decomposition of k . Obviously, a column has maximal depth m iff its index is odd. We introduce the notions of the left and right wing of column k in order to distinguish guard positions: The *left wing* $W_L(k)$ is the set of all points strictly on the left side of the midline of column k and the *right wing* is the complement $W_R(k) = S_m \setminus W_L(k)$ including the midline.



■ **Figure 7** Spike polygons S_1 and S_2 (left), left wing and right wing of column $k = 6$ in S_3 (middle), blocks and subblocks (right).

We define the *block* $B(k)$ of column k as the interval of all neighboring columns of depth at least $d_m(k)$, see Figure 7:

$$B(k) = \left[k - \left(2^{\pi_2(k)} - 1 \right), k + \left(2^{\pi_2(k)} - 1 \right) \right]$$

Geometrically, a block is nothing but a smaller spike polygon. Deleting its central column a block splits into a left and a right subblock:

$$B_L(k) = \left[k - \left(2^{\pi_2(k)} - 1 \right), k - 1 \right] \quad , \quad B_R(k) = \left[k + 1, k + \left(2^{\pi_2(k)} - 1 \right) \right]$$

For odd k we have $B(k) = \{k\}$ and $B_L(k) = B_R(k) = \emptyset$. Later it will be necessary to subdivide a left or right subblock again into its left and right subblocks. These *quarter-subblocks* can be described making use of the definition above together with the central column $l(k) = k - 2^{\pi_2(k)-1}$ in block $B_L(k)$ and column $r(k) = k + 2^{\pi_2(k)-1}$ in block $B_R(k)$:

$$B_{LL}(k) = B_L(l(k)) \quad , \quad B_{LR}(k) = B_R(l(k)) \quad , \quad B_{RL}(k) = B_L(r(k)) \quad , \quad B_{RR}(k) = B_R(r(k)).$$

Next we introduce a vertically stretched, but combinatorially equivalent version S_m^\downarrow of S_m with the following properties:

1. The width of each column is 1 and hence the total width of S_m^\downarrow is again $2^m - 1$.
2. We will distinguish between combinatorial and geometric depth of a column: While $d_m(k) = m - \pi_2(k)$ is still the combinatorial depth, we want the *geometric depth* to be $d_m^\downarrow(k) = 2^{(d_m(k)-1)m}$. That means that the height of the first row is $h_1 = 1$ and the height of the i -th row $h_i = 2^{im} - 2^{(i-1)m}$.

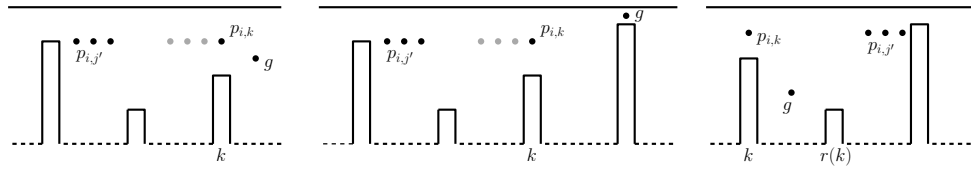
Consider the r-visibility(!) arrangement of S_m^\downarrow with the rectangular r-visibility cells $r_{i,k}$. Next we discretize the conflict-free l-guarding problem. Let $p_{i,k}$ be the bottom side midpoint of cell $r_{i,k}$, that is the cell in row i and column k . If $\gamma : G \rightarrow [t]$ for a guard set G is a conflict-free l-guarding of S_m^\downarrow , then let $M_{i,k}$ be the multiset of all colors of guards that see point $p_{i,k}$. By $m_{i,k}(c)$ we denote the multiplicity of color c in this multiset.

We call $\mathcal{M}(\gamma) = \{M_{i,k} \mid k \in [2^m - 1], i \in [d_m(k)]\}$ the corresponding *multicolor tableau*. The set of unique guard colors for point $p_{i,k}$ is defined by $U_{i,k} := \{c \in [t] \mid m_{i,k}(c) = 1\}$ and the standard inclusion relation $M_{i,k} \subseteq M_{j,l}$ for multisets by: $\forall c \in [t] \quad m_{i,k}(c) \leq m_{j,l}(c)$.

The next two lemmata state simple l-visibility properties in stretched spike polygons.

► **Lemma 9.** *Let g be an l-guard in S_m^\downarrow and k a column of this polygon with combinatorial depth $d = d_m(k)$ and geometric depth $d_m^\downarrow(k) = 2^{(d-1)m}$. If $g \in W_R(k)$ ($g \in W_L(k)$) then g cannot see any point p at depth $d^\downarrow(p) \geq 2^{dm}$ in the left (right) block of k . In particular, g cannot see any point $p_{i,j}$ with $j \in B_L(k)$ ($j \in B_R(k)$) and $i > d$.*

Proof. By symmetry it is sufficient to study the first case with $g \in W_R(k)$, $d^\downarrow(p) \geq 2^{dm}$ and p a point in the subpolygon $B_L(k)$. Let q_L be the left vertex of the horizontal polygon edge



■ **Figure 8** Possible guard positions with respect to the point $p_{i,k}$. Note that it is impossible to display the exponential growth of the row heights in the drawing.

in column k and consider the slopes s_1 and s_2 of the lines $\overline{pq_L}$ and $\overline{q_Lg}$. Since the width of $B_L(k)$ is $2^{m-d} - 1$ and $d^\downarrow(p) - d^\downarrow(q_L) \geq 2^{dm} - 2^{(d-1)m} = (2^m - 1) \cdot 2^{(d-1)m}$ we get

$$s_1 \geq \frac{(2^m - 1) \cdot 2^{(d-1)m}}{2^{m-d} - 1} = \frac{(2^m - 1) \cdot 2^{(d-1)m}}{2^{-d}(2^m - 2^d)} > \frac{2^{(d-1)m}}{2^{-d}} = 2^{(d-1)m+d}$$

Since g is in the right wing of k it is at least one half unit right of q_L and it is at most $d_m^\downarrow(k) = 2^{(d-1)m}$ units higher than q_L . We get

$$s_2 \leq \frac{2^{(d-1)m}}{1/2} = 2^{(d-1)m+1} \leq 2^{(d-1)m+d}$$

Thus $s_1 > s_2$, which shows that vertex q_l blocks the l-visibility between $p_{i,j}$ and g . ◀

► **Lemma 10.** *Let g be an l-guard watching a point $p_{i,k} \in S_m^\downarrow$. Then, for all $i' \leq i$ and for all $j \in B_L(k)$ or for all $j \in B_R(k)$, g sees also $p_{i',j}$.*

Proof. Let $d^\downarrow(g)$ be the geometric depth of g in S_m^\downarrow .

Case 1: (Fig. 8, left) If g is even an r-guard for $p_{i,k}$ then the rectangle spanned by g and $p_{i,k}$ can be horizontally extended over the whole block $B(k)$ as well as upwards to the top of S_m^\downarrow . Thus the claim holds for all $j \in B(k)$.

Otherwise there are two more cases, namely that $d^\downarrow(g)$ is strictly smaller or strictly larger than $2^{(i-1)m}$.

Case 2: (Fig. 8, middle) $d^\downarrow(g) < 2^{(i-1)m}$, i.e., g sees $p_{i,k}$ from above. If $g \in W_R(k)$ then g can see all $p_{i,j}$ with $j \in B_L(k)$ because the line segments $p_{i,j}p_{i,k}$ and $p_{i,k}g$ are contained in S_m^\downarrow and they form a chain that is convex from above. If $g \in W_L(k)$ then g can see all $p_{i,j}$ with $j \in B_R(k)$ because the line segments $gp_{i,k}$ and $p_{i,k}p_{i,j}$ are contained in S_m^\downarrow and they form a chain that is convex from above. Moreover it is clear that in S_m^\downarrow any guard that sees a point $p_{i,j}$ will see also all points directly above, especially the points $p_{i',j}$ with $i' < i$.

Case 3: (Fig. 8, right) $d^\downarrow(g) > 2^{(i-1)m}$, i.e., g sees $p_{i,k}$ from below. We can additionally assume $d^\downarrow(g) > d_m^\downarrow(k) = 2^{(d_m(k)-1)m}$ since otherwise we are in Case 1 again. By Lemma 9 (with the roles of guard and guarded point exchanged) we know that g is in row $i'' = d_m(k) + 1$ in some cell $r_{i'',j}$ with $j \in B_L(k)$ or $j \in B_R(k)$. It follows that, depending on whether g lies in $B_L(k)$ or $B_R(k)$, g sees all $p_{i'',j'}$ with $j' \in B_L(k)$ or $j' \in B_R(k)$ and all points above. ◀

A multicolor tableau $\mathcal{M}(\gamma)$ has standard size if it consists of m rows and $N = 2^m - 1$ columns. But by various constructions, for example restricting it to a single block, one creates a new tableau having m rows and $N' = 2^{m'} - 1$ columns for some $m' < m$.

The following central definition of t -conformity specifies some necessary, but not sufficient conditions a multicolor tableau has if it stems from a conflict-free t -chromatic l-guarding of a stretched spike polygon. Later we will show that t -conformity is preserved when acting on the tableau with various combinatorial operations defined below.

► **Definition 11.** Let $m' \leq m$ be natural numbers and $N' = 2^{m'} - 1$. A combinatorial scheme of multisets over the ground set $[t]$ of the form $\mathcal{M} = (M_{i,k} \mid k \in [N'], i \in [d_m(k)])$ is called a t -conform $m \times N'$ multicolor tableaux if the following properties hold:

1. cf-Property: $\forall k \in [N'] \forall i \in [d_m(k)] U_{i,k} \neq \emptyset$.
2. Monotonicity: $\forall k \in [N'] \forall 1 \leq i < i' \leq d_m(k) M_{i',k} \subseteq M_{i,k}$.
3. LR-quarter-block property: If c is a unique color for some point $p_{i,k}$ in column k then there exists a quarter-subblock $B_{XY}(k)$ with $XY \in \{LL, LR, RL, RR\}$ such that for all columns $j \in B_{XY}(k)$ the following predicate $Q(j)$ holds. $Q(j)$ is the conjunction of three conditions:
 - a. $c \in M_{i,j}$
 - b. $c \in U_{i,j} \rightarrow c \notin M_{d_m(k)+2,j}$
 - c. $c \notin U_{i,j} \rightarrow \exists Z \in \{L,R\} \forall j' \in B_Z(j) c \notin U_{i,j'}$.

► **Proposition 12.** *The multicolor tableau $\mathcal{M}(\gamma)$ for a conflict-free l -guarding γ of the polygon S_m^\downarrow with t colors is t -conform.*

Proof. There is nothing to prove for the cf-property and the monotonicity. Now let us assume $c \in U_{i,k}$ with a corresponding guard g . By symmetry we may suppose $g \in W_R(k)$. Again, there are three cases to distinguish (see Figure 8):

1. $p_{i,k}$ is r-visible from g .
2. $p_{i,k}$ is not r-visible from g and $p_{i,k}$ is deeper than g .
3. $p_{i,k}$ is not r-visible from g and g is deeper than $p_{i,k}$.

In Case 1 and Case 2 we choose $XY = LL$ (but $XY = LR$ would also work – the gray points). In Case 3 the choice depends on the position of g relative to the central column $r(k)$ of the block $B_R(k)$:

$$XY = \begin{cases} RL & \text{if } g \in W_R(r(k)) \\ RR & \text{if } g \in W_L(r(k)). \end{cases}$$

It remains to establish the three conditions of $Q(j)$ for all $j \in B_{XY}(k)$. Condition (a) is obvious in Case 1 and Case 2. In Case 3 it follows from the fact that g cannot be deeper than $d_m^\downarrow(r(k))$, see the proof of Lemma 9.

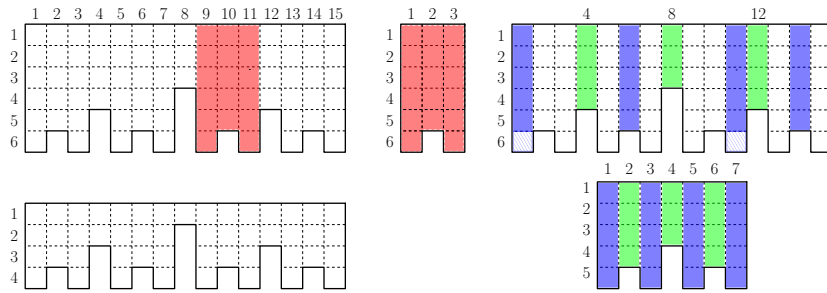
For condition (b) suppose that $c \in U_{i,j}$. This implies that g is the only guard with color c that sees $p_{i,j}$. However, in all three cases g is in the wing opposite to block $B_{XY}(k)$ and then g cannot see any point of combinatorial depth $d_m(k) + 2$ in $B_{XY}(k)$ by Lemma 9. It is worth observing that depth $d_m(k) + 1 = d_m(r(k))$ does not suffice in Case 3. Any other c -colored guard watching $p_{d_m(k)+2,j}$ would also watch $p_{i,j}$ and therefore contradicts the uniqueness of g . Thus $c \notin M_{d_m(k)+2,j}$.

Finally, let us suppose $c \notin U_{i,j}$, then there is a second c -colored guard g' for $p_{i,j}$. Now we can conclude from Lemma 10 that g' watches all points $p_{i,j'}$ for $j' \in B_L(j)$ or for all $j' \in B_R(j)$. This proves condition (c). ◀

4.3 The lower bound proof for l-visibility

We start with describing operations on t -conform multicolor tableaux that maintain this property. In Figure 9 the three operations are geometrically illustrated, however, the operations themselves are defined for the combinatorial tableaux.

► **Proposition 13.** *If $\mathcal{M} = (M_{i,k} \mid k \in [N'], i \in [d_m(k)])$ is a t -conform $m \times N'$ tableau with $N' = 2^{m'} - 1$ for some $m' \leq m$, then the following three constructions yield new t -conform tableaux $\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3$:*



■ **Figure 9** Illustrating horizontal (red), vertical (lower left), and selective (blue-green) truncation of a 6×15 multicolor tableau.

1. *Horizontal truncation:* \mathcal{M}_1 results from restricting \mathcal{M} to a block $B(k)$;
2. *Vertical truncation:* \mathcal{M}_2 results from deleting the top $m - m'$ rows of \mathcal{M} ;
3. *Selective truncation:* \mathcal{M}_3 results from selecting $2^{m^*} - 1$ columns for some $m^* < m'$ with respect to the following rules:
 - For all even $k \in [2^{m^*} - 1]$ choose column $k \cdot 2^{m' - m^*}$ of \mathcal{M} as column k of \mathcal{M}_3 .
 - For all odd $k \in [2^{m^*} - 1]$ choose any column j of \mathcal{M} with $(k - 1) \cdot 2^{m' - m^*} < j < (k + 1) \cdot 2^{m' - m^*}$, delete from that column all entries of depth $d > m^* + m - m'$ and use this truncated column as column k of \mathcal{M}_3 .

Proof. Recall, the width of $B(k)$ is $N'' = 2^{m''} - 1$ where $m'' = 2\pi_2(k)$. So the only thing that one has to do for \mathcal{M}_1 is shifting the column indexing from the interval $B(k) = [k - 2^{\pi_2(k)} + 1, k + 2^{\pi_2(k)} - 1]$ to $[N'']$. Then \mathcal{M}_1 is t -conform.

For the second construction it is sufficient to shift down by $m - m'$ the index of each row that is not deleted. Then \mathcal{M}_2 is an $m' \times N'$ tableau. Note that an old row index $d_m(k) = m - \pi_2(k)$ becomes $d_{m'}(k)$. Having that in mind, it is also trivial that \mathcal{M}_2 is t -conform.

The construction of \mathcal{M}_3 already contains the renumbering of indices. Again, it is not hard to conclude the t -conformity because the construction preserves the relation of being a column in the left (or right) subblock of another column. ◀

► **Theorem 14.** $\chi_{cf}^l(n) \in \Omega\left(\frac{\log \log n}{\log \log \log n}\right)$.

Proof. We define a recursive function $m(t)$ by $m(1) = 3$ and $m(t) = 1 + t \cdot (m(t - 1) + 1)$ for $t \geq 2$.

The inequality $m(t) \leq (t + 1)!$ holds for all $t \geq 5$ by induction. In fact, $m(5) = 651 < 720 = (5 + 1)!$ and the induction step works for any $t \geq 6$ as follows:

$$m(t) = t \cdot (m(t - 1) + 1) + 1 \leq t(t! + 1) + 1 = t \cdot t! + (t + 1) \leq t \cdot t! + t! = (t + 1)!$$

Claim: An $m(t) \times (2^{m(t)} - 1)$ tableau cannot be t -conform.

Before proving this claim we first show how it implies the Theorem. By Proposition 12 and the Claim we have $\chi_{cf}^l(n) > t$ for all $t \geq 5$ and some $n \leq 2^{(t+1)!+1}$, since $2^{(t+1)!+1}$ is an upper bound on the number of vertices in $S_{m(t)}^\downarrow$. This implies $\log \log n \in O(t \log t)$ and finally $t \in \Omega\left(\frac{\log \log n}{\log \log \log n}\right)$.

The proof of the Claim is by induction on t . The induction base is for $t = 1$. We show it by contradiction. Any 1-conform 3×7 tableau requires to set $U_{i,k} = \{1\}$ for all $k \in [7]$ and

all $i \in [d_3(k)]$. However, looking at the LR-quarter-block property for the situation $1 \in U_{1,4}$ yields a contradiction with condition (b).

The induction step is also proved by contradiction. Assume that there are no $(t - 1)$ -conform $m' \times N'$ tableaux with $m' = m(t - 1)$ and $N' = 2^{m'} - 1$, but there is a t -conform $m \times N$ tableau \mathcal{M} for $m = m(t)$ and $N = 2^m - 1$.

The following reasoning is a bit involved, so we first give an overview.

Outline: The proof by contradiction consists of s stages, for some $1 \leq s \leq t$. The precondition of stage s is the existence of a t -conform $m \times N_{s-1}$ tableau where $N_{s-1} = 2^{m-(s-1)(m'+1)} - 1$ such that the following additional property holds. There is a color set $C_{s-1} \subseteq [t]$ consisting of $s - 1$ colors, such that for all these $c \in C_{s-1}$ and for all columns $k \in [N_s]$ holds $c \notin U_{1,k}$. The precondition for the first stage is given by the tableau \mathcal{M} with $N_0 = N$ and $C_0 = \emptyset$. The tableau \mathcal{M} will change after every stage. The postcondition of the s -th stage is either a contradiction obtained by constructing a $(t - 1)$ -conform $m' \times N'$ tableau (this is Case 1: the stop condition) or the validation of the precondition for the next stage (this is Case 2). This will work in such a way that if the stop condition does not occur even after the t -th stage, then the derived condition also yields a contradiction. We would then have $C_t = [t]$ and $N_t = 2^1 - 1 = 1$, i.e., it results in a t -conform $m \times 1$ tableau (i.e., a single column) such that no color can be unique in $M_{1,1}$.

Proof details: Suppose that an $m \times N_{s-1}$ tableau \mathcal{M} with a color set C_{s-1} fulfills the precondition for stage s with $1 \leq s \leq t$. Let $k = \frac{N_{s-1}+1}{2}$ be the central column of \mathcal{M} and $c_s \in U_{1,k}$. Note that the precondition implies $c_s \notin C_{s-1}$. By the LR-quarter-block property of t -conform tableaux there is some $XY \in \{LL, LR, RL, RR\}$ such that predicate $Q(j)$ is true for all $j \in B_{XY}(k)$. We subdivide the block $B_{XY}(k)$ into $K = 2^{m'-1}$ subblocks of equal width. These subblocks are defined by their central columns j_l , where $l \in [K]$. Note that their width just fits to the precondition of the next stage because $B_{XY}(k)$ has width $\frac{N_{s-1}+1}{4} - 1$ and, consequently, all $B(j_l)$ have width:

$$\frac{N_{s-1} + 1}{4 \cdot 2^{m'-1}} - 1 = \frac{2^{m-(s-1)(m'+1)}}{4 \cdot 2^{m'-1}} - 1 = \frac{2^{m-(s-1)(m'+1)}}{2^{m'+1}} - 1 = 2^{m-s(m'+1)} - 1$$

Due to the conditions encoded in predicate $Q(j)$ for a given color $c = c_s$ and column k we make the following case distinction:

Case 1: $\forall l \in [K] \exists j' \in B(j_l) c_s \in U_{1,j'}$

Case 2: $\exists l \in [K] \forall j' \in B(j_l) c_s \notin U_{1,j'}$

In Case 1 we can immediately derive a contradiction using the constructions of Proposition 13: First we horizontally truncate the current tableau \mathcal{M} to the block $B_{XY}(k)$, then we use a selective truncation with $m^* = m'$, where the even columns (indexed by $2l$ for $l \in [K]$) of the new tableau are the ones that separate in \mathcal{M} the subblocks B_{j_l} and $B_{j_{l+1}}$ from each other and the odd columns (indexed $2l - 1$) are chosen from B_{j_l} with respect to fulfilling the property $c_s \in U_{1,j'}$. We show that c_s is also unique in the top set of an even column. Supposing that $c = c_s$ is not unique in that set contradicts condition (c) of predicate $Q(j)$. Thus c_s is unique everywhere in the first row of the new tableau. With respect to condition (b) it does not occur at all in the third row or deeper. Each column of this new tableau \mathcal{M}' has depth $d \geq 3$ because all columns of \mathcal{M}' have been selected from a quarter subblock $B_{XY}(k)$. Next we apply a vertical truncation (deletion of top rows) to \mathcal{M}' to obtain an $m' \times N'$ tableau \mathcal{M}^* . This way at least the two top rows of \mathcal{M}' are deleted and thus color c_s does not occur anymore in \mathcal{M}^* . As a result \mathcal{M}^* is a $(t - 1)$ -conform $m' \times N'$ tableau.

Case 2 is the easier one because horizontally truncating \mathcal{M} to a block $B(j_l)$ such that $\forall_{j' \in B(j_l)} c_s \notin U_{1,j'}$ yields the precondition for the next stage with $C_s = C_{s-1} \cup \{c_s\}$. ◀

5 Conclusion

We have shown tight bounds for the chromatic AGP for orthogonal simple polygons if based on r-visibility. While the upper bound proofs use known techniques, we consider our lower bound techniques to be the main technical contribution of the paper. The multicolor tableau technique used for l-visibility can also directly be applied to the r-visibility version of the problem, but does not result in a tight bound, see [8]. Our lower bound technique for r-visibility, however, does not easily generalize to the l-visibility version of the problem, as it relies on the bijection with monotone paths in the spine tree, which does not exist in that case. It would therefore be of interest to combine both techniques to obtain a stronger $\Omega(\log \log n)$ lower bound for $\chi_{cf}^l(n)$ as well. We conjecture that this is indeed the lower bound for stretched spike polygons. But one cannot hope for more, since $O(\log \log n)$ is also an upper bound for the conflict-free guarding of stretched spike polygons using line visibility. To improve this lower bound one has to look for other polygons.

References

- 1 A. Bärtschi and S. Suri. Conflict-free Chromatic Art Gallery Coverage. *Algorithmica* 68(1): 265–283, 2014.
- 2 A. Bärtschi, S.K. Ghosh, M. Mihalak, T. Tschager, and P. Widmayer. Improved bounds for the conflict-free chromatic art gallery problem. In *Proc. of 30th Symposium on Computational Geometry*, pages 144–153, 2014.
- 3 V. Chvátal. A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory, Series B*, 18(1):39–41, 1975.
- 4 L. H. Erickson und S. M. LaValle. An Art Gallery Approach to Ensuring that Landmarks are Distinguishable, in *Proc. Robotics: Science and Systems VII*, Los Angeles, pages 81–88, 2011.
- 5 S. Fisk. A short proof of Chvátal’s Watchman Theorem. *Journal of Combinatorial Theory, Series B*, 24(3):374–374, 1978.
- 6 S. P. Fekete, S. Friedrichs, and M. Hemmer. Complexity of the General Chromatic Art Gallery Problem. *arXiv 1403.2972 [cs.CG]*, 2014.
- 7 F. Hoffmann, On the Rectilinear Art Gallery Problem, *Proc. 17th ICALP*, Springer LNCS 443, 717–728, 1990.
- 8 F. Hoffmann, K. Kriegel, and M. Willert. Almost Tight Bounds for Conflict-Free Guarding of Orthogonal Art Galleries. *arXiv:1412.3984 [cs.CG]*, 2014.
- 9 R. Motwani, A. Raghunathan, and H. Saran. Covering orthogonal polygons with star polygons: The perfect graph approach. *Comput. Syst. Sci.* 40 (1990) 19–48.
- 10 J. O’Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, NY, 1987.
- 11 J. Pach and G. Tardos. Coloring axis-parallel rectangles. *J. Comb. Theory Ser. A*, 117(6):776–782, Aug 2010
- 12 T. Shermer. Recent results in art galleries (geometry). *Proceedings of the IEEE*, 80(9): 1383–1399, September 1992
- 13 D. D. Sleator and R. E. Tarjan. A data structure for dynamic trees. *Journal of Computer and System Sciences*, 26(3):362–391, 1983.

- 14 S. Smorodinski. Conflict-Free Coloring and its Applications. In *Geometry - Intuitive, Discrete, and Convex*, volume 24 of Bolyai Society Mathematical Studies, Springer Verlag, Berlin 2014
- 15 J. Urrutia. Art gallery and illumination problems, in J.-R. Sack and J. Urrutia, editors, *Handbook on Computational Geometry*, pages 973–1026, Elsevier Sc. Publishers, 2000
- 16 M. Willert. Schranken für eine orthogonale Variante des chromatischen Art Gallery Problems, Bachelor Thesis, FU Berlin, November 2014.
- 17 C. Worman and M. Keil. Polygon Decomposition and the Orthogonal Art Gallery Problem, *Int. J. Comput. Geometry Appl.* 17(2), 105–138, 2007

Low-Quality Dimension Reduction and High-Dimensional Approximate Nearest Neighbor*

Evangelos Anagnostopoulos¹, Ioannis Z. Emiris², and Ioannis Psarros¹

1 Department of Mathematics, University of Athens, Athens, Greece
aneva@math.uoa.gr, i.psarros@di.uoa.gr

2 Department of Informatics & Telecommunications, University of Athens, Athens, Greece
emiris@di.uoa.gr

Abstract

The approximate nearest neighbor problem (ϵ -ANN) in Euclidean settings is a fundamental question, which has been addressed by two main approaches: Data-dependent space partitioning techniques perform well when the dimension is relatively low, but are affected by the curse of dimensionality. On the other hand, locality sensitive hashing has polynomial dependence in the dimension, sublinear query time with an exponent inversely proportional to $(1 + \epsilon)^2$, and subquadratic space requirement.

We generalize the Johnson-Lindenstrauss Lemma to define “low-quality” mappings to a Euclidean space of significantly lower dimension, such that they satisfy a requirement weaker than approximately preserving all distances or even preserving the nearest neighbor. This mapping guarantees, with high probability, that an approximate nearest neighbor lies among the k approximate nearest neighbors in the projected space. These can be efficiently retrieved while using only linear storage by a data structure, such as BBD-trees. Our overall algorithm, given n points in dimension d , achieves space usage in $O(dn)$, preprocessing time in $O(dn \log n)$, and query time in $O(dn^\rho \log n)$, where ρ is proportional to $1 - 1/\log \log n$, for fixed $\epsilon \in (0, 1)$. The dimension reduction is larger if one assumes that pointsets possess some structure, namely bounded expansion rate. We implement our method and present experimental results in up to 500 dimensions and 10^6 points, which show that the practical performance is better than predicted by the theoretical analysis. In addition, we compare our approach with E2LSH.

1998 ACM Subject Classification F.2.2 [Analysis of algorithms and problem complexity] Geometrical problems and computations

Keywords and phrases Approximate nearest neighbor, Randomized embeddings, Curse of dimensionality, Johnson-Lindenstrauss Lemma, Bounded expansion rate, Experimental study

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.436

1 Introduction

Nearest neighbor searching is a fundamental computational problem. Let X be a set of n points in \mathbb{R}^d and let $d(p, p')$ be the (Euclidean) distance between any two points p and p' . The

* This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program “Education and Lifelong Learning” of the National Strategic Reference Framework (NSRF) – Research Funding Program: THALIS-UOA (MIS 375891). This work was done in part while Ioannis Z. Emiris was visiting the Simons Institute for the Theory of Computing, at UC Berkeley.

problem consists in reporting, given a query point q , a point $p \in X$ such that $d(p, q) \leq d(p', q)$, for all $p' \in X$ and p is said to be a “nearest neighbor” of q . For this purpose, we preprocess X into a structure called NN-structure. However, an exact solution to high-dimensional nearest neighbor search, in sublinear time, requires prohibitively heavy resources. Thus, many techniques focus on the less demanding task of computing the approximate nearest neighbor (ϵ -ANN). Given a parameter $\epsilon \in (0, 1)$, a $(1 + \epsilon)$ -approximate nearest neighbor to a query q is a point p in X such that $d(q, p) \leq (1 + \epsilon) \cdot d(q, p')$, $\forall p' \in X$. Hence, under approximation, the answer can be any point whose distance from q is at most $(1 + \epsilon)$ times larger than the distance between q and its nearest neighbor.

Our contribution

Tree-based space partitioning techniques perform well when the dimension is relatively low, but are affected by the curse of dimensionality. To address this issue, randomized methods such as Locality Sensitive Hashing are more efficient when the dimension is high. One may also apply the Johnson-Lindenstrauss Lemma followed by standard space partitioning techniques, but the properties guaranteed are stronger than what is required for efficient approximate nearest neighbor search (cf. 2).

We introduce a “low-quality” mapping to a Euclidean space of dimension $O(\log \frac{n}{k}/\epsilon^2)$, such that an approximate nearest neighbor lies among the k approximate nearest neighbors in the projected space. This leads to our main Theorem 10, which offers a new randomized algorithm for approximate nearest neighbor search with the following complexity: Given n points in \mathbb{R}^d , the data structure, which is based on Balanced Box-Decomposition (BBD) trees, requires $O(dn)$ space, and reports an $(1 + \epsilon)^2$ -approximate nearest neighbor in time $O(dn^\rho \log n)$, where function $\rho < 1$ is proportional to $1 - 1/\ln \ln n$ for fixed $\epsilon \in (0, 1)$ and shall be specified in Section 4. The total preprocessing time is $O(dn \log n)$. For each query $q \in \mathbb{R}^d$, the preprocessing phase succeeds with probability $> 1 - \delta$ for any constant $\delta \in (0, 1)$. The low-quality embedding is extended to pointsets with bounded expansion rate c (see Section 5 for definitions). The pointset is now mapped to a Euclidean space of dimension roughly $O(\log c/\epsilon^2)$, for large enough k .

We also present experiments, based on synthetic datasets that validate our approach and our analysis. One set of inputs, along with the queries, follow the “planted nearest neighbor model” which will be specified in Section 6. In another scenario, we assume that the near neighbors of each query point follow the Gaussian distribution. Apart from showing that the embedding has the desired properties in practice, we also implement our overall approach for computing ϵ -ANN using the ANN library and we compare with a LSH implementation, namely E2LSH.

The notation of key quantities is the same throughout the paper.

The paper extends and improves ideas from [25].

Paper organization

The next section offers a survey of existing techniques. Section 3 introduces our embeddings to dimension lower than predicted by the Johnson-Lindenstrauss Lemma. Section 4 states our main results about ϵ -ANN search. Section 5 generalizes our discussion so as to exploit bounded expansion rate, and Section 6 presents experiments to validate our approach. We conclude with open questions.

2 Existing work

As it was mentioned above, an exact solution to high-dimensional nearest neighbor search, in sublinear time, requires heavy resources. One notable solution to the problem [21] shows that nearest neighbor queries can be answered in $O(d^5 \log n)$ time, using $O(n^{d+\delta})$ space, for arbitrary $\delta > 0$.

One class of methods for ϵ -ANN may be called data-dependent, since the decisions taken for partitioning the space are affected by the given data points. In [8], they introduced the Balanced Box-Decomposition (BBD) trees. The BBD-trees data structure achieves query time $O(c \log n)$ with $c \leq d/2[1 + 6d/\epsilon]^d$, using space in $O(dn)$, and preprocessing time in $O(dn \log n)$. BBD-trees can be used to retrieve the $k \geq 1$ approximate nearest-neighbors at an extra cost of $O(d \log n)$ per neighbor. BBD-trees have proved to be very practical, as well, and have been implemented in software library ANN.

Another data structure is the Approximate Voronoi Diagrams (AVD). They are shown to establish a tradeoff between the space complexity of the data structure and the query time it supports [7]. With a tradeoff parameter $2 \leq \gamma \leq \frac{1}{\epsilon}$, the query time is $O(\log(n\gamma) + 1/(\epsilon\gamma)^{\frac{d-1}{2}})$ and the space is $O(n\gamma^{d-1} \log \frac{1}{\epsilon})$. They are implemented on a hierarchical quadtree-based subdivision of space into cells, each storing a number of representative points, such that for any query point lying in the cell, at least one of the representatives is an approximate nearest neighbor. Further improvements to the space-time trade offs for ANN, are obtained in [6].

One might directly apply the celebrated Johnson-Lindenstrauss Lemma and map the points to $O(\frac{\log n}{\epsilon^2})$ dimensions with distortion equal to $1 + \epsilon$ in order to improve space requirements. In particular, AVD combined with the Johnson-Lindenstrauss Lemma require $n^{O(\log \frac{1}{\epsilon}/\epsilon^2)}$ space which is prohibitive if $\epsilon \ll 1$ and query time polynomial in $\log n$, d and $1/\epsilon$. Notice that we relate the approximation error with the distortion for simplicity. Our approach (Theorem 10) requires $O(dn)$ space and has query time sublinear in n and polynomial in d .

In high dimensional spaces, data dependent data structures are affected by the curse of dimensionality. This means that, when the dimension increases, either the query time or the required space increases exponentially. An important method conceived for high dimensional data is locality sensitive hashing (LSH). LSH induces a data independent space partition and is dynamic, since it supports insertions and deletions. It relies on the existence of locality sensitive hash functions, which are more likely to map similar objects to the same bucket. The existence of such functions depends on the metric space. In general, LSH requires roughly $O(dn^{1+\rho})$ space and $O(dn^\rho)$ query time for some parameter $\rho \in (0, 1)$. In [4] they show that in the Euclidean case, one can have $\rho = \frac{1}{(1+\epsilon)^2}$ which matches the lower bound of hashing algorithms proved in [23]. Lately, it was shown that it is possible to overcome this limitation with an appropriate change in the scheme which achieves $\rho = \frac{1}{2(1+\epsilon)^2-1} + o(1)$ [5]. For comparison, in Theorem 10 we show that it is possible to use $O(dn)$ space, with query time roughly $O(dn^\rho)$ where $\rho < 1$ is now higher than the one appearing in LSH. One different approach [24] achieves near linear space but query time proportional to $O(dn^{\frac{2}{1+\epsilon}})$.

Exploiting the structure of the input is an important way to improve the complexity of nearest neighbor search. In particular, significant amount of work has been done for pointsets with low doubling dimension. In [14], they provide an algorithm for ANN with expected preprocessing time $O(2^{\dim(X)} n \log n)$, space $O(2^{\dim(X)} n)$ and query time $O(2^{\dim(X)} \log n + \epsilon^{-O(\dim(X))})$ for any finite metric space X of doubling dimension $\dim(X)$. In [16] they provide randomized embeddings that preserve nearest neighbor with constant probability, for points lying on low doubling dimension manifolds in Euclidean settings. Naturally, such an approach can be easily combined with any known data structure for ϵ -ANN.

In [10] they present random projection trees which adapt to pointsets of low doubling dimension. Like kd-trees, every split partitions the pointset into subsets of roughly equal cardinality; in fact, instead of splitting at the median, they add a small amount of “jitter”. Unlike kd-trees, the space is split with respect to a random direction, not necessarily parallel to the coordinate axes. Classic *kd*-trees also adapt to the doubling dimension of randomly rotated data [26]. However, for both techniques, no related theoretical arguments about the efficiency of ϵ -ANN search were given.

In [19], they introduce a different notion of intrinsic dimension for an arbitrary metric space, namely its expansion rate c ; it is formally defined in Section 5. The doubling dimension is a more general notion of intrinsic dimension in the sense that, when a finite metric space has bounded expansion rate, then it also has bounded doubling dimension, but the converse does not hold [13]. Several efficient solutions are known for metrics with bounded expansion rate, including for the problem of exact nearest neighbor. In [20], they present a data structure which requires $c^{O(1)}n$ space and answers queries in $c^{O(1)} \ln n$. Cover Trees [9] require $O(n)$ space and each query costs $O(c^{12} \log n)$ time for exact nearest neighbors. In Theorem 13, we provide a data structure for the ϵ -ANN problem with linear space and $O((C^{1/\epsilon^3} + \log n)d \log n / \epsilon^2)$ query time, where C depends on c . The result concerns pointsets in the d -dimensional Euclidean space.

3 Low Quality Randomized Embeddings

This section examines standard dimensionality reduction techniques and extends them to approximate embeddings optimized to our setting. In the following, we denote by $\|\cdot\|$ the Euclidean norm and by $|\cdot|$ the cardinality of a set.

Let us start with the classic Johnson-Lindenstrauss Lemma:

► **Proposition 1.** [18] *For any set $X \subset \mathbb{R}^d$, $\epsilon \in (0, 1)$ there exists a distribution over linear mappings $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$, where $d' = O(\log |X| / \epsilon^2)$, such that for any $p, q \in X$,*

$$(1 - \epsilon)\|p - q\|^2 \leq \|f(p) - f(q)\|^2 \leq (1 + \epsilon)\|p - q\|^2.$$

In the initial proof [18], they show that this can be achieved by orthogonally projecting the pointset on a random linear subspace of dimension d' . In [11], they provide a proof based on elementary probabilistic techniques. In [15], they prove that it suffices to apply a gaussian matrix G on the pointset. G is a $d \times d'$ matrix with each of its entries independent random variables given by the standard normal distribution $N(0, 1)$. Instead of a gaussian matrix, we can apply a matrix whose entries are independent random variables with uniformly distributed values in $\{-1, 1\}$ [2].

However, it has been realized that this notion of randomized embedding is somewhat stronger than what is required for approximate nearest neighbor searching. The following definition has been introduced in [16] and focuses only on the distortion of the nearest neighbor.

► **Definition 2.** Let (Y, d_Y) , (Z, d_Z) be metric spaces and $X \subseteq Y$. A distribution over mappings $f : Y \rightarrow Z$ is a *nearest-neighbor preserving embedding* with distortion $D \geq 1$ and probability of correctness $P \in [0, 1]$ if, $\forall \epsilon \geq 0$ and $\forall q \in Y$, with probability at least P , when $x \in X$ is such that $f(x)$ is a ϵ -ANN of $f(q)$ in $f(X)$, then x is a $(D \cdot (1 + \epsilon))$ -approximate nearest neighbor of q in X .

While in the ANN problem we search one point which is approximately nearest, in the k approximate nearest neighbors problem (ϵ - k ANNs) we seek an approximation of the k

nearest points, in the following sense. Let X be a set of n points in \mathbb{R}^d , let $q \in \mathbb{R}^d$ and $1 \leq k \leq n$. The problem consists in reporting a sequence $S = \{p_1, \dots, p_k\}$ of k distinct points such that the i -th point is an $(1 + \epsilon)$ -approximation to the i -th nearest neighbor of q . Furthermore, the following assumption is satisfied by the search routine of tree-based data structures such as BBD-trees.

► **Assumption 3.** Let $S' \subseteq X$ be the set of points visited by the ϵ - k ANNs search such that $S = \{p_1, \dots, p_k\} \subseteq S'$ is the (ordered w.r.t. distance from q) set of points which are the k nearest to the query point q among the points in S' . We assume that $\forall x \in X \setminus S'$, $d(x, q) > d(p_k, q)/(1 + \epsilon)$.

Assuming the existence of a data structure which solves ϵ - k ANNs, we can weaken Definition 2 as follows.

► **Definition 4.** Let (Y, d_Y) , (Z, d_Z) be metric spaces and $X \subseteq Y$. A distribution over mappings $f : Y \rightarrow Z$ is a *locality preserving embedding* with distortion $D \geq 1$, probability of correctness $P \in [0, 1]$ and locality parameter k , if $\forall \epsilon \geq 0$ and $\forall q \in Y$, with probability at least P , when $S = \{f(p_1), \dots, f(p_k)\}$ is a solution to ϵ - k ANNs for q , under Assumption 3 then there exists $f(x) \in S$ such that x is a $(D \cdot (1 + \epsilon))$ -approximate nearest neighbor of q in X .

According to this definition we can reduce the problem of ϵ -ANN in dimension d to the problem of computing k approximate nearest neighbors in dimension $d' < d$.

We use the Johnson-Lindenstrauss dimensionality reduction technique and more specifically the proof obtained in [11]. As it was previously discussed, there also exist alternative proofs which correspond to alternative randomized mappings.

► **Lemma 5.** [11] *There exists a distribution over linear maps $A : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ s.t., for any $p \in \mathbb{R}^d$ with $\|p\| = 1$:*

- if $\beta^2 < 1$ then $\Pr[\|Ap\|^2 \leq \beta^2 \cdot \frac{d'}{d}] \leq \exp(\frac{d'}{2}(1 - \beta^2 + 2 \ln \beta))$,
- if $\beta^2 > 1$ then $\Pr[\|Ap\|^2 \geq \beta^2 \cdot \frac{d'}{d}] \leq \exp(\frac{d'}{2}(1 - \beta^2 + 2 \ln \beta))$.

We prove the following lemma which will be useful.

► **Lemma 6.** *For all $i \in \mathbb{N}$, $\epsilon \in (0, 1)$, the following holds:*

$$\frac{(1 + \epsilon/2)^2}{(2^i(1 + \epsilon))^2} - 2 \ln \frac{(1 + \epsilon/2)}{2^i(1 + \epsilon)} - 1 > 0.05(i + 1)\epsilon^2.$$

Proof. For $i = 0$, it can be checked that if $\epsilon \in (0, 1)$ then, $\frac{(1 + \epsilon/2)^2}{(1 + \epsilon)^2} - 2 \ln \frac{1 + \epsilon/2}{1 + \epsilon} - 1 > 0.05\epsilon^2$. This is our induction basis. Let $j \geq 0$ be such that the induction hypothesis holds. Then, to prove the induction step

$$\begin{aligned} \frac{1}{4} \frac{(1 + \epsilon/2)^2}{(2^j(1 + \epsilon))^2} - 2 \ln \frac{(1 + \epsilon/2)}{2^j(1 + \epsilon)} + 2 \ln 2 - 1 &> 0.05(j + 1)\epsilon^2 - \frac{3}{4} \frac{(1 + \epsilon/2)^2}{(2^j(1 + \epsilon))^2} + 2 \ln 2 > \\ &> 0.05(j + 1)\epsilon^2 - \frac{3}{4} + 2 \ln 2 > 0.05(j + 2)\epsilon^2, \end{aligned}$$

since $\epsilon \in (0, 1)$. ◀

A simple calculation shows the following.

► **Lemma 7.** For all $x > 0$, it holds:

$$\frac{(1+x)^2}{(1+2x)^2} - 2 \ln\left(\frac{1+x}{1+2x}\right) - 1 < (1+x)^2 - 2 \ln(1+x) - 1. \tag{1}$$

► **Theorem 8.** Under the notation of Definition 4, there exists a randomized mapping $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ which satisfies Definition 4 for $d' = O(\log \frac{2n}{\delta k} / \epsilon^2)$, $\epsilon > 0$, distortion $D = 1 + \epsilon$ and probability of success $1 - \delta$, for any constant $\delta \in (0, 1)$.

Proof. Let X be a set of n points in \mathbb{R}^d and consider map

$$f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'} : v \mapsto \sqrt{d/d'} \cdot A v,$$

where A is a matrix chosen from a distribution as in Lemma 5. Wlog the query point q lies at the origin and its nearest neighbor u lies at distance 1 from q . We denote by $c \geq 1$ the approximation ratio guaranteed by the assumed data structure. That is, the assumed data structure solves the $(c - 1)$ - k ANNs problem. For each point x , $L_x = \|Ax\|^2 / \|x\|^2$. Let N be the random variable whose value indicates the number of “bad” candidates, that is

$$N = |\{x \in X : \|x - q\| > \gamma \wedge L_x \leq \frac{\beta^2}{\gamma^2} \cdot \frac{d'}{d}\}|,$$

where we define $\beta = c(1 + \epsilon/2)$, $\gamma = c(1 + \epsilon)$. Hence, by Lemma 5,

$$\mathbb{E}[N] \leq n \cdot \exp\left(\frac{d'}{2} \left(1 - \frac{\beta^2}{\gamma^2} + 2 \ln \frac{\beta}{\gamma}\right)\right).$$

By Markov’s inequality,

$$\Pr[N \geq k] \leq \frac{\mathbb{E}[N]}{k} \implies \Pr[N \geq k] \leq n \cdot \exp\left(\frac{d'}{2} \left(1 - \frac{\beta^2}{\gamma^2} + 2 \ln \frac{\beta}{\gamma}\right)\right) / k.$$

The event of failure is defined as the disjunction of two events:

$$[N \geq k] \vee [L_u \geq (\beta/c)^2 \frac{d'}{d}], \tag{2}$$

and its probability is at most equal to

$$\Pr[N \geq k] + \exp\left(\frac{d'}{2} (1 - (\beta/c)^2 + 2 \ln(\beta/c))\right),$$

by applying again Lemma 5. Now, we bound these two terms. For the first one, we choose d' such that

$$d' \geq 2 \frac{\ln \frac{2n}{\delta k}}{\frac{\beta^2}{\gamma^2} - 1 - 2 \ln \frac{\beta}{\gamma}}. \tag{3}$$

Therefore,

$$\frac{\exp\left(\frac{d'}{2} \left(1 - \frac{\beta^2}{\gamma^2} + 2 \ln \frac{\beta}{\gamma}\right)\right)}{k} \leq \frac{\delta}{2n} \implies \Pr[N \geq k] \leq \frac{\delta}{2}. \tag{4}$$

Notice that $k \leq n$ and due to expression (1), we obtain $(\beta/\gamma)^2 - 2 \ln(\beta/\gamma) - 1 < (\beta/c)^2 - 2 \ln(\beta/c) - 1$. Hence, inequality (3) implies the following inequality:

$$d' \geq 2 \frac{\ln \frac{2}{\delta}}{(\beta/c)^2 - 1 - 2 \ln(\beta/c)}.$$

Therefore, the second term in expression (2) is bounded as follows:

$$\exp\left(\frac{d'}{2}\left(1 - \left(\frac{\beta}{c}\right)^2 + 2\ln\frac{\beta}{c}\right)\right) \leq \frac{\delta}{2}. \quad (5)$$

Inequalities (4), (5) imply that the total probability of failure in expression (2) is at most δ .

Using Lemma 6 for $i = 0$, we obtain, that there exists d' such that

$$d' = O\left(\log\frac{n}{\delta k}/\epsilon^2\right)$$

and with probability at least $1 - \delta$, these two events occur:

- $\|f(q) - f(u)\| \leq (1 + \frac{\epsilon}{2})\|u - q\|$.
- $|\{p \in X \mid \|p - q\| > c(1 + \epsilon)\|u - q\|\} \implies \|f(q) - f(p)\| \leq c(1 + \epsilon/2)\|u - q\}| < k$.

Now consider the case when the random experiment succeeds and let $S = \{f(p_1), \dots, f(p_k)\}$ a solution of the $(c-1)$ - k ANNs problem in the projected space, given by a data-structure which satisfies Assumption 3. We have that $\forall f(x) \in f(X) \setminus S'$, $\|f(x) - f(q)\| > \|f(p_k) - f(q)\|/c$ where S' is the set of all points visited by the search routine.

Now, if $f(u) \in S$ then S contains the projection of the nearest neighbor. If $f(u) \notin S$ then if $f(u) \notin S'$ we have the following:

$$\|f(u) - f(q)\| > \|f(p_k) - f(q)\|/c \implies \|f(p_k) - f(q)\| < c(1 + \epsilon/2)\|u - q\|,$$

which means that there exists at least one point $f(p^*) \in S$ s.t. $\|q - p^*\| \leq c(1 + \epsilon)\|u - q\|$. Finally, if $f(u) \notin S$ but $f(u) \in S'$ then

$$\|f(p_k) - f(q)\| \leq \|f(u) - f(q)\| \implies \|f(p_k) - f(q)\| \leq (1 + \epsilon/2)\|u - q\|,$$

which means that there exists at least one point $f(p^*) \in S$ s.t. $\|q - p^*\| \leq c(1 + \epsilon)\|u - q\|$.

Hence, f satisfies Definition 4 for $D = 1 + \epsilon$. \blacktriangleleft

4 Approximate Nearest Neighbor Search

This section combines tree-based data structures which solve ϵ - k ANNs with the results above, in order to obtain an efficient randomized data structure which solves ϵ -ANN.

BBD-trees [8] require $O(dn)$ space, and allow computing k points, which are $(1 + \epsilon)$ -approximate nearest neighbors, within time $O((\lceil 1 + 6\frac{d}{\epsilon} \rceil^d + k)d \log n)$. The preprocessing time is $O(dn \log n)$. Notice, that BBD-trees satisfy the Assumption 3. The algorithm for the ϵ - k ANNs search, visits cells in increasing order with respect to their distance from the query point q . If the current cell lies at distance more than r_k/c where r_k is the current distance to the k th nearest neighbor, the search terminates. We apply the random projection for distortion $D = 1 + \epsilon$, thus relating approximation error to the allowed distortion; this is not required but simplifies the analysis.

Moreover, $k = n^\rho$; the formula for $\rho < 1$ is determined below. Our analysis then focuses on the asymptotic behaviour of the term $O(\lceil 1 + 6\frac{d}{\epsilon} \rceil^d + k)$.

► **Lemma 9.** *With the above notation, there exists $k > 0$ s.t., for fixed $\epsilon \in (0, 1)$, it holds that $\lceil 1 + 6\frac{d}{\epsilon} \rceil^d + k = O(n^\rho)$, where $\rho \leq 1 - \epsilon^2/\hat{c}(\epsilon^2 + \log(\max\{\frac{1}{\epsilon}, \log n\})) < 1$ for some appropriate constant $\hat{c} > 1$.*

Proof. Recall that $d' \leq \frac{\tilde{c}}{\epsilon^2} \ln \frac{n}{k}$ for some appropriate constant $\tilde{c} > 0$. The constant δ is hidden in \tilde{c} . Since $(\frac{d'}{\epsilon})^{d'}$ is a decreasing function of k , we need to choose k s.t. $(\frac{d'}{\epsilon})^{d'} = \Theta(k)$.

Let $k = n^\rho$. Obviously $\lceil 1 + 6\frac{d'}{\epsilon} \rceil^{d'} \leq (c'\frac{d'}{\epsilon})^{d'}$, for some appropriate constant $c' \in (1, 7)$. Then, by substituting d', k we have:

$$(c'\frac{d'}{\epsilon})^{d'} = n^{\frac{\tilde{c}(1-\rho)}{\epsilon^2} \ln(\frac{\tilde{c}c'(1-\rho)\ln n}{\epsilon^3})}. \tag{6}$$

We assume $\epsilon \in (0, 1)$ is a fixed constant. Hence, it is reasonable to assume that $\frac{1}{\epsilon} < n$. We consider two cases when comparing $\ln n$ to ϵ :

- $\frac{1}{\epsilon} \leq \ln n$. Substituting $\rho = 1 - \frac{\epsilon^2}{2\tilde{c}(\epsilon^2 + \ln(c'\ln n))}$ into equation (6), the exponent of n is bounded as follows:

$$\begin{aligned} & \frac{\tilde{c}(1-\rho)}{\epsilon^2} \ln(\frac{\tilde{c}c'(1-\rho)\ln n}{\epsilon^3}) = \\ & = \frac{\tilde{c}}{2\tilde{c}(\epsilon^2 + \ln(c'\ln n))} \cdot [\ln(c'\ln n) + \ln \frac{1}{\epsilon} - \ln(2\epsilon^2 + 2\ln(c'\ln n))] < \rho. \end{aligned}$$

- $\frac{1}{\epsilon} > \ln n$. Substituting $\rho = 1 - \frac{\epsilon^2}{2\tilde{c}(\epsilon^2 + \ln \frac{c'}{\epsilon})}$ into equation (6), the exponent of n is bounded as follows:

$$\begin{aligned} & \frac{\tilde{c}(1-\rho)}{\epsilon^2} \ln(\frac{\tilde{c}c'(1-\rho)\ln n}{\epsilon^3}) = \\ & = \frac{\tilde{c}}{2\tilde{c}(\epsilon^2 + \ln \frac{c'}{\epsilon})} \cdot [\ln \ln n + \ln \frac{c'}{\epsilon} - \ln(2\epsilon^2 + 2\ln \frac{c'}{\epsilon})] < \rho. \end{aligned}$$



Notice that for both cases $d' = O(\frac{\log n}{\epsilon^2 + \log \log n})$.

Combining Theorem 8 with Lemma 9 yields the following main theorem.

► **Theorem 10 (Main).** *Given n points in \mathbb{R}^d , there exists a randomized data structure which requires $O(dn)$ space and reports an $(1 + \epsilon)^2$ -approximate nearest neighbor in time*

$$O(dn^\rho \log n), \text{ where } \rho \leq 1 - \epsilon^2 / \hat{c}(\epsilon^2 + \log(\max\{\frac{1}{\epsilon}, \log n\}))$$

for some appropriate constant $\hat{c} > 1$. The preprocessing time is $O(dn \log n)$. For each query $q \in \mathbb{R}^d$, the preprocessing phase succeeds with any constant probability.

Proof. The space required to store the dataset is $O(dn)$. The space used by BBD-trees is $O(d'n)$ where d' is defined in Lemma 9. We also need $O(dd')$ space for the matrix A as specified in Theorem 8. Hence, since $d' < d$ and $d' < n$, the total space usage is bounded above by $O(dn)$.

The preprocessing consists of building the BBD-tree which costs $O(d'n \log n)$ time and sampling A . Notice that we can sample a d' -dimensional random subspace in time $O(dd'^2)$ as follows. First, we sample in time $O(dd')$, a $d \times d'$ matrix where its elements are independent random variables with the standard normal distribution $N(0, 1)$. Then, we orthonormalize using Gram-Schmidt in time $O(dd'^2)$. Since $d' = O(\log n)$, the total preprocessing time is bounded by $O(dn \log n)$.

For each query we use A to project the point in time $O(dd')$. Next, we compute its n^ρ approximate nearest neighbors in time $O(d'n^\rho \log n)$ and we check its neighbors with their real coordinates in time $O(dn^\rho)$. Hence, each query costs $O(d \log n + d'n^\rho \log n + dn^\rho) = O(dn^\rho \log n)$ because $d' = O(\log n)$, $d' = O(d)$. Thus, the query time is dominated by the time required for ϵ - k ANNs search and the time to check the returned sequence of k approximate nearest neighbors.



To be more precise, the probability of success, which is the probability that the random projection succeeds according to Theorem 8, is greater than $1 - \delta$, for any constant $\delta \in (0, 1)$. Notice that the preprocessing time for BBD-trees has no dependence on ϵ .

5 Bounded Expansion Rate

This section models the structure that the data points may have so as to obtain more precise bounds.

The bound on the dimension obtained in Theorem 8 is quite pessimistic. We expect that, in practice, the space dimension needed in order to have a sufficiently good projection is less than what Theorem 8 guarantees. Intuitively, we do not expect to have instances where all points in X , which are not approximate nearest neighbors of q , lie at distance almost equal to $(1 + \epsilon)d(q, X)$. To this end, we consider the case of pointsets with bounded expansion rate.

► **Definition 11.** Let M a metric space and $X \subseteq M$ a finite pointset and let $B_p(r) \subseteq X$ denote the points of X lying in the closed ball centered at p with radius r . We say that X has (ρ, c) -expansion rate if and only if, $\forall p \in M$ and $r > 0$,

$$|B_p(r)| \geq \rho \implies |B_p(2r)| \leq c \cdot |B_p(r)|.$$

► **Theorem 12.** Under the notation introduced in the previous definitions, there exists a randomized mapping $f : \mathbb{R}^d \rightarrow \mathbb{R}^{d'}$ which satisfies Definition 4 for dimension $d' = O(\frac{\log(c + \frac{\rho}{\delta k})}{\epsilon^2})$, distortion $D = 1 + \epsilon$ and probability of success $1 - \delta$, for any constant $\delta \in (0, 1)$, for pointsets with (ρ, c) -expansion rate.

Proof. We proceed in the same spirit as in the proof of Theorem 8, and using the notation from that proof. Let r_0 be the distance to the ρ -th nearest neighbor, excluding neighbors at distance $\leq 1 + \epsilon$. For $i > 0$, let $r_i = 2 \cdot r_{i-1}$ and set $r_{-1} = 1 + \epsilon$. Clearly,

$$\begin{aligned} \mathbb{E}[N] &\leq \sum_{i=0}^{\infty} |B_p(r_i)| \cdot \exp\left(\frac{d'}{2} \left(1 - \frac{(1 + \epsilon/2)^2}{r_{i-1}^2} + 2 \ln \frac{1 + \epsilon/2}{r_{i-1}}\right)\right) \\ &\leq \sum_{i=0}^{\infty} c^i \rho \cdot \exp\left(\frac{d'}{2} \left(1 - \frac{(1 + \epsilon/2)^2}{2^{2i}(1 + \epsilon)^2} + 2 \ln \frac{1 + \epsilon/2}{2^i(1 + \epsilon)}\right)\right). \end{aligned}$$

Now, using Lemma 6,

$$\mathbb{E}[N] \leq \sum_{i=0}^{\infty} c^i \rho \cdot \exp\left(-\frac{d'}{2} 0.05(i + 1)\epsilon^2\right),$$

and for $d' \geq 40 \cdot \ln(c + \frac{2\rho}{k\delta})/\epsilon^2$,

$$\mathbb{E}[N] \leq \rho \cdot \sum_{i=0}^{\infty} c^i \cdot \left(\frac{1}{c + \frac{2\rho}{k\delta}}\right)^{i+1} = \rho \cdot \sum_{i=0}^{\infty} c^i \cdot \left(\frac{1}{c}\right)^{i+1} \cdot \left(\frac{1}{1 + \frac{2\rho}{kc\delta}}\right)^{i+1} = \frac{\rho}{c} \cdot \sum_{i=0}^{\infty} \left(\frac{1}{1 + \frac{2\rho}{kc\delta}}\right)^{i+1} = \frac{k\delta}{2}.$$

Finally,

$$\Pr[N \geq k] \leq \frac{\mathbb{E}[N]}{k} \leq \frac{\delta}{2}. \quad \blacktriangleleft$$

Employing Theorem 12 we obtain a result analogous to Theorem 10 which is weaker than those in [20, 9] but underlines the fact that our scheme shall be sensitive to structure in the input data, for real world assumptions.

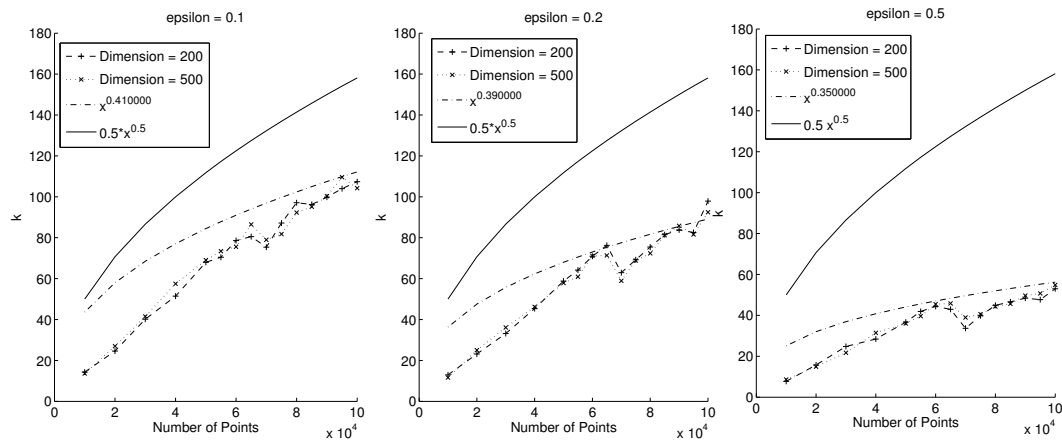


Figure 1 Plot of k as n increases for the “planted nearest neighbor model” datasets. The highest line corresponds to $\frac{\sqrt{n}}{2}$ and the dotted line to a function of the form n^ρ , where $\rho = 0.41, 0.39, 0.35$ that best fits the data.

► **Theorem 13.** Given n points in \mathbb{R}^d with $(\log n, c)$ -expansion rate, there exists a randomized data structure which requires $O(dn)$ space and reports an $(1+\epsilon)^2$ -approximate nearest neighbor in time $O((C^{1/\epsilon^3} + \log n)d \log n / \epsilon^2)$, for some constant C depending on c . The preprocessing time is $O(dn \log n)$. For each query $q \in \mathbb{R}^d$, the preprocessing phase succeeds with any constant probability.

Proof. Set $k = \log n$. Then $d' = O(\frac{\log c}{\epsilon^2})$ and $(\frac{d'}{\epsilon})^{d'} = O(c^{\frac{1}{\epsilon^2} \log[\frac{\log c}{\epsilon^3}]})$. Now the query time is

$$O((c^{\frac{1}{\epsilon^2} \log[\frac{\log c}{\epsilon^3}]} + \log n)d \frac{\log c}{\epsilon^2} \log n) = O((C^{1/\epsilon^3} + \log n)d \frac{\log n}{\epsilon^2}),$$

for some constant C such that $c^{\log(\log c / \epsilon^3) / \epsilon^2} = O(C^{1/\epsilon^3})$. ◀

6 Experiments

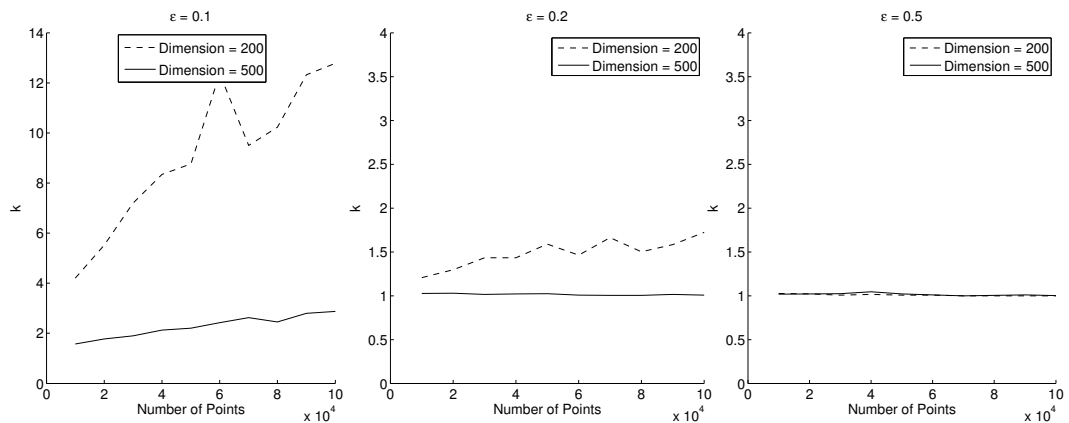
In the following two sections we present and discuss the two experiments we performed. In the first one we computed the average value of k in a worst-case dataset and we validated that it is indeed sublinear. In the second one we made an ANN query time and memory usage comparison against a LSH implementation using both artificial and real life datasets.

6.1 Validation of k

In this section we present an experimental verification of our approach. We show that the number k of the nearest neighbors in the random projection space that we need to examine in order to find an approximate nearest neighbor in the original space depends sublinearly on n . Recall that we denote by $\|\cdot\|$ the Euclidean norm.

Dataset

We generated our own synthetic datasets and query points to verify our results. We decided to follow two different procedures for data generation in order to be as complete as possible. First of all, as in [12], we followed the “planted nearest neighbor model” for our datasets. This model guarantees for each query point q the existence of a few approximate nearest



■ **Figure 2** Plot of k as n increases for the gaussian datasets. We see how increasing the number of approximate nearest neighbors in this case decreases the value of k .

neighbors while keeping all other points sufficiently far from q . The benefit of this approach is that it represents a typical ANN search scenario, where for each point there exist only a handful approximate nearest neighbors. In contrast, in a uniformly generated dataset, all the points will tend to be equidistant to each other in high dimensions, which is quite unrealistic.

In order to generate such a dataset, first we create a set Q of query points chosen uniformly at random in \mathbb{R}^d . Then, for each point $q \in Q$, we generate a single point p at distance R from q , which will be its single (approximate) nearest neighbor. Then, we create more points at distance $\geq (1 + \epsilon)R$ from q , while making sure that they shall not be closer than $(1 + \epsilon)R$ to any other query point q' . This dataset now has the property that every query point has exactly one approximate nearest neighbor, while all other points are at distance $\geq (1 + \epsilon)R$.

We fix $R = 2$, let $\epsilon \in \{0.1, 0.2, 0.5\}$, $d = \{200, 500\}$ and the total number of points $n \in \{10^4, 2 \times 10^4, \dots, 5 \times 10^4, 5.5 \times 10^4, 6 \times 10^4, 6.5 \times 10^4, \dots, 10^5\}$. For each combination of the above we created a dataset X from a set Q of 100 query points where each query coordinate was chosen uniformly at random in the range $[-20, 20]$.

The second type of datasets consisted again of sets of 100 query points in \mathbb{R}^d where each coordinate was chosen uniformly at random in the range $[-20, 20]$. Each query point was paired with a random variable σ_q^2 uniformly distributed in $[15, 25]$ and together they specified a gaussian distribution in \mathbb{R}^d of mean value $\mu = q$ and variance σ_q^2 per coordinate. For each distribution we drew n points in the same set as was previously specified.

Scenario

We performed the following experiment for the “planted nearest neighbor model”. In each dataset X , we consider, for every query point q , its unique (approximate) nearest neighbor $p \in X$. Then we use a random mapping f from \mathbb{R}^d to a Euclidean space of lower dimension $d' = \frac{\log n}{\log \log n}$ using a gaussian matrix G , where each entry $G_{ij} \sim N(0, 1)$. This matrix guarantees a low distortion embedding [15]. Then, we perform a range query centered at $f(q)$ with radius $\|f(q) - f(p)\|$ in $f(X)$: we denote by $rank_q(p)$ the number of points found. Then, exactly $rank_q(p)$ points are needed to be selected in the worst case as k -nearest neighbors of $f(q)$ in order for the approximate nearest neighbor $f(p)$ to be among them, so $k = rank_q(p)$.

For the datasets with the gaussian distributions we compute again the maximum number of points k needed to visit in the lower-dimensional space in order to find an ϵ -approximate nearest neighbor of each query point q in the original space. In this case the experiment

works as follows: we find all the ϵ -approximate nearest neighbors of a query point q . Let S_q be the set containing for each query q its ϵ - k ANNs. Next, let $p_q = \arg \min_{p \in S} \|f(p) - f(q)\|$. Now as before we perform a range query centered at $f(q)$ with radius $\|f(q) - f(p_q)\|$. We consider as k the number of points returned by this query.

Results

The “planted nearest neighbor model” datasets constitute a worst-case input for our approach since every query point has only one approximate nearest neighbor and has many points lying near the boundary of $(1 + \epsilon)$. We expect that the number of k approximate nearest neighbors needed to consider in this case will be higher than in the case of the gaussian distributions, but still expect the number to be considerably sublinear.

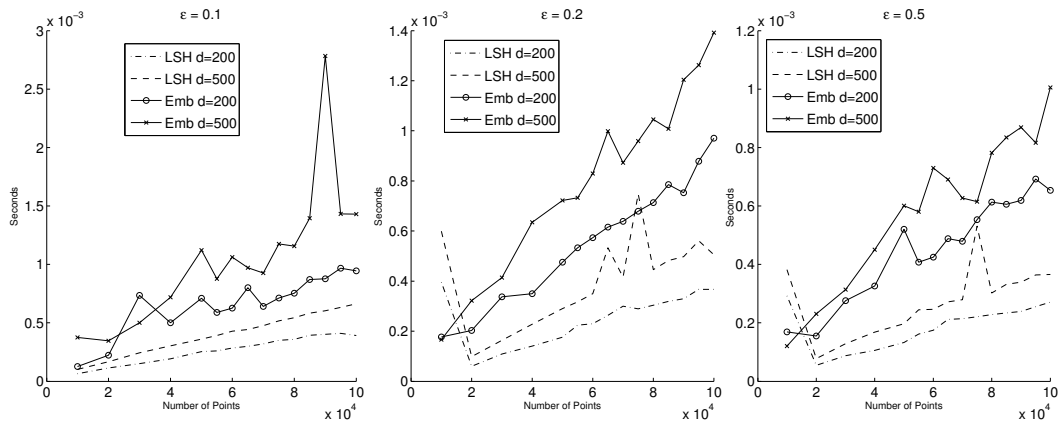
In Figure 1 we present the average value of k as we increase the number of points n for the planted nearest neighbor model. We can see that k is indeed significantly smaller than n . The line corresponding to the averages may not be smooth, which is unavoidable due to the random nature of the embedding, but it does have an intrinsic concavity, which shows that the dependency of k on n is sublinear. For comparison we also display the function $\sqrt{n}/2$, as well as a function of the form n^ρ , $\rho < 1$ which was computed by SAGE that best fits the data per plot. The fitting was performed on the points in the range $[50000, 100000]$ as to better capture the asymptotic behaviour. In Figure 2 we show again the average value of k as we increase the number of points n for the gaussian distribution datasets. As expected we see that the expected value of k is much smaller than n and also smaller than the expected value of k in the worst-case scenario, which is the planted nearest neighbor model.

6.2 ANN experiments

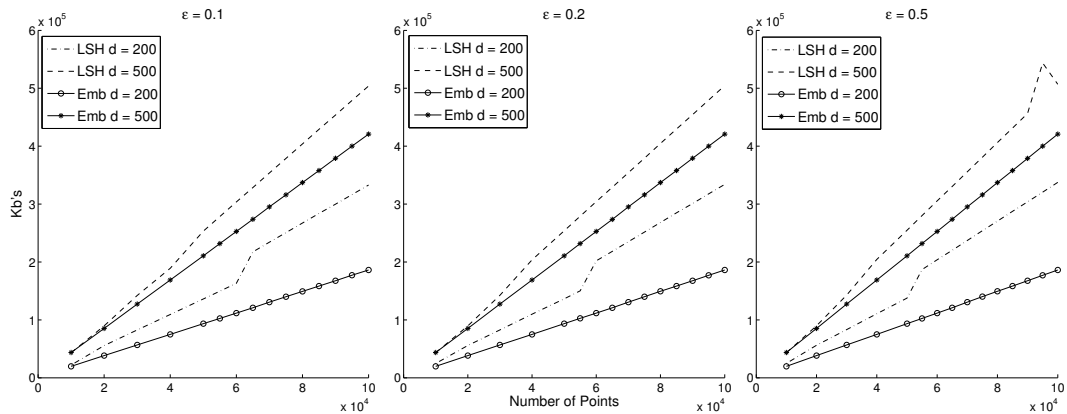
In this section we present a naive comparison between our algorithm and the E2LSH [3] implementation of the LSH framework for approximate nearest neighbor queries.

Experiment Description

We projected all the “planted nearest neighbor” datasets, down to $\frac{\log n}{\log \log n}$ dimensions. We remind the reader that these datasets were created to have a single approximate nearest neighbor for each query at distance R and all other points at distance $> (1 + \epsilon)R$. We then built a BBD-tree data structure on the projected space using the ANN library [22] with the default settings. Next, we measured the average time needed for each query q to find its ϵ - k ANNs, for $k = \sqrt{n}$, using the BBD-Tree data structure and then to select the first point at distance $\leq R$ out of the k in the original space. We compare these times to the average times reported by E2LSH range queries for $R = 2$, when used from its default script for probability of success 0.95. The script first performs an estimation of the best parameters for the dataset and then builds its data structure using these parameters. We required from the two approaches to have accuracy > 0.90 , which in our case means that in at least 90 out of the 100 queries they would manage to find the approximate nearest neighbor. We also measured the maximum resident set size of each approach which translates to the maximum portion of the main memory (RAM) occupied by a process during its lifetime. This roughly corresponds to the size of the dataset plus the size of the data structure for the E2LSH implementation and to the size of the dataset plus the size of the embedded dataset plus the size of the data structure for our approach.



■ **Figure 3** Comparison of average query time of our embedding approach against the E2LSH implementation.



■ **Figure 4** Comparison of memory usage of our embedding approach against the E2LSH implementation.

ANN Results

It is clear from Figure 3 that E2LSH is faster than our approach by a factor of 3. However in Figure 4, where we present the memory usage comparison between the two approaches, it is obvious that E2LSH also requires more space. Figure 4 also validates the linear space dependency of our embedding method. A few points can be raised here. First of all, we supplied the appropriate range to the LSH implementation, which gave it an advantage, because typically that would have to be computed empirically. To counter that, we allowed our algorithm to stop its search in the original space when it encountered a point that was at distance $\leq R$ from the query point. Our approach was simpler and the bottleneck was in the computation of the closest point out of the k returned from the BBD-Tree. We conjecture that we can choose better values for our parameters d' and k . Lastly, the theoretical guarantees for the query time of LSH are better than ours, but we did perform better in terms of space usage as expected.

Real life dataset

We also compared the two approaches using the ANN_SIFT1M [17] dataset which contains a collection of 1000000 vectors in 128 dimensions. This dataset also provides a query file containing 10000 vectors and a groundtruth file, which contains for each query the IDs of its 100 nearest neighbors. These files allowed us to estimate the accuracy for each approach, as the fraction $\frac{\#hits}{10000}$ where $\#hits$ denotes, for some query, the number of times one of its 100 nearest neighbors were returned. The parameters of the two implementations were chosen empirically in order to achieve an accuracy of about 85%. For our approach we set the projection dimension $d' = 25$ and for the BBD-trees we specified 100 points per leaf and $\epsilon = 0.5$ for the ϵ - k ANNs queries. We also used $k = \sqrt{n}$. For the E2LSH implementation we specified the radius $R = 240$, $k = 18$ and $L = 250$. As before we measured the average query time and the maximum resident set size. Our approach required an average of 0.171588s per query, whilst E2LSH required 0.051957s. However our memory footprint was 1255948 kbytes and E2LSH used 4781400 kbytes.

7 Open questions

In terms of practical efficiency it is obvious that checking the real distance to the neighbors while performing an ϵ - k ANNs search in the reduced space, is more efficient in practice than naively scanning the returned sequence of k -approximate nearest neighbors and looking for the best in the initial space. Moreover, we do not exploit the fact that BBD-trees return a sequence and not simply a set of neighbors.

Our embedding possibly has further applications. One possible application is the problem of computing the k -th approximate nearest neighbor. The problem may reduce to computing all neighbors between the i -th and the j -th nearest neighbors in a space of significantly smaller dimension for some appropriate $i < k < j$. Other possible applications include computing the approximate minimum spanning tree or the closest pair of points.

Our embedding approach could be possibly applied in other metrics or exploit other properties of the pointset. We also intend to seek connections between our work and the notion of local embeddings introduced in [1].

References

- 1 I. Abraham, Y. Bartal, and O. Neiman. Local embeddings of metric spaces. In *Proc. 39th ACM Symposium on Theory of Computing*, pages 631–640. ACM Press, 2007.
- 2 D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.
- 3 A. Andoni and P. Indyk. E²LSH 0.1 User Manual, Implementation of LSH: E2LSH, <http://www.mit.edu/~andoni/LSH>, 2005.
- 4 A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, 51(1):117–122, 2008.
- 5 A. Andoni and I. Razenshteyn. Optimal data-dependent hashing for approximate near neighbors. In *arXiv:1501.01062, to appear in the Proc. 47th ACM Symp. Theory of Computing*, STOC'15, 2015.
- 6 S. Arya, G. D. da Fonseca, and D. M. Mount. Approximate polytope membership queries. In *Proc. 43rd Annual ACM Symp. Theory of Computing*, STOC'11, pages 579–586, 2011.
- 7 S. Arya, T. Malamatos, and D. M. Mount. Space-time tradeoffs for approximate nearest neighbor searching. *J. ACM*, 57(1):1:1–1:54, 2009.

- 8 S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, 1998.
- 9 A. Beygelzimer, S. Kakade, and J. Langford. Cover trees for nearest neighbor. In *Proc. 23rd Intern. Conf. Machine Learning, ICML'06*, pages 97–104, 2006.
- 10 S. Dasgupta and Y. Freund. Random projection trees and low dimensional manifolds. In *Proc. 40th Annual ACM Symp. Theory of Computing, STOC'08*, pages 537–546, 2008.
- 11 S. Dasgupta and A. Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003.
- 12 M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni. Locality-sensitive hashing scheme based on p-stable distributions. In *Proc. 20th Annual Symp. Computational Geometry, SCG'04*, pages 253–262, 2004.
- 13 A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proc. 44th Annual IEEE Symp. Foundations of Computer Science, FOCS'03*, pages 534–, 2003.
- 14 S. Har-Peled and M. Mendel. Fast construction of nets in low dimensional metrics, and their applications. In *Proc. 21st Annual Symp. Computational Geometry, SCG'05*, pages 150–158, 2005.
- 15 P. Indyk and R. Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. 30th Annual ACM Symp. Theory of Computing, STOC'98*, pages 604–613, 1998.
- 16 P. Indyk and A. Naor. Nearest-neighbor-preserving embeddings. *ACM Trans. Algorithms*, 3(3), 2007.
- 17 H. Jegou, M. Douze, and C. Schmid. Product quantization for nearest neighbor search. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2011.
- 18 W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.
- 19 D. R. Karger and M. Ruhl. Finding nearest neighbors in growth-restricted metrics. In *Proc. 34th Annual ACM Symp. Theory of Computing, STOC'02*, pages 741–750, 2002.
- 20 R. Krauthgamer and J. R. Lee. Navigating nets: Simple algorithms for proximity search. In *Proc. 15th Annual ACM-SIAM Symp. Discrete Algorithms, SODA'04*, pages 798–807, 2004.
- 21 S. Meiser. Point location in arrangements of hyperplanes. *Inf. Comput.*, 106(2):286–303, 1993.
- 22 D. M. Mount. ANN programming manual: <http://www.cs.umd.edu/~mount/ANN/>, 2010.
- 23 R. O'Donnell, Yi Wu, and Y. Zhou. Optimal lower bounds for locality-sensitive hashing (except when q is tiny). *ACM Trans. Comput. Theory*, 6(1):5:1–5:13, 2014.
- 24 R. Panigrahy. Entropy based nearest neighbor search in high dimensions. In *Proc. 17th Annual ACM-SIAM Symp. Discrete Algorithms, SODA'06*, pages 1186–1195, 2006.
- 25 I. Psarros. Low quality embeddings and approximate nearest neighbors, MSc Thesis, Dept. of Informatics & Telecommunications, University of Athens, 2014.
- 26 S. Vempala. Randomly-oriented k-d trees adapt to intrinsic dimension. In *Proc. Foundations of Software Technology & Theor. Computer Science*, pages 48–57, 2012.

Restricted Isometry Property for General p -Norms*

Zeyuan Allen-Zhu, Rati Gelashvili, and Ilya Razenshteyn

MIT CSAIL, Cambridge, MA, USA
{zeyuan, gelash, ilyaraz}@csail.mit.edu

Abstract

The Restricted Isometry Property (RIP) is a fundamental property of a matrix which enables sparse recovery. Informally, an $m \times n$ matrix satisfies RIP of order k for the ℓ_p norm, if $\|Ax\|_p \approx \|x\|_p$ for every vector x with at most k non-zero coordinates.

For every $1 \leq p < \infty$ we obtain almost tight bounds on the minimum number of rows m necessary for the RIP property to hold. Prior to this work, only the cases $p = 1$, $1 + 1/\log k$, and 2 were studied. Interestingly, our results show that the case $p = 2$ is a “singularity” point: the optimal number of rows m is $\tilde{\Theta}(k^p)$ for all $p \in [1, \infty) \setminus \{2\}$, as opposed to $\tilde{\Theta}(k)$ for $k = 2$.

We also obtain almost tight bounds for the column sparsity of RIP matrices and discuss implications of our results for the Stable Sparse Recovery problem.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, G.3 Probability and Statistics

Keywords and phrases compressive sensing, dimension reduction, linear algebra, high-dimensional geometry

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.451

1 Introduction

The main object of our interest is a matrix with *Restricted Isometry Property for the ℓ_p norm* (RIP- p). Informally speaking, we are interested in a linear map from \mathbb{R}^n to \mathbb{R}^m with $m \ll n$ that approximately preserves ℓ_p norms for *all* vectors that have only few non-zero coordinates.

More precisely, an $m \times n$ matrix $A \in \mathbb{R}^{m \times n}$ is said to have (k, D) -RIP- p property for sparsity $k \in [n] \stackrel{\text{def}}{=} \{1, \dots, n\}$, distortion $D > 1$, and the ℓ_p norm for $p \in [1, \infty)$, if for every vector $x \in \mathbb{R}^n$ with at most k non-zero coordinates one has

$$\|x\|_p \leq \|Ax\|_p \leq D \cdot \|x\|_p .$$

In this work we investigate the following question: given $p \in [1, \infty)$, $n \in \mathbb{N}$, $k \in [n]$, and $D > 1$,

What is the smallest $m \in \mathbb{N}$ so that there exists a (k, D) -RIP- p matrix $A \in \mathbb{R}^{m \times n}$?

Besides that, the following question arises naturally from the complexity of computing Ax :

What is the smallest column sparsity d for such a (k, D) -RIP- p matrix $A \in \mathbb{R}^{m \times n}$?

(Above, we denote by column sparsity the maximum number of non-zero entries in a column of A .)

* The full version of this paper can be found at <http://arxiv.org/abs/1407.2178> [2].



1.1 Motivation

Why are RIP matrices important? RIP-2 matrices were introduced by Candès and Tao [7] for decoding a vector f from corrupted linear measurements $Bf + e$ under the assumption that the vector of errors e is sufficiently sparse (has only few non-zero entries). Later Candès, Romberg and Tao [6] used RIP-2 matrices to solve *the (Noisy) Stable Sparse Recovery* problem, which has since found numerous applications in areas such as compressive sensing of signals [6, 11], genetic data analysis [16], and data stream algorithms [19, 12].

The (noisy) stable sparse recovery problem is defined as follows. The input signal $x \in \mathbb{R}^n$ is assumed to be close to k -sparse, that is, to have most of the “mass” concentrated on k coordinates. The goal is to design a set of m linear measurements that can be represented as a single $m \times n$ matrix A such that, given a *noisy sketch* $y = Ax + e \in \mathbb{R}^m$, where $e \in \mathbb{R}^m$ is a noise vector, one can “approximately” recover x . Formally, the recovered vector $\hat{x} \in \mathbb{R}^n$ is required to satisfy

$$\|x - \hat{x}\|_p \leq C_1 \min_{k\text{-sparse } x^*} \|x - x^*\|_1 + C_2 \cdot \|e\|_p \quad (1.1)$$

for some $C_1, C_2 > 0$, $p \in [1, \infty)$, and $k \in [n]$.

(In order for (1.1) to be meaningful, we also require $\|A\|_p \leq 1$ – or equivalently, $\|Ax\|_p \leq \|x\|_p$ for all x – since otherwise, by scaling A up, the noise vector e will become negligible.)

We refer to (1.1) as the ℓ_p/ℓ_1 *guarantee*. The parameters of interest include: the number of measurements m , the column sparsity of the measurement matrix A , the approximation factors C_1, C_2 and the complexity of the recovery procedure.

Candès, Romberg and Tao [6] proved that if A is $(O(k), 1 + \varepsilon)$ -RIP-2 for a sufficiently small $\varepsilon > 0$, then one can achieve the ℓ_2/ℓ_1 guarantee with $C_1 = O(k^{-1/2})$ and $C_2 = O(1)$ in polynomial time.

The $p = 1$ case was first studied by Berinde *et al.* [4]. They prove that if A is $(O(k), 1 + \varepsilon)$ -RIP-1 for a sufficiently small $\varepsilon > 0$ and has a certain additional property, then one can achieve the ℓ_1/ℓ_1 guarantee with $C_1 = O(1)$, $C_2 = O(1)$.

We note that *any* matrix A that allows the (noisy) stable sparse recovery with the ℓ_p/ℓ_1 guarantee *must have the* (k, C_2) -RIP- p *property*. For the proof see the full version.

Known constructions and limitations. Candès and Tao [7] proved that for every $\varepsilon > 0$, a matrix with $m = O(k \log(n/k)/\varepsilon^2)$ rows and n columns whose entries are sampled from i.i.d. Gaussians is $(k, 1 + \varepsilon)$ -RIP-2 with high probability. Later, a simpler proof of the same result was discovered by Baraniuk *et al.* [3]¹. Berinde *et al.* [4] showed that a (scaled) *random sparse binary matrix* with $m = O(k \log(n/k)/\varepsilon^2)$ rows is $(k, 1 + \varepsilon)$ -RIP-1 with high probability².

Since the number of measurements is very important in practice, it is natural to ask, how optimal is the dimension bound $m = O(k \log(n/k))$ that the above constructions achieve? The results of Do Ba *et al.* [10] and Candès [8] imply the lower bound $m = \Omega(k \log(n/k))$ for $(k, 1 + \varepsilon)$ -RIP- p matrices for $p \in \{1, 2\}$, provided that $\varepsilon > 0$ is sufficiently small.

Another important parameter of a measurement matrix A is its *column sparsity*: the maximum number of non-zero entries in a single column of A . If A has column sparsity d , then we can perform multiplication $x \mapsto Ax$ in time $O(nd)$ as opposed to the naive $O(nm)$ bound. Moreover, for sparse matrices A , one can maintain the sketch $y = Ax$ very efficiently

¹ This proof has an advantage that it works for any subgaussian random variables, such as random ± 1 's.

² In the same paper [4] it is observed that the same construction works for $p = 1 + 1/\log k$.

■ **Table 1** Prior and new bounds on RIP- p matrices.

p	rows m	column sparsity d	references
1	$\Theta(k \log(n/k))$	$\Theta(\log(n/k))$	[4, 10, 20, 14]
$1 + \frac{1}{\log k}$	$O(k \log(n/k))$	$O(\log(n/k))$	[4]
(1, 2)	$\tilde{\Theta}(k^p)$	$\tilde{\Theta}(k^{p-1})$	this work
2	$\Theta(k \log(n/k))$	$\Theta(k \log(n/k))$	[7, 6, 8, 3, 10, 9, 23]
(2, ∞)	$\tilde{\Theta}(k^p)$	$\tilde{\Theta}(k^{p-1})$	this work

if we update x . Namely, if we set $x \leftarrow x + \alpha \cdot e_i$, where $\alpha \in \mathbb{R}$ and $e_i \in \mathbb{R}^n$ is a basis vector, then we can update y in time $O(d)$ instead of the naive bound $O(m)$.

The aforementioned constructions of RIP matrices exhibit very different behavior with respect to column sparsity. RIP-2 matrices obtained from random Gaussian matrices are obviously dense, whereas the construction of RIP-1 matrices of Berinde *et al.* [4] gives very small column sparsity $d = O(\log(n/k)/\varepsilon)$. It is known that in both cases the bounds on column sparsity are essentially tight.

Indeed, Nelson and Nguyễn showed [23] that any non-trivial column sparsity is impossible for RIP-2 matrices unless m is much larger than $O(k \log(n/k))$. Nachin showed [20] that any RIP-1 matrix with $O(k \log(n/k))$ rows must have column sparsity $\Omega(\log(n/k))$. Besides that, Indyk and Razenshteyn showed [14] that every RIP-1 matrix ‘must be sparse’: any RIP-1 matrix with $O(k \log(n/k))$ rows can be perturbed slightly and made $O(\log(n/k))$ -sparse.

Another notable difference between RIP-1 and RIP-2 matrices is the following. The construction of Berinde *et al.* [4] provides RIP-1 matrices with non-negative entries, whereas Chandar proved [9] that any RIP-2 matrix with non-negative entries must have $m = \Omega(k^2)$ (and this was later improved to $m = \Omega(k^2 \log(n/k))$ [23, 1]). In other words, negative signs are crucial in the construction of RIP-2 matrices but not for the RIP-1 case.

1.2 Our results

Motivated by these discrepancies between the optimal constructions for RIP- p matrices with $p \in \{1, 1 + \frac{1}{\log k}, 2\}$, we initiate the study of RIP- p matrices for the general $p \in [1, \infty)$.

Having in mind that the upper bound $m = O(k \log(n/k))$ holds for RIP- p matrices with $p \in \{1, 1 + \frac{1}{\log k}, 2\}$, it would be natural to conjecture that the same bound holds at least for every $p \in (1, 2)$. As we will see, surprisingly, this conjecture is very far from being true.

Also, knowing that the column sparsity $d = O(k \log(n/k))$ can be obtained for $p = 2$ while $d = O(\log(n/k))$ can be obtained for $p = 1$, it is interesting to “interpolate” these two bounds.

Besides the mathematical interest, a more “applied” reason to study RIP- p matrices for the general p is to get new guarantees for the stable sparse recovery. Indeed, we obtain new results in this direction.

Our upper bounds. On the positive side, for all $\varepsilon > 0$ and all $p \in (1, \infty)$, we construct $(k, 1 + \varepsilon)$ -RIP- p matrices with $m = \tilde{O}(k^p)$ rows. Here, we use the $\tilde{O}(\cdot)$ -notation to hide factors that depend on ε, p , and are polynomial in $\log n$. More precisely, we show that a (scaled) *random sparse 0/1 matrix* with $\tilde{O}(k^p)$ rows and column sparsity $\tilde{O}(k^{p-1})$ has the desired RIP property with high probability.

This construction essentially matches that of Berinde *et al.* [4] when p approaches 1. At the same time, when $p = 2$, our result matches known constructions of non-negative RIP-2

matrices based on the incoherence argument.³

Our lower bounds. Surprisingly, we show that, despite our upper bounds being suboptimal for $p = 2$, they are essentially tight for every constant $p \in (1, \infty)$ except 2. Namely, they are optimal both in terms of the dimension m and the column sparsity d .

More formally, on the dimension side, for every $p \in (1, \infty) \setminus \{2\}$, distortion $D > 1$, and (k, D) -RIP- p matrix $A \in \mathbb{R}^{m \times n}$, we show that $m = \Omega(k^p)$, where $\Omega(\cdot)$ hides factors that depend on p and D . Note that, it is not hard to extend an argument of Chandar [9] and obtain a lower bound $m = \Omega(k^{p-1})$.⁴ This additional factor k is exactly what makes our lower bound non-trivial and tight for $p \in (1, \infty) \setminus \{2\}$, and thus enables us to conclude that $p = 2$ is a “singularity”.⁵

As for the column sparsity, we present a simple extension of the argument of Chandar [9] and prove that for every $p \in [1, \infty)$ any (k, D) -RIP- p matrix must have column sparsity $\Omega(k^{p-1})$.

RIP matrices and sparse recovery. We extend the result of Candès, Romberg and Tao [6] to show that, for every $p > 1$, RIP- p matrices allow the stable sparse recovery with the ℓ_p/ℓ_1 guarantee and approximation factors $C_1 = O(k^{-1+1/p})$, $C_2 = O(1)$ in polynomial time. This extension is quite straightforward and seems to be folklore, but, to the best of our knowledge, it is not recorded anywhere.

On the other hand, for every $p \geq 1$, it is almost immediate that *any* matrix A that allows the stable sparse recovery with the ℓ_p/ℓ_1 guarantee – even if it works only for k -sparse signals – *must have the (k, C_2) -RIP- p property*. For the sake of completeness, we have included both the above proofs in the full version.

Implications to sparse recovery. Using the above equivalent relationship between the stable sparse recovery problem and the RIP- p matrices, we conclude that the stable sparse recovery with the ℓ_p/ℓ_1 guarantee requires $m = \tilde{\Theta}(k^p)$ measurements for every $p \in [1, \infty) \setminus \{2\}$, and requires $d = \tilde{\Theta}(k^{p-1})$ column sparsity for every $p \in [1, \infty)$. Our results together draw tradeoffs between the following three parameters in stable sparse recovery:

- p , the ℓ_p/ℓ_1 guarantee for the stable sparse recovery,⁶
- m , the number of measurements needed for sketching, and
- d , the running time (per input coordinate) needed for sketching.

It was pointed out by an anonymous referee that for the *noiseless* case – that is, when the noise vector e is always zero – better upper bounds are possible. Using the result of Gilbert *et al.* [13], one can obtain, for every $p \geq 2$, the noiseless stable sparse recovery procedure

³ That is, a (scaled) random $m \times n$ binary matrix with $m = O(\varepsilon^{-2} k^2 \log(n/k))$ rows and sparsity $d = O(\varepsilon^{-1} k \log(n/k))$ satisfies the $(k, 1 + \varepsilon)$ -RIP-2 property. This can be proved using for instance the incoherence argument from [24]: any incoherent matrix satisfies the RIP-2 property with certain parameters.

⁴ Also, the same argument gives the lower bound $\Omega(k^p)$ for *binary* RIP- p matrices for every $p \in [1, \infty)$.

⁵ A similar singularity is known to exist for linear dimension reduction for arbitrary point sets with respect to ℓ_p norms [18]; alas, tight bounds for that problem are not known.

⁶ We note that the ℓ_p/ℓ_1 and the ℓ_q/ℓ_1 guarantees are incomparable. However, it is often more desirable to have larger p in this ℓ_p/ℓ_1 guarantee to ensure a better recovery quality. This is because, if the noise vector $e = 0$, the ℓ_q/ℓ_1 guarantee (with $C_1 = O(k^{-1+1/q})$) can be shown to be stronger than the ℓ_p/ℓ_1 one (with $C_1 = O(k^{-1+1/p})$) whenever $q > p$. However, when there is a noise term, the guarantee $\|x - \hat{x}\|_p \leq O(1) \cdot \|e\|_p$ is incomparable to $\|x - \hat{x}\|_q \leq O(1) \cdot \|e\|_q$ for $p \neq q$.

with the ℓ_p/ℓ_1 guarantee using only $m = \tilde{O}(k^{2-2/p})$ measurements. Therefore, our results also imply a very large gap, both in terms of m and d , between the *noiseless* and the *noisy* stable sparse recovery problems.

2 Overview of the Proofs

2.1 Upper bounds

We construct RIP- p matrices as follows. Beginning with a zero matrix A with $m = \tilde{O}(k^p)$ rows and n columns, independently for each column of A , we choose $d = \tilde{O}(k^{p-1})$ out of m entries uniformly at random (without replacement), and assign the value $d^{-1/p}$ to those selected entries. For this construction, we have two very different analyses of its correctness: one works only for $p \geq 2$, and the other works only for $1 < p < 2$.

For $p \geq 2$, the most challenging part is to show that $\|Ax\|_p \leq (1 + \varepsilon)\|x\|_p$ holds with high probability, for all k -sparse vectors x . We reduce this problem to a probabilistic question *similar in spirit* to the following “balls and bins” question. Consider n bins in which we throw n balls uniformly and independently. As a result, we get n numbers X_1, X_2, \dots, X_n , where X_i is the number of balls falling into the i -th bin. We would like to upper bound the tail $\Pr[S \geq 1000 \cdot \mathbb{E}[S]]$ for the random variable $S = \sum_{i=1}^n X_i^{p-1}$. (Here, the constant 1000 can be replaced with any large enough one since we do not care about constant factors in this paper.) The first challenge is that X_i 's are not independent. To deal with this issue we employ the notion of *negative association* of random variables introduced by Joag-Dev and Proschan [15]. The second problem is that the random variables X_i^{p-1} are heavy tailed: they have tails of the form $\Pr[X_i^{p-1} \geq t] \approx \exp(-t^{\frac{1}{p-1}})$, so the standard technique of bounding the moment-generating function does not work. Instead, we bound the high moments of S directly, which introduces certain technical challenges. Let us remark that sums of i.i.d. heavy-tailed variables were thoroughly studied by Nagaev [21, 22], but it seems that for the results in these papers the independence of summands is crucial.

One major reason the above approach fails to work for $1 < p < 2$ is that, in this range, even the best possible tail inequality for S is too weak for our purposes. Another challenge in this regime is that, to bound the “lower tail” of $\|Ax\|_p^p$ (that is, to prove that $\|Ax\|_p \geq (1 - \varepsilon)\|x\|_p$ holds for all k -sparse x), the simple argument used for $p \geq 2$ no longer works. Our solution to both problems above is to instead build our RIP matrices based on the following general notion of bipartite expanders.

► **Definition 2.1.** Let $G = (U, V, E)$ with $|U| = n$, $|V| = m$ and $E \subseteq U \times V$ be a bipartite graph such that all vertices from U have the same degree d . We say that G is an (ℓ, d, δ) -*expander*, if for every $S \subseteq U$ with $|S| \leq \ell$ we have

$$|\{v \in V \mid \exists u \in S (u, v) \in E\}| \geq (1 - \delta)d|S| .$$

It is known that random d -regular graphs are good expanders, and we can take the (scaled) adjacency matrix of such an expander and prove that it satisfies the desired RIP- p property for $1 < p < 2$. Our argument can be seen as a subtle interpolation between the argument from [4], which proves that (scaled) adjacency matrices of $(k, d, \Theta(\varepsilon))$ -expanders (with $\tilde{O}(k)$ rows) are $(k, 1 + \varepsilon)$ -RIP-1 and the one using incoherence argument,⁷ which shows that $(2, d, \Theta(\varepsilon/k))$ -expanders give $(k, 1 + \varepsilon)$ -RIP-2 matrices (with $\tilde{O}(k^2)$ rows).

⁷ It is known [24] that an incoherent matrix satisfies the RIP-2 property with certain parameters. At the same time, the notion of incoherence can be interpreted as expansion for $\ell = 2$.

2.2 Lower bounds

In the full version of our paper [2], we derive our dimension lower bound $m = \Omega(k^p)$ essentially from norm inequalities. The high-level idea can be described in four simple steps. Consider any (k, D) -RIP- p matrix $A \in \mathbb{R}^{n \times m}$, and assume that D is very close to 1 in this high-level description.

In the first three steps, we deduce from the RIP property that (a) the sum of the p -th powers of all entries in A is approximately n , (b) the largest entry in A (i.e., the vector ℓ_∞ -norm of A) is essentially at most $k^{1/p-1}$, and (c) the sum of squares of all entries in A is at least $n(\frac{k}{m})^{2/p-1}$ if $p \in (1, 2)$, or at most $n(\frac{k}{m})^{2/p-1}$ if $p > 2$. In the fourth step, we combine (a) (b) and (c) together by arguing about the relationships between the ℓ_p , ℓ_∞ and ℓ_2 norms of entries of A , and prove the desired lower bound on m .

The sparsity lower bound $d = \Omega(k^{p-1})$ can be obtained via a simple extension of the argument of Chandar [9]. It is possible to extend the techniques of Nelson and Nguyễn [23] to obtain a slightly better sparsity lower bound. However, since we were unable to obtain a *tight* bound this way, we decided not to include it.

3 RIP Construction for $p \geq 2$

In this section, we construct $(k, 1 + \varepsilon)$ -RIP- p matrices for $p \geq 2$ by proving the following theorem.

► **Definition 3.1.** We say that an $m \times n$ matrix A is a *random binary matrix with sparsity* $d \in [m]$, if A is generated by assigning $d^{-1/p}$ to d random entries per column (selected uniformly at random without replacement), and assigning 0 to the remaining entries.

► **Theorem 3.2.** For all $n \in \mathbb{Z}_+$, $k \in [n]$, $\varepsilon \in (0, \frac{1}{2})$ and $p \in [2, \infty)$, there exist $m, d \in \mathbb{Z}_+$ with

$$m = p^{O(p)} \cdot \frac{k^p}{\varepsilon^2} \cdot \log^{p-1} n \quad \text{and} \quad d = p^{O(p)} \cdot \frac{k^{p-1}}{\varepsilon} \cdot \log^{p-1} n \leq m$$

such that, letting A be a random binary $m \times n$ matrix of sparsity d , with probability at least 98%, A satisfies $(1 - \varepsilon)\|x\|_p^p \leq \|Ax\|_p^p \leq (1 + \varepsilon)\|x\|_p^p$ for all k -sparse vectors $x \in \mathbb{R}^n$.

Our proof is divided into two steps: (1) the “lower-tail step”, that is, with probability at least 0.99 we have $\|Ax\|_p^p \geq (1 - \varepsilon)\|x\|_p^p$ for all k -sparse x , and (2) the “upper-tail step”, that is, with probability at least 0.99, we have $\|Ax\|_p^p \leq (1 + \varepsilon)\|x\|_p^p$.

For every $j \in [n]$, let us denote by $S_j \subseteq [m]$ the set of non-zero rows of the j -th column of A .

3.1 The Lower-Tail Step

The lower-tail step is very simple. It suffices to show that, with high probability, $|S_i \cap S_j|$ is small for every pair of different $i, j \in [n]$, which will then imply that if only k columns of A are considered, every S_i has to be almost disjoint from the union of the S_j of the $k - 1$ remaining columns. This can be summarized by the following claim, whose proof is deferred to the full version of this paper.

► **Claim 3.3.** If $d \geq C\varepsilon^{-1}k \log n$ and $m \geq 2dk/\varepsilon$, where C is some large enough constant, then

$$\Pr \left[\forall 1 \leq i < j \leq n \quad |S_i \cap S_j| \leq \frac{\varepsilon d}{k} \right] \geq 0.99 .$$

Now, to prove the lower tail, without loss of generality, let us assume that x is supported on $[k]$, the first k coordinates. For every $j \in [k]$, we denote by $S'_j = S_j \setminus \bigcup_{j' \in [k] \setminus \{j\}} S_{j'}$, the set of non-zero rows in column j that are not shared with the supports of other columns in $[k] \setminus \{j\}$. If the event in Claim 3.3 holds, then for every $j \in [k]$, we have $|S'_j| \geq (1 - \varepsilon)d$. Thus, we can lower bound $\|Ax\|_p$ as

$$\|Ax\|_p^p = \frac{1}{d} \cdot \sum_{i=1}^m \left| \sum_{j \in [k]: i \in S_j} x_j \right|^p \geq \frac{1}{d} \cdot \sum_{i=1}^m \left| \sum_{j \in [k]: i \in S'_j} x_j \right|^p = \frac{1}{d} \cdot \sum_{j \in [k]} |S'_j| \cdot |x_j|^p \geq (1 - \varepsilon) \|x\|_p^p . \tag{3.1}$$

► **Remark.** The above claim only works when $m = \Omega(k^2 \log n / \varepsilon^2)$, and therefore we cannot use it in for the case of $1 < p < 2$.

3.2 The Upper-Tail Step

Below we describe the framework of our proof for the upper-tail step, deferring all technical details to the full version of this paper.

Suppose again that x is supported on $[k]$. Then, we upper bound $\|Ax\|_p^p$ as

$$\begin{aligned} \|Ax\|_p^p &= \frac{1}{d} \cdot \sum_{i=1}^m \left| \sum_{j \in [k]: i \in S_j} x_j \right|^p \leq \frac{1}{d} \cdot \sum_{i=1}^m |\{j' \in [k] \mid i \in S_{j'}\}|^{p-1} \cdot \sum_{j \in [k]: i \in S_j} |x_j|^p \\ &= \frac{1}{d} \cdot \sum_{j=1}^k |x_j|^p \cdot \sum_{i \in S_j} |\{j' \in [k] \mid i \in S_{j'}\}|^{p-1} , \end{aligned} \tag{3.2}$$

where the first inequality follows from the fact that $(a_1 + \dots + a_N)^p \leq N^{p-1}(a_1^p + \dots + a_N^p)$ for any sequence of N non-negative reals a_1, \dots, a_N . Note that the quantity $|\{j' \in [k] \mid i \in S_{j'}\}| \in [k]$ captures the number of non-zeros of A in the i -th row and the first k columns. From now on, in order to prove the desired upper tail, it suffices to show that, with high probability

$$\forall j \in [k], \quad \sum_{i \in S_j} |\{j' \in [k] \mid i \in S_{j'}\}|^{p-1} \leq (1 + \varepsilon)d . \tag{3.3}$$

To prove this, let us fix some $j^* \in [k]$ and upper bound the probability that (3.3) holds for $j = j^*$, and then take a union bound over the choices of j^* . Without loss of generality, assume that $S_{j^*} = \{1, 2, \dots, d\}$, consisting of the first d rows. For every $i \in S_{j^*}$, define a random variable $X_i \stackrel{\text{def}}{=} |\{j' \in [k] \mid i \in S_{j'}\}| - 1$. It is easy to see that X_i is distributed as $\text{Bin}(k - 1, d/m)$, the binomial distribution that is the sum of $k - 1$ i.i.d. random 0/1 variables, each being 1 with probability d/m . For notational simplicity, let us define $\delta \stackrel{\text{def}}{=} dk/m$. We will later choose $\delta < \varepsilon$ to be very small. Our goal in (3.3) can now be reformulated as follows: upper bound the probability

$$\Pr \left[\sum_{i=1}^d ((X_i + 1)^{p-1} - 1) > \varepsilon d \right] .$$

We begin with a lemma showing an upper bound on the moments of each $Y_i \stackrel{\text{def}}{=} (X_i + 1)^{p-1} - 1$.

► **Lemma 3.4.** *There exists a constant $C \geq 1$ such that, if X is drawn from the binomial distribution $\text{Bin}(k - 1, \delta/k)$ for some $\delta < 1/(2e^2)$, and $p \geq 2$, then for any real $\ell \geq 1$,*

$$\mathbb{E}[(X + 1)^{p-1} - 1]^\ell \leq C \cdot \delta(\ell(p - 1) + 1)^{\ell(p-1)+1} .$$

Next, we note that although the random variables X_i 's are dependent, they can be verified to be *negatively associated*, a notion introduced by Joag-Dev and Proschan [15]. This theory allows us to conclude the following bound on the moments.

► **Lemma 3.5.** *Let $\tilde{X}_1, \dots, \tilde{X}_d$ be d random variables, each drawn independently from $\text{Bin}(k-1, \delta/k)$. Then, for every integer $t \geq 1$ we have*

$$\mathbb{E} \left[\left(\sum_{i=1}^d ((X_i + 1)^{p-1} - 1) \right)^t \right] \leq \mathbb{E} \left[\left(\sum_{i=1}^d ((\tilde{X}_i + 1)^{p-1} - 1) \right)^t \right].$$

Now, using the moments of random variables $Y_i = (X_i + 1)^{p-1} - 1$ from Lemma 3.4, as well as Lemma 3.5, we can compute the tail bound of the sum $\sum_{i=1}^d Y_i$. Our proof of the following Lemma uses the result of Latała [17].

► **Lemma 3.6.** *There exists constants $C \geq 1$ such that, whenever $\delta \leq \varepsilon/p^{Cp}$ and $d \geq p^{Cp}/\varepsilon$, we have*

$$\Pr \left[\sum_{i=1}^d ((X_i + 1)^{p-1} - 1) > \varepsilon d \right] \leq e^{-\Omega\left(\frac{(\varepsilon d)^{1/(p-1)}}{p}\right)}.$$

Finally, we are ready to prove Theorem 3.2.

Proof of Theorem 3.2. We can choose $d = \Theta(p)^{p-1} \cdot \frac{k^{p-1}}{\varepsilon} \cdot \log^{p-1} n$ so that $e^{-\Omega\left(\frac{(\varepsilon d)^{1/(p-1)}}{p}\right)} < \frac{1}{100} \frac{1}{k \binom{n}{k}}$. Since our choice of $m = \frac{dkp^{\Theta(p)}}{\varepsilon}$ ensures that $\delta = dk/m \leq \varepsilon/p^{Cp}$, and our choice of d ensures $d \geq p^{Cp}/\varepsilon$, we can apply Lemma 3.6 and conclude that with probability at least $1 - \frac{1}{100} \frac{1}{k \binom{n}{k}}$ one has

$$\sum_{i \in S_{j^*}} |\{j' \in [k] \mid i \in S_{j'}\}|^{p-1} = \sum_{i=1}^d (X_i + 1)^{p-1} \leq (1 + \varepsilon)d.$$

Therefore, by applying the union bound over all $j^* \in [k]$, we conclude that with probability at least $1 - \frac{1}{100} \frac{1}{\binom{n}{k}}$, the desired inequality (3.3) is satisfied for all $j \in [k]$.

Recall that, owing to (3.2), the inequality (3.3) implies that $\|Ax\|_p^p \leq (1 + \varepsilon)\|x\|_p^p$ for every $x \in \mathbb{R}^n$ that is supported on the *first* k coordinates. By another union bound over the choices of all possible $\binom{n}{k}$ subsets of $[n]$, we conclude that with probability at least 0.99, we have $\|Ax\|_p^p \leq (1 + \varepsilon)\|x\|_p^p$ for all k -sparse vectors x .

On the other hand, since our choice of d and m satisfies the assumptions $d \geq \Omega(k \log n/\varepsilon)$ and $m \geq 2dk/\varepsilon$ in Claim 3.3, the lower tail $\|Ax\|_p^p \geq (1 - \varepsilon)\|x\|_p^p$ also holds with probability at least 0.99. Overall we conclude that with probability at least 0.98, we have $\|Ax\|_p^p \in (1 \pm \varepsilon)\|x\|_p^p$ for every k -sparse vector $x \in \mathbb{R}^n$. ◀

4 RIP Construction for $1 < p < 2$

In this section, we construct $(k, 1 + \varepsilon)$ -RIP- p matrices for $1 < p < 2$ by proving the following theorem.

We assume that $1 + \tau \leq p \leq 2 - \tau$ for some $\tau > 0$, and whenever we write $O_\tau(\cdot)$, we assume that some factor that depends on τ is hidden. (For instance, factors of $p/(1-p)$ may be hidden.)

► **Theorem 4.1.** *For every $n \in \mathbb{Z}_+$, $k \in [n]$, $0 < \varepsilon < 1/2$ and $1 + \tau \leq p \leq 2 - \tau$, there exist $m, d \in \mathbb{Z}_+$ with*

$$m = O_\tau \left(k^p \frac{\log n}{\varepsilon^2} + k^{4-2/p-p} \frac{\log n}{\varepsilon^{2/(p-1)}} \right) \quad \text{and} \quad d = O_\tau \left(\frac{k^{p-1} \cdot \log n}{\varepsilon} + \frac{k^{(p-1)/p} \cdot \log n}{\varepsilon^{1/(p-1)}} \right)$$

such that, letting A be a random binary $m \times n$ matrix of sparsity d , with probability at least 98%, A satisfies $(1 - \varepsilon)\|x\|_p^p \leq \|Ax\|_p^p \leq (1 + \varepsilon)\|x\|_p^p$ for all k -sparse vectors $x \in \mathbb{R}^n$.

Note that, when $k \geq \varepsilon^{-\frac{p(2-p)}{(p-1)^3}}$, the above bounds on m and k can be simplified as

$$m = O_\tau\left(\frac{k^p \cdot \log n}{\varepsilon^2}\right) \quad \text{and} \quad d = O_\tau\left(\frac{k^{p-1} \cdot \log n}{\varepsilon}\right).$$

Our proof of the above theorem is based on the existence of (ℓ, d, δ) bipartite expanders (recall the definition of such expanders from Definition 2.1):

► **Lemma 4.2** ([5, Lemma 3.10]). *For every $\delta \in (0, \frac{1}{2})$, and $\ell \in [n]$, there exist (ℓ, d, δ) -expanders with $d = O(\frac{\log n}{\delta})$ and $m = O(d\ell/\delta) = O(\frac{\ell \log n}{\delta^2})$.*

In fact, the proof of Lemma 4.2 implies a simple probabilistic construction of such expanders: with probability at least 98%, a random binary matrix A of sparsity d is the adjacency matrix of a $(2\ell, d, \delta)$ -expander scaled by $d^{-1/p}$, for $\delta = \Theta(\frac{\log n}{d})$ and $\ell = \Theta(\frac{\delta m}{d})$.

In the full version of this paper [2] we argue that, when A is the (scaled) adjacency matrix of a $(2\ell, d, \delta)$ -expander, for parameters choices $\ell = \Theta_\tau(k^{2-p})$ and $\delta = \Theta_\tau(\min\{\frac{\varepsilon}{k^{p-1}}, \frac{\varepsilon^{1/(p-1)}}{k^{(p-1)/p}}\})$, it satisfies that $\|Ax\|_p^p = 1 \pm \varepsilon$. This proof is very technical, but we have included a high-level description of its idea in the full version of this paper.

It is perhaps interesting to be noted that, our construction confirms our description in the introduction: it interpolates between the expander construction of RIP-1 matrices from [4] that uses $\ell = k$, and the construction of RIP-2 matrices using incoherence argument that essentially corresponds to $\ell = 2$.

Acknowledgments. We thank Piotr Indyk for encouraging us to work on this project and for many valuable conversations. We are grateful to Piotr Indyk and Ronitt Rubinfeld for teaching “Sublinear Algorithms”, where parts of this work appeared as a final project. We thank Artūrs Bačkurs, Chinmay Hegde, Gautam Kamath, Sepideh Mahabadi, Jelani Nelson, Huy Nguyễn, Eric Price and Ludwig Schmidt for useful conversations and feedback. Thanks to Leonid Boytsov for pointing us to [21, 22]. We are grateful to anonymous referees for pointing out some relevant literature. The first author is partly supported by a Simons Graduate Student Award under grant no. 284059.

References

- 1 Zeyuan Allen-Zhu, Rati Gelashvili, Silvio Micali, and Nir Shavit. Johnson-Lindenstrauss Compression with Neuroscience-Based Constraints. *ArXiv e-prints*, abs/1411.5383, November 2014. Also appeared in the Proceedings of the National Academy of Sciences of the USA, vol 111, no 47.
- 2 Zeyuan Allen-Zhu, Rati Gelashvili, and Ilya Razenshteyn. Restricted Isometry Property for General p -Norms. *ArXiv e-prints*, abs/1407.2178v3, February 2015.
- 3 Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263, 2008.
- 4 Radu Berinde, Anna C. Gilbert, Piotr Indyk, Howard Karloff, and Martin J. Strauss. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing (Allerton 2008)*, pages 798–805, 2008.
- 5 Harry Buhrman, Peter Bro Miltersen, Jaikumar Radhakrishnan, and Srinivasan Venkatesh. Are bitvectors optimal? *SIAM Journal on Computing*, 31(6):1723–1744, 2002.

- 6 Emmanuel Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.
- 7 Emmanuel Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- 8 Emmanuel J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathématique*, 346(9–10):589–592, 2008.
- 9 Venkat B. Chandar. *Sparse Graph Codes for Compression, Sensing, and Secrecy*. PhD thesis, Massachusetts Institute of Technology, 2010.
- 10 Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. Lower bounds for sparse recovery. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'10)*, pages 1190–1197, 2010.
- 11 David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006.
- 12 Anna C. Gilbert and Piotr Indyk. Sparse recovery using sparse matrices. *Proceedings of IEEE*, 98(6):937–947, 2010.
- 13 Anna C. Gilbert, Martin J. Strauss, Joel A. Tropp, and Roman Vershynin. One sketch for all: fast algorithms for compressed sensing. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC 2007)*, pages 237–246, 2007.
- 14 Piotr Indyk and Ilya Razenshteyn. On model-based RIP-1 matrices. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP'13)*, pages 564–575, 2013.
- 15 Kumar Joag-Dev and Frank Proschan. Negative association of random variables with applications. *Annals of Statistics*, 11(1):286–295, 1983.
- 16 Raghunandan M. Kainkaryam, Angela Bruex, Anna C. Gilbert, John Schiefelbein, and Peter J. Woolf. poolMC: Smart pooling of mRNA samples in microarray experiments. *BMC Bioinformatics*, 11(299), 2010.
- 17 Rafał Łatała. Estimation of moments of sums of independent real random variables. *Annals of Probability*, 25(3):1502–1513, 1997.
- 18 James R. Lee, Manor Mendel, and Assaf Naor. Metric structures in L_1 : dimension, snowflakes, and average distortion. *European Journal of Combinatorics*, 26(8):1180–1190, 2005.
- 19 S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1(2):117–236, 2005.
- 20 Mergen Nachin. Lower bounds on the column sparsity of sparse recovery matrices. undergraduate thesis, MIT, 2010.
- 21 A.V. Nagaev. Integral limit theorems taking large deviations into account when Cramér's condition does not hold. I. *Theory of Probability and Its Applications*, 14(1):51–64, 1969.
- 22 A.V. Nagaev. Integral limit theorems taking large deviations into account when Cramér's condition does not hold. II. *Theory of Probability and Its Applications*, 14(2):193–208, 1969.
- 23 Jelani Nelson and Huy L. Nguyễn. Sparsity lower bounds for dimensionality reducing maps. In *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC'13)*, pages 101–110, 2013.
- 24 Holger Rauhut. Compressive sensing and structured random matrices. *Theoretical foundations and numerical methods for sparse recovery*, 9:1–92, 2010.

Strong Equivalence of the Interleaving and Functional Distortion Metrics for Reeb Graphs

Ulrich Bauer¹, Elizabeth Munch², and Yusu Wang³

- 1 Department of Mathematics, Technische Universität München (TUM), Germany
mail@ulrich-bauer.org
- 2 Department of Mathematics & Statistics, University at Albany – SUNY, USA
emunch@albany.edu
- 3 Department of Computer Science and Engineering, The Ohio State University, USA
yusu@cse.ohio-state.edu

Abstract

The Reeb graph is a construction that studies a topological space through the lens of a real valued function. It has been commonly used in applications, however its use on real data means that it is desirable and increasingly necessary to have methods for comparison of Reeb graphs. Recently, several metrics on the set of Reeb graphs have been proposed. In this paper, we focus on two: the functional distortion distance and the interleaving distance. The former is based on the Gromov–Hausdorff distance, while the latter utilizes the equivalence between Reeb graphs and a particular class of cosheaves. However, both are defined by constructing a near-isomorphism between the two graphs of study. In this paper, we show that the two metrics are strongly equivalent on the space of Reeb graphs. Our result also implies the bottleneck stability for persistence diagrams in terms of the Reeb graph interleaving distance.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems: Geometrical problems and computations

Keywords and phrases Reeb graph, interleaving distance, functional distortion distance

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.461

1 Introduction

The Reeb graph is a construction that can be used to study a topological space with a real valued function by tracking the relationships between connected components of level sets. It was originally developed in the context of Morse theory [21], and was later introduced for shape analysis by Shinagawa et al. [23]. Since then, it has attracted much attention due to its wide use for various data analysis applications, such as shape comparison [15, 11], denoising [25], and shape understanding [7, 14]; see [2] for a survey. Recently, the applications of Reeb graphs have been further broadened to summarizing high-dimensional and/or complex data, in particular, reconstructing non-linear 1-dimensional structure in data [18, 12, 4] and summarizing collections of trajectory data [3]. Its practical applications have also been facilitated by the availability of efficient algorithms for computing the Reeb graph from a piecewise-linear function defined on a simplicial complex [20, 13, 9].

In addition to the standard construction, a generalization of the Reeb graph construction, known as Mapper, [24], has proven extremely useful in the field of topological data analysis [26, 19]. A variant of Mapper for real-valued functions, called the α -Reeb graph, was used in [4] to study data sets with 1-dimensional structure.



© Ulrich Bauer, Elizabeth Munch, and Yusu Wang;
licensed under Creative Commons License CC-BY

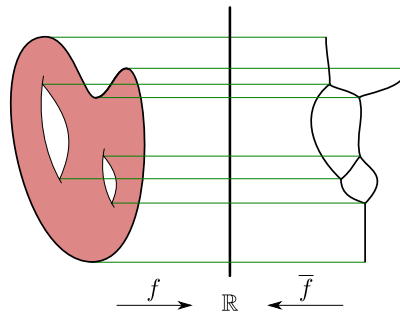
31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 461–475



Leibniz International Proceedings in Informatics

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



■ **Figure 1** A simple example of the Reeb graph (right) of a space (left). Here and in all other drawn examples in this paper, the real valued function is indicated by vertical height.

Given the popularity of the Reeb graph and related constructions for practical data analysis applications, it is desirable and increasingly necessary to understand how robust (stable) these structures are in the presence of noise. Consequently, several metrics for comparing Reeb graphs have been proposed recently. These include the functional distortion distance [1], the interleaving distance [6], and the combinatorial edit distance [8]. We note that the latter is limited to Reeb graphs resulting from Morse functions defined on surfaces. In addition, Morozov et. al proposed an interleaving distance for a simpler variant of the Reeb graph, the *merge tree* [17].

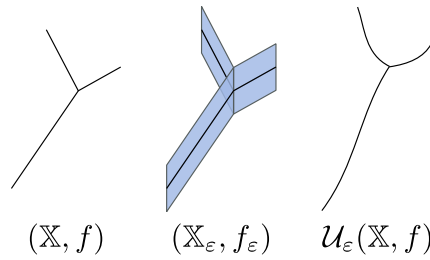
In this paper, we study the relation between two recently proposed distances for general Reeb graphs: the functional distortion distance of [1] and the interleaving distance of [6]. The former is based on concepts from metric geometry, and is defined by treating both graphs as metric spaces and inspecting continuous maps between them. The latter, on the other hand, is defined using ideas of category theory, utilizing the equivalence between Reeb graphs and a particular class of cosheaves. However, in essence, both construct a near-isomorphism between the two input graphs of study. In Sections 3 and 4, we explore this connection between the two distances, and show that indeed, the functional distortion distance and the interleaving distances are strongly equivalent on the space of Reeb graphs, meaning that they are within a constant factor of each other. This immediately leads to the bottleneck stability result for the Reeb graph interleaving distance.

2 Definitions

Given a topological space \mathbb{W} with a real valued function $f : \mathbb{W} \rightarrow \mathbb{R}$, we define the Reeb graph of (\mathbb{W}, f) as follows. We say that two points in \mathbb{W} are equivalent if they are in the same path component of a level set $f^{-1}(a)$ for $a \in \mathbb{R}$. This is denoted as $x \sim_f y$, or $x \sim y$ if the function is obvious. Then the Reeb graph is the quotient space \mathbb{W} / \sim_f . Note that the Reeb graph inherits a real valued function from its parent space. See Fig. 1 for an example.

2.1 Category of Reeb Graphs

For nice enough functions $f : \mathbb{W} \rightarrow \mathbb{R}$, such as Morse functions on compact manifolds or PL functions on finite simplicial complexes, the Reeb graph is, in fact, a finite graph [6]. We will tacitly make this assumption on the Reeb graph throughout the paper. Thus, we will define the category of Reeb graphs, following [6], intuitively to be finite graphs with real valued functions that are strictly monotonic on the edges. Morphisms will be given by function preserving maps between the underlying spaces as given in the following definition.



■ **Figure 2** An example of a smoothed Reeb graph. Shown on the left is the original graph \mathbb{X} , with the function f given by height. The middle space is $\mathbb{X}_\varepsilon = \mathbb{X} \times [-\varepsilon, \varepsilon]$ with the function $f_\varepsilon(x, t) = f(x) + t$ still given by height. On the right is the Reeb graph of $(\mathbb{X}_\varepsilon, f_\varepsilon)$, which is the smoothed Reeb graph $\mathcal{U}_\varepsilon(\mathbb{X})$.

► **Definition 1.** An object of the category **Reeb** is a finite graph, seen as a topological space \mathbb{X} (specifically, as a regular CW complex of dimension 1), together with a real valued function that is strictly monotonic on edges. This will equivalently be written as either $f : \mathbb{X} \rightarrow \mathbb{R}$ or (\mathbb{X}, f) . A morphism between (\mathbb{X}, f) and (\mathbb{Y}, g) is a function preserving map $\varphi : \mathbb{X} \rightarrow \mathbb{Y}$, i.e., the following diagram commutes:

$$\begin{array}{ccc}
 \mathbb{X} & \xrightarrow{\varphi} & \mathbb{Y} \\
 & \searrow f & \downarrow g \\
 & & \mathbb{R}
 \end{array}$$

Note that since we assume that the function is strictly monotonic when restricted to the edges, it is defined up to isomorphism by the values on the vertices. As an aside, notice that the quotient map sending a space with a function to its Reeb graph is an isomorphism when the space is in **Reeb**.

2.2 Interleaving Distance

Given a Reeb graph (\mathbb{X}, f) , let \mathbb{X}_ε denote the space $\mathbb{X} \times [-\varepsilon, \varepsilon]$, and define the ε -smoothing of (\mathbb{X}, f) as the Reeb graph of the function

$$\begin{aligned}
 f_\varepsilon : \mathbb{X}_\varepsilon &\rightarrow \mathbb{R}, \\
 (x, t) &\mapsto f(x) + t.
 \end{aligned}$$

That is, the ε -smoothing is the quotient space $\mathbb{X}_\varepsilon / \sim_{f_\varepsilon}$. Denote this space by $\mathcal{U}_\varepsilon(\mathbb{X}, f)$ and note that $\mathcal{U}_\varepsilon(\mathcal{U}_\varepsilon(\mathbb{X}, f)) \cong \mathcal{U}_{2\varepsilon}(\mathbb{X}, f)$ [6]. Sometimes when we are focusing on the underlying topological space and the function is obvious, we will denote this as $\mathcal{U}_\varepsilon(\mathbb{X})$. See Fig. 2 for an example.

An ε -interleaving of (\mathbb{X}, f) and (\mathbb{Y}, g) is a pair of function preserving maps (as in Definition 1) $\varphi : (\mathbb{X}, f) \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y}, g)$ and $\psi : (\mathbb{Y}, g) \rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f)$ with the following requirements. Consider the maps

$$\begin{aligned}
 \iota : (\mathbb{X}, f) &\rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f), & x &\mapsto [x, 0], \\
 \iota_\varepsilon : \mathcal{U}_\varepsilon(\mathbb{Y}, g) &\rightarrow \mathcal{U}_{2\varepsilon}(\mathbb{Y}, g), & [x, t] &\mapsto [x, t], \\
 \varphi_\varepsilon : \mathcal{U}_\varepsilon(\mathbb{X}, f) &\rightarrow \mathcal{U}_{2\varepsilon}(\mathbb{Y}, g), & [x, t] &\mapsto [\varphi(x), t],
 \end{aligned}$$

where $[x, t] = q(x, t)$ is the equivalence class of (x, t) under the quotient map $q : \mathbb{X}_\varepsilon \rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f)$.

Note that the diagram

$$\begin{array}{ccc} (\mathbb{X}, f) & \xrightarrow{\iota} & \mathcal{U}_\varepsilon(\mathbb{X}, f) \\ \varphi \downarrow & & \downarrow \varphi_\varepsilon \\ \mathcal{U}_\varepsilon(\mathbb{Y}, g) & \xrightarrow{\iota_\varepsilon} & \mathcal{U}_{2\varepsilon}(\mathbb{Y}, g) \end{array}$$

commutes. Analogously defining maps $\iota : (\mathbb{Y}, g) \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y}, g)$, $\iota_\varepsilon : \mathcal{U}_\varepsilon(\mathbb{X}, f) \rightarrow \mathcal{U}_{2\varepsilon}(\mathbb{X}, f)$, and $\psi_\varepsilon : \mathcal{U}_\varepsilon(\mathbb{Y}, g) \rightarrow \mathcal{U}_{2\varepsilon}(\mathbb{X}, f)$, we have the following definition of an ε -interleaving.

► **Definition 2** (ε -Interleaving). The maps $\varphi : (\mathbb{X}, f) \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y}, g)$ and $\psi : (\mathbb{Y}, g) \rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f)$ are an ε -interleaving if both of them are function preserving, and the following diagram

$$\begin{array}{ccccc} (\mathbb{X}, f) & \xrightarrow{\iota} & \mathcal{U}_\varepsilon(\mathbb{X}, f) & \xrightarrow{\iota_\varepsilon} & \mathcal{U}_{2\varepsilon}(\mathbb{X}, f) \\ & \searrow \varphi & \nearrow \varphi_\varepsilon & & \nearrow \varphi_\varepsilon \\ & & & & \\ & \swarrow \psi & \searrow \psi_\varepsilon & & \searrow \psi_\varepsilon \\ (\mathbb{Y}, g) & \xrightarrow{\iota} & \mathcal{U}_\varepsilon(\mathbb{Y}, g) & \xrightarrow{\iota_\varepsilon} & \mathcal{U}_{2\varepsilon}(\mathbb{Y}, g) \end{array}$$

commutes.

We can use this definition of interleavings to define a distance on Reeb graphs.

► **Definition 3** (Interleaving Distance, [6]). The *interleaving distance* between two Reeb graphs (\mathbb{X}, f) and (\mathbb{Y}, g) is defined to be

$$d_I((\mathbb{X}, f), (\mathbb{Y}, g)) = \inf \{ \varepsilon \mid \text{there exists an } \varepsilon\text{-interleaving between } (\mathbb{X}, f), (\mathbb{Y}, g) \}.$$

The definition of the interleaving distance was motivated by the cosheaf structure of Reeb graphs. It was shown in [6] that the category of Reeb graphs is equivalent to a particular class of cosheaves, which can be thought of as functors $\mathbf{F} : \mathbf{Int} \rightarrow \mathbf{Set}$ giving a set for each open interval. Specifically, given a real-valued function $f : \mathbb{X} \rightarrow \mathbb{R}$, we can construct the associated functor $\mathbf{F} = \pi_0 \circ f^{-1}$, where π_0 sends a topological space to its set of path components. This equivalence allows us to work with either the topological construction or the category theoretic one, whichever is easier or more appropriate. An excellent introduction to cellular cosheaves can be found in [5].

2.3 Functional Distortion Distance

For a given path π from u to v in $(\mathbb{X}, f) \in \mathbf{Reeb}$, we define the height of the path to be

$$\text{height}(\pi) = \max_{x \in \pi} f(x) - \min_{x \in \pi} f(x).$$

Then we define the distance

$$d_f(u, v) = \min_{\pi: u \rightsquigarrow v} \text{height}(\pi)$$

where π ranges over all paths from u to v in \mathbb{X} . Note that this can be equivalently defined by the minimum length of any closed interval I such that u and v are in the same path component of $f^{-1}(I)$.

The functional distortion distance between (\mathbb{X}, f) and (\mathbb{Y}, g) is now defined as follows:

► **Definition 4** (Functional Distortion Distance, [1]). Given $(\mathbb{X}, f), (\mathbb{Y}, g) \in \mathbf{Reeb}$ and maps $\Phi : \mathbb{X} \rightarrow \mathbb{Y}$ and $\Psi : \mathbb{Y} \rightarrow \mathbb{X}$, let

$$C(\Phi, \Psi) = \{(x, y) \in \mathbb{X} \times \mathbb{Y} \mid \Phi(x) = y \text{ or } x = \Psi(y)\}$$

and

$$D(\Phi, \Psi) = \sup_{\substack{(x,y), (x',y') \\ \in C(\Phi, \Psi)}} \frac{1}{2} |d_f(x, x') - d_g(y, y')|.$$

Then the *functional distortion distance* is defined to be

$$d_{FD}(f, g) = \inf_{\Phi, \Psi} \max\{D(\Phi, \Psi), \|f - g \circ \Phi\|_\infty, \|g - f \circ \Psi\|_\infty\}.$$

Note that since the maps Φ, Ψ are not required to preserve the function values, they are not necessarily Reeb graph morphisms in the sense of Definition 1.

2.4 Multivalued Maps and Continuous Selections

In order to prove our main result, we will make heavy use of the theory of multivalued maps and the notion of a selection of such a map. We briefly introduce the required definitions and a central result asserting the existence of a continuous selection.

A multivalued map (or multimap) $F : X \rightarrow Y$ is a relation $F \subseteq X \times Y$ that sends a point $x \in X$ to a nonempty set $F(x) = \{y \in Y \mid \exists x \in X : (x, y) \in F\} \subset Y$. A selection of a multimap is a map $f : X \rightarrow Y$ such that $f(x) \in F(x)$ for every $x \in X$. See [22] for an introduction to multimaps.

Note that using the axiom of choice, a selection always exists; the difficulty is in finding a continuous selection. The Michael selection theorem gives a criterion for a multimap to have a continuous selection. However, in order to state it, we will need several definitions.

► **Definition 5.** A family \mathcal{S} of subsets of a topological space Y is *equi-locally n -connected* if for every $S \in \mathcal{S}$, every $y \in S$, and every neighborhood W of y , there is a neighborhood V of y such that $V \subset W$ and for every $S' \in \mathcal{S}$ such that $V \cap S' \neq \emptyset$, every continuous mapping of the m -sphere \mathbb{S}^m into $S' \cap V$ is null-homotopic in $S' \cap W$ for $m \leq n$. This is denoted by $\mathcal{S} \in \text{ELC}^n$.

In particular, we will be requiring the case where $\mathcal{S} \in \text{ELC}^0$. A sufficient condition for this to hold is that in the above definition, V can be chosen such that for any $S' \in \mathcal{S}$, the intersection $S' \cap V$ is either empty or path connected.

► **Definition 6.** A multivalued map $F : X \rightarrow Y$ is *lower semicontinuous (LSC)* if for every open set $U \subset Y$ the set $F^{-1}(U) = \{x \in X \mid F(x) \cap U \neq \emptyset\}$ is open in X .

Finally we can state the Michael selection theorem. Since we are working with a space of covering dimension 1, we paraphrase the more general theorem here to relate it to our context.

► **Theorem 7** (Michael 1956[16]). *A multivalued mapping $F : X \rightarrow Y$ admits a continuous single-valued selection provided that the following conditions are satisfied:*

1. X is a paracompact space with covering dimension $\dim(X) \leq 1$;
2. Y is a completely metrizable space;
3. F is an LSC mapping;
4. for every $x \in X$, $F(x)$ is a path connected subspace of Y ; and
5. the family of values $\{F(x)\}_{x \in X}$ is ELC^0 .

3 ε -Interleaving and Functional Distortion

In order to prove the main result, Theorem 16, we will prove each inequality separately as Lemmas 8 and 15 .

3.1 The Easy Direction

► **Lemma 8.** *Let $(\mathbb{X}, f), (\mathbb{Y}, g) \in \text{Reeb}$. Then*

$$d_I(f, g) \leq d_{FD}(f, g).$$

Proof. Let $\varepsilon > d_{FD}(f, g)$. By definition of the functional distortion metric, there are maps

$$\mathbb{X} \begin{array}{c} \xrightarrow{\Phi} \\ \xleftarrow{\Psi} \end{array} \mathbb{Y}$$

that satisfy the requirements of Definition 4. In particular, x and $\Psi \circ \Phi(x)$ are connected by a path γ of height 2ε . This path is thus contained in the preimage $f^{-1}[f(x) - 2\varepsilon, f(x) + 2\varepsilon]$. As a consequence, the points $(x, 0)$ and $(\Psi \circ \Phi(x), f(x) - f(\Psi \circ \Phi(x)))$ are in the same path component of the level set $f_{2\varepsilon}^{-1}(f(x))$.

Define

$$\begin{aligned} \varphi : (\mathbb{X}, f) &\rightarrow \mathcal{U}_\varepsilon(\mathbb{Y}, g), & x &\mapsto [\Phi(x), f(x) - g(\Phi(x))], \\ \psi : (\mathbb{Y}, g) &\rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f), & y &\mapsto [\Psi(y), g(y) - f(\Psi(y))], \end{aligned}$$

with the latter inducing the map

$$\psi_\varepsilon : \mathcal{U}_\varepsilon(\mathbb{Y}, g) \rightarrow \mathcal{U}_{2\varepsilon}(\mathbb{X}, f), \quad [y, t] \mapsto [\Psi(y), g(y) - f(\Psi(y)) + t]$$

appearing in the definition of an interleaving. A visual representation of the map φ is given in Figure 3. We then have

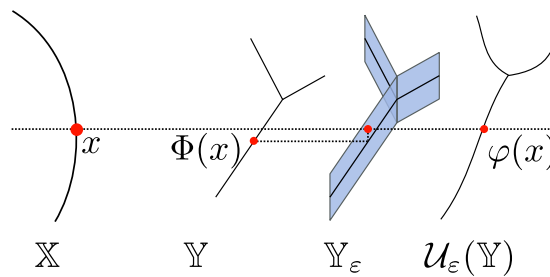
$$\begin{aligned} \psi_\varepsilon \circ \varphi(x) &= \psi_\varepsilon[\Phi(x), f(x) - g(\Phi(x))] \\ &= [\Psi \circ \Phi(x), g \circ \Phi(x) - f(\Psi \circ \Phi(x)) + f(x) - g(\Phi(x))] \\ &= [\Psi \circ \Phi(x), f(x) - f(\Psi \circ \Phi(x))] \\ &= [x, 0] = \iota_\varepsilon \circ \iota(x). \end{aligned}$$

By an analogous argument, we also have $\varphi_\varepsilon \circ \psi(y) = [y, 0] = \iota_\varepsilon \circ \iota(y)$, and hence φ and ψ are an ε -interleaving. Since the above holds for any $\varepsilon > d_{FD}(f, g)$, the claim is now immediate. ◀

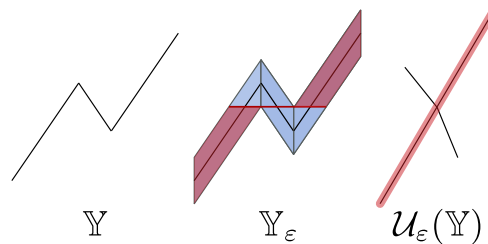
3.2 The Hard Direction

In order to show $d_{FD}((\mathbb{X}, f), (\mathbb{Y}, g)) \leq 3d_I((\mathbb{X}, f), (\mathbb{Y}, g))$, we need to start with an ε -interleaving, $\varphi : (\mathbb{X}, f) \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y}, g)$ and $\psi : (\mathbb{Y}, g) \rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f)$, and construct a pair of maps satisfying the requirements of the functional distortion distance. To do this, note that the map φ induces a multimap $\bar{\varphi} : \mathbb{X} \rightarrow \mathbb{Y}_\varepsilon$, which sends a point x to the entire equivalence class of $\varphi(x)$, thought of as a subset of \mathbb{Y}_ε . Concretely, letting $q : \mathbb{Y}_\varepsilon \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y})$ denote the Reeb graph quotient map, we have $\bar{\varphi} = q^{-1} \circ \varphi$.

This multimap, however, does not always have a continuous selection (see Figure 4 for a counterexample), so we will introduce a parameter δ to slightly enlarge the images of



■ **Figure 3** The definition of the map $\varphi : (\mathbb{X}, f) \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y}, g)$ as given in the proof of Lemma 8.



■ **Figure 4** The map $\bar{\varphi}$ is not enough for us to have a continuous selection as seen in this counterexample. The image under $\varphi : \mathbb{X} \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y})$ is the red line in the rightmost graph. However, this implies the image under $\bar{\varphi}$ is the red region in the middle space. Since with $\bar{\varphi}$, a selection may only choose one point from every level, we run into a problem in the center line since no choice of point will allow for a continuous selection.

$\bar{\varphi}$. First, note that we have metrics d_f and d_g for \mathbb{X} and \mathbb{Y} respectively. For an arbitrarily small $\delta > 0$, we can construct the multimap, $\bar{\varphi}_\delta : \mathbb{X} \rightarrow \mathbb{Y}_\varepsilon$ sending x to $\bar{\varphi}(B_\delta(x))$, where $B_\delta(x) = \{x' \mid d_f(x, x') < \delta\}$. Explicitly, we have

$$\bar{\varphi}_\delta(x) = \{(y', t') \in \mathbb{Y}_\varepsilon \mid x' \in \mathbb{X}, d_f(x, x') < \delta, (y', t') \in \bar{\varphi}(x')\}.$$

See Fig. 5 for an example. For technical reasons, we will assume that $\delta < L/4$, where L is the minimum height of any edge in $\mathcal{U}_\varepsilon(\mathbb{Y})$.

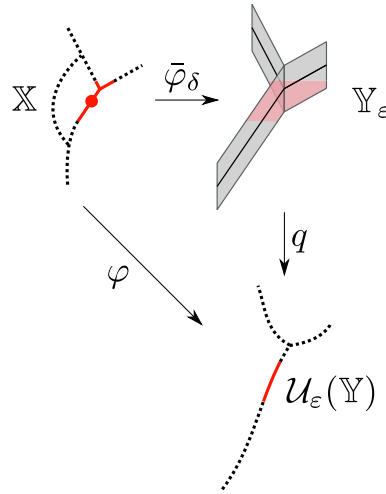
In order to assert the existence of a continuous selection, we now show that the multimap $\bar{\varphi}_\delta : \mathbb{X} \rightarrow \mathbb{Y}_\varepsilon$ satisfies the assumptions of Theorem 7:

1. Since \mathbb{X} is a finite CW complex, it is compact and thus trivially paracompact. In addition, because it is a graph, it has covering dimension 1.
2. Since \mathbb{Y} is a finite CW complex, it is completely metrizable. Therefore, \mathbb{Y}_ε is also completely metrizable, being the product of two completely metrizable spaces.
3. To show that $\bar{\varphi}_\delta$ is LSC, let $U \subset \mathbb{Y}_\varepsilon$ be open. We will show that any $x \in \bar{\varphi}_\delta^{-1}(U)$ has an open neighborhood in $\bar{\varphi}_\delta^{-1}(U)$, implying that $\bar{\varphi}_\delta^{-1}(U)$ is open. Expanding the definition of $x \in \bar{\varphi}_\delta^{-1}(U)$, there is an x' with $d_f(x, x') < \delta$ such that $\bar{\varphi}(x') \cap U \neq \emptyset$. Let $r = \delta - d_f(x, x')$. We now want to show that $B_r(x) \subseteq \bar{\varphi}_\delta^{-1}(U)$. Let $x'' \in B_r(x)$. We know that $x' \in B_\delta(x'')$ since

$$d_f(x', x'') \leq d_f(x', x) + d_f(x, x'') < (\delta - r) + r = \delta.$$

Since $\bar{\varphi}(x') \cap U \neq \emptyset$ and $x' \in B_\delta(x'')$, we must have $\bar{\varphi}_\delta(x'') \cap U = \bar{\varphi}(B_\delta(x'')) \cap U \neq \emptyset$ and hence $x'' \in \bar{\varphi}_\delta^{-1}(U)$.

4. Let $q : \mathbb{Y}_\varepsilon \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y})$ be the quotient map. Then $q \circ \bar{\varphi}_\delta(x) = \varphi(B_\delta(x))$ is the image of a path component under a continuous map and is therefore path connected. Since $\varphi(x) \subset \mathcal{U}_\varepsilon(\mathbb{Y})$



■ **Figure 5** An example for determining the map $\bar{\varphi}_\delta$. Given the red point $x \in \mathbb{X}$, the red solid region in \mathbb{X} is $B_\delta(x)$. Then we can look at $\varphi(B_\delta(x))$, the red region in $\mathcal{U}_\varepsilon(\mathbb{Y})$. The set $\bar{\varphi}_\delta(x)$ in \mathbb{Y}_ε consists of the points which map into $\varphi(B_\delta(x))$ in $\mathcal{U}_\varepsilon(\mathbb{Y})$ under the quotient map q .

is by definition the image of a path component of \mathbb{X} , it is also path connected. So $\bar{\varphi}_\delta(x)$ can be thought of as a fibration with base space $\varphi(B_\delta(x))$ and fibers $\bar{\varphi}(x')$, for $x' \in B_\delta(x)$. Since the fibers are path connected by definition of q , and the base is path connected, the total space is path connected.

5. As checking this property is by far the most complicated, we prove it in Lemma 9.

► **Lemma 9.** *The family of values $\{\bar{\varphi}_\delta(x)\}_{x \in \mathbb{X}}$ is ELC⁰.*

Proof. Fix $x \in \mathbb{X}$. Given an arbitrary $(y, t) \in \bar{\varphi}_\delta(x) \subset \mathbb{Y}_\varepsilon$ and a neighborhood W of (y, t) , let $0 < r \leq \delta$ be such that $V = B_r(y, t)$ is contained in W . Here $B_r(y, t)$ denotes the open ball of radius r around (y, t) in \mathbb{Y}_ε using the metric

$$d_{\mathbb{Y}_\varepsilon}((y, t), (y', t')) = d_g(y, y') + |t - t'|.$$

It suffices to show that for any \tilde{x} such that $\bar{\varphi}_\delta(\tilde{x}) \cap V \neq \emptyset$, the set $\bar{\varphi}_\delta(\tilde{x}) \cap V$ is path connected. For brevity, let $U = \bar{\varphi}_\delta(\tilde{x})$. Let (y_1, t_1) and (y_2, t_2) be in the intersection $U \cap V$ and, seeking a contradiction, assume that they are in different path components of $U \cap V$. Since U is path connected, there is a path γ_1 from (y_1, t_1) to (y_2, t_2) with $\text{Im } \gamma_1 \subset U = \bar{\varphi}_\delta(\tilde{x})$. Thus for every $s \in [0, 1]$ there is an $x_s \in B_\delta(\tilde{x})$ such that $\gamma_1(s) \in \bar{\varphi}(x_s)$. The map $\bar{\varphi}$ is function preserving, so $g_\varepsilon(\gamma_1(s)) = f(x_s)$. Moreover, as $x_s \in B_\delta(\tilde{x})$, we have $|f(\tilde{x}) - g_\varepsilon(\gamma_1(s))| < \delta$ and thus $\text{height}(\gamma_1) < 2\delta$. On the other hand, V is path connected and so there is a path γ_2 from (y_2, t_2) to (y_1, t_1) that stays completely inside of $V = B_r(y, t)$, and thus $\text{height}(\gamma_2) < 2r$.

We can now consider the paths $q \circ \gamma_1$ and $q \circ \gamma_2$ in $\mathcal{U}_\varepsilon(\mathbb{Y})$. As the endpoints of γ_2 are in different path components of $U \cap V$ and at the same time $\text{Im } \gamma_2 \subset V$, there must be a point $v \in \text{Im } \gamma_2$ that is not in U .

We want to show that $q(v) \notin q(\text{Im } \gamma_1) \subset q(U)$. By definition, $\bar{\varphi}$ is the map such that $q \circ \bar{\varphi}(z) = \varphi(z)$ for any $z \in \mathbb{X}$. Thus

$$q(U) = q \circ \bar{\varphi}_\delta(\tilde{x}) = q \circ \bar{\varphi}(B_\delta(\tilde{x})) = \varphi(B_\delta(\tilde{x})).$$

Again seeking a contradiction, assume $q(v) \in q(U) = \varphi(B_\delta(\tilde{x}))$. Then there is an $x_v \in B_\delta(\tilde{x})$ such that $\varphi(x_v) = q(v)$. But this implies that $v \in \bar{\varphi}(x_v)$ and thus $v \in \bar{\varphi}_\delta(\tilde{x}) = U$,

contradicting our assumption that $v \notin U$. We conclude that $q(v) \notin q(U)$; in particular, $q(v) \notin q(\text{Im } \gamma_1)$.

This implies that the loop $q(\gamma_1 \bullet \gamma_2)$ is not nullhomotopic in $\mathcal{U}_\varepsilon(\mathbb{Y})$, where $\gamma_1 \bullet \gamma_2$ denotes the concatenation of the two paths. However, since we assumed that $r \leq \delta < L/4$, where L is the minimum height of any edge in $\mathcal{U}_\varepsilon(\mathbb{Y})$, we have

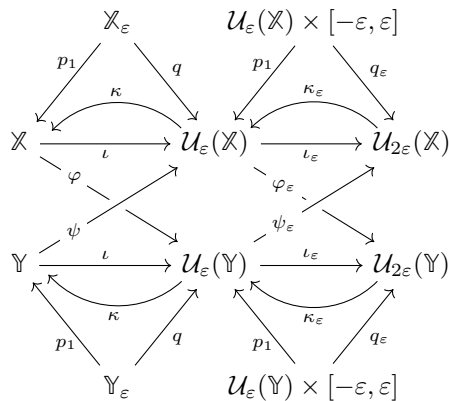
$$\begin{aligned} \text{height}(q(\gamma_1 \bullet \gamma_2)) &= \text{height}(\gamma_1 \bullet \gamma_2) \\ &\leq \text{height}(\gamma_1) + \text{height}(\gamma_2) \\ &< 2\delta + 2r \leq 4\delta < L, \end{aligned}$$

and therefore must be nullhomotopic in $\mathcal{U}_\varepsilon(\mathbb{Y})$. Thus, the original assumption that $\bar{\varphi}_\delta(\tilde{x}) \cap V$ is not path connected must be false. \blacktriangleleft

Thus, since $\bar{\varphi}$ satisfies the requirements for Theorem 7, there exists a continuous selection of $\bar{\varphi}_\delta$, that is, a map $\tilde{\varphi}_\delta : \mathbb{X} \rightarrow \mathbb{Y}_\varepsilon$ satisfying $\tilde{\varphi}_\delta(x) \in \bar{\varphi}(B_\delta(x))$ for all $x \in \mathbb{X}$. Likewise, there exists a continuous selection $\tilde{\psi}_\delta : \mathbb{Y} \rightarrow \mathbb{X}_\varepsilon$ for $\bar{\psi}_\delta$. Note however that the functional distortion distance requires a pair of maps $\mathbb{X} \rightarrow \mathbb{Y}$ and $\mathbb{Y} \rightarrow \mathbb{X}$. To get there, let p_1 be either the map $\mathbb{X}_\varepsilon \rightarrow \mathbb{X}$ or $\mathbb{Y}_\varepsilon \rightarrow \mathbb{Y}$, defined by projection onto the first factor. We define our maps for the functional distortion distance to be $\Phi = p_1 \circ \tilde{\varphi}_\delta : \mathbb{X} \rightarrow \mathbb{Y}$ and $\Psi = p_1 \circ \tilde{\psi}_\delta : \mathbb{Y} \rightarrow \mathbb{X}$. Note that Φ and Ψ depend on the choice of δ . The remainder of this section is devoted to showing that this pair of maps induces a functional distortion of at most $3(\varepsilon + \delta)$, establishing the upper bound on the functional distortion distance.

Bounding the functional distortion

In order to prove the main result of this section, we need to establish some notation and technical lemmas. Recall that $\iota : \mathbb{X} \rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f)$ is the map that sends x to $[x, 0] = q(x, 0)$. Moreover, let $\kappa = p_1 \circ q^{-1}$. Note that both ι^{-1} and κ are multimaps, and $\iota^{-1} \subseteq \kappa$ as relations of $\mathbb{X} \times \mathcal{U}_\varepsilon(\mathbb{X}, f)$. Similarly, define $\iota_\varepsilon : \mathcal{U}_\varepsilon(\mathbb{X}, f) \rightarrow \mathcal{U}_{2\varepsilon}(\mathbb{X}, f)$ and $\kappa_\varepsilon = p_1 q_\varepsilon^{-1}$, where $q_\varepsilon : \mathcal{U}_\varepsilon(\mathbb{X}, f) \times [-\varepsilon, \varepsilon] \rightarrow \mathcal{U}_{2\varepsilon}(\mathbb{X}, f)$ is the quotient map. We have analogous maps for (\mathbb{Y}, g) in place of (\mathbb{X}, f) , for which we use the same identifiers while ensuring that their domains will always be clear from the context. These maps are summarized in the following diagram. Note that not all parts of the diagram commute.



For $t \in \mathbb{R}$ and $s \geq 0$, let

$$I_s(t) := \{r \in \mathbb{R} \mid |r - t| \leq s\}$$

denote the thickening of t by s . Given any point $x \in \mathbb{X}$, we define

$$R_r(x) := \{x' \in \mathbb{X} \mid \exists \text{ path } \pi : x \rightsquigarrow x' \text{ such that } f(\text{Im } \pi) \subseteq I_r(f(x))\}.$$

That is, $R_r(x)$ is the path component of x in $f^{-1}(I_r(f(x)))$. For a subset $U \subseteq \mathbb{X}$, we define $R_r(U) := \cup_{x \in U} R_r(x)$. We can define R_r similarly for \mathbb{Y} , $\mathcal{U}_\varepsilon(\mathbb{X})$, or $\mathcal{U}_\varepsilon(\mathbb{Y})$. The following simple observations will be useful later; we omit the easy proof.

► **Lemma 10.** (i) $B_r(x) \subseteq R_r(x) \subseteq B_{2r}(x)$. (ii) $R_r(R_s(x)) \subseteq R_{r+s}(x)$.

We will now present several technical lemmas that establish how far the above diagram is from commuting.

► **Lemma 11.** $\kappa \circ R_r \circ \iota \subseteq R_{r+\varepsilon}$.

Proof. Given $x \in \mathbb{X}$, let $[\tilde{x}, \tilde{t}] \in R_r(\iota(x))$. We want to show that there exists a path $\pi : x \rightsquigarrow \tilde{x}$ such that $f(\text{Im } \pi) \subseteq I_{r+\varepsilon}(f(x))$. An analogous argument also holds for $y \in \mathbb{Y}$.

Since $f_\varepsilon(\iota(x)) = f(x)$ and $[\tilde{x}, \tilde{t}] \in R_r(\iota(x))$, there is a path γ from $\iota(x) = [x, 0]$ to $[\tilde{x}, \tilde{t}]$ in $\mathcal{U}_\varepsilon(\mathbb{X})$ satisfying $f_\varepsilon(\text{Im } \gamma) \subseteq I_r(f(x))$. Because the image of γ is path connected and q induces an isomorphism on path components, the subspace $q^{-1}(\text{Im } \gamma) \subset \mathbb{X}_\varepsilon$ is path connected as well. In particular, $(x, 0)$ and (\tilde{x}, \tilde{t}) are in this set, so there is a path ζ between them. As $\text{Im } \zeta \subseteq q^{-1}(\text{Im } \gamma)$, we have that $f_\varepsilon(\text{Im } \zeta) \subseteq f_\varepsilon(q^{-1}(\text{Im } \gamma)) = f_\varepsilon(\text{Im } \gamma) \subseteq I_r(f(x))$.

Finally, consider the path $\pi = p_1 \circ \zeta$ in \mathbb{X} from x to \tilde{x} . Since the projection p_1 changes the function value by at most ε , we have that $f(\text{Im } \pi) \subseteq I_{r+\varepsilon}(f(x))$, and thus $\tilde{x} \in R_{r+\varepsilon}(x)$. ◀

Note that the previous lemma can also be stated using ι_ε , so we have $\kappa_\varepsilon \circ R_r \circ \iota_\varepsilon \subseteq R_{r+\varepsilon}$. Since this lemma holds for $r = 0$ as well, this also implies that $\kappa \circ \iota \subseteq R_\varepsilon$.

► **Lemma 12.** $\psi \circ \kappa \subseteq \kappa_\varepsilon \circ \psi_\varepsilon$.

Proof. Let $y_\varepsilon = [y, s] \in \mathcal{U}_\varepsilon(\mathbb{Y})$. Note that $y \in \kappa[y, s]$ and thus $\psi(y) \in \psi \circ \kappa[y, s]$. Since every element of $\psi \circ \kappa(y_\varepsilon)$ can be represented in this form, it suffices to show that $\psi(y) \in \kappa_\varepsilon \circ \psi_\varepsilon(y_\varepsilon)$ as well. To see this, note that by definition of ψ_ε we have $\psi_\varepsilon[y, s] = [\psi(y), s]$. Moreover, we have $\psi(y) \in \kappa_\varepsilon[\psi(y), s]$, so the claim follows. ◀

► **Lemma 13.** $\Psi \circ \Phi \in R_{2\varepsilon+2\delta}$.

Proof. By definition of Φ and Ψ , for any $x \in \mathbb{X}$ we have

$$\begin{aligned} \Phi(x) &= p_1 \circ \tilde{\varphi}_\delta(x) \\ &\in p_1 \circ \tilde{\varphi}(B_\delta(x)) \\ &= p_1 \circ q^{-1} \circ \varphi(B_\delta(x)) \\ &= \kappa \circ \varphi \circ B_\delta(x), \end{aligned}$$

and similarly for any $y \in \mathbb{Y}$ we have

$$\Psi(y) \in \kappa \circ \psi \circ B_\delta(y).$$

The composition yields

$$\Psi \circ \Phi(x) \in \kappa \circ \psi \circ B_\delta \circ \kappa \circ \varphi \circ B_\delta(x).$$

Since ψ preserves function values, a path γ in \mathbb{Y} is sent to a path $\psi \circ \gamma$ of the same height in $\mathcal{U}_\varepsilon(\mathbb{X})$. Thus, for any $r \geq 0$, we have $\psi \circ B_r \subseteq B_r \circ \psi$, and so we obtain:

$$\begin{aligned}
 \Psi \circ \Phi(x) &\in \kappa \circ \psi \circ B_\delta \circ \kappa \circ \varphi \circ B_\delta(x) \\
 &\subseteq \kappa \circ (B_\delta \circ \psi) \circ \kappa \circ \varphi \circ B_\delta(x) && \text{since } \psi \circ B_r \subseteq B_r \circ \psi, \\
 &\subseteq \kappa \circ B_\delta \circ (\kappa_\varepsilon \circ \psi_\varepsilon) \circ \varphi \circ B_\delta(x) && \text{since } \psi \circ \kappa \subseteq \kappa_\varepsilon \circ \psi_\varepsilon \text{ by Lemma 12,} \\
 &\subseteq \kappa \circ B_\delta \circ \kappa_\varepsilon \circ (\iota_\varepsilon \circ \iota) \circ B_\delta(x) && \text{by the definition of an interleaving,} \\
 &\subseteq \kappa \circ B_\delta \circ (R_\varepsilon) \circ \iota \circ B_\delta(x) && \text{since } \kappa_\varepsilon \circ \iota_\varepsilon \subseteq R_\varepsilon \text{ by Lemma 11,} \\
 &\subseteq \kappa \circ (R_{\delta+\varepsilon}) \circ \iota \circ B_\delta(x) && \text{since } B_\delta \circ R_\varepsilon \subseteq R_{\delta+\varepsilon} \text{ by Lemma 10,} \\
 &\subseteq (R_{\delta+2\varepsilon}) \circ B_\delta(x) && \text{since } \kappa \circ R_{\delta+\varepsilon} \circ \iota \subseteq R_{\delta+2\varepsilon} \text{ by Lemma 11,} \\
 &\subseteq R_{2\delta+2\varepsilon}(x) && \text{since } R_{2\varepsilon} \circ B_\delta \subseteq R_{\delta+2\varepsilon} \text{ by Lemma 10.} \quad \blacktriangleleft
 \end{aligned}$$

► **Lemma 14.** (i) $\|f - g \circ \Phi\|_\infty \leq \varepsilon + \delta$. (ii) $\|g - f \circ \Psi\|_\infty \leq \varepsilon + \delta$.

Proof. For any $x \in \mathbb{X}$, the image $\Phi(x)$ is a point in \mathbb{Y} such that there is a $\tilde{x} \in \mathbb{X}$ with $d_f(x, \tilde{x}) < \delta$ and a $t \in [-\varepsilon, \varepsilon]$ with $(\Phi(x), t) \in \bar{\varphi}(\tilde{x})$. So $|f(x) - f(\tilde{x})| < \delta$ and $f(\tilde{x}) = g(\Phi(x)) + t$. Thus

$$|f(x) - g(\Phi(x))| = |f(x) - (f(\tilde{x}) - t)| = |f(x) - f(\tilde{x}) + t| \leq \delta + \varepsilon$$

and hence $\|f - g \circ \Phi\|_\infty \leq \varepsilon + \delta$. Likewise, $\|g - f \circ \Psi\|_\infty \leq \varepsilon + \delta$. ◀

Finally, we can prove the main result of this section.

► **Lemma 15.** Let $f : \mathbb{X} \rightarrow \mathbb{R}$ and $g : \mathbb{Y} \rightarrow \mathbb{R}$. Then

$$d_{FD}(f, g) \leq 3d_I(f, g).$$

Proof. Let $\varphi : (\mathbb{X}, f) \rightarrow \mathcal{U}_\varepsilon(\mathbb{Y}, g)$ and $\psi : (\mathbb{Y}, g) \rightarrow \mathcal{U}_\varepsilon(\mathbb{X}, f)$ be an ε -interleaving, and thus $d_I(f, g) \leq \varepsilon$. As shown above, there exist continuous maps $\Phi : \mathbb{X} \rightarrow \mathbb{Y}$ and $\Psi : \mathbb{Y} \rightarrow \mathbb{X}$, constructed from selections for the multimaps $\bar{\varphi}_\delta$ and $\bar{\psi}_\delta$. Let $(x, y), (x', y') \in C(\Phi, \Psi)$. There are two cases to consider; either the pairs are of the same type (e.g., $(x, \Phi(x))$ and $(x', \Phi(x'))$), or they are different.

First assume that they are of the same type, $(x, \Phi(x))$ and $(x', \Phi(x'))$. Let γ be a minimum height path in \mathbb{X} from x to x' . Then $\Phi(\gamma)$ is a path in \mathbb{Y} from $\Phi(x)$ to $\Phi(x')$. Since $\|f - g \circ \Phi\|_\infty \leq \varepsilon + \delta$, the height of $\Phi(\gamma)$ exceeds the height of γ by at most $2(\varepsilon + \delta)$. So

$$\begin{aligned}
 d_g(\Phi(x), \Phi(x')) &\leq \text{height}(\Phi(\gamma)) \\
 &\leq \text{height}(\gamma) + 2(\varepsilon + \delta) \\
 &= d_f(x, x') + 2(\varepsilon + \delta).
 \end{aligned} \tag{1}$$

Conversely, to get an upper bound for $d_f(x, x')$ in terms of $d_g(\Phi(x), \Phi(x'))$, let ζ be a minimum height path in \mathbb{Y} between $\Phi(x)$ and $\Phi(x')$, i.e., $\text{height}(\zeta) = d_g(\Phi(x), \Phi(x'))$. Note that $\Psi \circ \zeta$ is a path in \mathbb{X} from $\Psi \circ \Phi(x)$ to $\Psi \circ \Phi(x')$. Since $\|g - f \circ \Psi\|_\infty \leq \varepsilon + \delta$ (Lemma 14), we have that

$$f(\Psi(\zeta)) \subseteq I_{\varepsilon+\delta}(g(\text{Im } \zeta)), \tag{2}$$

where $I_s(A) := \{r \in \mathbb{R} \mid \exists r' \in A : |r - r'| \leq s\}$ denotes the thickening of an interval $A \subseteq \mathbb{R}$ by a real number $s \geq 0$. Since $g(\Phi(x)), g(\Phi(x')) \in g(\text{Im } \zeta)$, we conclude from Lemma 14 that both $f(x)$ and $f(x')$ are contained in $I_{\varepsilon+\delta}(g(\text{Im } \zeta))$. Now consider the path $\hat{\gamma} = \gamma_1 \bullet \gamma_2 \bullet \gamma_3$ in

\mathbb{X} connecting x to x' , where γ_1 is a minimum height path in \mathbb{X} from x to $\Psi \circ \Phi(x)$, $\gamma_2 = \Psi \circ \zeta$ connects $\Psi \circ \Phi(x)$ to $\Psi \circ \Phi(x')$ as described above, and γ_3 is a minimum height path in \mathbb{X} connecting $\Psi \circ \Phi(x')$ to x' . Combining Lemma 13 and (2), we obtain:

$$\begin{aligned} f(\text{Im } \hat{\gamma}) &\subseteq f(\text{Im } \gamma_1) \cup f(\text{Im } \gamma_2) \cup f(\text{Im } \gamma_3) \\ &\subseteq I_{2\varepsilon+2\delta}(f(x)) \cup I_{\varepsilon+\delta}(g(\text{Im } \zeta)) \cup I_{2\varepsilon+2\delta}(f(x')) \\ &\subseteq I_{3\varepsilon+3\delta}(g(\text{Im } \zeta)) \cup I_{\varepsilon+\delta}(g(\text{Im } \zeta)) \cup I_{3\varepsilon+3\delta}(g(\text{Im } \zeta)) \\ &= I_{3\varepsilon+3\delta}(g(\text{Im } \zeta)). \end{aligned}$$

We thus conclude

$$d_f(x, x') \leq \text{height}(\hat{\gamma}) \leq d_g(\Phi(x), \Phi(x')) + 6\varepsilon + 6\delta. \tag{3}$$

Combining the two bounds (1) and (3), we obtain

$$|d_f(x, x') - d_g(\Phi(x), \Phi(x'))| \leq 6(\varepsilon + \delta).$$

Analogously, if we are given two pairs $(\Psi(y), y), (\Psi(y'), y') \in C(\Phi, \Psi)$, we can show that

$$|d_f(\Psi(y), \Psi(y')) - d_g(y, y')| \leq 6(\varepsilon + \delta).$$

What remains to consider is the case of two pairs $(x, \Phi(x)), (\Psi(y), y) \in C(\Phi, \Psi)$. Let ξ be a minimum height path in \mathbb{Y} between $\Phi(x)$ and y . By Lemma 14, $\pi_1 = \Psi \circ \xi$ is a path $\Psi(y)$ to $\Psi \circ \Phi(x)$ in \mathbb{X} such that

$$f(\pi_1) \subseteq I_{\varepsilon+\delta}(g(\text{Im } \xi)).$$

Since $g(\Phi(x)) \in g(\text{Im } \xi)$, we also have $f(x) \in I_{\varepsilon+\delta}(g(\text{Im } \xi))$. Now let π_2 be a minimum height path in \mathbb{X} connecting x to $\Psi \circ \Phi(x)$; by Lemma 13 we have $f(\pi_2) \subseteq I_{2\varepsilon+2\delta}(f(x))$. Concatenating the two, we obtain a path $\pi = \pi_1 \bullet \pi_2$ from x to $\Psi(y)$ such that

$$\begin{aligned} f(\text{Im } \pi) &\subseteq f(\text{Im } \pi_1) \cup f(\text{Im } \pi_2) \\ &\subseteq I_{\varepsilon+\delta}(g(\text{Im } \xi)) \cup I_{2\varepsilon+2\delta}(f(x)) \\ &\subseteq I_{\varepsilon+\delta}(g(\text{Im } \xi)) \cup I_{3\varepsilon+3\delta}(g(\text{Im } \xi)) \\ &= I_{3\varepsilon+3\delta}(g(\text{Im } \xi)). \end{aligned}$$

We conclude that

$$d_f(x, \Psi(y)) \leq d_g(\Phi(x), y) + 6\varepsilon + 6\delta.$$

Likewise, by a symmetric argument, we can show that

$$d_g(\Phi(x), y) \leq d_f(x, \Psi(y)) + 6\varepsilon + 6\delta.$$

Hence $|d_f(x, \Psi(y)) - d_g(\Phi(x), y)| \leq 6(\varepsilon + \delta)$.

Combining all of these bounds gives

$$D(\Phi, \Psi) = \sup_{\substack{(x,y),(x',y') \\ \in C(\Phi,\Psi)}} \frac{1}{2} |d_f(x, x') - d_g(y, y')| \leq 3(\varepsilon + \delta).$$

and therefore, together with Lemma 14,

$$d_{FD}(f, g) = \inf_{\Phi, \Psi} \max\{D(\Phi, \Psi), \|f - g \circ \Phi\|_\infty, \|g - f \circ \Psi\|_\infty\} \leq 3(\varepsilon + \delta).$$

Since the above holds for any $\varepsilon > d_I(f, g)$ and for any $\delta > 0$, this completes the proof. ◀

Putting together Lemmas 8 and 15, our main result is immediate.

► **Theorem 16.** *The functional distortion metric and the interleaving metric are strongly equivalent. That is, given any Reeb graphs (\mathcal{X}, f) and (\mathcal{Y}, g) ,*

$$d_I(f, g) \leq d_{FD}(f, g) \leq 3d_I(f, g).$$

4 Relationship Between the Interleaving and Bottleneck Distances

Having strongly equivalent metrics means that we can quickly pass back and forth many of the properties associated to the metrics. For example, the bottleneck stability bound for persistence diagrams in terms of the functional distortion distance [1] says the following (for the definitions of the persistence diagrams $Dg_0(f)$, $ExDg_1(f)$ associated to a function f and of the bottleneck distance d_B we refer the reader to [10]):

► **Theorem 17** (Bauer, Ge, Wang [1]). *Given two Reeb graphs (\mathcal{X}, f) and (\mathcal{Y}, g) ,*

$$d_B(Dg_0(f), Dg_0(g)) \leq d_{FD}(f, g)$$

and

$$d_B(ExDg_1(f), ExDg_1(g)) \leq 3d_{FD}(f, g).$$

Combining this result with Theorem 16 gives an immediate stability result relating the interleaving distance with the bottleneck distance.

► **Corollary 18.** *Given two Reeb graphs (\mathcal{X}, f) and (\mathcal{Y}, g) ,*

$$d_B(Dg_0(f), Dg_0(g)) \leq 3d_I(f, g)$$

and

$$d_B(ExDg_1(f), ExDg_1(g)) \leq 9d_I(f, g).$$

5 Discussion

In this paper, we study the relationship between two existing distances for Reeb graphs, and show that they are strongly equivalent on the set of Reeb graphs. This relationship will be a powerful tool for understanding convergence properties of the different metrics. For example, if we have a Cauchy sequence in one metric, we have a Cauchy sequence in the other and can therefore pass completeness results back and forth. This relationship also means that algorithms for approximation of the metrics can be written using whichever method is most helpful and applicable to the context.

These two distances may in general not be the same. However, we have yet to find an example for which it can be shown that the two distances are actually different. It is easy to construct examples where the bound $d_I(f, g) \leq d_{FD}(f, g)$ of Lemma 8 is tight; the status of the bound $d_{FD}(f, g) \leq 3d_I(f, g)$ of Lemma 15 is unclear. While that bound is obtained using an arbitrary selection, a better bound may be achievable using a particular optimal selection. In addition, this may shed light on whether the bounds given between the bottleneck distance of the extended persistence diagrams and the two Reeb graph distances are tight. Finally, we plan to explore the use of these distances for studying the stability of Reeb-like structures, such as Mapper and α -Reeb graphs [24, 4].

References

- 1 Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring distance between Reeb graphs. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, New York, NY, USA, 2014. ACM.
- 2 Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1-3):5–22, February 2008.
- 3 Kevin Buchin, Maike Buchin, Marc van Kreveld, Bettina Speckmann, and Frank Staals. Trajectory grouping structure. In Frank Dehne, Roberto Solis-Oba, and Jörg-Rüdiger Sack, editors, *Algorithms and Data Structures*, volume 8037 of *Lecture Notes in Computer Science*, pages 219–230. Springer Berlin Heidelberg, 2013.
- 4 Frédéric Chazal and Jian Sun. Gromov-Hausdorff approximation of filament structure using Reeb-type graph. In *Proceedings of the Thirtieth Annual Symposium on Computational Geometry*, SOCG'14, pages 491:491–491:500, New York, NY, USA, 2014. ACM.
- 5 Justin Curry. *Sheaves, Cosheaves and Applications*. PhD thesis, University of Pennsylvania, December 2014.
- 6 Vin de Silva, Elizabeth Munch, and Amit Patel. Categorized Reeb graphs, January 2015.
- 7 Tamal K. Dey, Fengtao Fan, and Yusu Wang. An efficient computation of handle and tunnel loops via Reeb graphs. *ACM Trans. Graph.*, 32(4):32:1–32:10, July 2013.
- 8 Barbara Di Fabio and Claudia Landi. The edit distance for Reeb graphs of surfaces, November 2014. <http://arxiv.org/abs/1411.1544>.
- 9 Harish Doraiswamy and Vijay Natarajan. Output-Sensitive construction of Reeb graphs. *Visualization and Computer Graphics, IEEE Transactions on*, 18(1):146–159, January 2012.
- 10 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. Amer. Math. Soc., Providence, Rhode Island, 2010.
- 11 Francisco Escolano, Edwin R. Hancock, and Silvia Biasotti. Complexity fusion for indexing Reeb digraphs. In Richard Wilson, Edwin Hancock, Adrian Bors, and William Smith, editors, *Computer Analysis of Images and Patterns*, volume 8047 of *Lecture Notes in Computer Science*, pages 120–127. Springer Berlin Heidelberg, 2013.
- 12 Xiaoyin Ge, Issam I. Safa, Mikhail Belkin, and Yusu Wang. Data skeletonization via Reeb graphs. *Advances in Neural Information Processing Systems*, 24:837–845, 2011.
- 13 William Harvey, Yusu Wang, and Rephael Wenger. A randomized $O(m \log m)$ time algorithm for computing Reeb graphs of arbitrary simplicial complexes. In *Proceedings of the Twenty Sixth Annual Symposium on Computational Geometry*, SoCG'10, pages 267–276, New York, NY, USA, 2010. ACM.
- 14 Franck Hétroy and Dominique Attali. Topological quadrangulations of closed triangulated surfaces using the Reeb graph. *Graphical Models*, 65(1-3):131–148, May 2003.
- 15 Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Toshiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, SIGGRAPH'01, pages 203–212, New York, NY, USA, 2001. ACM.
- 16 Ernest Michael. Continuous selections II. *Annals of Mathematics*, 64(3):pp. 562–580, 1956.
- 17 Dmitriy Morozov, Kenes Beketayev, and Gunther Weber. Interleaving distance between merge trees. Manuscript, 2013.
- 18 Mattia Natali, Silvia Biasotti, Giuseppe Patanè, and Bianca Falcidieno. Graph-based representations of point clouds. *Graphical Models*, 73(5):151–164, September 2011.
- 19 Monica Nicolau, Arnold J. Levine, and Gunnar Carlsson. Topology based data analysis identifies a subgroup of breast cancers with a unique mutational profile and excellent survival. *Proceedings of the National Academy of Sciences*, 108(17):7265–7270, 2011.

- 20 Salman Parsa. A deterministic $O(m \log m)$ time algorithm for the Reeb graph. *Discrete & Computational Geometry*, 49(4):864–878, 2013.
- 21 Georges Reeb. Sur les points singuliers d’une forme de Pfaff complètement intégrable ou d’une fonction numérique. *Comptes Rendus de L’Académie ses Séances*, 222:847–849, 1946.
- 22 Dušan Repovš and Pavel V. Semenov. *Continuous Selections of Multivalued Mappings*. Kluwer Academic Publishers, 1998.
- 23 Yoshihisa Shinagawa, Toshiyasu L. Kunii, and Yannick L. Kergosien. Surface coding based on Morse theory. *IEEE Comput. Graph. Appl.*, 11(5):66–78, September 1991.
- 24 Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. Topological methods for the analysis of high dimensional data sets and 3D object recognition. In *Eurographics Symposium on Point-Based Graphics*, 2007.
- 25 Zoë Wood, Hugues Hoppe, Mathieu Desbrun, and Peter Schröder. Removing excess topology from isosurfaces. *ACM Transactions on Graphics*, 23(2):190–208, April 2004.
- 26 Yuan Yao, Jian Sun, Xuhui Huang, Gregory R. Bowman, Gurjeet Singh, Michael Lesnick, Leonidas J. Guibas, Vijay S. Pande, and Gunnar Carlsson. Topological methods for exploring low-density states in biomolecular folding pathways. *The Journal of Chemical Physics*, 130:144115, 2009.

On Generalized Heawood Inequalities for Manifolds: A Van Kampen–Flores-type Nonembeddability Result*

Xavier Goaoc¹, Isaac Mabillard², Pavel Paták³, Zuzana Patáková⁴, Martin Tancer^{4,2}, and Uli Wagner²

1 LIGM, Université Paris-Est Marne-la-Vallée, France

2 IST Austria, Klosterneuburg, Austria

3 Department of Algebra, Charles University, Czech Republic

4 Department of Applied Mathematics, Charles University, Czech Republic

Abstract

The fact that the complete graph K_5 does not embed in the plane has been generalized in two independent directions. On the one hand, the solution of the classical *Heawood problem* for graphs on surfaces established that the complete graph K_n embeds in a closed surface M if and only if $(n-3)(n-4) \leq 6b_1(M)$, where $b_1(M)$ is the first \mathbb{Z}_2 -Betti number of M . On the other hand, Van Kampen and Flores proved that the k -skeleton of the n -dimensional simplex (the higher-dimensional analogue of K_{n+1}) embeds in \mathbb{R}^{2k} if and only if $n \leq 2k+2$.

Two decades ago, Kühnel conjectured that the k -skeleton of the n -simplex embeds in a compact, $(k-1)$ -connected $2k$ -manifold with k th \mathbb{Z}_2 -Betti number b_k only if the following *generalized Heawood inequality* holds: $\binom{n-k-1}{k+1} \leq \binom{2k+1}{k+1} b_k$. This is a common generalization of the case of graphs on surfaces as well as the Van Kampen–Flores theorem.

In the spirit of Kühnel’s conjecture, we prove that if the k -skeleton of the n -simplex embeds in a $2k$ -manifold with k th \mathbb{Z}_2 -Betti number b_k , then $n \leq 2b_k \binom{2k+2}{k} + 2k + 5$. This bound is weaker than the generalized Heawood inequality, but does not require the assumption that M is $(k-1)$ -connected. Our proof uses a result of Volovikov about maps that satisfy a certain homological triviality condition.

1998 ACM Subject Classification G. Mathematics of Computing, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Heawood Inequality, Embeddings, Van Kampen–Flores, Manifolds

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.476

1 Introduction

Given a closed surface M , it is a natural question to determine the maximum integer n such that the complete graph K_n can be embedded (drawn without crossings) into M (e.g., $n = 4$ if $M = S^2$ is the 2-sphere, and $n = 7$ if M is a torus). This classical problem was raised in the late 19th century by Heawood [9] and Heffter [10] and completely settled in the

* The work by Z. P. was partially supported by the Charles University Grant SVV-2014-260103. The work by Z. P. and M. T. was partially supported by the project CE-ITI (GACR P202/12/G061) of the Czech Science Foundation and by the ERC Advanced Grant No. 267165. Part of the research work of M. T. was conducted at IST Austria, supported by an *IST Fellowship*. The work by U. W. was partially supported by the Swiss National Science Foundation (grants SNSF-200020-138230 and SNSF-PP00P2-138948).



1950–60’s through a sequence of works by Gustin, Guy, Mayer, Ringel, Terry, Welch, and Youngs (see [22, Ch. 1] for a discussion of the history of the problem and detailed references). Heawood already observed that if K_n embeds into M then

$$(n - 3)(n - 4) \leq 6b_1(M) = 12 - 6\chi(M), \quad (1)$$

where $\chi(M)$ is the Euler characteristic of M and $b_1(M) = 2 - \chi(M)$ is the first \mathbb{Z}_2 -Betti number of M , i.e., the dimension of the first homology group $H_1(M; \mathbb{Z}_2)$ (here and throughout the paper, we work with homology with \mathbb{Z}_2 -coefficients).¹ Conversely, for surfaces M other than the Klein bottle, the inequality is tight, i.e., K_n embeds into M if and only if (1) holds; this is a hard result, the bulk of the monograph [22] is devoted to its proof. (The exceptional case, the Klein bottle, has $b_1 = 2$, but does not admit an embedding of K_7 , only of K_6 .)

The question naturally generalizes to higher dimension: Let $\Delta_n^{(k)}$ denote the k -skeleton of the n -simplex, the natural higher-dimensional generalization of $K_{n+1} = \Delta_n^{(1)}$ (by definition $\Delta_n^{(k)}$ has $n + 1$ vertices and every subset of at most $k + 1$ vertices form a face). Given a $2k$ -dimensional manifold M , what is the largest n such that $\Delta_n^{(k)}$ embeds (topologically) into M ? This line of enquiry started in the 1930’s when Van Kampen [23] and Flores [5] showed that $\Delta_{2k+2}^{(k)}$ does not embed into \mathbb{R}^{2k} (the case $k = 1$ corresponding to the non-planarity of K_5). Somewhat surprisingly, little else seems to be known, and the following conjecture of Kühnel [12, Conjecture B] regarding a *generalized Heawood inequality* remains unresolved:

► **Conjecture 1** (Kühnel). *Let $n, k \geq 1$ be integers. If $\Delta_n^{(k)}$ embeds in a compact, $(k - 1)$ -connected $2k$ -manifold M with k th \mathbb{Z}_2 -Betti number $b_k(M)$ then*

$$\binom{n - k - 1}{k + 1} \leq \binom{2k + 1}{k + 1} b_k(M) \quad (2)$$

The classical Heawood inequality (1) and the Van Kampen–Flores Theorem correspond to the special cases $k = 1$ and $b_k = 0$, respectively. Kühnel states Conjecture 1 in slightly different form in terms of Euler characteristic of M rather than $b_k(M)$. Our formulation is an equivalent form. The \mathbb{Z}_2 -coefficients are not important in the statement of the conjecture but they are convenient for our further progress.

New result. Here, we prove an estimate in the spirit of the generalized Heawood inequality (2), with a quantitatively weaker bound. Note that our bound holds (at no extra cost) under weaker hypotheses.

A somewhat technical but useful relaxation is that instead of embeddings, we consider the following slightly more general notion (which also helps with setting up our proof method). Let K be a finite simplicial complex and let $|K|$ be its underlying space (geometric

¹ The inequality (1), which by a direct calculation is equivalent to $n \leq c(M) := \lfloor (7 + \sqrt{1 + \beta_1(M)})/2 \rfloor$, is closely related to the *Map Coloring Problem* for surfaces (which is the context in which Heawood originally considered the question). Indeed, it turns out that for surfaces M other than the Klein bottle, $c(M)$ is the maximum chromatic number of any graph embeddable into M . For $M = S^2$ the 2-sphere (i.e., $b_1(M) = 0$), this is the *Four-Color Theorem* [1, 2]; for other surfaces (i.e., $b_1(M) > 0$) this was originally stated (with an incomplete proof) by Heawood and is now known as the *Map Color Theorem* or *Ringel–Youngs Theorem* [22]. Interestingly, for surfaces $M \neq S^2$, there is a fairly short proof, based on edge counting and Euler characteristic, that the chromatic number of any graph embeddable into M is at most $c(M)$ (see [22, Thms. 4.2 and 4.8]). The hard part of the proof of the Ringel–Youngs Theorem is to show that for every M (except for the Klein bottle) $K_{c(M)}$ embeds into M .

realization). We define an *almost-embedding* of K into a (Hausdorff) topological space X to be a continuous map $f : |K| \rightarrow X$ such that any two disjoint simplices $\sigma, \tau \in K$ have disjoint images, $f(\sigma) \cap f(\tau) = \emptyset$. We stress that the condition for being an almost-embedding depends on the actual simplicial complex (the triangulation), not just the underlying space. That is, if K and L are two different complexes with $|K| = |L|$ then a map $f : |K| = |L| \rightarrow X$ may be an almost-embedding of K into X but not an almost-embedding of L into X . Note also that every embedding is an almost-embedding as well. Our main result is as follows:

► **Theorem 2.** *Let $n, k \geq 1$ be integers. If $\Delta_n^{(k)}$ almost-embeds into a $2k$ -manifold M with k th \mathbb{Z}_2 -Betti number $b_k(M)$, then*

$$n \leq 2 \binom{2k+2}{k} b_k(M) + 2k + 5. \quad (3)$$

As remarked above, this bound is weaker than the conjectured generalized Heawood inequality (2) and is clearly not optimal (as we already see in the special cases $k = 1$ or $b_k = 0$). On the other hand, apart from applying more generally to almost-embeddings, the hypotheses of Theorem 2 are weaker than those of Conjecture 1 in that we do not assume the manifold M to be $(k-1)$ -connected. We conjecture that this connectedness assumption is not necessary for Conjecture 1, i.e., that (2) holds whenever $\Delta_n^{(k)}$ almost-embeds into a $2k$ -manifold M . The intuition is that $\Delta_n^{(k)}$ is $(k-1)$ -connected and therefore the image of an almost-embedding cannot “use” any parts of M on which nontrivial homotopy classes of dimension less than k are supported.

Previous work. The following special case of Conjecture 1 was proved by Kühnel [12, Thm. 2] (and served as a motivation for the general conjecture): Suppose that P is an n -dimensional simplicial convex polytope, and that there is a subcomplex of the boundary ∂P of P that is k -Hamiltonian (i.e., that contains the k -skeleton of P) and that is a triangulation of M , a $2k$ -dimensional manifold. Then inequality (2) holds. To see that this is indeed a special case of Conjecture 1, note that ∂P is a *piecewise linear (PL)* sphere of dimension $n-1$, i.e., ∂P is combinatorially isomorphic to some subdivision of $\partial\Delta_n$ (and, in particular, $(n-2)$ -connected). Therefore, the k -skeleton of P , and hence M , contains a subdivision of $\Delta_n^{(k)}$ and is $(k-1)$ -connected.

In this special case and for $n \geq 2k+2$, equality in (2) is attained if and only if P is a simplex. More generally, equality is attained whenever M is a triangulated $2k$ -manifold on $n+1$ vertices that is $k+1$ -neighborly (i.e., any subset of at most $k+1$ vertices form a face, in which case $\Delta_n^{(k)}$ is a subcomplex of M). Some examples of $(k+1)$ -neighborly $2k$ -manifolds are known, e.g., for $k=1$ (the so-called *regular cases* of equality for the Heawood inequality [22]), for $k=2$ [15, 14] (e.g., a 3-neighborly triangulation of the complex projective plane) and for $k=4$ [3], but in general, a characterization of the higher-dimensional cases of equality for (2) (or even of those values of the parameters for which equality is attained) seems rather hard (which is maybe not surprising, given how difficult the construction of examples of equality is already for $k=1$).

Proof technique. Our proof of Theorem 2 strongly relies on a different generalization of the Van Kampen–Flores Theorem, due to Volovikov [24], regarding maps into general manifolds but under an additional homological triviality condition:

► **Theorem 3 (Volovikov).** *Let M be a $2k$ -dimensional manifold and let $f : |\Delta_{2k+2}^{(k)}| \rightarrow M$ be a continuous map such that the induced homomorphism $f_* : H_k(\Delta_{2k+2}^{(k)}; \mathbb{Z}_2) \rightarrow H_k(M; \mathbb{Z}_2)$ is trivial. Then f is not an almost-embedding, i.e., there exist two disjoint simplices $\sigma, \tau \in \Delta_{2k+2}^{(k)}$ such that $f(\sigma) \cap f(\tau) \neq \emptyset$.*

Note that the homological triviality condition is automatically satisfied if $H_k(M; \mathbb{Z}_2) = 0$, e.g., if $M = \mathbb{R}^{2k}$ or $M = S^{2k}$. On the other hand, without the homological triviality condition, the assertion is in general not true for other manifolds (e.g., K_5 embeds into every closed surface different from the sphere, or $\Delta_8^{(2)}$ embeds into the complex projective plane).

Theorem 3 is only a special of the main result in [24]; it is obtained by setting $j = q = 2$, $m = 2k$, $s = k + 1$ and $N = 2k + 2$ in item 3 of Volovikov’s main result (beware that k from Volovikov’s condition “there exists a natural number k ” is different from our k).

In addition, Volovikov [24] formulates the triviality condition in terms of cohomology, i.e., he requires that $f^* : H^k(M; \mathbb{Z}_2) \rightarrow H^k(\Delta_{2k+2}^{(k)}; \mathbb{Z}_2)$ is trivial. However, since we are working with field coefficients and the (co)homology groups in question are finitely generated, the homological triviality condition (which is more convenient for us to work with) and the cohomological one are equivalent.²

The key idea of our approach is to show that if n is large enough and f is a mapping from $\Delta_n^{(k)}$ to M , then there is an almost-embedding g from $\Delta_s^{(k)}$ to $|\Delta_n^{(k)}|$ for some prescribed value of s such that the composed map $f \circ g : \Delta_s \rightarrow M$ satisfies Volovikov’s condition. More specifically, the following is our main technical lemma:

► **Lemma 4.** *Let $k, s \geq 1$ and $b \geq 0$ be integers. There exists a value $n_0 := n_0(k, b, s)$ with the following property. Let $n \geq n_0$ and let f be a mapping of $|\Delta_n^{(k)}|$ into a manifold M with k th \mathbb{Z}_2 -Betti number at most b . Then there exists a subdivision D of $\Delta_s^{(k)}$ and a simplicial map $g_{\text{simp}} : D \rightarrow \Delta_n^{(k)}$ with the following properties.*

1. *The induced map on the geometric realizations $g : |D| \rightarrow |\Delta_n^{(k)}|$ is an almost-embedding from $\Delta_s^{(k)}$ to $|\Delta_n^{(k)}|$ (note that $|D| = |\Delta_s^{(k)}|$).*
2. *The homomorphism $(f \circ g)_* : H_k(\Delta_s^{(k)}) \rightarrow H_k(M)$ is trivial (see Section 2 below for the precise interpretation of $(f \circ g)_*$).*

The value n_0 can be taken as $\binom{s}{k}b(s - 2k) + 2s - 2k + 1$.

Therefore, if $s \geq 2k + 2$, then $f \circ g$ cannot be an almost-embedding by Volovikov’s theorem. We deduce that f is not an almost-embedding either, and Theorem 2 immediately follows. This deduction requires the following lemma as in general, a composition of two almost-embeddings needs not be an almost-embedding.

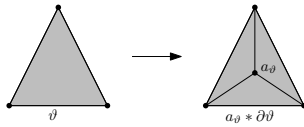
► **Lemma 5.** *Let K and L be simplicial complexes and X a topological space. Suppose g is an almost-embedding of K into $|L|$ and f is an almost-embedding of L into X . Then $f \circ g$ is an almost-embedding of K into X , provided that g is the realization of a simplicial map g_{simp} from some subdivision K' of K to L .*

We prove Lemma 4 in Section 4 thus completing the proof of Theorem 2. Before that, in Section 3 we first present a simpler version of that proof that introduces the main ideas in a simpler setting, and yields a weaker bound for n_0 (see Equation(4)). Further related questions and problems will be discussed in Section 5.

² More specifically, by the Universal Coefficient Theorem [21, 53.5], $H_k(\cdot; \mathbb{Z}_2)$ and $H^k(\cdot; \mathbb{Z}_2)$ are dual vector spaces, and f^* is the adjoint of f_* , hence triviality of f_* implies that of f^* . Moreover, if the homology group $H_k(X; \mathbb{Z}_2)$ of a space X is finitely generated (as is the case for both $\Delta_n^{(k)}$ and M , by assumption) then it is (non-canonically) isomorphic to its dual vector space $H^k(X; \mathbb{Z}_2)$. Therefore, f_* is trivial if and only if f^* is.

2 Preliminaries

We begin by fixing some terminology and notation. We will use $\text{card}(U)$ to denote the cardinality of a set U .



We also recall that the *stellar subdivision* of a maximal face ϑ in a simplicial complex K is obtained by removing ϑ from K and adding a cone $a_{\vartheta} * (\partial\vartheta)$, where a_{ϑ} is a newly added vertex, the apex of the cone (see the figure on the left).

Throughout this paper we only work with homology groups and Betti numbers over \mathbb{Z}_2 , and for simplicity, we will for the most part drop the coefficient group \mathbb{Z}_2 from the notation. Moreover, we will need to switch back and forth between singular and simplicial homology. More precisely, if K is a simplicial complex then $H_*(K)$ will mean the simplicial homology of K , whereas $H_*(X)$ will mean the singular homology of a topological space X . In particular, $H_*(|K|)$ denotes the singular homology of the underlying space $|K|$ of a complex K . We use analogous conventions for $C_*(K), C_*(X)$ and $C_*(|K|)$ on the level of chains, and likewise for the subgroups of cycles and boundaries, respectively.³ Given a cycle c , we denote by $[c]$ the homology class it represents.

A mapping $h: |K| \rightarrow X$ induces a chain map $h_{\#}^{\text{sing}}: C_*(|K|) \rightarrow C_*(X)$ on the level of singular chains; see [8, Chapter 2.1]. There is also a canonical chain map $\iota_K: C_*(K) \rightarrow C_*(|K|)$ inducing the isomorphism of $H_*(K)$ and $H_*(|K|)$, see again [8, Chapter 2.1]. We define $h_{\#}: C_*(K) \rightarrow C_*(X)$ as $h_{\#} := h_{\#}^{\text{sing}} \circ \iota_K$. The three chain maps mentioned above also induce maps $h_*^{\text{sing}}, (\iota_K)_*$, and h_* on the level of homology satisfying $h_* = h_*^{\text{sing}} \circ (\iota_K)_*$.

We also need a technical lemma saying that our maps compose, in a right way, on the level of homology.

► **Lemma 6.** *Let K and L be simplicial complexes and X a topological space. Let j_{simp} be a simplicial map for K to L , $j: |K| \rightarrow |L|$ be the continuous map induced by j_{simp} and $h: |L| \rightarrow X$ be another continuous map. Then $h_* \circ (j_{\text{simp}})_* = (h \circ j)_*$ where $(j_{\text{simp}})_*: H_*(K) \rightarrow H_*(L)$ is the map induced by j_{simp} on the level of simplicial homology and h_* and $(h \circ j)_*$, as explained above.*

3 Proof of Lemma 4 with a weaker bound on n_0

Let k, b, s be fixed integers. We consider a $2k$ -manifold M with k th Betti number b , a map $f: |\Delta_n^{(k)}| \rightarrow M$. The strategy of our proof of Lemma 4 is to start by designing an auxiliary chain map

$$\varphi: C_*\left(\Delta_s^{(k)}\right) \rightarrow C_*\left(\Delta_n^{(k)}\right).$$

that behaves as an almost-embedding, in the sense that whenever σ and σ' are disjoint k -faces of Δ_s , $\varphi(\sigma)$ and $\varphi(\sigma')$ have disjoint supports, and such that for every $(k+1)$ -face τ of Δ_s the homology class $[(f_{\#} \circ \varphi)(\partial\tau)]$ is trivial. We then use φ to design a subdivision D of $\Delta_s^{(k)}$ and a simplicial map $g_{\text{simp}}: D \rightarrow \Delta_n^{(k)}$ that induces a map $g: |D| \rightarrow |\Delta_n^{(k)}|$ with the

³ We remark that throughout this paper, we will only work with spaces that are either (underlying spaces of) simplicial complexes or topological manifolds. Such spaces are homotopy equivalent to CW complexes [20, Corollary 1], and so on the matter of homology, it does not really matter which (ordinary, i.e., satisfying the dimension axiom) homology theory we use as they are all naturally equivalent for CW complexes [8, Thm. 4.59]. However the distinction between the simplicial and the singular setting will be relevant on the level of chains.

desired properties: g is an almost-embedding and $(f \circ g)_*([\partial\tau])$ is trivial for all $(k + 1)$ -faces τ of Δ_s . Since the cycles $\partial\tau$, for $(k + 1)$ -faces τ of Δ_s , generate all k -cycles of $\Delta_s^{(k)}$, this implies that $(f \circ g)_*$ is trivial.

The purpose of this section is to give a first implementation of the above strategy that proves Lemma 4 with a bound of

$$n_0 \geq \left(\binom{s+1}{k+1} - 1 \right) 2^{b(s+1)} + s + 1. \tag{4}$$

In Section 4 we then improve this bound to $\binom{s}{k}b(s - 2k) + 2s - 2k + 1$ at the cost of some technical complications.

Throughout the rest of this paper we use the following notations. We let $\{v_1, v_2, \dots, v_{n+1}\}$ denote the set of vertices of Δ_n and we assume that Δ_s is the induced subcomplex of Δ_n on $\{v_1, v_2, \dots, v_{s+1}\}$. We let $U = \{v_{s+2}, v_{s+3}, \dots, v_{n+1}\}$ denote the set of vertices of Δ_n unused by Δ_s . We let $m = \binom{s+1}{k+1}$ and denote by $\sigma_1, \sigma_2, \dots, \sigma_m$ the k -faces of Δ_s .

3.1 Construction of φ

For every face ϑ of Δ_s of dimension at most $k - 1$ we set $\varphi(\vartheta) = \vartheta$. We then “route” each σ_i by mapping it to its stellar subdivision with an apex $u \in U$, i.e. by setting $\varphi(\sigma_i)$ to $\sigma_i + z(\sigma_i, u)$ where $z(\sigma_i, u)$ denotes the cycle $\partial(\sigma_i \cup \{u\})$. The picture on the left shows the case $k = 1$, the support of $z(\sigma_i, u)$ is dashed on the left, and the support of the resulting $\varphi(\sigma_i)$ is on the right.

We ensure that φ behave as an almost-embedding by using a different apex $u \in U$ for each σ_i . The difficulty is to choose these m apices in a way that $[f_{\#}(\varphi(\partial\tau))]$ is trivial for every $(k + 1)$ -face τ of Δ_s . To that end we associate to each $u \in U$ the sequence

$$\mathbf{v}(u) := ([f_{\#}(z(\sigma_1, u))], [f_{\#}(z(\sigma_2, u))], \dots, [f_{\#}(z(\sigma_m, u))]) \in H_k(M)^m,$$

and we denote by $\mathbf{v}_i(u)$ the i th element of $\mathbf{v}(u)$. We work with \mathbb{Z}_2 -homology, so $H_k(M)^m$ is finite; more precisely, its cardinality equals 2^{bm} . From $n \geq n_0 = (m - 1)2^{bm} + s + 1$ we get that $\text{card}(U) \geq (m - 1)\text{card}(H_k(M)^m) + 1$. The pigeonhole principle then guarantees that there exist m distinct vertices u_1, u_2, \dots, u_m of U such that $\mathbf{v}(u_1) = \mathbf{v}(u_2) = \dots = \mathbf{v}(u_m)$. We use u_i to “route” σ_i and put

$$\varphi(\sigma_i) := \sigma_i + z(\sigma_i, u_i). \tag{5}$$

We finally extend φ linearly to $C_* \left(\Delta_s^{(k)} \right)$.

► **Lemma 7.** φ is a chain map and $[f_{\#}(\varphi(\partial\tau))] = 0$ for every $(k + 1)$ -face $\tau \in \Delta_s$.

Before proving the lemma, we establish a simple claim that will be also useful later on.

► **Claim 8.** Let τ be a $(k + 1)$ -face of Δ_s and let $u \in U$. Let $\sigma_{i_1}, \dots, \sigma_{i_{k+2}}$ be all the k -faces of τ . Then

$$\partial\tau + z(\sigma_{i_1}, u) + z(\sigma_{i_2}, u) + \dots + z(\sigma_{i_{k+2}}, u) = 0. \tag{6}$$

Proof. This follows from expanding the equation $0 = \partial^2(\tau \cup \{u\})$. ◀

Proof of Lemma 7. The map φ is the identity on ℓ -chains with $\ell \leq k - 1$ and Equation (5) immediately implies that $\partial\varphi(\sigma) = \partial\sigma$ for every k -simplex σ . It follows that φ is a chain map.

Now let τ be a $(k + 1)$ -simplex of Δ_s and let $\sigma_{i_1}, \dots, \sigma_{i_{k+2}}$ be its k -faces. We have

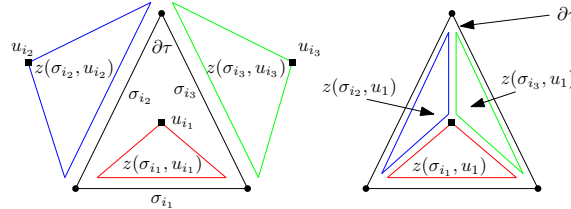
$$f_{\#} \circ \varphi(\partial\tau) = f_{\#} \left(\sum_{j=1}^{k+2} \sigma_{i_j} + z(\sigma_{i_j}, u_{i_j}) \right) = f_{\#}(\partial\tau) + \sum_{j=1}^{k+2} f_{\#}(z(\sigma_{i_j}, u_{i_j})).$$

The u_i 's are chosen in such a way that the homology class $[f_{\#}(z(\sigma_{i_j}, u_{i_j}))] = \mathbf{v}_{i_j}(u_{i_j})$ is independent of the value ℓ . When passing to the homology classes in the above identity, we can therefore replace each u_{i_j} with u_1 , and obtain,

$$[f_{\#} \circ \varphi(\partial\tau)] = [f_{\#}(\partial\tau)] + \sum_{j=1}^{k+2} [f_{\#}(z(\sigma_{i_j}, u_1))] = \left[f_{\#} \left(\partial\tau + \sum_{j=1}^{k+2} z(\sigma_{i_j}, u_1) \right) \right].$$

This class is trivial by Claim 8.

Here is the idea behind the proof with $k = 1$ and $u_{i_1} = u_1$ (same colors represent same homology classes; the class on the right is trivial, because each edge appears twice):



3.2 Construction of D and g

The definition of φ , in particular Equation (5), suggests to construct our subdivision D of $\Delta_s^{(k)}$ by simply replacing every k -face of $\Delta_s^{(k)}$ by its stellar subdivision. Let a_i denote the new vertex introduced when subdividing σ_i .

We define a simplicial map $g_{\text{simp}}: D \rightarrow \Delta_n^{(k)}$ by putting $g_{\text{simp}}(v) = v$ for every original vertex v of $\Delta_s^{(k)}$, and $g_{\text{simp}}(a_i) = u_i$ for $i \in [m]$. This g_{simp} induces a map $g: |\Delta_s^{(k)}| \rightarrow |\Delta_n^{(k)}|$ on the geometric realizations. Since the u_i 's are pairwise distinct, g is an embedding⁴, so Condition 1 of Lemma 4 holds.

On principle, we would like to derive Condition 2 of Lemma 4 by observing that g ‘induces’ a chain map from $C_*(\Delta_s^{(k)})$ to $C_*(\Delta_n^{(k)})$ that coincides with φ . Making this a formal statement is thorny because g , as a continuous map, naturally induces a chain map $g_{\#}$ on singular rather than simplicial chains. We can’t use directly g_{simp} either, since we are interested in a map from $C_*(\Delta_s^{(k)})$ and not from $C_*(D)$.

We handle this technicality as follows. Let $\rho: C_*(\Delta_s^{(k)}) \rightarrow C_*(D)$ be the chain map that sends each simplex ϑ of $\Delta_s^{(k)}$ to the sum of simplices of D of the same dimension that subdivide it. This map induces an isomorphism ρ_* in homology, and $\varphi = (g_{\text{simp}})_{\#} \circ \rho_*$ where $(g_{\text{simp}})_{\#}: C_*(D) \rightarrow C_*(\Delta_n^{(k)})$ denotes the (simplicial) chain map induced by g_{simp} . We thus have in homology

$$f_* \circ \varphi_* = f_* \circ (g_{\text{simp}})_* \circ \rho_*$$

⁴ We use the full strength of almost-embeddings when proving Lemma 4 with the better bound on n_0 .

and since ρ_* is an isomorphism and $f_* \circ \varphi_*$ is trivial, Lemma 7 yields that $f_* \circ (g_{\text{simp}})_*$ is also trivial. Since $f_* \circ (g_{\text{simp}})_* = (f \circ g)_*$ by Lemma 6, $(f \circ g)_*$ is trivial as well. This concludes the proof of Lemma 4 with the weaker bound.

4 Proof of Lemma 4

We now prove Lemma 4 with the bound claimed in the statement, namely

$$n_0 = \binom{s}{k} b(s - 2k) + 2s - 2k + 1.$$

Let k, b, s be fixed integers. We consider a $2k$ -manifold M with k th Betti number b , a map $f : |\Delta_n^{(k)}| \rightarrow M$, and we assume that $n \geq n_0$.

The proof follows the same strategy as in Section 3 : we construct a chain map $\varphi : C_*(\Delta_s^{(k)}) \rightarrow C_*(\Delta_n^{(k)})$ such that the homology class $[(f_{\#} \circ \varphi)(\partial\tau)]$ is trivial for all $(k + 1)$ -faces τ of Δ_s , then upgrade φ to a continuous map $g : |\Delta_s^{(k)}| \rightarrow |\Delta_n^{(k)}|$ with the desired properties.

When constructing φ , we refine the arguments of Section 3 to “route” each k -face using not only one, but several vertices from U ; this makes finding “collisions” easier, as we can use linear algebra arguments instead of the pigeonhole principle. This comes at the cost that when upgrading g , we must content ourselves with proving that it is an almost-embedding. This is sufficient for our purpose and has an additional benefit: the same group of vertices from U may serve to route several k -faces provided they pairwise intersect in $\Delta_k^{(s)}$.

4.1 Construction of φ

We use the same notation regarding v_1, \dots, v_{n+1} , Δ_n , Δ_s , U , $m = \binom{s+1}{k+1}$ and $\sigma_1, \sigma_2, \dots, \sigma_m$ as in Section 3.

Definition of multipoints and the map \mathbf{v} . As we said we plan to route k -faces of Δ_s through collections of vertices from U , we will call these collections multipoints. It turns out that this is useful for our needs only if these multipoints have an odd cardinality. In order to easily proceed with later computations, we define multipoints as vectors rather than subsets of U as below.

Let $C_0(U)$ denote the \mathbb{Z}_2 -vector space of formal linear combinations of vertices from U . A *multipoint* is an element of $C_0(U)$ with an odd number of non-zero coefficients. The multipoints form an affine subspace of $C_0(U)$ which we denote by \mathcal{M} . The *support*, $\text{sup}(\mu)$, of a multipoint $\mu \in \mathcal{M}$ is the set of vertices $v \in U$ with non-zero coefficient in μ . We say that two multipoints are *disjoint* if their supports are disjoint.

For any k -face σ_i and any multipoint μ we define:

$$z(\sigma_i, \mu) := \sum_{u \in \text{sup}(\mu)} z(\sigma_i, u) = \sum_{u \in \text{sup}(\mu)} \partial(\sigma_i \cup \{u\}).$$

Now, we proceed as in Section 3 but replace the unused points by the multipoints of \mathcal{M} and the cycles $z(\sigma_i, u)$ with the cycles $z(\sigma_i, \mu)$. Since \mathbb{Z}_2 is a field, $H_k(M)^m$ is a vector space and we can replace the sequences $\mathbf{v}(u)$ of Section 3 by the linear map

$$\mathbf{v} : \begin{cases} C_0(U) & \rightarrow H_k(M)^m \\ \mu & \mapsto ([f_{\#}(z(\sigma_1, \mu))], [f_{\#}(z(\sigma_2, \mu))], \dots, [f_{\#}(z(\sigma_m, \mu))]) \end{cases}$$

Finding collisions. The following lemma takes advantage of the vector space structure of $H_k(M)^m$ to find disjoint multipoints μ_1, μ_2, \dots to route the σ_i 's more effectively than by simple pigeonhole.

► **Lemma 9.** *For any $r \geq 1$, any \mathbb{Z}_2 -vector space V , and any linear map $\psi: C_0(U) \rightarrow V$, if $\text{card}(U) \geq (\dim(\psi(\mathcal{M}) + 1)(r - 1) + 1$ then \mathcal{M} contains r disjoint multipoints $\mu_1, \mu_2, \dots, \mu_r$ such that $\psi(\mu_1) = \psi(\mu_2) = \dots = \psi(\mu_r)$.*

Proof. Let us write $U = \{v_{s+2}, v_{s+3}, \dots, v_{n+1}\}$ and $d = \dim(\psi(\mathcal{M}))$. We first prove by induction on r the following statement:

If $\text{card}(U) \geq (d + 1)(r - 1) + 1$ there exist r pairwise disjoint subsets $I_1, I_2, \dots, I_r \subseteq U$ whose image under ψ have affine hulls with non-empty intersection.

(This is, in a sense, a simple affine version of Tverberg's theorem.) The statement is obvious for $r = 1$, so assume that $r \geq 2$ and that the statement holds for $r - 1$. Let A denote the affine hull of $\{\psi(v_{s+2}), \psi(v_{s+3}), \dots, \psi(v_{n+1})\}$ and let I_r denote a minimal cardinality subset of U such that the affine hull of $\{\psi(v) : v \in I_r\}$ equals A . Since $\dim A \leq d$ the set I_r has cardinality at most $d + 1$. The cardinality of $U \setminus I_r$ is at least $(d + 1)(r - 2) + 1$ so we can apply the induction hypothesis for $r - 1$ to $U \setminus I_r$. We thus obtain $r - 1$ disjoint subsets I_1, I_2, \dots, I_{r-1} whose images under ψ have affine hulls with non-empty intersection. Since the affine hull of $\psi(U \setminus I_r)$ is contained in the affine hull of $\psi(I_r)$, the claim follows.

Now, let $a \in V$ be a point common to the affine hulls of $\psi(I_1), \psi(I_2), \dots, \psi(I_r)$. Writing a as an affine combination in each of these spaces, we get

$$a = \sum_{u \in J_1} \psi(u) = \sum_{u \in J_2} \psi(u) = \dots = \sum_{u \in J_r} \psi(u)$$

where $J_j \subseteq I_j$ and $|J_j|$ is odd for any $j \in [r]$. Setting $\mu_j = \sum_{u \in J_j} u$ finishes the proof. ◀

Computing the dimension of $\mathbf{v}(\mathcal{M})$. Having in mind to apply Lemma 9 with $V = H_k(M)^m$ and $\psi = \mathbf{v}$, we now need to bound from above the dimension of $\mathbf{v}(\mathcal{M})$. An obvious upper bound is $\dim H_k(M)^m$, which equals $bm = b\binom{s+1}{k+1}$. A better bound can be obtained by an argument analogous to the proof of Lemma 7. We first extend Claim 8 to multipoints.

► **Claim 10.** *Let τ be a $(k + 1)$ -face of Δ_s and let $\mu \in \mathcal{M}$. Let $\sigma_{i_1}, \dots, \sigma_{i_{k+2}}$ be all the k -faces of τ . Then*

$$\partial\tau + z(\sigma_{i_1}, \mu) + z(\sigma_{i_2}, \mu) + \dots + z(\sigma_{i_{k+2}}, \mu) = 0. \tag{7}$$

Proof. By Claim 8 we know that (7) is true for points. For a multipoint μ , we get (7) as a linear combination of equations for the points in $\text{sup}(\mu)$ (using that $\text{card}(\text{sup}(\mu))$ is odd). ◀

► **Lemma 11.** $\dim(\mathbf{v}(\mathcal{M})) \leq b\binom{s}{k}$.

Proof. Let τ be a $(k + 1)$ -face of Δ_s and let $\sigma_{i_1}, \dots, \sigma_{i_{k+2}}$ denote its k -faces.

For any multipoint μ , Claim 10 implies

$$[f_{\sharp}(\partial\tau)] = \sum_{j=1}^{k+2} [f_{\sharp}(z(\sigma_{i_j}, \mu))] = \sum_{j=1}^{k+2} \mathbf{v}_{i_j}(\mu) \quad \text{so} \quad \mathbf{v}_{i_{k+2}}(\mu) = [f_{\sharp}(\partial\tau)] + \sum_{j=1}^{k+1} \mathbf{v}_{i_j}(\mu).$$

(Remember that homology is computed over \mathbb{Z}_2 .) Each vector $\mathbf{v}(\mu)$ is thus determined by the values of the $\mathbf{v}_j(\mu)$'s where σ_j contains the vertex v_1 . Indeed, the vectors $[f_{\sharp}(\partial\tau)]$ are

independent of μ , and for any σ_i not containing v_1 we can eliminate $\mathbf{v}_i(\mu)$ by considering $\tau := \sigma_i \cup \{v_1\}$ (and setting $\sigma_{i_{k+2}} = \sigma_i$). For each of the $\binom{s}{k}$ faces σ_j that contain v_1 , the vector $\mathbf{v}_j(\mu)$ takes values in $H_k(M)$ which has dimension at most b . It follows that $\dim \mathbf{v}(\mathcal{M}) \leq b \binom{s}{k}$. ◀

Coloring hypergraphs to reduce the number of multipoints used. We could now apply Lemma 9 with $r = m$ to obtain one multipoints per k -face, all pairwise disjoint, to proceed with our “routing”. As mentioned above, however, we only need that φ is an almost-embedding, so we can use the same multipoint for several k -faces provided they pairwise intersect. Optimizing the number of multipoints used reformulates as the following hypergraph coloring problem:

Assign to each k -face σ_i of Δ_s some color $c(i) \in \mathbb{N}$ such that $\text{card}\{c(i) : 1 \leq i \leq m\}$ is minimal and disjoint faces use distinct colors.

This question is classically known as Kneser’s hypergraph coloring problem and an optimal solution uses $s - 2k + 1$ colors [17, 18]. Let us spell out one such coloring (proving its optimality is considerably more difficult, but we do not need to know that it is optimal). For every k -face σ_i we let $\min \sigma_i$ denote the smallest index of a vertex in σ_i . When $\min \sigma_i \leq s - 2k$ we set $c(i) = \min \sigma_i$, otherwise we set $c(i) = s - 2k + 1$. Observe that any k -face with color $c \leq s - 2k$ contains vertex v_c . Moreover, the k -faces with color $s - 2k + 1$ consist of $k + 1$ vertices each, all from a set of $2k + 1$ vertices. It follows that any two k -faces with the same color have some vertex in common.

Defining φ . We are finally ready to define the chain map $\varphi : C_*(\Delta_s^{(k)}) \rightarrow C_*(\Delta_n^{(k)})$. Recall that we assume that $n \geq n_0 = \left(\binom{s}{k}b + 1\right)(r - 1) + s + 1$. Using the bound of Lemma 11 we can apply Lemma 9 with $r = s - 2k + 1$, obtaining $s - 2k + 1$ multipoints $\mu_1, \mu_2, \dots, \mu_{s-2k+1} \in \mathcal{M}$. We set $\varphi(\vartheta) = \vartheta$ for any face ϑ of Δ_s of dimension less than k . We then “route” each k -face σ_i through the multipoint $\mu_{c(i)}$ by putting

$$\varphi(\sigma_i) := \sigma_i + z(\sigma_i, \mu_{c(i)}), \tag{8}$$

where $c(i)$ is the color of σ_i in the coloring of the Kneser hypergraph proposed above. We finally extend φ linearly to $C_*(\Delta_s)$.

We need the following analogue of Lemma 7.

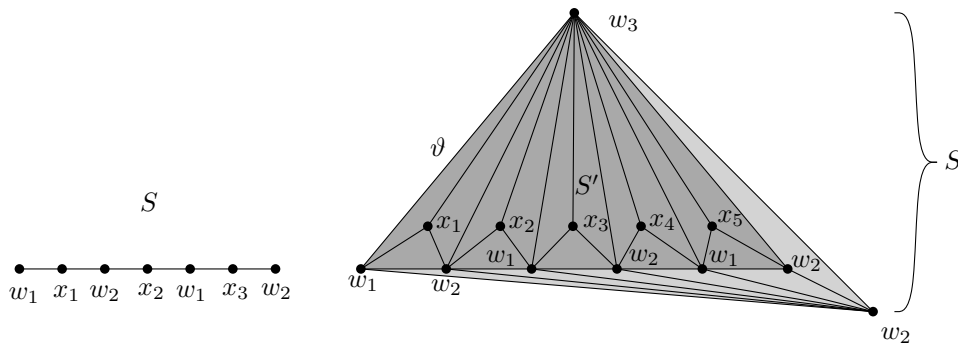
► **Lemma 12.** φ is a chain map and $[f_{\#}(\varphi(\partial\tau))] = 0$ for every $(k + 1)$ -face $\tau \in \Delta_s$.

The proof of Lemma 12 is very similar to the proof of Lemma 7; it just replaces points with multipoints and Claim 8 with Claim 10.

We next argue that φ behaves like an almost embedding.

► **Lemma 13.** For any two disjoint faces ϑ, η of $\Delta_s^{(k)}$, the supports of $\varphi(\vartheta)$ and $\varphi(\eta)$ use disjoint sets of vertices.

Proof. Since φ is the identity on chains of dimension at most $(k - 1)$, the statement follows if neither face has dimension k . For any k -chain σ_i , the support of $\varphi(\sigma_i)$ uses only vertices from σ_i and from the support of $\mu_{c(i)}$. Since each $\mu_{c(i)}$ has support in U , which contains no vertex of Δ_s , the statement also holds when exactly one of ϑ or η has dimension k . When both ϑ and η are k -faces, their disjointness implies that they use distinct μ_j ’s, and the statement follows from the fact that distinct μ_j ’s have disjoint supports. ◀



■ **Figure 1** Examples of subdivisions for $k = 1$ and $\ell = 3$ (left) and for $k = 2$ and $\ell = 5$ (right).

4.2 Construction of D and g

We define D and g similarly as in Section 3, but the switch from points to multipoints requires to replace stellar subdivisions by a slightly more complicated decomposition.

The subdivision D . We define D so that it coincides with Δ_s on the faces of dimension at most $(k - 1)$ and decomposes each face of dimension k independently. The precise subdivision of a k -face σ_i depends on the cardinality of the support of the multipoint $\mu_{c(i)}$ used to “route” σ_i under φ , but the method is generic and spelled out in the next lemma; refer to Figure 1.

► **Lemma 14.** *Let $k \geq 1$ and $\sigma = \{w_1, w_2, \dots, w_{k+1}\}$ be a k -simplex. For any odd integer $\ell \geq 1$ there exists a subdivision S of σ in which no face of dimension $k - 1$ or less is subdivided, and a labelling of the vertices of S by $\{w_1, w_2, \dots, w_{k+1}, x_1, x_2, \dots, x_\ell\}$ (some labels may appear several times) such that:*

1. Every vertex in S corresponding to an original vertex w_i of σ is labelled by w_i ,
2. no k -face of S has its vertices labelled w_1, w_2, \dots, w_{k+1} ,
3. for every $(i, j) \in [k + 1] \times [\ell]$ there exists a unique k -face of S that is labelled by $w_1, w_2, \dots, w_{i-1}, w_{i+1}, \dots, w_{k+1}, x_j$,
4. no edge of S has its two vertices labelled in $\{x_1, x_2, \dots, x_\ell\}$,

Proof. This proof is done in the language of geometric simplicial complexes (rather than abstract ones).

The case $\ell = 1$ can be done by a stellar subdivision and labelling the added apex x_1 . The case $k = 1$ is easy, as illustrated in Figure 1 (left). We therefore assume that $k \geq 2$ and build our subdivision and labelling in four steps:

- We start with the boundary of our simplex σ where each vertex w_i is labelled by itself. Let ϑ be the $(k - 1)$ -face of $\partial\sigma$ opposite vertex w_2 , ie labelled by $w_1, w_3, w_4, \dots, w_{k+1}$. We create a vertex in the interior of σ , label it w_2 , and construct a new simplex σ' as the join of ϑ and this new vertex; this is the dark simplex in Figure 1 (right).
- We then subdivide σ' by considering $\ell - 1$ distinct hyperplanes passing through the vertices of σ' labelled w_3, w_4, \dots, w_{k+1} and through an interior points of the edge of σ' labelled w_1, w_2 . These hyperplanes subdivide σ' into ℓ smaller simplices. We label the new interior vertices on the edge of σ' labelled w_1, w_2 by alternatively, w_1 and w_2 ; since ℓ is odd we can do so in a way that every sub-edge is bounded by two vertices labelled w_1, w_2 .
- We operate a stellar subdivision of each of the ℓ smaller simplices subdividing σ' , and label the added apices x_1, x_2, \dots, x_ℓ . This way we obtain a subdivision S' of σ' .

- We finally consider each face η of S' subdividing $\partial\sigma'$ and other than ϑ and add the simplex formed by η and the (original) vertex w_2 of σ . These simplices, together with S' , form the desired subdivision S of σ .

It follows from the construction that no face of $\partial\sigma$ was subdivided.

Property 1 is enforced in the first step and preserved throughout. We can ensure that Property 2 holds in the following way. First, we have that any k -simplex of S' contains a vertex x_j for some $j \in [\ell]$. Next, if we consider a k -simplex of S which is not in S' it is a join of a certain $(k - 1)$ -simplex η of S' , with $\eta \subset \partial\sigma'$, and the vertex w_2 of σ . However, the only such $(k - 1)$ -simplex labelled by $w_1, w_3, w_4, \dots, w_{k+1}$ is ϑ , but the join of ϑ and w_2 does not belong to S .

Properties 3 and 4 are enforced by the stellar subdivisions of the third step, and no other step creates, destroys or modifies any simplex involving a vertex labelled x_i . ◀

The subdivision D of $\Delta_s^{(k)}$ is now defined as follows. First, we leave the $(k - 1)$ -skeleton untouched. Next, for each k -simplex σ_i we let ℓ_i denote the number of points in the support of $\mu_{c(i)}$; since we work with \mathbb{Z}_2 coefficients, ℓ_i is odd. We then compute some subdivision $S(i)$ of σ_i using Lemma 14 with $\ell := \ell_i$.

We let $\rho: C_*(\Delta_s^{(k)}) \rightarrow C_*(D)$ denote the map that is the identity on $\Delta_s^{(k-1)}$ and that maps each σ_i to the sum of the k -dimensional simplices of $S(i)$. This maps induces an isomorphism ρ_* in homology.

The simplicial map g_{simp} . We now define a simplicial map $g_{\text{simp}}: D \rightarrow \Delta_n^{(k)}$. We first set $g_{\text{simp}}(v) = v$ for every vertex v of Δ_s . Consider next some k -face $\sigma_i = \{w_1(i), w_2(i), \dots, w_{k+1}(i)\}$. We denote by $v_1(i), v_2(i), \dots, v_{k+1}(i)$ the vertices on the boundary of $S(i)$, being understood that each $v_j(i)$ is labelled by w_j , and let $u_1(i), u_2(i), \dots, u_{\ell(i)}(i)$ denote the vertices of the support of $\mu_{c(i)}$. We map each interior vertex of $S(i)$ to either some $w_j(i)$ if its label, as given by Lemma 14, is $w_j(i)$, or some $u_j(i)$ if that label is x_j .

► **Lemma 15.** $(g_{\text{simp}})_\# \circ \rho = \varphi$.

Proof. All three maps are the identity on $\Delta_s^{(k-1)}$ so let us focus on the k -faces. Since ρ maps σ_i to the formal sum of the k -faces of $S(i)$. Each k -face of $S(i)$ is mapped, under g_{simp} , to a k -face with labels $v_1(i), v_2(i), \dots, v_{j-1}(i), v_{j+1}(i), \dots, v_{k+1}(i), u_{j'}(i)$ for some $(j, j') \in [k + 1] \times [\ell(i)]$. Although tedious, it is elementary to check that the chain $(g_{\text{simp}})_\# \circ \rho(\sigma_i)$ has the same support as $\varphi(\sigma_i)$. Since we are working with \mathbb{Z}_2 coefficients, the chains are therefore equal. ◀

The continuous map g . Since D is a subdivision of $\Delta_s^{(k)}$, we have $|\Delta_s^{(k)}| = |D|$ and the simplicial map $g_{\text{simp}}: D \rightarrow \Delta_n^{(k)}$ induces a continuous map $g: |\Delta_s^{(k)}| \rightarrow |\Delta_n^{(k)}|$. All that remains to do is check that g satisfies the two conditions of Lemma 4. Condition 1 follows from a direct translation of Lemma 13. Condition 2 can be verified by a computation in the same way as in Section 3. Specifically, in homology we have

$$f_* \circ \varphi_* = f_* \circ (g_{\text{simp}})_* \circ \rho_*$$

and we know that $f_* \circ \varphi_*$ is trivial on $\Delta_s^{(k)}$ by Lemma 12. As ρ_* is an isomorphism, this implies that $f_* \circ (g_{\text{simp}})_*$ is trivial. Lemma 6 then implies that $(f \circ g)_*$ is trivial. This concludes the proof of Lemma 4.

5 Related Questions and Outlook

While we consider Conjecture 1 natural and interesting in its own right, there are a number of connections to other problems that are worth mentioning and provide additional motivation.

5.1 Topological Helly-Type Theorems for subsets of Manifolds

In [6, Thm. 1], we use the Van Kampen–Flores Theorem to prove the following topological *Helly-type theorem* for finite families \mathcal{F} of subsets of \mathbb{R}^d , under the assumption that for every proper subfamily $\mathcal{G} \subsetneq \mathcal{F}$, the \mathbb{Z}_2 -Betti numbers $b_i(\bigcup \mathcal{G})$, $0 \leq i \leq \lceil d/2 \rceil - 1$, are bounded.

More precisely, our proof heavily relies the fact that the Van Kampen–Flores Theorem also applies to the following generalization of almost-embeddings: We define a *homological almost-embedding* of a finite simplicial complex K into a topological space X as a chain map φ from the simplicial chain complex $C_*(K; \mathbb{Z}_2)$ to the singular chain complex $C_*(X; \mathbb{Z}_2)$ with the properties that (i) for every vertex v of K , $\varphi(v)$ consists of an odd number of points in X and (ii) for any pair σ, τ of disjoint simplices of K , the image chains $\varphi(\sigma)$ and $\varphi(\tau)$ have disjoint underlying point sets.

One can show that Volovikov’s theorem, and consequently Theorem 2 extend to homological almost-embeddings; we plan to discuss this in more detail in the full version of the present paper. Thus, (3) holds whenever $\Delta_n^{(k)}$ homologically almost-embeds into M .

As a consequence, the Helly-type result in [6, Thm. 1] generalizes (with an appropriate change in the constants) to families of subsets of an arbitrary d -dimensional manifold.

5.2 Extremal Problems for Embeddings

Closely related to the classical Heawood inequality is the well-known fact that for a (simple) graph embedded into a surface M , the number of edges of G is at most linear in the number of vertices of G (see, e.g., [22, Thm. 4.2]). More specifically, if G embeds into a surface M with first \mathbb{Z}_2 -Betti number $b_1(M)$, and if $f_1(G)$ and $f_0(G)$ denote the number of vertices and of edges of G , respectively, then

$$f_1(G) \leq 3f_0(G) - 6 + 3b_1(M).$$

Note that this immediately implies (1) when applied to $G = K_n$.

This question also naturally generalizes to higher dimensions:

► **Conjecture 16.** *Let M be a $2k$ -dimensional manifold with k th \mathbb{Z}_2 -Betti number $b_k(M)$. If K is a finite k -dimensional simplicial complex that embeds into M then*

$$f_k(K) \leq C \cdot f_{k-1}(K),$$

where f_i denotes the number of i -dimensional faces of K , $-1 \leq i \leq k$, and C is a constant that depends only on k and on $b_k(M)$.⁵

The special case $M = \mathbb{R}^{2k}$ of the problem was first raised by Grünbaum [7] more than forty years ago, and has since then been rediscovered and posed independently by a number of authors (see, e.g., Dey [4], where the problem is motivated by the question of counting

⁵ In the spirit of the bound for graphs on surfaces, it is also natural to wonder if there might be a bound of the form $f_k(K) \leq C \cdot f_{k-1}(K) + B$, with C depending only on the dimension k and the additive term B depending on k and $b_k(M)$.

triangulations of higher-dimensional point sets), and the problem remains wide open even in that case. Moreover, there is a beautiful conjecture, due to Kalai and Sarkaria [11, Conjecture 27] that gives a necessary condition for embeddability into \mathbb{R}^{2k} in terms of *algebraic shifting* and would, in particular, imply that the constant C in Conjecture 16 can be taken to be $k + 2$ if $M = \mathbb{R}^{2k}$.

The aforementioned extension of Theorem 2 to homological almost-embeddings together with [25, Thm. 7] imply the following result for random complexes:

► **Corollary 17.** *Let $X^k(n, p)$ denote the Linial–Meshulam model [16, 19] of k -dimensional random complexes on n vertices.⁶ Given integers $k \geq 1$ and $b \geq 0$, there exists a constant $C = C(k, b)$ with the following property: If M is a $2k$ -dimensional manifold with \mathbb{Z}_2 -Betti number $b_k(M) \leq b$ and if $p \geq C/n$ then asymptotically almost surely, $X^k(n, p)$ does not embed into M .*

This generalizes [25, Thm. 2] and can be viewed as evidence for Conjecture 16 (in a sense, it shows that the conjecture holds for “almost all complexes”).

The arguments in [25, Thm. 7] are based on the following notion closely related to homological almost-embeddings: If K and L are simplicial complexes, we say that K is a homological minor of L if there is a chain map φ from the simplicial chain complex $C_*(K; \mathbb{Z}_2)$ into the simplicial chain complex of $C_*(L; \mathbb{Z}_2)$ that satisfies conditions (i) and (ii) in the definition of a homological almost-embedding (one might call φ a *simplicial homological almost-embedding*). In [25, Conj. 6], we propose a conjectural generalization of Mader’s theorem to the extent that a finite k -dimensional simplicial complex K contains $\Delta_t^{(k)}$ as a homological minor provided that $f_k(K) \geq C \cdot f_{k-1}(K)$ for some suitable constant $C = C(k, t)$. If true, this conjecture, together with the extension of Theorem 2 to homological almost-embeddings, would imply Conjecture 16.

We remark that Conjecture 1 is also closely related to the combinatorial theory of *face numbers* of triangulated spheres and manifolds, in particular the *Generalized Lower Bound Theorem* for polytopes (which is the main ingredient in Kühnel’s proof of his special case) and conjectured generalizations thereof to triangulated spheres and manifolds. A detailed discussion of these questions goes beyond the scope of this extended abstract, and we refer the reader to [12] and [13, Ch. 4].

Acknowledgement. U. W. learned about Conjecture 1 from Wolfgang Kühnel when attending the *Mini Symposia on Discrete Geometry and Discrete Topology* at the *Jahrestagung der Deutschen Mathematiker-Vereinigung* in München in 2010. He would like to thank the organizers Frank Lutz and Achill Schürmann for the invitation, and Prof. Kühnel for stimulating discussions.

References

- 1 K. Appel and W. Haken. Every planar map is four colorable. I. Discharging. *Illinois J. Math.*, 21(3):429–490, 1977.
- 2 K. Appel, W. Haken, and J. Koch. Every planar map is four colorable. II. Reducibility. *Illinois J. Math.*, 21(3):491–567, 1977.

⁶ By definition, $X^k(n, p)$ has n vertices, a complete $(k - 1)$ -skeleton, and every subset of $k + 1$ vertices is chosen independently with probability p as a k -simplex.

- 3 U. Brehm and W. Kühnel. 15-vertex triangulations of an 8-manifold. *Math. Ann.*, 294(1):167–193, 1992.
- 4 T. K. Dey. On counting triangulations in d dimensions. *Comput. Geom.*, 3(6):315–325, 1993.
- 5 A. I. Flores. Über die Existenz n -dimensionaler Komplexe, die nicht in den \mathbb{R}^{2n} topologisch einbettbar sind. *Ergeb. Math. Kolloqu.*, 5:17–24, 1933.
- 6 X. Goaoc, P. Paták, Z. Patáková, M. Tancer, and U. Wagner. Bounding Helly numbers via Betti numbers. Preprint, [arXiv:1310.4613](https://arxiv.org/abs/1310.4613), 2013.
- 7 B. Grünbaum. Imbeddings of simplicial complexes. *Comment. Math. Helv.*, 44:502–513, 1969.
- 8 A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, UK, 2002.
- 9 P. J. Heawood. Map-colour theorem. *Quart. J.*, 24:332–338, 1890.
- 10 L. Heffter. Ueber das Problem der Nachbargebiete. *Math. Ann.*, 38:477–508, 1891.
- 11 G. Kalai. Algebraic shifting. In *Computational commutative algebra and combinatorics (Osaka, 1999)*, volume 33 of *Adv. Stud. Pure Math.*, pages 121–163. Math. Soc. Japan, Tokyo, 2002.
- 12 W. Kühnel. Manifolds in the skeletons of convex polytopes, tightness, and generalized Heawood inequalities. In *Polytopes: abstract, convex and computational (Scarborough, ON, 1993)*, volume 440 of *NATO Adv. Sci. Inst. Ser. C Math. Phys. Sci.*, pages 241–247. Kluwer Acad. Publ., Dordrecht, 1994.
- 13 W. Kühnel. *Tight polyhedral submanifolds and tight triangulations*, volume 1612 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1995.
- 14 W. Kühnel and T. F. Banchoff. The 9-vertex complex projective plane. *Math. Intelligencer*, 5(3):11–22, 1983.
- 15 W. Kühnel and G. Lassmann. The unique 3-neighborly 4-manifold with few vertices. *J. Combin. Theory Ser. A*, 35(2):173–184, 1983.
- 16 N. Linial and R. Meshulam. Homological connectivity of random 2-complexes. *Combinatorica*, 26(4):475–487, 2006.
- 17 L. Lovász. Kneser’s conjecture, chromatic number, and homotopy. *J. Combin. Theory Ser. A*, 25(3):319–324, 1978.
- 18 J. Matoušek. *Using the Borsuk-Ulam Theorem*. Springer-Verlag, Berlin, 2003.
- 19 R. Meshulam and N. Wallach. Homological connectivity of random k -dimensional complexes. *Random Structures Algorithms*, 34(3):408–417, 2009.
- 20 J. Milnor. On spaces having the homotopy type of a CW-complex. *Trans. Amer. Math. Soc.*, 90:272–280, 1959.
- 21 J. R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Menlo Park, CA, 1984.
- 22 G. Ringel. *Map Color Theorem*. Springer-Verlag, New York-Heidelberg, 1974. Die Grundlehren der mathematischen Wissenschaften, Band 209.
- 23 E. R. van Kampen. Komplexe in euklidischen Räumen. *Abh. Math. Sem. Univ. Hamburg*, 9:72–78, 1932.
- 24 A. Yu. Volovikov. On the van Kampen-Flores theorem. *Mat. Zametki*, 59(5):663–670, 797, 1996.
- 25 U. Wagner. Minors in random and expanding hypergraphs. In *Proceedings of the 27th Annual Symposium on Computational Geometry (SoCG)*, pages 351–360, 2011.

Comparing Graphs via Persistence Distortion*

Tamal K. Dey, Dayu Shi, and Yusu Wang

Computer Science and Engineering Department, The Ohio State University, USA
tamaldey,shiday,yusu@cse.ohio-state.edu

Abstract

Metric graphs are ubiquitous in science and engineering. For example, many data are drawn from hidden spaces that are graph-like, such as the cosmic web. A metric graph offers one of the simplest yet still meaningful ways to represent the non-linear structure hidden behind the data. In this paper, we propose a new distance between two finite metric graphs, called the persistence-distortion distance, which draws upon a topological idea. This topological perspective along with the metric space viewpoint provide a new angle to the graph matching problem. Our persistence-distortion distance has two properties not shared by previous methods: First, it is stable against the perturbations of the input graph metrics. Second, it is a *continuous* distance measure, in the sense that it is defined on an alignment of the underlying spaces of input graphs, instead of merely their nodes. This makes our persistence-distortion distance robust against, for example, different discretizations of the same underlying graph.

Despite considering the input graphs as continuous spaces, that is, taking all points into account, we show that we can compute the persistence-distortion distance in polynomial time. The time complexity for the discrete case where only graph nodes are considered is much faster.

1998 ACM Subject Classification F.2.2 Geometric problems and computations, G.2.2 Graph algorithms

Keywords and phrases Graph matching, metric graphs, persistence distortion, topological method

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.491

1 Introduction

Many data in science and engineering are drawn from a hidden space which are graph-like, such as the cosmic web [28] and road networks [1, 5]. Furthermore, as modern data becomes increasingly complex, understanding them with a simple yet still meaningful structure becomes important. Metric graphs equipped with a metric derived from the data can provide such a simple structure [18, 27]. They are graphs where each edge is associated with a length inducing the metric of shortest path distance. The comparison of the representative metric graphs can benefit classification of data, a fundamental task in processing them. This motivates the study of metric graphs in the context of matching or comparison.

To compare two objects, one needs a notion of distance in the space where the objects are coming from. Various distance measures for graphs and their metric versions have been proposed in the literature with associated matching algorithms. We approach this problem with two new perspectives: (i) We aim to develop a distance measure which is both meaningful and stable against metric perturbations, and at the same time amenable to polynomial time computations. (ii) Unlike most previous distance measures which are *discrete* in the sense that only graph nodes alignments are considered, we aim for a distance

* This work is partially supported by NSF under grants CCF-0747082, CCF-1064416, CCF-1319406, CCF1318595. See [11] for the full version of this paper.



measure that is *continuous*, that is, alignment for all points in the underlying space of the metric graphs are considered.

Related work. To date, the large number of proposed graph matching algorithms fall into two broad categories: exact graph matching methods and inexact graph matching (distances between graphs) methods.

The exact graph matching, also called the graph isomorphism problem, checks whether there is a bijection between the node sets of two input graphs that also induces a bijection in their edge sets. While polynomial time algorithms exist for many special cases, e.g., [2, 21, 25], for general graphs, it is not known whether the graph isomorphism problem is NP complete or not [17]. Nevertheless, given the importance of this problem, there are various exact graph matching algorithms developed in practice. Usually, these methods employ some pruning techniques aiming to reduce the search space for identifying graph isomorphisms. See [15] for comparisons of various graph isomorphism testing methods.

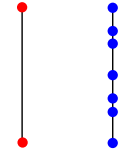
In real world applications, input graphs often suffer from noise and deformation, and it is highly desirable to obtain a *distance* between two input graphs beyond the binary decision of whether they are the same (isomorphic) or not. This is referred to as inexact graph matching in the field of pattern recognition, and various distance measures have been proposed. One line of work is based on graph edit distance which is NP-hard to compute [32]. Many heuristic methods, using for example A^* algorithms, have been proposed to address the issue of high computational complexity, see the survey [16] and references within. One of the main challenges in comparing two graphs is to determine how “good” a given alignment of graph nodes is in terms of the quality of the pairwise relations between those nodes. Hence matching two graphs naturally leads to an integer quadratic programming problem (IQP), which is a NP-hard problem. Several heuristic methods have been proposed to approach this optimization problem, such as the annealing approach of [19], iterative methods of [24, 30] and probabilistic approach in [31]. Finally, there have been several methods that formulate the optimization problem based on spectral properties of graphs. For example, in [29], the author uses the eigendecomposition of adjacency matrices of the input graphs to derive an expression of an orthogonal matrix which optimizes the objective function. In [9, 23], the principal eigenvector of a “compatibility” matrix of the input graphs is used to obtain correspondences between input graph nodes. Recently in [22], Hu et. al proposed the general and descriptive *Laplacian family signatures* to build the compatibility matrix and model the graph matching problem as an integer quadratic program.

New work. Different from previous approaches, we view input graphs as *continuous* metric spaces. Intuitively, we assume that our input is a finite graph $G = (V, E)$ where each edge is assigned a positive length value. We now consider G as a metric space $(|G|, d_G)$ on the underlying space $|G|$ of G , with metric d_G being the shortest path metric in $|G|$. Given two metric graphs G_1 and G_2 , a natural way to measure their distance is to use the so-called Gromov-Hausdorff distance [20, 26] to measure the metric distortion between these two metric spaces. Unfortunately, it is NP-hard to even approximate the Gromov-Hausdorff distance for graphs within a constant factor¹. Instead, we propose a new metric, called the *persistence-distortion distance* $d_{PD}(G_1, G_2)$, which draws upon a topological idea and is

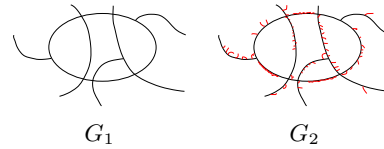
¹ This result is very recently obtained by two groups of researchers independently: Agarwal, Fox and Nath from Duke U., and Sidiropoulos and Wang from Ohio State U.

computable in polynomial time with techniques from computational geometry. This provides a new angle to the graph comparison problem, and our distance has several nice properties:

1. The persistence-distortion distance takes all points in the input graphs into account, while previous graph matching algorithms align only graph nodes. Thus our persistence-distortion distance is insensitive to different discretization of the same graph: For example, the two geometric graphs on the right are equivalent as metric graphs, and thus the persistence-distortion between them is zero.



2. In Section 3, we show that our persistence-distortion distance $d_{PD}(G_1, G_2)$ is stable w.r.t. changes to input metric graphs as measured by the Gromov-Hausdorff distance. For example, the two geometric graphs on the right have small persistence-distortion distance. (Imagine that they are the reconstructed road networks from noisy data sampled from the same road systems.)



3. Despite that our persistence-distortion distance is a *continuous* measure which considers all points in the input graphs, we show in Section 5 that it can be computed in polynomial time ($O(m^{12} \log m)$ where m is the total complexity of input graphs). We note that the *discrete* version of our persistence-distortion distance, where only graph nodes are considered (much like in previous graph matching algorithms), can be computed much more efficiently in $O(n^2 m^{1.5} \log m)$ time, where n is the number of graph nodes in input graphs.

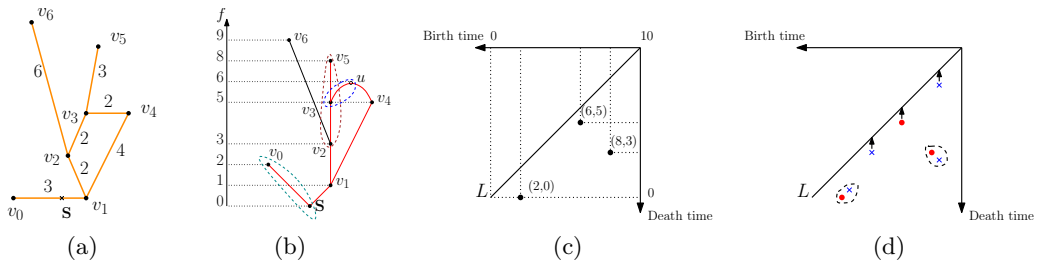
All technical details omitted from this extended abstract due to lack of space can be found in the full version of the paper at [11]. Some preliminary experimental results to demonstrate the use of the persistence-distortion distance are also included in the full version.

2 Notations and Proposed Distance Measure for Graphs

Metric graphs. A metric graph is a metric space (M, d) where M is the underlying space of a finite 1-dimensional simplicial complex. Given a graph $G = (V, E)$ and a weight function $\text{Len} : E \rightarrow \mathbb{R}^+$ on its edge set E (assigning length to edges in E), we can associate a metric graph $(|G|, d_G)$ to it as follows. The space $|G|$ is a geometric realization of G . Let $|e|$ denote the image of an edge $e \in E$ in $|G|$. To define the metric d_G , we consider the arclength parameterization $e : [0, \text{Len}(e)] \rightarrow |e|$ for every edge $e \in E$ and define the distance between any two points $x, y \in |e|$ as $d_G(x, y) = |e^{-1}(y) - e^{-1}(x)|$. This in turn provides the length of a path $\pi(z, w)$ between two points $z, w \in |G|$ that are not necessarily on the same edge in $|G|$, by simply summing up the lengths of the restrictions of this path to edges in G . Finally, given any two points $z, w \in |G|$, the distance $d_G(z, w)$ is given by the minimum length of any path connecting z to w in $|G|$.

In what follows, we do not distinguish between $|\cdot|$ and its argument and write (G, d_G) to denote the metric graph $(|G|, d_G)$ for simplicity. Furthermore, for simplicity in presentation, we abuse the notations slightly and refer to the metric graph as $G = (V, E)$, with the understanding that (V, E) refers to the topological graph behind the metric space (G, d_G) . Finally, we refer to any point $x \in G$ as a point, while a point $x \in V$ as a *graph node*.

Background on persistent homology. The definition of our proposed distance measure for two metric graphs relies on the so-called persistence diagram induced by a scalar function. We



■ **Figure 1** (a) A graph with basepoint s : edge length is marked for each edge. (b) The function $f = d_G(s, \cdot)$. We also indicate critical-pairs. (c) Persistence diagram $Dg_0 f$: E.g, the persistence-point $(6, 5)$ is generated by critical-pair (u, v_3) . (d) shows a partial matching between the red points and blue points (representing two persistence diagrams). Some points are matched to the diagonal L .

refer the readers to resources such as [12, 13] for formal discussions on persistent homology and related developments. Below we only provide an intuitive and informal description of the persistent homology induced by a function under our simple setting.

Let $f : X \rightarrow \mathbb{R}$ be a continuous real-valued function defined on a topological space X . We want to understand the structure of X from the perspective of the scalar function f : Specifically, let $X^\alpha := \{x \in X \mid f(x) \geq \alpha\}$ denote the *super-level set*² of X w.r.t. $\alpha \in \mathbb{R}$. Now as we sweep X top-down by decreasing the α value, the sequence of super-level sets connected by natural inclusion maps gives rise to a *filtration of X induced by f* :

$$X^{\alpha_1} \subseteq X^{\alpha_2} \subseteq \dots \subseteq X^{\alpha_m} = X, \quad \text{for } \alpha_1 > \alpha_2 > \dots > \alpha_m. \tag{1}$$

We track how the topological features captured by the so-called homology classes of the super-level sets change. In particular, as α decreases, sometimes new topological features are “born” at time α , that is, new families of homology classes are created in $H_k(X^\alpha)$, the k -th homology group of X^α . Sometimes, existing topological features disappear, i.e, some homology classes become trivial in $H_k(X^\beta)$ for some $\beta < \alpha$. The *persistent homology* captures such *birth* and *death* events, and summarizes them in the so-called *persistence diagram* $Dg_k(f)$. Specifically, $Dg_k(f)$ consists of a set of points $\{(\alpha, \beta) \in \mathbb{R}^2\}$ in the plane, where each (α, β) indicates a homological feature created at time α and killed at time β .

In our setting, the domain X will be the underlying space of a metric graph G . The specific function that we use later is the geodesic distance to a fixed basepoint $s \in G$, that is, we consider $f : G \rightarrow \mathbb{R}$ where $f(x) = d_G(s, x)$ for any $x \in G$. We are only interested in the 0th-dimensional persistent homology ($k = 0$ in the above description), which simply tracks the connected components in the super-level set as we vary α .

Figure 1 gives an example of the 0-th persistence diagram $Dg_0(f)$ with the basepoint s in edge (v_0, v_1) . As we sweep the graph top-down in terms of the geodesic function f , a new connected component is created as we pass through a *local maximum* u_b of the function $f = d_G(s, \cdot)$. A local maximum of f , such as u in Figure 1 (b), is not necessarily a graph node from V . Two connected components in the super-level set can only merge at an *up-fork saddle* u_d of the function f : The up-fork saddle u_d is a point such that within a sufficiently small neighborhood of u_d , there are at least two branches incident on u_d with function values larger than u_d . Each point (b, d) in the persistence diagram is called a *persistence point*, corresponding to the creation and death of some connected component: At time b , a new

² In the standard formulation of persistent homology of a scalar field, the *sub-level set* $X_\alpha = \{x \in X \mid f(x) \leq \alpha\}$ is often used. We use super-level sets which suit the specific functions that we use.

component is created in X^b at a local maximum $u_b \in G$ with $f(u_b) = b$. At time d and at an up-fork saddle $u_d \in G$ with $f(u_d) = d$, this component merges with another component created earlier. We refer to the pair of points (u_b, u_d) from the graph G as the *critical-pair* corresponding to persistent point (b, d) . We call b and d the *birth-time* and *death-time*, respectively. The plane containing the persistence diagram is called the *birth-death plane*.

Finally, given two finite persistence diagrams $Dg = \{p_1, \dots, p_\ell \in \mathbb{R}^2\}$ and $Dg' = \{q_1, \dots, q_k \in \mathbb{R}^2\}$, a common distance measure for them, the *bottleneck distance* $d_B(Dg, Dg')$ [6], is defined as follows: Consider Dg and Dg' as two finite sets of points in the plane (where points may overlap). Call $L = \{(x, x) \in \mathbb{R}^2\}$ the *diagonal* of the birth-death plane.

► **Definition 1.** A *partial matching* C of Dg and Dg' is a relation $C : (Dg \cup L) \times (Dg' \cup L)$ such that each point in Dg is either matched to a unique point in Dg' , or mapped to its closest point (under L_∞ -norm) in the diagonal L ; and the same holds for points in Dg' . See Figure 1 (d). The bottleneck distance is defined as $d_B(Dg, Dg') = \min_C \max_{(p,q) \in C} \|p - q\|_\infty$, where C ranges over all possible partial matchings of Dg and Dg' . We call the partial matching that achieves the bottleneck distance $d_B(Dg, Dg')$ as the *bottleneck matching*.

Proposed persistence-distortion distance for metric graphs. Suppose we are given two metric graphs (G_1, d_{G_1}) and (G_2, d_{G_2}) .

Choose any point $s \in G_1$ as the base point, and consider the shortest path distance function $d_{G_1,s} : G_1 \rightarrow \mathbb{R}$ defined as $d_{G_1,s}(x) = d_{G_1}(s, x)$ for any point $x \in G_1$. Let P_s denote the 0-th dimensional persistence diagram $Dg_0(d_{G_1,s})$ induced by the function $d_{G_1,s}$. Define $d_{G_2,t}$ and Q_t similarly for any base point $t \in G_2$ for the graph G_2 . We map the graph G_1 to the set of (infinite number of) points in the space of persistence diagrams \mathbb{D} , denoted by $\mathcal{C} := \{P_s \mid s \in G_1\}$. Similarly, map the graph G_2 to $\mathcal{F} := \{Q_t \mid t \in G_2\}$.

► **Definition 2.** The *persistence-distortion distance between G_1 and G_2* , denoted by $d_{PD}(G_1, G_2)$, is the Hausdorff distance $d_H(\mathcal{C}, \mathcal{F})$ between the two sets \mathcal{C} and \mathcal{F} where the distance between two persistence diagrams is measured by the bottleneck distance. In other words,

$$d_{PD}(G_1, G_2) = d_H(\mathcal{C}, \mathcal{F}) = \max \left\{ \max_{P \in \mathcal{C}} \min_{Q \in \mathcal{F}} d_B(P, Q), \max_{Q \in \mathcal{F}} \min_{P \in \mathcal{C}} d_B(P, Q) \right\}.$$

► **Remark.** (1) We note that if two graphs are isomorphic, then $d_{PD}(G_1, G_2) = 0$. The inverse unfortunately is not true (an example is shown in the full version [11]). Hence d_{PD} is a pseudo-metric (it inherits the triangle-inequality property from the Hausdorff distance). (2) While the above definition uses only the 0-th persistence diagram for the geodesic distance functions, all our results hold with the same time complexity when we also include the *1st-extended persistence diagram* [7] or equivalently *1st-interval persistence diagram* [10] for each geodesic distance function $d_{G_1,s}$ (resp. $d_{G_2,t}$).

3 Stability of persistence-distortion distance

Gromov-Hausdorff distance. There is a natural way to measure metric distortion between metric spaces (thus for metric graphs), called the Gromov-Hausdorff distance [20, 4]. Given two metric spaces $\mathcal{X} = (X, d_X)$ and $\mathcal{Y} = (Y, d_Y)$, a *correspondence* between \mathcal{X} and \mathcal{Y} is a relation $\mathcal{M} : X \times Y$ such that (i) for any $x \in X$, there exists $(x, y) \in \mathcal{M}$ and (ii) for any $y' \in Y$, there exists $(x', y') \in \mathcal{M}$. The *Gromov-Hausdorff* distance between \mathcal{X} and \mathcal{Y} is

$$d_{GH}(\mathcal{X}, \mathcal{Y}) = \frac{1}{2} \inf_{\mathcal{M}} \max_{(x_1, y_1), (x_2, y_2) \in \mathcal{M}} |d_X(x_1, x_2) - d_Y(y_1, y_2)|, \tag{2}$$

where \mathcal{M} ranges over all correspondences of $X \times Y$. The Gromov-Hausdorff distance is a natural measurement for distance between two metric spaces; see [26] for more discussions. Unfortunately, so far, there is no efficient (polynomial-time) algorithm to compute nor approximate this distance, even for special metric spaces – In fact, it has been recently shown that even the *discrete* Gromov-Hausdorff distance for metric trees (where only tree nodes are considered) is NP-hard to compute, as well as to approximate within a constant factor (see footnote 1). In contrast, as we show in Section 4 and 5, the persistence-distortion distance can be computed in polynomial time.

On the other hand, we have the following stability result, which intuitively suggests that the persistence-distortion distance is a weaker relaxation of the Gromov-Hausdorff distance. The proof of this theorem leverages a recent result on measuring distances between the Reeb graphs [3] and can be found in the full version.

► **Theorem 3 (Stability).** $d_{\text{PD}}(G_1, G_2) \leq 6d_{\text{GH}}(G_1, G_2)$.

By triangle inequality, this also implies that given two metric graphs G_1 and G_2 and their perturbations G'_1 and G'_2 , respectively, we have that:

$$d_{\text{PD}}(G'_1, G'_2) \leq d_{\text{PD}}(G_1, G_2) + 6d_{\text{GH}}(G_1, G'_1) + 6d_{\text{GH}}(G_2, G'_2).$$

4 Discrete PD-Distance

Suppose we are given two metric graphs $(G_1 = (V_1, E_1), d_{G_1})$ and $(G_2 = (V_2, E_2), d_{G_2})$, where the shortest distance metrics d_{G_1} and d_{G_2} are induced by lengths associated with the edges in $E_1 \cup E_2$. As a simple warm-up, we first compute the following discrete version of persistence-distortion distance where only graph nodes in V_1 and V_2 are considered:

► **Definition 4.** Let $\hat{\mathcal{C}} := \{P_v \mid v \in V(G_1)\}$ and $\hat{\mathcal{F}} := \{Q_u \mid u \in V(G_2)\}$ be two discrete sets of persistence diagrams. The *discrete persistence-distortion distance* between G_1 and G_2 , denoted by $\hat{d}_{\text{PD}}(G_1, G_2)$, is given by the Hausdorff distance $d_H(\hat{\mathcal{C}}, \hat{\mathcal{F}})$.

We note that while we only consider graph nodes as base points, the local maxima of the resulting geodesic function may still occur in the middle of an edge. Nevertheless, for a fixed base point, each edge could have at most one local maximum, and its location can be decided in $O(1)$ time once the shortest-path distance from the base point to the endpoints of this edge are known. The observation below follows from the fact that geodesic distance is 1-Lipschitz (as the basepoint moves) and the stability of persistence diagrams.

► **Observation 5.** $d_{\text{PD}}(G_1, G_2) \leq \hat{d}_{\text{PD}}(G_1, G_2) \leq d_{\text{PD}}(G_1, G_2) + \frac{\ell}{2}$, where ℓ is the largest length of any edge in $E_1 \cup E_2$.

► **Lemma 6.** Given metric graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, $\hat{d}_{\text{PD}}(G_1, G_2)$ can be computed in $O(n^2 m^{1.5} \log m)$ time, where $n = \max\{|V_1|, |V_2|\}$ and $m = \max\{|E_1|, |E_2|\}$.

Proof. For a given base point $\mathbf{s} \in V_1$ (or $\mathbf{t} \in V_2$), computing the shortest path distance from \mathbf{s} to all other graph nodes, as well as the persistence diagram $P_{\mathbf{s}}$ (or $Q_{\mathbf{t}}$) takes $O(m \log n)$ time. Hence it takes $O(mn \log n)$ total time to compute the two collections of persistence diagrams $\hat{\mathcal{C}} = \{P_{\mathbf{s}} \mid \mathbf{s} \in V(G_1)\}$ and $\hat{\mathcal{F}} = \{Q_{\mathbf{t}} \mid \mathbf{t} \in V(G_2)\}$.

Each persistence diagram $P_{\mathbf{s}}$ has $O(m)$ number of points in the plane – it is easy to show that there are $O(m)$ number of local maxima of the geodesic function $d_{G_1, \mathbf{s}}$ (some of which may occur in the interior of graph edges). Since the birth time \mathbf{b} of every persistence point

(b, d) corresponds to a unique local maximum u_b with $f(u_b) = b$, there can be only $O(m)$ points (some of which may overlap each other) in the persistence diagram P_s .

Next, given two persistence diagrams P_s and Q_t , we need to compute the bottleneck distance between them. In [14], Efrat et al. gives an $O(k^{1.5} \log k)$ time algorithm to compute the optimal bijection between two input sets of k points P and Q in \mathbb{R}^2 such that the maximum distance between any mapped pair of points $(p, q) \in P \times Q$ is minimized. This distance is also called the bottleneck distance, and let us denote it by \hat{d}_B . The bottleneck distance between two persistence diagrams P_s and Q_t is similar to the bottleneck distance \hat{d}_B , with the extra addition of diagonals. However, let P' and Q' denote the vertical projection of points in P_s and Q_t , respectively, onto the diagonal L . It is easy to show that $d_B(P, Q) = \hat{d}_B(P_s \cup Q', Q_t \cup P')$. Hence $d_B(P_s, Q_t)$ can be computed by the algorithm of [14] in $O(m^{1.5} \log m)$ time. Finally, to compute the Hausdorff distance between the two sets of persistence diagrams $\hat{\mathcal{C}}$ and $\hat{\mathcal{F}}$, one can check for all pairs of persistence diagrams from these two sets, which takes $O(n^2 m^{1.5} \log m)$ time since the $|\hat{\mathcal{C}}| \leq n$ and $|\hat{\mathcal{F}}| \leq n$. The lemma then follows. ◀

By Observation 5, $\hat{d}_{PD}(G_1, G_2)$ only provides an approximation of $d_{PD}(G_1, G_2)$ with an additive error as decided by the longest edge in the input graphs. For unweighted graphs (where all edges have length 1), this gives an additive error of 1. This in turns provides a factor-2 approximation of the continuous persistence-distortion distance, since $d_{PD}(G_1, G_2)$ is necessarily an integer in this setting.

► **Corollary 7.** *The discrete persistence-distortion distance provides a factor-2 approximation of the continuous persistence-distortion distance for two graphs G_1 and G_2 with unit edge length; that is, $d_{PD}(G_1, G_2) \leq \hat{d}_{PD}(G_1, G_2) \leq 2d_{PD}(G_1, G_2)$.*

One may add additional (steiner) nodes to edges of input graphs to reduce the longest edge length, so that the discrete persistence-distortion distance approximates the continuous one within a smaller additive error. But it is not clear how to bound the number of steiner nodes necessary for approximating the continuous distance within a multiplicative error, even for the case when all edges weights are approximately 1. Below we show how to directly compute the continuous persistence-distortion distance *exactly* in polynomial time.

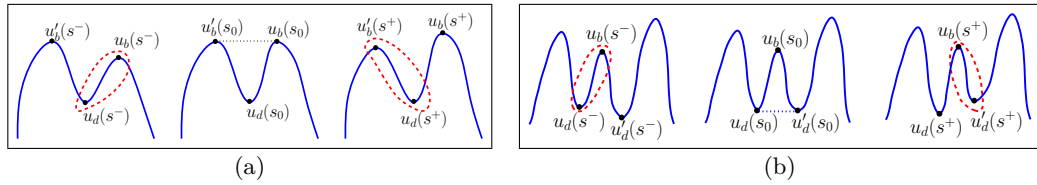
5 Computation of Continuous Persistence-distortion Distance

We now present a polynomial-time algorithm to compute the (continuous) persistence-distortion distance between two metric graphs $(G_1 = (V_1, E_1), d_{G_1})$ and $(G_2 = (V_2, E_2), d_{G_2})$. As before, set $n = \max\{|V_1|, |V_2|\}$ and $m = \max\{|E_1|, |E_2|\}$. Below we first analyze how points in the persistence diagram change as we move the basepoint in G_1 and G_2 continuously.

5.1 Changes of persistence diagrams

We first consider the scenario where the basepoint s moves within a fixed edge $\sigma \in E_1$ of G_1 , and analyze how the corresponding persistence diagram P_s changes. Using notations from Section 2, let (u_b, u_d) be the critical-pair in G_1 that gives rise to the persistence point $(b, d) \in P_s$. Then u_b is a maximum for the distance function $d_{G_1, s}$, while u_d is an up-fork saddle for $d_{G_1, s}$. We call u_b and u_d from G_1 the *birth point* and *death point* w.r.t. the persistence-point (b, d) in the persistence diagram.

As the basepoint s moves to $s' \in \sigma$ within ε distance along the edge σ for any $\varepsilon \geq 0$, the distance function is perturbed by at most ε ; that is, $\|d_{G_1, s} - d_{G_1, s'}\|_\infty \leq \varepsilon$. By the



■ **Figure 2** For better illustration of ideas, we use height function defined on a line to show: (a) a max-max critical event at s_0 ; and (b) a saddle-saddle critical event at s_0 .

Stability Theorem of the persistence diagrams [6], we have that $d_B(P_s, P_{s'}) \leq \varepsilon$. Hence as the basepoint \mathbf{s} moves continuously along σ , points in the persistence diagram P_s move continuously³. We now analyze how a specific point (b, d) may change its trajectory as \mathbf{s} moves from one endpoint v_1 of $\sigma = (v_1, v_2) \in E_1$ to the other endpoint v_2 .

Specifically, we use the arc-length parameterization of σ for \mathbf{s} , that is, $\mathbf{s} : [0, \text{Len}(\sigma)] \rightarrow \sigma$. For any object $X \in \{b, d, u_b, u_d\}$, we use $X(s)$ to denote the object X w.r.t. basepoint $\mathbf{s}(s)$. For example, $(b(s), d(s))$ is the persistence-point w.r.t. basepoint $\mathbf{s}(s)$, while $u_b(s)$ and $u_d(s)$ are the corresponding pair of local maximum and up-fork saddle that give rise to $(b(s), d(s))$. We specifically refer to $b : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ and $d : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ as the *birth-time function* and the *death-time function*, respectively. By the discussion from the previous paragraph, these two functions are continuous.

Critical events. To describe the birth-time and death-time functions, we need to understand how the corresponding birth-point and death-point $u_b(s)$ and $u_d(s)$ in G_1 change as the basepoint \mathbf{s} varies. Recall that as \mathbf{s} moves, the birth-time and death-time change continuously. However, the critical points $u_b(s)$ and $u_d(s)$ in G_1 may (i) stay the same or move continuously, or (ii) have discontinuous jumps. Informally, if it is case (i), then we show below that we can describe $b(s)$ and $d(s)$ using a piecewise linear function with $O(1)$ complexity. Case (ii) happens when there is a *critical event* where two critical-pairs (u_b, u_d) and (u'_b, u'_d) swap their pairing partners to (u_b, u'_d) and (u'_b, u_d) . Specifically, at a critical event, since the birth-time and death-time functions are still continuous, it is necessary that either $d_{G_1, \mathbf{s}}(u_b) = d_{G_1, \mathbf{s}}(u'_b)$ or $d_{G_1, \mathbf{s}}(u_d) = d_{G_1, \mathbf{s}}(u'_d)$; we call the former a *max-max critical event* and the latter a *saddle-saddle critical event*. See Figure 2 for an illustration. It turns out that the birth-time function $b : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ (resp. death-time function d) is a piecewise linear function whose complexity depends on the number of critical events, which we analyze below.

5.1.1 The death-time function $d : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$

The analysis of death-time function is simpler than that of the birth-time function; so we describe it first. Given that $d_{G_1, \mathbf{s}}$ is the geodesic distance to the base point \mathbf{s} , a merging of two components at an up-fork saddle cannot happen in the interior of an edge, unless at the basepoint \mathbf{s} itself.

► **Observation 8.** *An up-fork saddle $u \in G_1$ is necessarily a graph node from V_1 with degree at least 3 unless $u = \mathbf{s}$.*

³ There could be new persistence points appearing or current points disappearing in the persistence diagram as \mathbf{s} moves. Both creation and deletion necessarily happen on the diagonal of the diagram as $d_B(P_s, P_{s'})$ necessarily tends to 0 as s' approaches s . Nevertheless, for simplicity of presentation, below we track the movement of persistence points ignoring their creation and deletion for the time being.

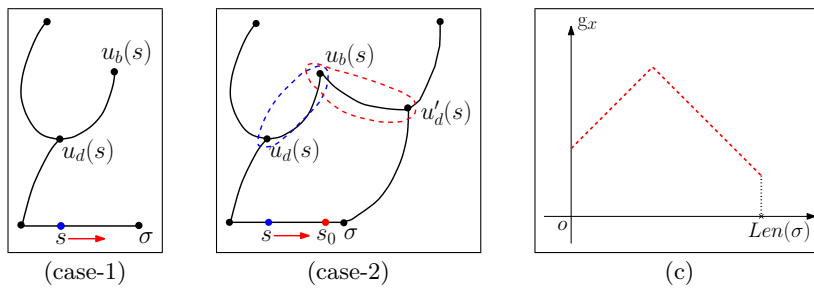


Figure 3 (c) Graph of function $g_x : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$.

To simplify the exposition, we omit the case of $u = \mathbf{s}$ (which is an easier case) in our discussions below. Since the up-fork saddles now can only be graph nodes, as the basepoint $\mathbf{s}(s)$ moves, the death-point $u_d(s)$ either (case-1) stays at the same graph node, or (case-2) switches to a different up-fork saddle u'_d (i.e, a saddle-saddle critical event); see Figure 3.

Now for any point $x \in G_1$, we introduce the function $g_x : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ which is the distance function from x to the moving basepoint $\mathbf{s}(s)$ for $s \in [0, L_\sigma]$; that is, $g_x(s) := d_{G_1, \mathbf{s}(s)}(x)$. Intuitively, as the basepoint $\mathbf{s}(s)$ moves along σ , the distance from $\mathbf{s}(s)$ to a fixed point x either increases or decreases at unit speed, until it reaches a point where the shortest path from $\mathbf{s}(s)$ to x changes discontinuously. We have the following observation.

► **Claim 9.** For any point $x \in G_1$, as the basepoint \mathbf{s} moves in an edge $\sigma \in E$, the distance function $g_x : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ defined as $g_x(s) := d_{G_1, \mathbf{s}(s)}(x)$ is a piecewise linear function with at most 2 pieces, where each piece has slope either ‘1’ or ‘-1’. See Figure 3 (c).

As $\mathbf{s}(s)$ moves, if the death-point $u_d(s)$ stays at the same up-fork saddle u , then by the above claim, the death-time function d (which locally equals g_u) is a piecewise linear function with at most 2 pieces.

Now we consider (case-2) when a saddle-saddle critical event happens: Assume that as s passes value s_0 , $u_d(s)$ switches from a graph node u to another one u' . At the time s_0 when this swapping happens, we have that $d_{G_1, \mathbf{s}(s_0)}(u) = d_{G_1, \mathbf{s}(s_0)}(u')$. In other words, the graph for function g_u and the graph for function $g_{u'}$ intersect at s_0 . Before s_0 , d follows the graph for the distance function g_u , while after time s_0 , u_d changes its identity to u' and thus the movement of d will then follow the distance function $g_{u'}$ for $s > s_0$. Since the function g_x is PL with at most 2 pieces as shown in Figure 3 (c) for any point $x \in G_1$, the switching for a fixed pair of nodes u and u' can happen at most once (as the graph of g_u and that of $g_{u'}$ intersect at most once). Overall, since there are $|V_1| \leq n$ graph nodes, we conclude that:

► **Lemma 10.** As \mathbf{s} moves along σ , there are $O(n^2)$ number of saddle-saddle critical events in the persistence diagram $P_{\mathbf{s}}$.

For our later arguments, we need a stronger version of the above result. Specifically, imagine that we track the trajectory of the death-time d for a persistence pair (\mathbf{b}, d) .

► **Proposition 11.** For a fixed persistent point $(\mathbf{b}(0), d(0)) \in P_{\mathbf{s}(0)}$, the corresponding death-time function $d : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ is piecewise linear with at most $O(n)$ pieces, and each linear piece has slope either ‘1’ or ‘-1’. This also implies that the function d is 1-Lipschitz.

Proof. By Observation 8, $u_d(s)$ is always a graph node from V_1 . For any node u , recall $g_u(s) = d_{G_1, \mathbf{s}(s)}(u)$. As described above, $d(s)$ will follow certain g_u with $u = u_d(s)$ till the identify of $u_d(s)$ changes at a saddle-saddle critical event between u with another up-fork

saddle u' . Afterwards, $d(s)$ will follow $g_{u'}$ till the next critical event. Since each piece of g_v has slope either '1' or '-1', the graph of d consists of linear pieces of slope '1' or '-1'. Note that this implies that the function d is a 1-Lipschitz function.

On the other hand, for a specific graph node $u \in V$, each linear piece in g_u has slope '1' or '-1'. This means that one linear piece in g_u can intersect the graph of d at most once for $s \in [0, \text{Len}(\sigma)]$ as d is 1-Lipschitz. Hence the graph of g_u can intersect the graph of d at most twice; implying that the node u can appear as $u_d(s)$ for at most two intervals of s values. Thus the total descriptive complexity of d is $O(|V_1|) = O(n)$, which completes the proof. ◀

5.1.2 The birth-time function $b : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$.

To track the trajectory of the birth-time b of a persistence pair $(b(0), d(0)) \in P_{s(0)}$, we need to study the movements of its corresponding birth-point (which is a maximum) $u_b : [0, \text{Len}(\sigma)] \rightarrow G_1$ in the graph. However, unlike up-fork saddles (which must be graph nodes), maxima of the distance function $d_{G_1, s}$ can also appear in the interior of a graph edge. Roughly speaking, in addition to degree-1 graph nodes, which must be local maxima of the distance function $d_{G_1, s}$, imagine the shortest path tree with s being the root (source), then any non-tree edge will generate a local maximum of the distance function $d_{G_1, s}$. (Recall the maximum u in Figure 1 (b), which lies in the interior of edge (v_3, v_4) .) Nevertheless, the following result states there can be at most one local maximum associated with each edge.

► **Lemma 12.** *Given an arbitrary basepoint s , a maximum for the distance function $d_{G_1, s} : G_1 \rightarrow \mathbb{R}$ is either a degree-1 graph node, or a point v with at least two shortest paths to the basepoint s which are disjoint in a small neighborhood around v .*

Furthermore, there can be at most one maximum of $d_{G_1, s}$ in each edge in E_1 .

This lemma suggests that we can now associate each local maximum with an edge in E_1 , and analyze the changes of such an edge e_b containing the birth-point u_b (instead of the birth-point itself). Specifically, using approaches similar to the tracking of death-point as in Section 5.1.1, we study, for a fixed edge $e \in E_1$ the function $g_e : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ where, for any $s \in [0, \text{Len}(\sigma)]$, $g_e(s)$ is the distance from the basepoint $s(s)$ to the unique maximum (if it exists) in e ; $g_e(s) = +\infty$ if the distance function $d_{G_1, s(s)}$ does not have a local maximum in e . We refer to the portion of g_e with finite value as *well-defined*. Intuitively, the function g_e serves as the same role as the distance function g_x in Section 5.1.1, and similar to Claim 9, we have the following characterization for this distance function.

► **Proposition 13.** *For any edge $e \in E_1$, the well-defined portion of the function g_e is a piecewise-linear function with $O(1)$ pieces, where each piece is of slope '1', '-1' or '0'.*

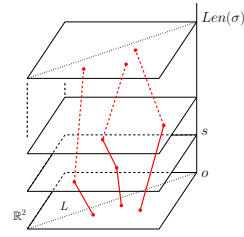
Using argument similar to, but more involved than that of Section 5.1.1, we obtain the following result about the birth-time function, analogous to Proposition 11.

► **Proposition 14.** *For a fixed $(b(0), d(0)) \in P_{s(0)}$, the birth-time function $b : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}$ is piecewise linear with at most $O(m)$ pieces, and each linear piece has slope either '1', '-1', or '0'. Note that this also implies that the function b is 1-Lipschitz.*

5.1.3 Tracking the persistence pair $(b, d) : [0, \text{Len}(\sigma)] \rightarrow \mathbb{R}^2$.

Now consider the space $\Pi_\sigma := [0, \text{Len}(\sigma)] \times \mathbb{R}^2$, where \mathbb{R}^2 denotes the birth-death plane: We can think of Π_σ as the stacking of all the planes containing persistence diagrams $P_{s(s)}$ for all $s \in [0, \text{Len}(\sigma)]$. Hence we refer to Π_σ as the *stacked persistence-space*. For a fixed

persistence pair $(b, d) \in P_{\mathbf{s}(s)}$, as we vary $s \in [0, \text{Len}(\sigma)]$, it traces out a trajectory $\pi = \{(s, b(s), d(s)) \mid s \in [0, \text{Len}(\sigma)]\} \in \Pi_\sigma$, which is the same as the “vines” introduced by Cohen-Steiner et al. [8]. By Propositions 11 and 14, the trajectory π is a polygonal curve with $O(n+m) = O(m)$ linear pieces. See the right figure for an illustration, where there are three trajectories in the stacked persistence diagrams.



In general, a trajectory (a vine) could appear or terminate within the range $(0, \text{Len}(\sigma))$. Specifically, as we track a specific point in the persistence diagram, it is possible that the pair of critical points giving rise to this persistent-point may coincide and cease to exist afterwards. In this case, the corresponding trajectory (vine) hits the diagonal of the persistence diagram (since as the two critical points coincide with $u_b = u_d$, we have that $b = d$) and terminates. The inverse of this procedure indicates the creation of a new trajectory. Nevertheless, we can show that there can be $O(n+m) = O(m)$ total number of trajectories in the stacked persistence diagrams (whether they span the entire range of $s \in [0, \text{Len}(\sigma)]$ or not). We conclude with the following result.

► **Theorem 15.** *Let $\sigma \in E_1$ be an arbitrary edge from the metric graph (G_1, d_{G_1}) . As the basepoint \mathbf{s} moves from one endpoint to another endpoint of σ by $\mathbf{s} : [0, \text{Len}(\sigma)] \rightarrow \sigma$, the persistence-points in the persistence diagram $P_{\mathbf{s}(s)}$ of the distance function $d_{G_1, \mathbf{s}(s)}$ form $O(m)$ number of trajectories in the stacked persistence-space Π_σ . Each trajectory is a polygonal curve of $O(m)$ number of linear segments.*

A symmetric statement holds for metric graph (G_2, d_{G_2}) .

5.2 Computing $d_{PD}(G_1, G_2)$

Given a pair of edges $\sigma_s \in G_1$ and $\sigma_t \in G_2$, as before, we parameterize the basepoints \mathbf{s} and \mathbf{t} by the arc-length parameterization of σ_s and σ_t ; that is: $\mathbf{s} : [0, L_s] \rightarrow \sigma_s$ and $\mathbf{t} : [0, L_t] \rightarrow \sigma_t$ where $L_s = \text{Len}(\sigma_s)$ and $L_t = \text{Len}(\sigma_t)$. We now introduce the following function to help compute $d_{PD}(G_1, G_2)$:

► **Definition 16.** The bottleneck distance function $F_{\sigma_s, \sigma_t} : \Omega \rightarrow \mathbb{R}$ is defined as $F_{\sigma_s, \sigma_t}(s, t) \mapsto d_B(P_{\mathbf{s}(s)}, Q_{\mathbf{t}(t)})$. For simplicity, we sometimes omit σ_s, σ_t from the subscript when their choices are clear from the context.

Recall that $\mathcal{C} = \{P_s \mid \mathbf{s} \in G_1\}$, $\mathcal{F} = \{Q_t \mid \mathbf{t} \in G_2\}$, and by Definition 2:

$$d_{PD}(G_1, G_2) = \max\{\max_{P \in \mathcal{C}} \min_{Q \in \mathcal{F}} d_B(P, Q), \max_{P \in \mathcal{F}} \min_{P \in \mathcal{C}} d_B(P, Q)\}.$$

Below we focus on computing $\vec{d}_H(\mathcal{C}, \mathcal{F}) := \max_{P \in \mathcal{C}} \min_{Q \in \mathcal{F}} d_B(P, Q)$, and the treatment of $\vec{d}_H(\mathcal{F}, \mathcal{C}) := \max_{P \in \mathcal{F}} \min_{P \in \mathcal{C}} d_B(P, Q)$ is symmetric. It is easy to see:

$$\vec{d}_H(\mathcal{C}, \mathcal{F}) = \max_{P \in \mathcal{C}} \min_{Q \in \mathcal{F}} d_B(P, Q) = \max_{\sigma_s \in G_1} \max_{s \in [1, L_s]} \min_{\sigma_t \in G_2} \min_{t \in [1, L_t]} F_{\sigma_s, \sigma_t}(s, t). \tag{3}$$

In what follows, we present the descriptive complexity of F_{σ_s, σ_t} for a fixed pair of edges $\sigma_s \in G_1$ and $\sigma_t \in G_2$ in Section 5.2.1, and show how to use it to compute the persistence-distortion distance between G_1 and G_2 in Section 5.2.2.

5.2.1 One pair of edges $\sigma_s \in G_1$ and $\sigma_t \in G_2$.

Recall that we call the plane containing the persistence diagrams as the birth-death plane, and for persistence-points in this plane, we follow the literature and measure their distance

under the L_∞ -norm (recall Definition 1). From now on, we refer to persistence-points in $P_{\mathbf{s}(s)}$ as *red points*, while persistence-points in $Q_{\mathbf{t}(t)}$ as *blue points*. As s and t vary, the red and blue points move in the birth-death plane. By Theorem 15, the movement of each red (or blue) point traces out a polygonal curve with $O(m)$ segments (which are the projections of the trajectories from the stacked persistence diagrams onto the birth-death plane).

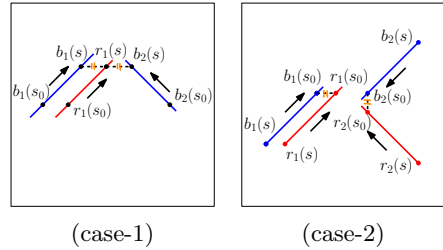
Set $\Omega := [0, L_s] \times [0, L_t]$ and we refer to it as the *s-t domain*. For a point $(s, t) \in \Omega$, the function value $F(s, t) (= F_{\sigma_s, \sigma_t}(s, t)) = d_B(P_{\mathbf{s}(s)}, Q_{\mathbf{t}(t)})$ is the bottleneck distance between the set of red and the set of blue points (with the addition of diagonals) in the birth-death plane. To simplify the exposition, in what follows we ignore the diagonals from the two persistence diagrams and only consider the bottleneck matching between red and blue points.

Let $r^*(s) \in P_{\mathbf{s}(s)}$ and $b^*(t) \in Q_{\mathbf{t}(t)}$ be the pair of red-blue points from the bottleneck matching between $P_{\mathbf{s}(s)}$ and $Q_{\mathbf{t}(t)}$ such that $d_\infty(r^*(s), b^*(t)) = d_B(P_{\mathbf{s}(s)}, Q_{\mathbf{t}(t)})$. We call $(r^*(s), b^*(t))$ the *bottleneck pair (of red-blue points) w.r.t. (s, t)*. As s and t vary continuously, red and blue points move continuously in the birth-death plane. The distance between any pair of red-blue points change continuously. The bottleneck pair between $P_{\mathbf{s}(s)}$ and $Q_{\mathbf{t}(t)}$ typically remains the same till certain *critical values* of the parameters (s, t) .

Characterizing critical (s, t) values. Given (s, t) , consider the optimal bottleneck matching $C^*(s, t) : P_s \times Q_t$. For any corresponding pair $(r(s), b(t)) \in C^*(s, t)$, $d_\infty(r(s), b(t)) \leq d_\infty(r^*(s), b^*(t))$. Suppose $r^*(s) = r_1(s)$ and $b^*(t) = b_1(t)$. As (s, t) varies in Ω , the bottleneck pair $(r^*(s), b^*(t))$ may change only when:

- (case-1): $(r_1(s), b_1(t))$ ceases to be a matched pair in the optimal matching $C^*(s, t)$; or
- (case-2): $(r_1(s), b_1(t))$ is still in C^* , but another matched pair $(r_2(s), b_2(t))$ becomes the bottleneck pair.

At the time (s_0, t_0) that either cases above happens, it is necessary that there are two red-blue pairs, one of which being (r_1, b_1) , and denoting the other one by (r_2, b_2) , such that $d_\infty(r_1(s_0), b_1(t_0)) = d_\infty(r_2(s_0), b_2(t_0))$. (For case-1, we have that either $r_2 = r_1$ or $b_2 = b_1$.) Hence all critical (s, t) values are included in those (s, t) values for which two red-blue pairs of persistence-points acquire equal distance in the birth-death plane. Let



$$X_{(r_1, b_1), (r_2, b_2)} := \{(s, t) \mid d_\infty(r_1(s), b_1(t)) = d_\infty(r_2(s), b_2(t))\}$$

denote the set of *potential critical (s,t)-values generated by (r_1, b_1) and (r_2, b_2)* . To describe $X_{(r_1, b_1), (r_2, b_2)}$, we first consider, for a fixed pair of red-blue points (r, b) , the distance function $D_{r,b} : [0, L_s] \times [0, L_t] \rightarrow \mathbb{R}$ defined as the distance between this pair of red and blue points in the birth-death plane, that is, $D_{r,b}(s, t) := d_\infty(r(s), b(t))$ for any $(s, t) \in \Omega$.

In particular, recall that by Theorem 15, $r : [0, L_s] \rightarrow \mathbb{R}^2$ (resp. $b : [0, L_t] \rightarrow \mathbb{R}^2$) is continuous and piecewise-linear with $O(m)$ segments. In other words, the range $[0, L_s]$ (resp. $[0, L_t]$) can be decomposed to $O(m)$ intervals such that within each interval, r moves (resp. b moves) along a line in the birth-death plane with fixed speed. Hence combining Propositions 11 and 14, we have the following:

► **Proposition 17.** *The s-t domain Ω can be decomposed into an $O(m) \times O(m)$ grid such that, within each of the $O(m^2)$ grid cell, $D_{r,b}$ is piecewise-linear with $O(1)$ linear pieces, and the partial derivative of each piece w.r.t. s or w.r.t. t is either ‘1’, ‘-1’, or ‘0’.*

Given two pairs of red-blue pairs (r_1, b_1) and (r_2, b_2) , the set $X_{(r_1, b_1), (r_2, b_2)}$ of potential critical (s, t) values generated by them corresponds to the intersection of the graph of D_{r_1, b_1} and that of D_{r_2, b_2} . By overlaying the two $O(m) \times O(m)$ grids corresponding to D_{r_1, b_1} and D_{r_2, b_2} as specified by Proposition 17, we obtain another grid of size $O(m) \times O(m)$ and within each cell, the intersection of the graphs of D_{r_1, b_1} and D_{r_2, b_2} has $O(1)$ complexity. Hence,

► **Corollary 18.** *The set $X_{(r_1, b_1), (r_2, b_2)} \subseteq \Omega$ consists of a set of polygonal curves in the s - t domain Ω with $O(m^2)$ total complexity.*

Consider the arrangement $Arr(\Omega)$ of the set of curves in $\mathcal{X} = \{X_{(r_1, b_1), (r_2, b_2)} \mid r_1, r_2 \in P_s, b_1, b_2 \in Q_t\}$. Since there are altogether $O(m^4) \times O(m^2) = O(m^6)$ segments in \mathcal{X} , we have that the arrangement $Arr(\Omega)$ has $O(m^{12})$ complexity; that is, there are $O(m^{12})$ number of vertices, edges and polygonal cells. However, this arrangement $Arr(\Omega)$ is more refined than necessary. Specifically, within a single cell $c \in Arr(\Omega)$, the *entire* bottleneck matching C^* does not change. By a much more sophisticated argument, we can prove the following (see the full version [11] for details):

► **Proposition 19.** *There is a planar decomposition $\Lambda(\Omega)$ of the s - t domain Ω with $O(m^8)$ number of vertices, edges and polygonal cells such that as (s, t) varies within in each cell $c \in \Lambda(\Omega)$, the pair of red-blue persistence points that generates the bottleneck pair (r^*, b^*) remains the same.*

Furthermore, the decomposition $\Lambda(\Omega)$, as well as the bottleneck pair (r^*, b^*) associated to each cell, can be computed in $O(m^{9.5} \log m)$ time.

Our goal is to compute the bottleneck distance function $F : \Omega \rightarrow \mathbb{R}$ introduced at the beginning of this subsection where $F(s, t) \mapsto d_B(P_s(s), Q_t(t)) = d_\infty(r^*(s), b^*(t))$, so as to further compute persistence-distortion distance using Eqn (3). To do this, we need to further refine the decomposition $\Lambda(\Omega)$ from Proposition 19 to another decomposition $\widehat{\Lambda}(\Omega)$ as described below so that within each cell, the bottleneck distance function F_{σ_s, σ_t} can be described by a single linear function. The proof can be found in the full version [11].

► **Theorem 20.** *For a fixed pair of edges $\sigma_s \in G_1$ and $\sigma_t \in G_2$, there is a planar polygonal decomposition $\widehat{\Lambda}(\Omega)$ of the s - t domain Ω of $O(m^{10})$ complexity such that within each cell, the bottleneck distance function F_{σ_s, σ_t} is linear. Furthermore, one can compute this decomposition $\widehat{\Lambda}(\Omega)$ as well as the function F_{σ_s, σ_t} in $O(m^{10} \log m)$ time.*

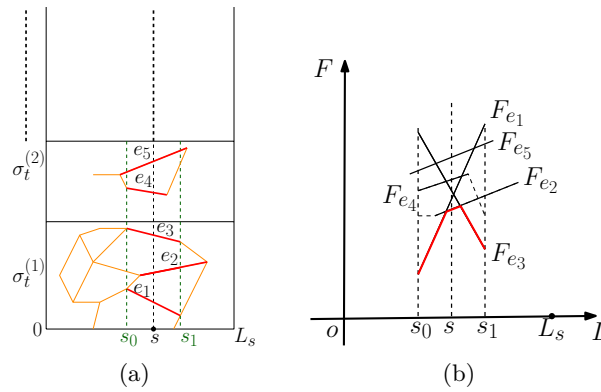
5.2.2 Final algorithm and analysis.

We now aim to compute $\vec{d}_H(\mathcal{C}, \mathcal{F})$ using Eqn (3). First, for a fixed edge $\sigma_s \in G_1$, consider the following *lower-envelop function*

$$\mathcal{L} : [0, L_s] \rightarrow \mathbb{R} \text{ where } \mathcal{L}(s) \mapsto \min_{\sigma_t \in G_2} \min_{t \in [1, L_t]} F(s, t), \tag{4}$$

where recall L_s and L_t denote the length of edge σ_s and σ_t respectively. The reason behind the name "lower-envelop function" will become clear shortly.

Now for each $\sigma_t \in G_2$, consider the polygonal decomposition $\widehat{\Lambda}(\Omega)$ as described in Theorem 20. Since within each cell the bottleneck distance function F is a linear piece, we know that for any s , the extreme of $F(s, t)$ for all possible $t \in [0, L_t]$ must come from some edge in $\widehat{\Lambda}(\Omega)$. In other words, to compute the function $\min_{t \in [0, L_t]} F(s, t)$ at any $s \in [1, L_s]$, we only need to inspect the function F restricted to edges in the refined decomposition $\widehat{\Lambda}(\Omega_{\sigma_s, \sigma_t})$ for the s - t domain $\Omega_{\sigma_s, \sigma_t} = [0, L_s] \times [0, L_t]$. Take any edge e of $\widehat{\Lambda}(\Omega_{\sigma_s, \sigma_t})$, define



■ **Figure 4** (a) s-t domains for $\sigma_s \in E_1$ and edges $\sigma_t^{(j)} \in E_2$. (b) $\mathcal{L}(s)$ is the lowest value along any F_{e_t} .

$\pi_e : [0, L_s] \rightarrow [0, L_t]$ such that $(s, \pi_e(s)) \in e$. Now denote by the function $F_e : [0, L_s] \rightarrow \mathbb{R}$ as the projection of F onto the first parameter $[0, L_s]$; that is, $F_e(s) := F(s, \pi_e(s))$. Let $E_{\sigma_s} := \{e \in \widehat{\Lambda}(\Omega_{\sigma_s, \sigma_t}) \mid \sigma_t \in G_2\}$ be the union of edges from the refined decompositions of the s-t domain formed by σ_s and any edge σ_t from G_2 . It is easy to see that (see Figure 4):

$$\mathcal{L}(s) = \min_{e \in E_{\sigma_s}} F_e(s); \text{ that is, } \mathcal{L} \text{ is the lower-envelop of linear functions } F_e \text{ for all } e \in E_{\sigma_s}.$$

There are $O(m)$ edges in G_2 , thus by Theorem 20 we have $|E_{\sigma_s}| = O(m^{11})$. The lower envelop \mathcal{L} of $|E_{\sigma_s}|$ number of linear functions (linear segments), is a piecewise-linear function with $O(|E_{\sigma_s}| = O(m^{11}))$ complexity and can be computed in $O(|E_{\sigma_s}| \log |E_{\sigma_s}|) = O(m^{11} \log m)$ time. Finally, from Eqn (3), $d_H(\mathcal{C}, \mathcal{F}) = \max_{\sigma_s \in G_1} \max_{s \in [1, L_s]} \mathcal{L}(s)$. Since there are $O(m)$ choices for σ_s , we conclude with the following main result.

► **Theorem 21.** *Given two metric graphs (G_1, d_{G_1}) and (G_2, d_{G_2}) with n total vertices and m total edges, we can compute the persistence-distortion distance $d_{PD}(G_1, G_2)$ between them in $O(m^{12} \log n)$ time.*

We remark that if both input graphs are metric trees, then we can compute their persistence-distortion distance more efficiently in $O(n^8 \log n)$ time.

6 Future directions

The time complexity for computing the (continuous) persistence-distortion distance is high. A worthwhile endeavor will be to bring it down with more accurate analysis. In particular, the geodesic distance function (to a basepoint) in the graph has many special properties, some of which we already leverage. It will be interesting to see whether we can further leverage these properties to reduce the bound on the decomposition $\widehat{\Lambda}(\Omega)$ as used in Theorem 20. Developing efficient approximation algorithms for computing the persistence-distortion distance is also an interesting question. Also, the special case of metric trees is worthwhile to investigate. Notice that even discrete tree matching is still a hard problem for unlabeled trees, i.e., when no correspondences between tree nodes are given.

Acknowledgment. We thank anonymous reviewers for very helpful comments, including the suggestion that $d_B(P_s, Q_t)$ can be computed directly using the algorithm of [14], which simplifies our original approach based on modifying the algorithm of [14].

References

- 1 M. Aanjaneya, F. Chazal, D. Chen, M. Glisse, L. Guibas, and D. Morozov. Metric graph reconstruction from noisy data. *Int. J. Comput. Geom. Appl.*, pages 305–325, 2012.
- 2 A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison Wesley, 1974.
- 3 Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring distance between Reeb graphs. In *Proc. 30th SoCG*, pages 464–473, 2014.
- 4 D. Burago, Y. Burago, and S. Ivanov. *A course in metric geometry*. volume 33 of *AMS Graduate Studies in Math*. American Mathematics Society, 2001.
- 5 Frédéric Chazal and Jian Sun. Gromov-Hausdorff Approximation of Filament Structure Using Reeb-type Graph. In *Proc. 30th SoCG*, pages 491–500, 2014.
- 6 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.
- 7 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*, 9(1):79–103, 2009.
- 8 David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *Proc. 22nd SoCG*, pages 119–126, 2006.
- 9 T. Cour, P. Srinivasan, and J. Shi. Balanced Graph Matching. In *Advances in Neural Information Processing Systems 19*, pages 313–320. MIT Press, 2007.
- 10 T. K. Dey and R. Wenger. Stability of critical points with interval persistence. *Discrete Comput. Geom.*, 38:479–512, 2007.
- 11 Tamal K. Dey, Dayu Shi, and Yusu Wang. Comparing graphs via persistence distortion, 2015. arXiv:1503.07414.
- 12 H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Amer. Math. Soc., Providence, Rhode Island, 2009.
- 13 H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.
- 14 A. Efrat, M. Katz, and A. Itai. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 1:1–28, 2001.
- 15 P. Foggia, C. Sansone, and M. Vento. A Performance Comparison of Five Algorithms for Graph Isomorphism. In *Proc. of the 10th ICIAP*, Italy, 2001.
- 16 Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Anal. Appl.*, 13(1):113–129, January 2010.
- 17 M. R. Garey and D. S. Johnson. *Computers and Intractability: a guide to the theory of NP-completeness*. W. H. Freeman & Co, New York, NY, USA, 1990.
- 18 X. Ge, I. Safa, M. Belkin, and Y. Wang. Data skeletonization via Reeb graphs. In *Proc. 25th NIPS*, pages 837–845, 2011.
- 19 S. Gold and A. Rangarajan. A Graduated Assignment Algorithm for Graph Matching. In *IEEE Trans. on PAMI*, volume 18, pages 377–388, 1996.
- 20 M. Gromov. *Metric structures for Riemannian and non-Riemannian spaces*. volume 152 of *Progress in Mathematics*. Birkhäuser Boston Inc., 1999.
- 21 J. E. Hopcroft and J. K. Wong. Linear Time Algorithm for Isomorphism of Planar Graphs (Preliminary Report). In *Proc. of the ACM STOC*, STOC’74, pages 172–184, New York, NY, USA, 1974. ACM.
- 22 N. Hu, R.M. Rustamov, and L. Guibas. Graph Matching with Anchor Nodes: A Learning Approach. In *IEEE Conference on CVPR*, pages 2906–2913, 2013.
- 23 M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *IEEE International Conference on ICCV*, pages 1482–1489, 2005.

- 24 M. Leordeanu, M. Hebert, and R. Sukthankar. An Integer Projected Fixed Point Method for Graph Matching and MAP Inference. In *Proc. NIPS*. Springer, December 2009.
- 25 E. M. Luks. Isomorphism of Graphs of Bounded Valence Can be Tested in Polynomial Time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982.
- 26 Facundo Mémoli. On the use of Gromov-Hausdorff Distances for Shape Comparison. In *Symposium on Point Based Graphics*, pages 81–90, 2007.
- 27 U. Ozertem and D. Erdogmus. Locally defined principal curves and surfaces. *Journal of Machine Learning Research*, 12:1249–1286, 2011.
- 28 T. Sousbie, C. Pichon, and H. Kawahara. The persistent cosmic web and its filamentary structure – II. Illustrations. *Mon. Not. R. Astron. Soc.*, 414:384–403, 2011.
- 29 S. Umeyama. An eigendecomposition approach to weighted graph matching problems. In *IEEE Trans. on PAMI*, volume 10, pages 695–703, 1998.
- 30 B. J. van Wyk and M. A. van Wyk. A pocs-based graph matching algorithm. In *IEEE Trans. on PAMI*, volume 26, pages 1526–1530, 2004.
- 31 R. Zass and A. Shashua. Probabilistic graph and hypergraph matching. In *IEEE Conference on CVPR*, pages 1–8, June 2008.
- 32 Zhiping Zeng, Anthony K. H. Tung, Jianyong Wang, Jianhua Feng, and Lizhu Zhou. Comparing stars: On approximating graph edit distance. *Proc. VLDB Endow.*, 2(1):25–36, August 2009.

Bounding Helly Numbers via Betti Numbers*

Xavier Goaoc¹, Pavel Paták², Zuzana Patáková³, Martin Tancer³,
and Uli Wagner⁴

- 1 UPEM, Université Paris-Est Marne-la-Vallée, France
- 2 Department of Algebra, Charles University in Prague, Czech Republic
- 3 Department of Applied Mathematics, Charles University in Prague, Czech Republic
- 4 IST Austria, Klosterneuburg, Austria

Abstract

We show that very weak topological assumptions are enough to ensure the existence of a Helly-type theorem. More precisely, we show that for any non-negative integers b and d there exists an integer $h(b, d)$ such that the following holds. If \mathcal{F} is a finite family of subsets of \mathbb{R}^d such that $\tilde{\beta}_i(\bigcap \mathcal{G}) \leq b$ for any $\mathcal{G} \subseteq \mathcal{F}$ and every $0 \leq i \leq \lceil d/2 \rceil - 1$ then \mathcal{F} has Helly number at most $h(b, d)$. Here $\tilde{\beta}_i$ denotes the reduced \mathbb{Z}_2 -Betti numbers (with singular homology). These topological conditions are sharp: not controlling any of these $\lceil d/2 \rceil$ first Betti numbers allow for families with unbounded Helly number.

Our proofs combine homological non-embeddability results with a Ramsey-based approach to build, given an arbitrary simplicial complex K , some well-behaved chain map $C_*(K) \rightarrow C_*(\mathbb{R}^d)$. Both techniques are of independent interest.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling, G.2 Discrete Mathematics

Keywords and phrases Helly-type theorem, Ramsey's theorem, Embedding of simplicial complexes, Homological almost-embedding, Betti numbers

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.507

*Dedicated to the memory of Jiří Matoušek,
wonderful teacher, mentor, collaborator, and friend.*

1 Introduction

Helly's classical theorem [13], a cornerstone of convex geometry, asserts that if a finite family of convex subsets of \mathbb{R}^d has the property that any $d + 1$ of the sets have a point in common then the whole family must have a point in common. Stated in the contrapositive, if \mathcal{F} is a finite family of convex subsets of \mathbb{R}^d with empty intersection then \mathcal{F} contains a sub-family \mathcal{G} of size at most $d + 1$ that already has empty intersection. This inspired the definition of the *Helly number* of a family \mathcal{F} of arbitrary sets. If \mathcal{F} has empty intersection then its Helly

* PP, ZP and MT were partially supported by the Charles University Grant GAUK 421511. ZP was partially supported by the Charles University Grant SVV-2014-260103. ZP and MT were partially supported by the ERC Advanced Grant No. 267165 and by the project CE-ITI (GACR P202/12/G061) of the Czech Science Foundation. UW was partially supported by the Swiss National Science Foundation (grants SNSF-200020-138230 and SNSF-PP00P2-138948). Part of this work was done when XG was affiliated with INRIA Nancy Grand-Est and when MT was affiliated with Institutionen för matematik, Kungliga Tekniska Högskolan, then IST Austria.



number is defined as the size of the largest sub-family $\mathcal{G} \subseteq \mathcal{F}$ with the following properties: \mathcal{G} has empty intersection and any proper sub-family of \mathcal{G} has nonempty intersection; if \mathcal{F} has nonempty intersection then its Helly number is, by convention, 1. With this terminology, Helly's theorem simply states that any finite family of convex sets in \mathbb{R}^d has Helly number at most $d + 1$.

In the spirit of Helly's theorem, bounds on Helly numbers, typically independent of the cardinality of the family, were given for a variety of situations in discrete geometry (such bounds are often referred to as *Helly-type theorems*); we refer to the surveys [9, 30, 27] for an overview of the abundant literature on this topic. Part of the interest for Helly numbers in discrete and computational geometry also stems from their interpretation in optimization problems. In short, a crucial step in applying the framework of generalized linear programming [1] to a geometric problem is to bound the size of so-called *feasible basis*; such bounds are Helly numbers in disguise. We come back to this question when we discuss some consequences of our main result.

Problem statement and results. The classical questions on Helly numbers are of two types, existential and quantitative: identify conditions under which Helly numbers can be bounded uniformly, and obtain sharp bounds. In this paper, we focus on the existential question and give the following new homological sufficient condition for bounding Helly numbers. Note that we consider homology with coefficients over \mathbb{Z}_2 , denote by $\tilde{\beta}_i(X)$ the i th reduced Betti number (over \mathbb{Z}_2) of a space X , and use the notation $\bigcap \mathcal{F} := \bigcap_{U \in \mathcal{F}} U$ as a shorthand for the intersection of a family of sets.

► **Theorem 1.** *For any non-negative integers b and d there exists an integer $h(b, d)$ such that the following holds. If \mathcal{F} is a finite family of subsets of \mathbb{R}^d such that $\tilde{\beta}_i(\bigcap \mathcal{G}) \leq b$ for any $\mathcal{G} \subseteq \mathcal{F}$ and every $0 \leq i \leq \lceil d/2 \rceil - 1$ then \mathcal{F} has Helly number at most $h(b, d)$.*

Our proof hinges on a general principle, which we learned from Matoušek [19] but already underlies the classical proof of Helly's theorem from Radon's lemma, to derive Helly-type theorems from results of non-embeddability of certain simplicial complexes. The novelty of our approach is to examine these non-embeddability arguments from a homological point of view. This turns out to be a surprisingly effective idea, as homological analogues of embeddings appear to be much richer and easier to build than their homotopic counterparts. More precisely, our proof of Theorem 1 builds on two contributions of independent interest:

- We reformulate some non-embeddability results in homological terms. We obtain a homological analogue of the Van Kampen-Flores Theorem (Corollary 7) and, as a side-product, a homological version of Radon's lemma (Lemma 8). This is part of a systematic effort to translate various homotopy technique to a more tractable homology setting. It builds on, and extends, previous work on homological minors [29].
- By working with homology rather than homotopy, we can generalize a technique of Matoušek [19] that uses Ramsey's theorem to find embedded structures.

Theorem 1 is “qualitatively sharp”, in the sense that all (reduced) Betti numbers $\tilde{\beta}_i$ with $0 \leq i \leq \lceil d/2 \rceil - 1$ need to be bounded to obtain a bounded Helly number. To see this, fix some k with $0 \leq k \leq \lceil d/2 \rceil - 1$. For n arbitrarily large, consider a geometric realization in \mathbb{R}^d of the k -skeleton of the $(n - 1)$ -dimensional simplex (see [18, Section 1.6]); more specifically, let $V = \{v_1, \dots, v_n\}$ be a set of points in general position in \mathbb{R}^d (for instance, n points on the moment curve) and consider all geometric simplices $\sigma_A := \text{conv}(A)$ spanned by subsets $A \subseteq V$ of cardinality $|A| \leq k + 1$. By general position, $\sigma_A \cap \sigma_B = \sigma_{A \cap B}$, so this yields indeed a geometric realization.

For $1 \leq j \leq n$, let U_j be the union of all the simplices not containing the vertex v_j . We set $\mathcal{F} = \{U_1, \dots, U_n\}$. Then, $\bigcap \mathcal{F} = \emptyset$, and for any proper sub-family $\mathcal{G} \subsetneq \mathcal{F}$, the intersection $\bigcap \mathcal{G}$ is either \mathbb{R}^d (if $\mathcal{G} = \emptyset$) or (homeomorphic to) the k -dimensional skeleton of a $(n - 1 - |\mathcal{G}|)$ -dimensional simplex. Thus, the Helly number of \mathcal{F} equals n . Moreover, the k -skeleton $\Delta_{m-1}^{(k)}$ of an $(m - 1)$ -dimensional simplex has reduced Betti numbers $\tilde{\beta}_i = 0$ for $i \neq k$ and $\tilde{\beta}_k = \binom{m-1}{k+1}$. Thus, we can indeed obtain arbitrarily large Helly number as soon as at least one $\tilde{\beta}_k$ is unbounded. In particular, setting $k = 0$ yields the lower bound $h(b, d) \geq b + 1$.

Relation to previous work. The study of *topological conditions* (as opposed to more geometric ones like convexity) ensuring bounded Helly numbers started with *Helly's topological theorem* [14] (see also [8] for a modern version of the proof), which states that a finite family of open subsets of \mathbb{R}^d has Helly number at most $d + 1$ if the intersection of any sub-family of at most d members of the family is either empty or a *homology cell*.¹ This includes the case of finite open *good cover*² in \mathbb{R}^d , where the same bound follows easily from the classical *Nerve theorem* [6, 5].

The “good cover” condition was subsequently relaxed by Matoušek [19] who showed that it is sufficient to control the low-dimensional homotopy of intersections: for any non-negative integers b and d there exists a constant $c(b, d)$ such that any finite family of subsets of \mathbb{R}^d in which every sub-family intersects in at most b connected components, each $(\lceil d/2 \rceil - 1)$ -connected,³ has Helly number at most $c(b, d)$.

By Hurewicz' Theorem and the Universal Coefficient Theorem [12, Theorem 4.37 and Corollary 3A.6], a k -connected space X satisfies $\tilde{\beta}_i(X) = 0$ for all $i \leq k$. Thus, our condition indeed relaxes Matoušek's, in two ways: by using \mathbb{Z}_2 -homology instead of the homotopy-theoretic assumptions of k -connectedness⁴, and by allowing an arbitrary fixed bound b instead of $b = 0$.

Quantitatively, the bound on $h(b, d)$ that we obtain is very large as it follows from successive applications of Ramsey's theorems. However, as far as only the existence of uniform bounds is concerned, Theorem 1 not only generalizes Matoušek's result (which also uses Ramsey's theorem), but also subsumes a series of Helly-type theorems due to Amenta [2], Kalai and Meshulam [16], Colin de Verdière et al. [7], and Montejano [21]. Note that for results that hold in rather general ambient spaces, e.g. [16, 7, 21], Theorem 1 only subsumes the case of \mathbb{R}^d .

Our method also proves a bound of $d + 1$ on the Helly number of any family \mathcal{F} such that $\tilde{\beta}_i(\bigcap \mathcal{G}) = 0$ for all $i \leq d$ and all $\mathcal{G} \subsetneq \mathcal{F}$ (see Corollary 10), which generalizes Helly's

¹ By definition, a homology cell is a topological space X all of whose (reduced, singular, integer coefficient) homology groups are trivial, as is the case if $X = \mathbb{R}^d$ or X is a single point. Here and in what follows, we refer the reader to standard textbooks like [12, 22] for further topological background and various topological notions that we leave undefined.

² An open good cover is a finite family of open subsets of \mathbb{R}^d such that the intersection of any sub-family of at most d members is either empty or is contractible (and hence, in particular, a homology cell).

³ We recall that a topological space X is k -connected, for some integer $k \geq 0$, if every continuous map $S^i \rightarrow X$ from the i -dimensional sphere to X , $0 \leq i \leq k$, can be extended to a map $D^{i+1} \rightarrow X$ from the $(i + 1)$ -dimensional disk to X .

⁴ We also remark that our condition can be verified algorithmically since Betti numbers are easily computable, at least for sufficiently nice spaces that can be represented by finite simplicial complexes, say. By contrast, it is algorithmically undecidable whether a given 2-dimensional simplicial complex is 1-connected, see, e.g., the survey [26].

topological theorem as the sets of \mathcal{F} are, for instance, not assumed to be open.⁵ Under the weaker assumption that $\tilde{\beta}_i(\bigcap \mathcal{G}) = 0$ for all subfamilies $\mathcal{G} \subsetneq \mathcal{F}$ but only for $i \leq \lceil d/2 \rceil - 1$, our method still yields a bound of $d + 2$ on the Helly number (see Corollary 9). In both cases the bounds are tight.

Note that Theorem 1 is similar, in spirit, to some of the general relations between the growth of Betti numbers and *fractional* Helly theorems conjectured by Kalai and Meshulam [15, Conjectures 6 and 7]. Kalai and Meshulam, in their conjectures, allow a polynomial growth of the Betti numbers in $|\bigcap \mathcal{G}|$. We remark that Theorem 1 is also sharp in the sense that even a linear growth of Betti number, already in \mathbb{R}^1 , may yield unbounded Helly numbers.

Indeed, consider a positive integer n and open intervals $I_i := (i - 1.1; i + 0.1)$ for $i \in [n]$. Let $X_i := [0, n] \setminus X_i$. The intersection of all X_i is empty but the intersection of any proper subfamily is nonempty. In addition, the intersection of k such X_i can be obtained from $[0, n]$ by removing at most k open intervals, thus the reduced Betti numbers of such intersection are bounded by k .

In particular, the conjectures of Kalai and Meshulam cannot be strengthened to include Theorem 1.

Further consequences. The main strength of our result is to show that very weak assumptions on families of sets are enough to guarantee a bounded Helly number. A first natural application is as a tool to identify concrete situations in which Helly numbers are bounded. Let us give an example which, to the best of our knowledge, is not covered by any other Helly-type theorem appearing in the literature.

By an *affine k -sphere* in \mathbb{R}^d for $0 \leq k \leq d - 1$ we simply mean a geometric sphere of arbitrary center and radius inside some affine $(k + 1)$ -space of \mathbb{R}^d . An *affine sphere* is an affine k -sphere for some $k \in \{0, \dots, d - 1\}$. Theorem 1 implies that the Helly number of an arbitrary family of affine spheres in \mathbb{R}^d is bounded since an arbitrary intersection of affine spheres is an empty set, singleton, or an affine sphere, all of them having bounded Betti numbers. A careful analysis can of course lead to a much better bound on the Helly number than the one given by Theorem 1; see for instance [17] for sharp bounds for the case of $(d - 1)$ -dimensional spheres in \mathbb{R}^d . However, note that Theorem 1 immediately reveals that the Helly number is bounded.

Theorem 1 also has consequences in the direction of optimization problems. Various optimization problems can be formulated as the minimization of some function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ over some intersection $\bigcap_{i=1}^n C_i$ of subsets C_1, C_2, \dots, C_n of \mathbb{R}^d . If, for $t \in \mathbb{R}$, we let $L_t = f^{-1}((-\infty, t])$ and $\mathcal{F}_t = \{C_1, C_2, \dots, C_n, L_t\}$ then

$$\min_{x \in \bigcap_{i=1}^n C_i} f(x) = \min \left\{ t \in \mathbb{R} : \bigcap \mathcal{F}_t \neq \emptyset \right\}.$$

If the Helly number of the families \mathcal{F}_t can be bounded *uniformly* in t by some constant h then there exists a subset of $h - 1$ constraints $C_{i_1}, C_{i_2}, \dots, C_{i_{h-1}}$ that suffice to define the minimum of f :

$$\min_{x \in \bigcap_{i=1}^n C_i} f(x) = \min_{x \in \bigcap_{j=1}^{h-1} C_{i_j}} f(x).$$

⁵ In the original proof, this assumption is crucial and used to ensure that the union of the sets must have trivial homology in dimensions larger than d ; this may fail if the sets are not open.

A consequence of this observation, noted by Amenta [1], is that the minimum of f over $C_1 \cap C_2 \cap \dots \cap C_n$ can⁶ be computed in randomized $O(n)$ time by *generalized linear programming* [25]. Together with Theorem 1, this implies that an optimization problem of the above form can be solved in randomized linear time if it has the property that every intersection of some subset of the constraints with a level set of the function has bounded “topological complexity” (measured in terms of the sum of the first $\lceil d/2 \rceil$ Betti numbers). Let us emphasize that this linear-time bound holds in a real-RAM model of computation, where any constant-size subproblems can be solved in $O(1)$ -time; it therefore concerns the *combinatorial difficulty* of the problem and says nothing about its *numerical difficulty*.

Organization, notation, etc. We prove Theorem 1 in three steps. We first set up our homological machinery in Section 2 (homological almost-embeddings, homological Van Kampen-Flores Theorem, and homological Radon lemma). We then present, in Section 3, variations of the technique that derives Helly-type theorems from non-embeddability. We finally introduce our refinement of this technique and the proof of Theorem 1 in Section 4. Due to space constraint, various proofs are only sketched and we refer to [11] for the full details.

We assume that the reader is familiar with basic topological notions and facts concerning simplicial complexes and singular and simplicial homology, as described in textbooks like [12, 22]. As remarked above, throughout this paper we will work with homology with \mathbb{Z}_2 -coefficients unless explicitly stated otherwise. Moreover, while we will consider singular homology groups for topological spaces in general, for simplicial complexes we will work with simplicial homology groups. In particular, if X is a topological space then $C_*(X)$ will denote the singular chain complex of X , while if K is a simplicial complex, then $C_*(K)$ will denote the simplicial chain complex of K (both with \mathbb{Z}_2 -coefficients).

We use the following notation. Let K be a (finite, abstract) simplicial complex. The *underlying topological space* of K is denoted by $|K|$. Moreover, we denote by $K^{(i)}$ the i -*dimensional skeleton* of K , i.e., the set of simplices of K of dimension at most i ; in particular $K^{(0)}$ is the set of vertices of K . For an integer $n \geq 0$, let Δ_n denote the n -dimensional simplex.

Given a set X we let 2^X and $\binom{X}{k}$ denote, respectively, the set of all subsets of X (including the empty set) and the set of all k -element subsets of X . If $f : X \rightarrow Y$ is an arbitrary map between sets then we abuse the notation by writing $f(S)$ for $\{f(s) \mid s \in S\}$ for any $S \subseteq X$; that is, we implicitly extend f to a map from 2^X to 2^Y whenever convenient.

2 Homological Almost-Embeddings

In this section, we define *homological almost-embedding*, an analogue of topological embeddings on the level of chain maps, and show that certain simplicial complexes do not admit homological almost-embeddings in \mathbb{R}^d , in analogy to classical non-embeddability results due to Van Kampen and Flores.

Recall that an *embedding* of a finite simplicial complex K into \mathbb{R}^d is simply an injective continuous map $|K| \rightarrow \mathbb{R}^d$. The fact that the complete graph on five vertices cannot be embedded in the plane has the following generalization.

⁶ This requires f and C_1, C_2, \dots, C_n to be generic in the sense that the number of minima of f over $\bigcap_{i \in I} C_i$ is bounded uniformly for $I \subseteq \{1, 2, \dots, n\}$.

► **Proposition 2** (Van Kampen [28], Flores [10]). *For $k \geq 0$, the complex $\Delta_{2k+2}^{(k)}$, the k -dimensional skeleton of the $(2k + 2)$ -dimensional simplex, does not embed in \mathbb{R}^{2k} .*

A basic tool for proving the non-embeddability of a simplicial complex is the so-called *Van Kampen obstruction*. Given a simplicial complex K , one can define, for each $d \geq 0$, a certain cohomology class $\mathfrak{o}^d(K)$ that resides in the cohomology group $H^d(\overline{K})$ of a certain auxiliary complex \overline{K} (the quotient of the combinatorial deleted product by the natural \mathbb{Z}_2 -action, see below); this cohomology class $\mathfrak{o}^d(K)$ is called the Van Kampen obstruction to embeddability into \mathbb{R}^d because of the following fact:

► **Proposition 3** ([24, 31]). *A finite simplicial complex K with $\mathfrak{o}^d(K) \neq 0$ does not embed into \mathbb{R}^d .*

A slightly stronger conclusion actually holds: there is no *almost-embedding* $f: |K| \rightarrow \mathbb{R}^d$, i.e., no continuous map such that the images of disjoint simplices of K are disjoint. Proposition 2, and in fact the slightly stronger statement that $\Delta_{2k+2}^{(k)}$ does not admit an almost-embedding into \mathbb{R}^{2k} , then follows from the next result (for a short proof see, for instance, [20, Example 3.5]).

► **Proposition 4** ([28, 10]). *For every $k \geq 0$, $\mathfrak{o}^{2k}(\Delta_{2k+2}^{(k)}) \neq 0$.*

A close examination of the standard proof of Proposition 3 reveals that it is based on (co)homological arguments so that maps can be replaced by suitable chain maps at every step.⁷ The appropriate analogue of an almost-embedding is the following:

► **Definition 5.** Let K be a simplicial complex, and consider a chain map⁸ $\gamma: C_*(K) \rightarrow C_*(\mathbb{R}^d)$ from the simplicial chains in K to singular chains in \mathbb{R}^d .

- (i) The chain map γ is called *nontrivial*⁹ if the image of every vertex of K is a finite set of points in \mathbb{R}^d (a 0-chain) of *odd* cardinality.
- (ii) The chain map γ is called a *homological almost-embedding* of a simplicial complex K in \mathbb{R}^d if it is nontrivial and if, additionally, the following holds: whenever σ and τ are disjoint simplices of K , their image chains $\gamma(\sigma)$ and $\gamma(\tau)$ have disjoint supports, where the support of a chain is the union of (the images of) the singular simplices with nonzero coefficient in that chain.

Definition 5 generalizes classical homotopic notions. Indeed, if $f: |K| \rightarrow \mathbb{R}^d$ is a continuous map then the induced chain map¹⁰ $f_\# : C_*(K) \rightarrow C_*(\mathbb{R}^d)$ is nontrivial. Moreover, if f is an almost-embedding then the induced chain map is a homological almost-embedding. We can generalize Proposition 3 as follows:

► **Proposition 6.** *A finite simplicial complex K with $\mathfrak{o}^d(K) \neq 0$ has no homological almost-embedding in \mathbb{R}^d .*

⁷ This observation was already used in [29] to study the (non-)embeddability of certain simplicial complexes. What we call a *homological almost-embedding* in the present paper corresponds to the notion of a *homological minor* used in [29].

⁸ We recall that a chain map $\gamma: C_* \rightarrow D_*$ between chain complexes is simply a sequence of homomorphisms $\gamma_n: C_n \rightarrow D_n$ that commute with the respective boundary operators, $\gamma_{n-1} \circ \partial_C = \partial_D \circ \gamma_n$.

⁹ If we consider augmented chain complexes with chain groups also in dimension -1 , then being nontrivial is equivalent to requiring that the generator of $C_{-1}(K) \cong \mathbb{Z}_2$ (this generator corresponds to the empty simplex in K) is mapped to the generator of $C_{-1}(\mathbb{R}^d) \cong \mathbb{Z}_2$.

¹⁰ The induced chain map is defined as follows: We assume that we have fixed a total ordering of the vertices of K . For a p -simplex σ of K , the ordering of the vertices induces a homeomorphism $h_\sigma: |\Delta_p| \rightarrow |\sigma| \subseteq |K|$. The image $f_\#(\sigma)$ is defined as the singular p -simplex $f \circ h_\sigma$.

Sketch of proof: Like in the standard proof of Proposition 3, we construct given a homological almost-embedding of a complex K into \mathbb{R}^d a non-trivial equivariant chain map from the (combinatorial) deleted product of that complex into \mathbb{S}^{d-1} , then into \mathbb{S}^∞ through the inclusion $\mathbb{S}^{d-1} \rightarrow \mathbb{S}^\infty$. We can then interpret $\mathfrak{o}^d(K)$ in terms of the d -dimensional cohomology of $\mathbb{R}\mathbb{P}^\infty$, the \mathbb{Z}_2 quotient of \mathbb{S}^∞ , and show that it should vanish. In one of the steps we need to replace (classical) equivariant homotopy with equivariant chain homotopy, which is somewhat technical. We refer to [11, Proposition 7] for a complete proof. ◀

As a consequence we obtain a homological analogue of the Van Kampen-Flores theorem:

▶ **Corollary 7.** *For $d \geq 0$, $\Delta_{d+2}^{\lceil d/2 \rceil}$ has no homological almost-embedding in \mathbb{R}^d .*

Proof. Propositions 4 and 6 together imply that for any $k \geq 0$, the k -skeleton $\Delta_{2k+2}^{(k)}$ of the $(2k + 2)$ -dimensional simplex has no homological almost-embedding in \mathbb{R}^{2k} . This proves the statement when d is even.

Assume that d is odd and write $d = 2k + 1$. If K is a finite simplicial complex with $\mathfrak{o}^d(K) \neq 0$ and if CK is the cone over K then $\mathfrak{o}^{d+1}(CK) \neq 0$ (for a proof, see, for instance, [4, Lemma 8]). Since we know that $\mathfrak{o}^{2k}(\Delta_{2k+2}^{(k)}) \neq 0$ it follows that $\mathfrak{o}^{2k+1}(C\Delta_{2k+2}^{(k)}) \neq 0$. Consequently, $\mathfrak{o}^{2k+1}(\Delta_{2k+3}^{(k+1)}) \neq 0$ since $C\Delta_{2k+2}^{(k)}$ is a subcomplex of $\Delta_{2k+3}^{(k+1)}$ (and since there is an equivariant map from the deleted product of the subcomplex to the deleted product of the complex). Proposition 6 then implies that $\Delta_{2k+3}^{(k+1)}$ admits no homological almost-embedding in \mathbb{R}^{2k+1} . This proves the statement when d is odd. ◀

We also deduce a homological Radon lemma (note that $\partial\Delta_{d+1} = \Delta_{d+1}^{(d)}$); see [11, Lemma 10] for a proof.

▶ **Corollary 8.** *For $d \geq 0$, $\partial\Delta_{d+1}$ has no homological almost-embedding in \mathbb{R}^d .*

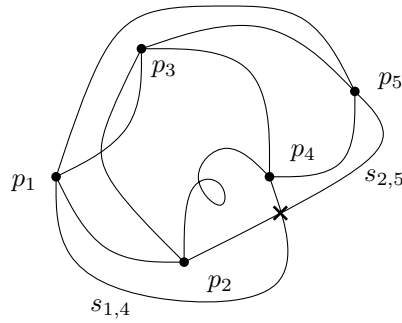
3 Helly-type theorems from non-embeddability

In this section, we review various applications, and formalize the ingredients, of a technique to prove Helly-type theorems from obstructions to embeddability. This technique was already present in the classical proof of Helly’s convex theorem from Radon’s lemma and was made more transparent by Matoušek [19].

3.1 Homotopic assumptions

Let $\mathcal{F} = \{U_1, U_2, \dots, U_n\}$ denote a family of subsets of \mathbb{R}^d . We assume that \mathcal{F} has empty intersection and that any proper subfamily of \mathcal{F} has nonempty intersection. Our goal is to show how various conditions on the topology of the intersections of the subfamilies of \mathcal{F} imply bounds on the cardinality of \mathcal{F} . For any (possibly empty) proper subset I of $[n] = \{1, 2, \dots, n\}$ we write $U_{\overline{I}}$ for $\bigcap_{i \in [n] \setminus I} U_i$. We also put $U_{\overline{[n]}} = \mathbb{R}^d$.

Path-connected intersections in the plane. Consider the case where $d = 2$ and the intersections $\bigcap \mathcal{G}$ are path-connected for all subfamilies $\mathcal{G} \subsetneq \mathcal{F}$. Since every intersection of $n - 1$ members of \mathcal{F} is nonempty, we can pick, for every $i \in [n]$, a point p_i in $U_{\overline{\{i\}}}$. Moreover, as every intersection of $n - 2$ members of \mathcal{F} is connected, we can connect any pair of points p_i and p_j by an arc $s_{i,j}$ inside $U_{\overline{\{i,j\}}}$. We thus obtain a drawing of the complete graph on $[n]$ in the plane in a way that the edge between i and j is contained in $U_{\overline{\{i,j\}}}$ (see Figure 1). If $n \geq 5$ then the stronger form of non-planarity of K_5 implies that there exist two edges



■ **Figure 1** Two edges (arcs) with no common vertices intersect (in this case $s_{1,4}$ and $s_{2,5}$). The point in the intersection then belongs to all sets in \mathcal{F} .

$\{i, j\}$ and $\{k, \ell\}$ with no vertex in common and whose images intersect (see Proposition 3 and Lemma 4). Since $U_{\{i,j\}} \cap U_{\{k,\ell\}} = \bigcap \mathcal{F} = \emptyset$, this cannot happen and \mathcal{F} has cardinality at most 4.

$\lceil d/2 \rceil$ -connected intersections in \mathbb{R}^d . The previous argument generalizes to higher dimension as follows. Assume that the intersections $\bigcap \mathcal{G}$ are $\lceil d/2 \rceil$ -connected¹¹ for all subfamilies $\mathcal{G} \subseteq \mathcal{F}$. Then we can build by induction a function f from the $\lceil d/2 \rceil$ -skeleton of Δ_{n-1} to \mathbb{R}^d in a way that for any simplex σ , the image $f(\sigma)$ is contained in $U_{\bar{\sigma}}$. The previous case shows how to build such a function from the 1-skeleton of Δ_{n-1} . Assume that a function f from the ℓ -skeleton of Δ_{n-1} is built. For every $(\ell + 1)$ -simplex σ of Δ_{n-1} , for every facet τ of σ , we have $f(\tau) \subset U_{\bar{\tau}} \subseteq U_{\bar{\sigma}}$. Thus, the set

$$\bigcup_{\tau \text{ facet of } \sigma} f(\tau)$$

is the image of an ℓ -dimensional sphere contained in $U_{\bar{\sigma}}$, which has vanishing homotopy of dimension ℓ . We can extend f from this sphere to an $(\ell + 1)$ -dimensional ball so that the image is still contained in $U_{\bar{\sigma}}$. This way we extend f to the $(\ell + 1)$ -skeleton of Δ_{n-1} .

The Van Kampen-Flores theorem asserts that for any continuous function from $\Delta_{2k+2}^{(k)}$ to \mathbb{R}^{2k} there exist two disjoint faces of $\Delta_{2k+2}^{(k)}$ whose images intersect (see Proposition 3 and Lemma 4). So, if $n \geq 2\lceil d/2 \rceil + 3$, then there exist two disjoint simplices σ and τ of $\Delta_{2\lceil d/2 \rceil + 2}^{(\lceil d/2 \rceil)}$ such that $f(\sigma) \cap f(\tau)$ is nonempty. Since $f(\sigma) \cap f(\tau)$ is contained in $U_{\bar{\sigma}} \cap U_{\bar{\tau}} = \bigcap \mathcal{F} = \emptyset$, this is a contradiction and \mathcal{F} has cardinality at most $2\lceil d/2 \rceil + 2$.

By a more careful inspection of odd dimensions, the bound $2\lceil d/2 \rceil + 2$ can be improved to $d + 2$. We skip this in the homotopic setting, but we will do so in the homological setting (which is stronger anyway); see Corollary 9 below.

Contractible intersections. Of course, the previous argument works with other non-embeddability results. For instance, if the intersections $\bigcap \mathcal{G}$ are contractible for all subfamilies then the induction yields a map f from the d -skeleton of Δ_{n-1} to \mathbb{R}^d with the property that for any simplex σ , the image $f(\sigma)$ is contained in $U_{\bar{\sigma}}$. The topological Radon theorem [3] (see also [18, Theorem 5.1.2]) states that for any continuous function from Δ_{d+1} to \mathbb{R}^d there exist two disjoint faces of Δ_{d+1} whose images intersect. So, if $n \geq d + 2$ we again obtain

¹¹ Recall that a set is k -connected if it is connected and has vanishing homotopy in dimension 1 to k .

a contradiction (the existence of two disjoint simplices σ and τ such that $f(\sigma) \cap f(\tau) \neq \emptyset$ whereas $U_\sigma \cap U_\tau = \bigcap \mathcal{F} = \emptyset$), and the cardinality of \mathcal{F} must be at most $d + 1$.

3.2 From homotopy to homology

The previous reasoning can be transposed to homology as follows. Assume that for $i = 0, 1, \dots, k - 1$ and all subfamilies $\mathcal{G} \subsetneq \mathcal{F}$ we have $\tilde{\beta}_i(\bigcap \mathcal{G}) = 0$. We construct a nontrivial¹² chain map f from the simplicial chains of $\Delta_{n-1}^{(k)}$ to the singular chains of \mathbb{R}^d by increasing dimension:

- For every $\{i\} \subset [n]$ we let $p_i \in U_{\{i\}}$. This is possible since every intersection of $n - 1$ members of \mathcal{F} is nonempty. We then put $f(\{i\}) = p_i$ and extend it by linearity into a chain map from $\Delta_{n-1}^{(0)}$ to \mathbb{R}^d . Notice that f is nontrivial and that for any 0-simplex $\sigma \subseteq [n]$, the support of $f(\sigma)$ is contained in U_σ .
- Now, assume, as an induction hypothesis, that there exists a nontrivial chain map f from the simplicial chains of $\Delta_{n-1}^{(\ell)}$ to the singular chains of \mathbb{R}^d with the property that for any $(\leq \ell)$ -simplex $\sigma \subseteq [n]$, $\ell < k$, the support of $f(\sigma)$ is contained in U_σ . Let σ be a $(\ell + 1)$ -simplex in $\Delta_{n-1}^{(\ell+1)}$. For every ℓ -dimensional face τ of σ , the support of $f(\tau)$ is contained in $U_\tau \subseteq U_\sigma$. It follows that the support of $f(\partial\sigma)$ is contained in U_σ , which has trivial homology in dimension $\ell + 1$. As a consequence, $f(\partial\sigma)$ is a boundary in U_σ . We can therefore extend f to every simplex of dimension $\ell + 1$ and then, by linearity, to a chain map from the simplicial chains of $\Delta_{n-1}^{(\ell+1)}$ to the singular chains of \mathbb{R}^d . This chain map remains nontrivial and, by construction, for any $(\leq \ell + 1)$ -simplex $\sigma \subseteq [n]$, the support of $f(\sigma)$ is contained in U_σ .

If σ and τ are disjoint simplices of $\Delta_{n-1}^{(k)}$ then the intersection of the supports of $f(\sigma)$ and $f(\tau)$ is contained in $U_\sigma \cap U_\tau = \bigcap \mathcal{F} = \emptyset$ and these supports are disjoint. It follows that f is not only a nontrivial chain map, but also a homological almost-embedding in \mathbb{R}^d . We can then use obstructions to the existence of homological almost-embeddings to bound the cardinality of \mathcal{F} . Specifically, since we assumed that \mathcal{F} has empty intersection and any proper subfamily of \mathcal{F} has nonempty intersection, Corollary 7 implies:

► **Corollary 9.** *Let \mathcal{F} be a family of subsets of \mathbb{R}^d such that $\tilde{\beta}_i(\bigcap \mathcal{G}) = 0$ for every $\mathcal{G} \subsetneq \mathcal{F}$ and $i = 0, 1, \dots, \lceil d/2 \rceil - 1$. Then the Helly number of \mathcal{F} is at most $d + 2$.*

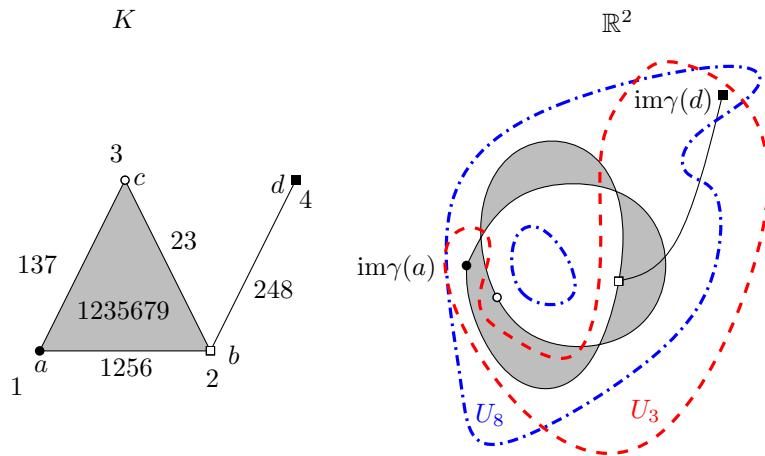
The homological Radon lemma (Lemma 8) yields:

► **Corollary 10.** *Let \mathcal{F} be a family of subsets of \mathbb{R}^d such that $\tilde{\beta}_i(\bigcap \mathcal{G}) = 0$ for every $\mathcal{G} \subsetneq \mathcal{F}$ and $i = 0, 1, \dots, d - 1$. Then the Helly number of \mathcal{F} is at most $d + 1$.*

The examples showing, in the introduction, that Theorem 1 is qualitatively sharp can be modified to show that the previous Corollaries are also sharp in various ways.

First assume that for some values k, n there exists an embedding f of $\Delta_{n-1}^{(k)}$ into \mathbb{R}^d . Let K_i be the simplicial complex obtained by deleting the i th vertex of $\Delta_{n-1}^{(k)}$ (as well as all simplices using that vertex) and put $U_i := f(K_i)$. The family $\mathcal{F} = \{U_1, \dots, U_n\}$ has Helly number exactly n , since it has empty intersection and all its proper subfamilies have nonempty intersection. Moreover, for every $\mathcal{G} \subseteq \mathcal{F}$, $\bigcap \mathcal{G}$ is the image through f of the k -skeleton of a simplex on $|\mathcal{F} \setminus \mathcal{G}|$ vertices, and therefore $\tilde{\beta}_i(\bigcap \mathcal{G}) = 0$ for every $\mathcal{G} \subseteq \mathcal{F}$ and $i = 0, \dots, k - 1$.

¹²See Definition 5.



■ **Figure 2** An example of a constrained map $\gamma: K \rightarrow \mathbb{R}^2$. A label at a face σ of K denotes $\Phi(\sigma)$. Note, for example, that the support of $\gamma(\{a, b, c\})$ needn't be a triangle since we work with chain maps. Constrains by Φ mean that a set U_i must contain cover images of all faces without label i . It is demonstrated by U_3 and U_8 for example.

Such an embedding exists when $k = d$ and $n = d + 1$, as the d -dimensional simplex easily embeds into \mathbb{R}^d . Consequently, the bound of $d + 1$ is best possible under the assumptions of Corollary 10. Such an embedding also exists for $k = d - 1$ and $n = d + 2$, as we can first embed the $(d - 1)$ -skeleton of the d -simplex linearly, then add an extra vertex at the barycentre of the vertices of that simplex and embed the remaining faces linearly. This implies that if we relax the condition of Corollary 10 by only controlling the first $d - 2$ Betti numbers then the bound of $d + 1$ becomes false. It also implies that the bound of $d + 2$ is best possible under (a strengthening of) the assumptions of Corollary 9.

Constrained chain map. Let us formalize the technique illustrated by the previous example. We focus on the homological setting, as this is what we use to prove Theorem 1, but this can be easily transposed to homotopy.

As above, we have a family $\mathcal{F} = \{U_1, U_2, \dots, U_n\}$ of subsets of \mathbb{R}^d and we keep the notation for $U_{\overline{I}}$ introduced in the beginning of this section.

Let K be a simplicial complex and let $\gamma: C_*(K) \rightarrow C_*(\mathbb{R}^d)$ be a chain map from the simplicial chains of K to the singular chains of \mathbb{R}^d . We say that γ is *constrained by* (\mathcal{F}, Φ) if:

- (i) Φ is a map from K to $2^{[n]}$ such that $\Phi(\sigma \cap \tau) = \Phi(\sigma) \cap \Phi(\tau)$ for all $\sigma, \tau \in K$ and $\Phi(\emptyset) = \emptyset$.
- (ii) For any simplex $\sigma \in K$, the support of $\gamma(\sigma)$ is contained in $U_{\overline{\Phi(\sigma)}}$.

See Figure 2. We also say that a chain map γ from K is *constrained by* \mathcal{F} if there exists a map Φ such that γ is constrained by (\mathcal{F}, Φ) . In the above constructions, we simply set Φ to be the identity. As we already saw, constrained chain maps relate Helly numbers to homological almost-embeddings (see Definition 5) via the following observation:

► **Lemma 11.** *Let $\gamma: C_*(K) \rightarrow C_*(\mathbb{R}^d)$ be a nontrivial chain map constrained by \mathcal{F} . If $\bigcap \mathcal{F} = \emptyset$ then γ is a homological almost-embedding of K .*

Proof. Let $\Phi: K \rightarrow 2^{[n]}$ be such that γ is constrained by (\mathcal{F}, Φ) . Since γ is nontrivial, it remains to check that disjoint simplices are mapped to chains with disjoint support. Let

σ and τ be two disjoint simplices of K . The supports of $\gamma(\sigma)$ and $\gamma(\tau)$ are contained, respectively, in $U_{\overline{\Phi(\sigma)}}$ and $U_{\overline{\Phi(\tau)}}$, and

$$U_{\overline{\Phi(\sigma)}} \cap U_{\overline{\Phi(\tau)}} = U_{\overline{\Phi(\sigma) \cap \Phi(\tau)}} = U_{\overline{\Phi(\sigma \cap \tau)}} = U_{\overline{\Phi(\emptyset)}} = U_{\emptyset} = \bigcap \mathcal{F}.$$

Therefore, if $\bigcap \mathcal{F} = \emptyset$ then γ is a homological almost-embedding of K . ◀

3.3 Relaxing the connectivity assumption

In all the examples listed so far, the intersections $\bigcap \mathcal{G}$ must be connected. Matoušek [19] relaxed this condition into “having a bounded number of connected components”, the assumptions then being on the topology of the components, by using Ramsey’s theorem. The gist of our proof is to extend his idea to allow a bounded number of homology classes not only in the first dimension but in *any* dimension. Let us illustrate how Matoušek’s idea works in dimension two:

► **Theorem 12** ([19, Theorem 2 with $d = 2$]). *For every positive integer b there is an integer $h(b)$ with the following property. If \mathcal{F} is a finite family of subsets of \mathbb{R}^2 such that the intersection of any subfamily has at most b path-connected components, then the Helly number of \mathcal{F} is at most $h(b)$.*

Let us fix b from above and assume that for any subfamily $\mathcal{G} \subseteq \mathcal{F}$ the intersection $\bigcap \mathcal{G}$ consists of at most b path-connected components and that $\bigcap \mathcal{F} = \emptyset$. We start, as before, by picking for every $i \in [n]$, a point p_i in $U_{\overline{\{i\}}}$. This is possible as every intersection of $n - 1$ members of \mathcal{F} is nonempty. Now, if we consider some pair of indices $i, j \in [n]$, the points p_i and p_j are still in $U_{\overline{\{i,j\}}}$ but may lie in different connected components. It may thus not be possible to connect p_i to p_j inside $U_{\overline{\{i,j\}}}$. If we, however, consider $b + 1$ indices i_1, i_2, \dots, i_{b+1} then all the points $p_{i_1}, p_{i_2}, \dots, p_{i_{b+1}}$ are in $U_{\overline{\{i_1, i_2, \dots, i_{b+1}\}}}$ which has at most b connected components, so at least one pair among of these points can be connected by a path inside $U_{\overline{\{i_1, i_2, \dots, i_{b+1}\}}}$. Thus, while we may not get a drawing of the complete graph on n vertices we can still draw many edges.

To find many vertices among which every pair can be connected we will use the hypergraph version of the classical theorem of Ramsey:

► **Theorem 13** (Ramsey [23]). *For any x, y and z there is an integer $R_x(y, z)$ such that any x -uniform hypergraph on at least $R_x(y, z)$ vertices colored with at most y colors contains a subset of z vertices inducing a monochromatic sub-hypergraph.*

From the discussion above, for any $b + 1$ indices $i_1 < i_2 < \dots < i_{b+1}$ there exists a pair $\{k, \ell\} \in \binom{[b+1]}{2}$ such that p_{i_k} and p_{i_ℓ} can be connected inside $U_{\overline{\{i_1, i_2, \dots, i_{b+1}\}}}$. Let us consider the $(b + 1)$ -uniform hypergraph on $[n]$ and color every set of indices $i_1 < i_2 < \dots < i_{b+1}$ by one of the pairs in $\binom{[b+1]}{2}$ that can be connected inside $U_{\overline{\{i_1, i_2, \dots, i_{b+1}\}}}$ (if more than one pair can be connected, we pick one arbitrarily). Let t be some integer to be fixed later. By Ramsey’s theorem, if $n \geq R_{b+1} \left(\binom{[b+1]}{2}, t \right)$ then there exist a pair $\{k, \ell\} \in \binom{[b+1]}{2}$ and a subset $T \subseteq [n]$ of size t with the following property: for any $(b + 1)$ -element subset $S \subset T$, the points whose indices are the k th and ℓ th indices of S can be connected inside $U_{\overline{S}}$.

Now, let us set $t = 5 + \binom{5}{2}(b - 1) = 10b - 5$. We claim that we can find five indices in T , denoted i_1, i_2, \dots, i_5 , and, for each pair $\{i_u, i_v\}$ among these five indices, some $(b + 1)$ -element subset $Q_{u,v} \subset T$ with the following properties:

- (i) i_u and i_v are precisely in the k th and ℓ th position in $Q_{u,v}$, and
- (ii) for any $1 \leq u, v, u', v' \leq 5$, $Q_{u,v} \cap Q_{u',v'} = \{i_u, i_v\} \cap \{i_{u'}, i_{v'}\}$.

We first conclude the argument, assuming that we can obtain such indices and sets. Observe that from the construction of T , the i_u 's and the $Q_{u,v}$'s we have the following property: for any $u, v \in [5]$, we can connect p_{i_u} and p_{i_v} inside $U_{Q_{u,v}}$. This gives a drawing of K_5 in the plane. Since K_5 is not planar, there exist two edges with no vertex in common, say $\{u, v\}$ and $\{u', v'\}$, that cross. This intersection point must lie in

$$U_{Q_{u,v}} \cap U_{Q_{u',v'}} = U_{Q_{u,v} \cap Q_{u',v'}} = U_{\{i_u, i_v\} \cap \{i_{u'}, i_{v'}\}} = U_\emptyset = \bigcap \mathcal{F} = \emptyset,$$

a contradiction. Hence the assumption that $n \geq R_{b+1} \binom{b+1}{2}, t$ is false and \mathcal{F} has cardinality at most $R_{b+1} \binom{b+1}{2}, 10b - 5 - 1$, which is our $h(b)$.

The selection trick. It remains to derive the existence of the i_u 's and the $Q_{u,v}$'s. It is perhaps better to demonstrate the method by a simple example to develop some intuition before we formalize it.

Example. Let us fix $b = 4$ and $\{k, \ell\} = \{2, 3\} \in \binom{[4+1]}{2}$. We first make a ‘blueprint’ for the construction inside the rational numbers. For any two indices $u, v \in [5]$ we form a totally ordered set $Q'_{u,v} \subseteq \mathbb{Q}$ of size $b+1 = 5$ by adding three rational numbers (different from $1, \dots, 5$) to the set $\{u, v\}$ in such a way that u appears at the second and v at the third position of $Q'_{u,v}$. For example, we can set $Q'_{1,4}$ to be $\{0.5; 1; 4; 4.7; 5.13\}$. Apart from this we require that we add a different set of rational numbers for each $\{u, v\}$. Thus $Q'_{u,v} \cap Q'_{u',v'} = \{u, v\} \cap \{u', v'\}$. Our blueprint now appears inside the set $T' := \bigcup_{1 \leq u < v \leq 5} Q'_{u,v}$; note that both this set T' and the set T in which we search for the sets $Q_{u,v}$ have 35 elements. To obtain the required indices i_u and sets $Q_{u,v}$ it remains to consider the unique strictly increasing bijection $\pi_0 : T' \rightarrow T$ and set $i_u := \pi_0(u)$ and $Q_{u,v} := \pi_0(Q'_{u,v})$.

The general case. Let us now formalize the generalization of this trick that we will use to prove Theorem 1. Let Q be a subset of $[w]$. If $e_1 < e_2 < \dots < e_w$ are the elements of a totally ordered set W then we call $\{e_i : i \in Q\}$ the *subset selected by Q in W* .

► **Lemma 14.** *Let $1 \leq q \leq w$ be integers and let Q be a subset of $[w]$ of size q . Let Y and Z be two finite totally ordered sets and let A_1, A_2, \dots, A_r be q -element subsets of Y . If $|Z| \geq |Y| + r(w - q)$, then there exist an injection $\pi : Y \rightarrow Z$ and r subsets $W_1, W_2, \dots, W_r \in \binom{Z}{w}$ such that for every $i \in [r]$, Q selects $\pi(A_i)$ in W_i . We can further require that $W_i \cap W_j = \pi(A_i \cap A_j)$ for any two $i, j \in [r]$, $i \neq j$.*

We refer to [11, Lemma 24] for a proof.

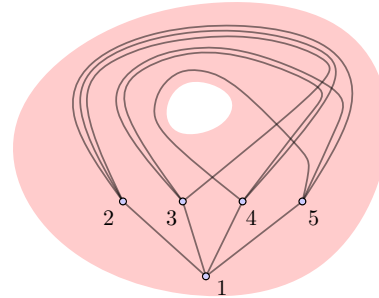
4 Constrained chain maps and Helly number

We now generalize the technique presented in Section 3 to obtain Helly-type theorems from non-embeddability results. We will construct constrained chain maps for arbitrary complexes. As above, $\mathcal{F} = \{U_1, U_2, \dots, U_n\}$ denotes a family of subsets of \mathbb{R}^d and for $I \subseteq [n]$ we keep the notation $U_{\bar{I}}$ as used in the previous section. Note that although so far we only used the *reduced* Betti numbers $\tilde{\beta}$, in this section it will be convenient to work with *standard* (non-reduced) Betti numbers β , starting with the following proposition.

► **Proposition 15.** *For any finite simplicial complex K and non-negative integer b there exists a constant $h_K(b)$ such that the following holds. For any finite family \mathcal{F} of at least $h_K(b)$ subsets of \mathbb{R}^d such that $\bigcap \mathcal{G} \neq \emptyset$ and $\beta_i(\bigcap \mathcal{G}) \leq b$ for any $\mathcal{G} \subsetneq \mathcal{F}$ and any $0 \leq i < \dim K$, there exists a nontrivial chain map $\gamma : C_*(K) \rightarrow C_*(\mathbb{R}^d)$ that is constrained by \mathcal{F} .*

Sketch of proof. We proceed by induction on the dimension k of K . The case $k = 0$ is straightforward. The case $k = 1$ is still slightly easier than the general case. It is a translation of Matoušek’s [19] approach, as summarized in Section 3.3 for the 2-dimensional case, in the language of constrained chain maps.

In the general case, we start with the $(k - 1)$ -skeleton of a (possibly large) simplicial complex and need to find enough $(k - 1)$ -cycles that are boundaries to build the k -skeleton of K . The difficulty of finding those boundaries is illustrated by the following observation: for arbitrarily large n , there exist maps from the complete graph on n vertices into an annulus such that no triangle is a boundary (see the figure on the right for an example with $n = 5$). We may thus have to examine complicated cycles when searching for boundaries, and the general case therefore requires new ingredients.



Let s denote some large enough integer, depending on the $(k - 1)$ -skeleton $K^{(k-1)}$. Assuming $h_k(b)$ is large enough, there exists, by the induction assumption, a nontrivial chain map $\gamma' : C_*(\Delta_s^{(k-1)}) \rightarrow C_*(\mathbb{R}^d)$ constrained by \mathcal{F} . One may hope to use Ramsey’s theorem to find a copy of $K^{(k-1)}$ inside $\Delta_s^{(k-1)}$ on which γ' can be extended to a constrained chain map on all of K . This turns out to be impossible: while Ramsey’s theorem may guarantee that the boundaries of the k -dimensional simplices all have the same homology class, we cannot prevent that common homology class to be non-zero!

We overcome this issue by considering the barycentric subdivision $\text{sd } K$ of K and finding a suitable injection β of $(\text{sd } K)^{(k-1)}$ into $\Delta_s^{(k-1)}$. We then consider the chain maps

$$C_*(K^{(k-1)}) \xrightarrow{\alpha} C_*((\text{sd } K)^{(k-1)}) \xrightarrow{\beta_{\sharp}} C_*(\Delta_s^{(k-1)}) \xrightarrow{\gamma'} C_*(\mathbb{R}^d)$$

where α is a natural chain map corresponding to the subdivision and β_{\sharp} is the chain map induced by β . We set $\gamma = \gamma' \circ \beta_{\sharp} \circ \alpha$. We use Ramsey’s theorem to set β in such a way that the boundaries of the k -dimensional simplices all have the same homology class under γ . Again, Ramsey’s theorem can only ensure that through $\gamma' \circ \beta_{\sharp}$ the boundaries of the k -simplices of $\text{sd } K$ have the same, possibly non-trivial, homology class. But since every k -simplex of K is the sum of an *even* number of simplices of $\text{sd } K$, and we compute homology over \mathbb{Z}_2 , this is good enough, and γ can be extended to K .

This outline brushes under the rug some technical difficulties raised by the use of barycentric subdivision; we refer the interested reader to [11, Proposition 25] for full details. ◀

The case $K = \Delta_{2k+2}^{(k)}$, with $k = \lceil d/2 \rceil$, of Proposition 15 finally implies Theorem 1.

Proof of Theorem 1. Let b and d be fixed integers, let $k = \lceil d/2 \rceil$ and let $K = \Delta_{2k+2}^{(k)}$. Let $h_K(b + 1)$ denote the constant from Proposition 15 (we plug in $b + 1$ because we need to switch between reduced and non-reduced Betti numbers). Let \mathcal{F} be a finite family of subsets of \mathbb{R}^d such that $\tilde{\beta}_i(\cap \mathcal{G}) \leq b$ for any $\mathcal{G} \subsetneq \mathcal{F}$ and every $0 \leq i \leq \dim K = \lceil d/2 \rceil - 1$, in particular $\beta_i(\cap \mathcal{G}) \leq b + 1$ for such \mathcal{G} . Let \mathcal{F}^* denote an inclusion-minimal sub-family of \mathcal{F} with empty intersection: $\cap \mathcal{F}^* = \emptyset$ and $\cap (\mathcal{F}^* \setminus \{U\}) \neq \emptyset$ for any $U \in \mathcal{F}^*$. If \mathcal{F}^* has size at least $h_K(b + 1)$, it satisfies the assumptions of Proposition 15 and there exists a

nontrivial chain map from K that is constrained by \mathcal{F}^* . Since \mathcal{F}^* has empty intersection, this chain map is a homological almost-embedding by Lemma 11. However, no such homological almost-embedding exists by Corollary 7, so \mathcal{F}^* must have size at most $h_K(b+1) - 1$. As a consequence, the Helly number of \mathcal{F} is bounded and the statement of Theorem 1 holds with $h(b, d) = h_K(b+1) - 1$. ◀

Acknowledgments. We would like to express our immense gratitude to Jiří Matoušek, not only for raising the problem addressed in the present paper and valuable discussions about it, but, much more generally, for the privilege of having known him, as our teacher, mentor, collaborator, and friend. Through his tremendous depth and insight, and the generosity with which he shared them, he greatly influenced all of us.

We further thank Jürgen Eckhoff for helpful comments on a preliminary version of the paper, and Andreas Holmsen and Gil Kalai for providing us with useful references.

References

- 1 N. Amenta. Helly-type theorems and generalized linear programming. *Discrete & Computational Geometry*, 12:241–261, 1994.
- 2 N. Amenta. A short proof of an interesting Helly-type theorem. *Discrete & Computational Geometry*, 15:423–427, 1996.
- 3 E. G. Bajmóczy and I. Bárány. On a common generalization of Borsuk’s and Radon’s theorem. *Acta Math. Acad. Sci. Hungar.*, 34(3-4):347–350, 1979.
- 4 M. Bestvina, M. Kapovich, and B. Kleiner. Van Kampen’s embedding obstruction for discrete groups. *Invent. Math.*, 150(2):219–235, 2002.
- 5 A. Björner. Nerves, fibers and homotopy groups. *Journal of Combinatorial Theory, Series A*, 102(1):88–93, 2003.
- 6 K. Borsuk. On the imbedding of systems of compacta in simplicial complexes. *Fundamenta Mathematicae*, 35:217–234, 1948.
- 7 É. Colin de Verdière, G. Ginot, and X. Goaoc. Multinerves and Helly numbers of acyclic families. In *Proceedings of the 2012 symposium on Computational Geometry*, SoCG’12, pages 209–218, 2012. <http://arxiv.org/abs/1101.6006>.
- 8 H. Debrunner. Helly type theorems derived from basic singular homology. *American Mathematical Monthly*, 77:375–380, 1970.
- 9 J. Eckhoff. Helly, Radon and Carathéodory type theorems. In P.M. Gruber and J.M. Wills, editors, *Handbook of Convex Geometry*, pages 389–448. North Holland, 1993.
- 10 A. I. Flores. Über die Existenz n -dimensionaler Komplexe, die nicht in den \mathbb{R}^{2n} topologisch einbettbar sind. *Ergeb. Math. Kolloqu.*, 5:17–24, 1933.
- 11 X. Goaoc, P. Paták, Z. Patáková, M. Tancer, and U. Wagner. Bounding Helly numbers via Betti numbers. <http://arxiv.org/abs/1310.4613>.
- 12 A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, UK, 2002.
- 13 E. Helly. Über Mengen konvexer Körper mit gemeinschaftlichen Punkten. *Jahresbericht Deutsch. Math. Verein.*, 32:175–176, 1923.
- 14 E. Helly. Über Systeme von abgeschlossenen Mengen mit gemeinschaftlichen Punkten. *Monats. Math. und Physik*, 37:281–302, 1930.
- 15 G. Kalai. Combinatorial expectations from commutative algebra. In I. Peeva and V. Welker, editors, *Combinatorial Commutative Algebra*, volume 1(3), pages 1729–1734. Oberwolfach Reports, 2004.
- 16 G. Kalai and R. Meshulam. Leray numbers of projections and a topological Helly-type theorem. *Journal of Topology*, 1:551–556, 2008.

- 17 H. Maehara. Helly-type theorems for spheres. *Discrete & Computational Geometry*, 4(1):279–285, 1989.
- 18 J. Matoušek. *Using the Borsuk-Ulam Theorem*. Springer-Verlag, Berlin, 2003.
- 19 J. Matoušek. A Helly-type theorem for unions of convex sets. *Discrete & Computational Geometry*, 18:1–12, 1997.
- 20 S. A. Melikhov. The van Kampen obstruction and its relatives. *Proc. Steklov Inst. Math.*, 266(1):142–176, 2009.
- 21 L. Montejano. A new topological Helly theorem and some transversal results. *Discrete & Computational Geometry*, 52(2):390–398, 2014.
- 22 J. R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Menlo Park, CA, 1984.
- 23 F. P. Ramsey. On a problem in formal logic. *Proc. London Math. Soc.*, 30:264—286, 1929.
- 24 A. Shapiro. Obstructions to the imbedding of a complex in a euclidean space. I. The first obstruction. *Ann. of Math. (2)*, 66:256–269, 1957.
- 25 M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. 9th Sympos. on Theo. Aspects of Comp. Science*, pages 569–579, 1992.
- 26 R. I. Soare. Computability theory and differential geometry. *Bull. Symbolic Logic*, 10(4):457–486, 2004.
- 27 M. Tancer. Intersection patterns of convex sets via simplicial complexes: A survey. In J. Pach, editor, *Thirty Essays on Geometric Graph Theory*, pages 521–540. Springer New York, 2013.
- 28 E. R. van Kampen. Komplexe in euklidischen Räumen. *Abh. Math. Sem. Univ. Hamburg*, 9:72–78, 1932.
- 29 U. Wagner. Minors in random and expanding hypergraphs. In *Proceedings of the 27th Annual Symposium on Computational Geometry (SoCG)*, pages 351—360, 2011.
- 30 R. Wenger. Helly-type theorems and geometric transversals. In Jacob E. Goodman and Joseph O’Rourke, editors, *Handbook of Discrete & Computational Geometry*, chapter 4, pages 73–96. CRC Press LLC, Boca Raton, FL, 2nd edition, 2004.
- 31 W.-T. Wu. On the realization of complexes in euclidean spaces. I, II, III. *Acta Math. Sinica (English transl. of I and III in Sci. Sinica)*, 5:505–552, 1955.

Polynomials Vanishing on Cartesian Products: The Elekes–Szabó Theorem Revisited*

Orit E. Raz¹, Micha Sharir¹, and Frank de Zeeuw²

- 1 School of Computer Science, Tel Aviv University
Tel Aviv 69978, Israel
{oritraz,michas}@post.tau.ac.il
- 2 École Polytechnique Fédérale de Lausanne
Lausanne, Switzerland
fdezeeuw@gmail.com

Abstract

Let $F \in \mathbb{C}[x, y, z]$ be a constant-degree polynomial, and let $A, B, C \subset \mathbb{C}$ with $|A| = |B| = |C| = n$. We show that F vanishes on at most $O(n^{11/6})$ points of the Cartesian product $A \times B \times C$ (where the constant of proportionality depends polynomially on the degree of F), unless F has a special group-related form. This improves a theorem of Elekes and Szabó [2], and generalizes a result of Raz, Sharir, and Solymosi [9]. The same statement holds over \mathbb{R} . When A, B, C have different sizes, a similar statement holds, with a more involved bound replacing $O(n^{11/6})$.

This result provides a unified tool for improving bounds in various Erdős-type problems in combinatorial geometry, and we discuss several applications of this kind.

1998 ACM Subject Classification G.2 Discrete Mathematics

Keywords and phrases Combinatorial geometry, incidences, polynomials

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.522

1 Introduction

In 2000, Elekes and Rónyai [1] proved the following result. Given a constant-degree real polynomial $f(x, y)$, and finite sets $A, B, C \subset \mathbb{R}$ each of size n , we have

$$|\{(x, y, z) \in \mathbb{R}^3 \mid z - f(x, y) = 0\} \cap (A \times B \times C)| = o(n^2),$$

unless f has one of the forms $f(x, y) = g(h(x) + k(y))$ or $f(x, y) = g(h(x)k(y))$, with univariate real polynomials g, h, k . Recently, Raz, Sharir, and Solymosi [9] extended an argument introduced in [11] to improve the upper bound (when f does not have one of the special forms) to $O(n^{11/6})$ (where the constant of proportionality depends polynomially on the degree of f).

Elekes and Szabó [2] generalized the result of [1] to any complex algebraic surface

$$Z(F) := \{(x, y, z) \in \mathbb{C}^3 \mid F(x, y, z) = 0\},$$

* Work on this paper by Orit E. Raz and Micha Sharir was supported by Grant 892/13 from the Israel Science Foundation and by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11). Work by Micha Sharir was also supported by Grant 2012/229 from the U.S.–Israel Binational Science Foundation and by the Hermann Minkowski-MINERVA Center for Geometry at Tel Aviv University. Work on this paper by Frank de Zeeuw was partially supported by Swiss National Science Foundation Grants 200020-144531 and 200021-137574.



where F is an irreducible polynomial in $\mathbb{C}[x, y, z]$. They showed that if $A, B, C \subset \mathbb{C}$ are finite sets, each of size n , then $|Z(F) \cap (A \times B \times C)|$ is subquadratic in n , unless F has a certain exceptional form. The exceptional form of F in this statement is harder to describe (see (ii) in Theorem 1.1 below), but is related to an underlying group structure that describes the dependencies of F on each of the variables (similar to the addition or multiplication that appear in the exceptional forms of $F(x, y, z) = z - f(x, y)$ in [1, 9]). The upper bound that Elekes and Szabó obtained, when F is not exceptional, was $|Z(F) \cap (A \times B \times C)| = O(n^{2-\eta})$, for a constant $\eta > 0$ that depends on the degree of F , and which they did not make explicit.

Our results. In this paper, we show that the theorem of Elekes and Szabó holds for $\eta = 1/6$, thereby extending the strengthened result of [9] to the generalized setup in [2]. More precisely, our main result is the following theorem.

► **Theorem 1.1 (Balanced case).** *Let $F \in \mathbb{C}[x, y, z]$ be an irreducible polynomial of degree d , and assume that none of the derivatives $\partial F/\partial x, \partial F/\partial y, \partial F/\partial z$ is identically zero. Then one of the following two statements holds.*

(i) *For all $A, B, C \subset \mathbb{C}$ with $|A| = |B| = |C| = n$ we have*

$$|Z(F) \cap (A \times B \times C)| = O(d^{13/2}n^{11/6}).$$

(ii) *There exists a one-dimensional subvariety $Z_0 \subset Z(F)$, such that for every $v \in Z(F) \setminus Z_0$, there exist open sets $D_1, D_2, D_3 \subset \mathbb{C}$ and analytic functions $\varphi_i : D_i \rightarrow \mathbb{C}$ for $i = 1, 2, 3$, such that $v \in D_1 \times D_2 \times D_3$, and, for every $(x, y, z) \in D_1 \times D_2 \times D_3$,*

$$(x, y, z) \in Z(F) \quad \text{if and only if} \quad \varphi_1(x) + \varphi_2(y) + \varphi_3(z) = 0.$$

When property (ii) holds, property (i) fails. Indeed, consider any $v = (x_0, y_0, z_0)$ and φ_i, D_i as in property (ii). If we set $t_1 = \varphi_1(x_0)$, $t_2 = \varphi_2(y_0)$, and $t_3 = \varphi_3(z_0)$, then we have $t_1 + t_2 + t_3 = 0$. Now choose $A \subset D_1$, $B \subset D_2$, and $C \subset D_3$ so that $\varphi_1(A) = \{t_1 + a, t_1 + 2a, \dots, t_1 + na\}$, $\varphi_2(B) = \{t_2 + a, t_2 + 2a, \dots, t_2 + na\}$, and $\varphi_3(C) = \{t_3 - a, t_3 - 2a, \dots, t_3 - na\}$; this is clearly possible for $a \in \mathbb{C}$ with a sufficiently small absolute value. Then $|Z(F) \cap (A \times B \times C)| \geq n^2/4$.

Our proof also works when the sets A, B, C do not have the same size. Such an “unbalanced” form was not considered in [1] or [2], but similar unbalanced bounds were obtained in [9], and they are useful in applications where the roles of A, B, C are not symmetric. We obtain the following result, which subsumes Theorem 1.1; we have stated both for clarity.

► **Theorem 1.2 (Unbalanced case).** *In Theorem 1.1, property (i) can be replaced by:*

(i*) *For all triples $A, B, C \subset \mathbb{C}$ of finite sets, we have*

$$|Z(F) \cap (A \times B \times C)| = O\left(\min\left\{d^{\frac{13}{2}}|A|^{\frac{1}{2}}|B|^{\frac{2}{3}}|C|^{\frac{2}{3}} + d^{\frac{17}{2}}|A|^{\frac{1}{2}}\left(|A|^{\frac{1}{2}} + |B| + |C|\right),\right.\right. \\ \left.\left. d^{\frac{13}{2}}|B|^{\frac{1}{2}}|A|^{\frac{2}{3}}|C|^{\frac{2}{3}} + d^{\frac{17}{2}}|B|^{\frac{1}{2}}\left(|B|^{\frac{1}{2}} + |A| + |C|\right),\right.\right. \\ \left.\left. d^{\frac{13}{2}}|C|^{\frac{1}{2}}|A|^{\frac{2}{3}}|B|^{\frac{2}{3}} + d^{\frac{17}{2}}|C|^{\frac{1}{2}}\left(|C|^{\frac{1}{2}} + |A| + |B|\right)\right\}\right).$$

We also have the following specialization of Theorem 1.2 when F is a real polynomial. Note that, when F is real, it does not immediately follow from Theorems 1.1 and 1.2 that, in property (ii) there, the functions φ_i can be chosen so that they map \mathbb{R} to \mathbb{R} . We write $Z_{\mathbb{R}}(F)$ for the real zero set of a real polynomial defined over \mathbb{R} .

► **Theorem 1.3** (Real case). *Let $F \in \mathbb{R}[x, y, z]$ be a polynomial of degree d that is irreducible over \mathbb{R} . Assume that $Z_{\mathbb{R}}(F)$ has dimension two. Then property (ii) in both Theorems 1.1 and 1.2 can be replaced by:*

(ii) $_{\mathbb{R}}$ *There exists a one-dimensional subvariety $Z_0 \subset Z_{\mathbb{R}}(F)$ (whose degree is polynomial in d), such that for every $v \in Z_{\mathbb{R}}(F) \setminus Z_0$, there exist open intervals $I_1, I_2, I_3 \subset \mathbb{R}$, and real-analytic functions $\varphi_i : I_i \rightarrow \mathbb{R}$ for $i = 1, 2, 3$, such that $v \in I_1 \times I_2 \times I_3$, and, for every $(x, y, z) \in I_1 \times I_2 \times I_3$,*

$$(x, y, z) \in Z(F) \quad \text{if and only if} \quad \varphi_1(x) + \varphi_2(y) + \varphi_3(z) = 0.$$

The proof of Theorem 1.3 is omitted in this version.

Discussion. Although the results in this paper generalize those of Raz et al. [9], the analysis here is quite different and considerably more involved. The overlap between the two studies is only in the initial reduction of the problem to an incidence problem between points and curves (see below). The remaining and major part of the paper applies totally different machinery. Instead of the purely algebraic study of properties of polynomials that was used in [9], the approach here requires more advanced tools from algebraic geometry, and applies them in a considerably more involved style, inspired in part by a technique used by Tao [14] for a problem in finite fields.

That the current problem is considerably more difficult than the Elekes–Rónyai problem (in spite of their similarities) can also be seen by comparing the original respective studies in [1] and in [2]. We regard the considerable simplification (on top of the improvement in the bound) of the analysis of Elekes and Szabó in [2] as a major outcome of this paper.

We note that the polynomial dependence of our bound on the degree of F is also a significant feature, because it allows us to obtain non-trivial bounds for polynomials of non-constant degree. This arises for example in the application of obtaining lower bounds for the number of distinct distances between points on an algebraic curve (as discussed below), where the bound is still non-trivial when the degree of the curve is non-constant. An improved dependence on d would allow us to treat more general sets of points, and get closer (and perhaps even reconstruct) the general lower bound of Guth and Katz [6].

Consequences. Besides being an interesting problem in itself, the Elekes–Szabó setup arises in many problems in combinatorial geometry. To demonstrate this, consider the problem of obtaining a lower bound for the number of distinct distances determined between three *non-collinear* points p_1, p_2, p_3 and a set P of n other points in the plane, studied in [2, 12]. To cast this problem into the Elekes–Szabó mold, let D denote the set of the squared distances between the points p_i and those of P . Write $p_i = (a_i, b_i)$, for $i = 1, 2, 3$. A point $q = (t, s) \in \mathbb{R}^2$ determines three squared distances to p_1, p_2, p_3 , given by

$$X = (t - a_1)^2 + (s - b_1)^2, \quad Y = (t - a_2)^2 + (s - b_2)^2, \quad Z = (t - a_3)^2 + (s - b_3)^2.$$

Eliminating t and s from these equations yields a quadratic equation $F(X, Y, Z) = 0$. By construction, for each point $q \in P$, each of the corresponding squared distances X, Y, Z belongs to D . Moreover, the resulting triples (X, Y, Z) are all distinct, and so F vanishes at n triples of $D \times D \times D$. Moreover, since p_1, p_2, p_3 are non-collinear, one can show that F does not have the special form in property (ii) $_{\mathbb{R}}$ of Theorem 1.3. So one gets $n = O(|D|^{11/6})$, or $|D| = \Omega(n^{6/11})$, which is the same lower bound obtained in [12], using a direct ad-hoc analysis. Note that for p_1, p_2 , and p_3 collinear, F becomes a linear polynomial, in which

case it certainly satisfies property (ii)_ℝ, and the above bound on $|D|$ does not hold – it can be $\Theta(n^{1/2})$ in this case.

Geometric questions which involve Euclidean distances, slopes, or collinearity often lead to polynomial relations of the form $F(x, y, z) = 0$, and can be reduced to studying the number of zeros of such polynomials attained on a Cartesian product. The following is a sample of problems that fit into this framework: (i) Bounding from below the number of distinct distances [8, 11] determined by a set of n points lying on a planar algebraic curve. (ii) Bounding from above the number of triple intersection points for three families of n unit circles, each consisting of circles that pass through a fixed point [4, 10]. (iii) Bounding from below the number of collinear triples among n points on an algebraic curve in \mathbb{R}^2 [3].

Due to lack of space, many details are omitted in this abstract and are given in the full version of the paper.

2 Proof of Theorem 1.2

In this section we prove Theorem 1.2, up to the crucial Proposition 2.3 that we prove in Section 3. Let $F \in \mathbb{C}[x, y, z]$ be an irreducible polynomial of degree d . Let $A, B, C \subset \mathbb{C}$ be finite, and put $M := |Z(F) \cap (A \times B \times C)|$; this is the quantity we wish to bound. The strategy of the proof is to transform the problem of bounding M into an incidence problem for points and curves in \mathbb{C}^2 . The latter problem can then be tackled using a Szemerédi-Trotter-like incidence bound, *provided* that the resulting curves have well-behaved intersections, in the following sense.

► **Definition 2.1.** We say that a system (Π, Γ) , where Π is a finite set of distinct points in \mathbb{C}^2 , and Γ is a finite multiset of curves in \mathbb{C}^2 , has (λ, μ) -bounded multiplicity if

- (a) for any curve $\gamma \in \Gamma$, there are at most λ curves $\gamma' \in \Gamma$ (counted with multiplicity) such that there are more than μ points contained in both γ and γ' ; and
- (b) for any point $p \in \Pi$, there are at most λ points $p' \in \Pi$ such that there are more than μ curves (counted with multiplicity) that contain both p and p' .

A major component of the proof is to show that if the points and curves that we are about to define fail to satisfy the conditions of (λ, μ) -bounded multiplicity, then $Z(F)$ must have the special form described in property (ii) of Theorem 1.2.

Quadruples. Define $Q := \{(b, b', c, c') \in B^2 \times C^2 \mid \exists a \in A \text{ s.t. } F(a, b, c) = F(a, b', c') = 0\}$. The following inequality bounds M in terms of $|Q|$.

► **Lemma 2.2.** We have $M = O\left(d^{1/2}|A|^{1/2}|Q|^{1/2} + d^2|A|\right)$.

Proof. For each $a \in A$, we write $(B \times C)_a := \{(b, c) \in B \times C \mid F(a, b, c) = 0\}$. Using the Cauchy-Schwarz inequality, we have

$$M = \sum_{a \in A} |(B \times C)_a| \leq |A|^{1/2} \left(\sum_{a \in A} |(B \times C)_a|^2 \right)^{1/2}.$$

Define $R := \{(a, b, b', c, c') \in A \times B^2 \times C^2 \mid F(a, b, c) = F(a, b', c') = 0\}$, and consider the standard projection $\tau : \mathbb{C} \times \mathbb{C}^4 \rightarrow \mathbb{C}^4$ (in which the first coordinate is discarded). We have $Q = \tau(R)$ and $M \leq |A|^{1/2}|R|^{1/2}$.

We claim that $|R| \leq d|Q| + d^4|A|$. To prove this, let

$$S := \{(b, b', c, c') \in B^2 \times C^2 \mid F(a, b, c) \equiv 0 \text{ and } F(a, b', c') \equiv 0 \text{ (as polynomials in } a)\}.$$

We prove in the full version that $|S| = O(d^4)$. Observe that for $(b, b', c, c') \in Q \setminus S$ we have $|\tau^{-1}(b, b', c, c') \cap R| \leq d$, while for $(b, b', c, c') \in S$ we have $|\tau^{-1}(b, b', c, c') \cap R| = |A|$. Thus

$$|R| = |\tau^{-1}(Q)| = |\tau^{-1}(Q \setminus S)| + |\tau^{-1}(S)| \leq d|Q| + d^4|A|,$$

which proves the claim and the lemma. \blacktriangleleft

In what follows, we derive an upper bound on $|Q|$. It will turn out that, when we fail to obtain the bound we are after, F must have the special form in property (ii).

Curves and dual curves. For every point $(y, y') \in \mathbb{C}^2$, we define

$$\gamma_{y, y'} := \text{Cl}(\{(z, z') \in \mathbb{C}^2 \mid \exists x \in \mathbb{C} \text{ such that } F(x, y, z) = F(x, y', z') = 0\}),$$

where $\text{Cl}(X)$ stands for the Zariski closure of X . We show in the full version that there exists an exceptional set $\mathcal{S} \subset \mathbb{C}^2$ of size $O(d^4)$, such that for every $(y, y') \in \mathbb{C}^2 \setminus \mathcal{S}$ the set $\gamma_{y, y'}$ is an algebraic curve of degree at most d^2 , or an empty set (a possibility we can safely ignore).

We define, in an analogous manner, a dual system of curves by switching the roles of the y - and z -coordinates, as follows. For every point $(z, z') \in \mathbb{C}^2$, we define

$$\gamma_{z, z'}^* := \text{Cl}(\{(y, y') \in \mathbb{C}^2 \mid \exists x \in \mathbb{C} \text{ such that } F(x, y, z) = F(x, y', z') = 0\}).$$

As above, here too our (omitted) analysis yields an exceptional set \mathcal{T} of size $O(d^4)$, such that for every $(z, z') \in \mathbb{C}^2 \setminus \mathcal{T}$ the set $\gamma_{z, z'}^*$ is an algebraic curve of degree at most d^2 (or empty).

By a standard argument (omitted here), the closure in the definitions of $\gamma_{y, y'}$ and $\gamma_{z, z'}^*$ adds only finitely many points. It follows that, for all but finitely many points $(z, z') \in \gamma_{y, y'}$, we have $(y, y') \in \gamma_{z, z'}^*$. Symmetrically, for all but finitely many $(y, y') \in \gamma_{z, z'}^*$ we have $(z, z') \in \gamma_{y, y'}$.

We set $m := d^4$ throughout this proof. We say that an irreducible algebraic curve $\gamma \subset \mathbb{C}^2$ is a *popular curve* if there exist at least $m + 1$ distinct points $(y, y') \in \mathbb{C}^2 \setminus \mathcal{S}$ such that $\gamma \subset \gamma_{y, y'}$. We denote by \mathcal{C} the set of all popular curves. Similarly, we say that an irreducible algebraic curve $\gamma^* \subset \mathbb{C}^2$ is a *popular dual curve*, if there exist at least $m + 1$ distinct points $(z, z') \in \mathbb{C}^2 \setminus \mathcal{T}$ such that $\gamma^* \subset \gamma_{z, z'}^*$. We denote by \mathcal{D} the set of all popular dual curves.

The main step in our proof is the following proposition, whose proof takes up Section 3. Note that its statement is only about F and does not involve the specific sets A, B, C .

► **Proposition 2.3.** *Either F satisfies property (ii) of Theorem 1.2, or the following holds.*

- (a) *There exists an algebraic curve $\mathcal{X} \subset \mathbb{C}^2$ of degree $O(d^{11})$ containing \mathcal{S} , such that for every $(y, y') \in \mathbb{C}^2 \setminus \mathcal{X}$, no irreducible component of $\gamma_{y, y'}$ is a popular curve.*
- (b) *There exists an algebraic curve $\mathcal{Y} \subset \mathbb{C}^2$ of degree $O(d^{11})$ containing \mathcal{T} , such that for every $(z, z') \in \mathbb{C}^2 \setminus \mathcal{Y}$, no irreducible component of $\gamma_{z, z'}^*$ is a popular dual curve.*

Incidences. We continue with the analysis, assuming the truth of Proposition 2.3. We introduce the following set of points and *multiset* of curves:

$$\Pi := (C \times C) \setminus \mathcal{Y} \quad \text{and} \quad \Gamma := \{\gamma_{b, b'} \mid (b, b') \in (B \times B) \setminus \mathcal{X}\}.$$

By definition, for every $(b, b', c, c') \in Q$, we have $(c, c') \in \gamma_{b, b'}$ and $(b, b') \in \gamma_{c, c'}^*$ (albeit not necessarily vice versa, because the definition of the curves involves a closure, and does not require x to be in A). This lets us relate $|Q|$ to $I(\Pi, \Gamma)$, the number of incidences between these points and curves; since Γ is a multiset, these incidences are counted with the multiplicity of the relevant curves. Specifically, we show in the full version:

► **Lemma 2.4.** *We have $|Q| \leq I(\Pi, \Gamma) + O(d^{13}|B||C| + d^4|B|^2 + d^4|C|^2)$.*

Bounded multiplicity. We claim that the system (Π, Γ) has (d^6, d^4) -bounded multiplicity. Indeed, by Proposition 2.3(a) and the fact that we have avoided \mathcal{X} when defining Γ , any component of a curve $\gamma \in \Gamma$ is not in \mathcal{C} , and is thus shared with at most $m = d^4$ other curves. The curve γ has at most d^2 irreducible components, so there are at most $md^2 = d^6$ curves $\gamma' \in \Gamma$ such that γ and γ' have a common component. Curves γ' that do not have a common component with γ intersect it in at most d^4 points by Bézout's inequality; thus condition (a) in the definition of (d^6, d^4) -bounded multiplicity is satisfied. The argument for condition (b) is fully symmetric.

Incidence bound. In the full version of this paper we derive an incidence bound, based on that of Solymosi and De Zeeuw [13], resembling the classical Szemerédi-Trotter point-line incidence bound. It applies to a set Π of points and a multiset Γ of algebraic curves, each of degree at most δ , in \mathbb{C}^2 , such that Π is a Cartesian product and (Π, Γ) have (λ, μ) -bounded multiplicity as in Definition 2.1. The analysis culminates in the incidence bound

$$I(\Pi, \Gamma) = O\left(\delta^{4/3}\lambda^{4/3}\mu^{1/3}|\Pi|^{2/3}|\Gamma|^{2/3} + \lambda^2\mu|\Pi| + \delta^4\lambda|\Gamma|\right).$$

Specializing this, with $\delta = d^2$, $\lambda = d^6$, and $\mu = d^4$, we get

$$\begin{aligned} I(\Pi, \Gamma) &= O\left((d^2)^{4/3}(d^6)^{4/3}(d^4)^{1/3}|B|^{4/3}|C|^{4/3} + (d^6)^2d^4|B|^2 + (d^2)^4d^6|C|^2\right) \\ &= O\left(d^{12}|B|^{4/3}|C|^{4/3} + d^{16}|B|^2 + d^{14}|C|^2\right), \end{aligned}$$

which, together with Lemma 2.4, gives

$$|Q| = I(\Pi, \Gamma) + O\left(d^{13}|B||C| + d^4|B|^2 + d^4|C|^2\right) = O\left(d^{12}|B|^{4/3}|C|^{4/3} + d^{16}|B|^2 + d^{14}|C|^2\right).$$

Then, from Lemma 2.2, we get

$$\begin{aligned} M &\leq d^{1/2}|A|^{1/2}|Q|^{1/2} + d^2|A| \\ &= O\left(d^{13/2}|A|^{1/2}|B|^{2/3}|C|^{2/3} + d^{17/2}|A|^{1/2}|B| + d^{15/2}|A|^{1/2}|C| + d^2|A|\right), \end{aligned}$$

which gives the first of the three bounds in Theorem 1.2(i). The other two follow symmetrically.

3 Proof of Proposition 2.3

3.1 Overview of the proof

We adapt an idea used by Tao [14] to study the expansion of a polynomial $P(x, y)$ over finite fields. As part of his analysis he considered the map $\Psi : \mathbb{C}^4 \rightarrow \mathbb{C}^4$ defined by

$$\Psi : (a, b, c, d) \mapsto (P(a, c), P(a, d), P(b, c), P(b, d)).$$

Tao showed that if the image $\Psi(\mathbb{C}^4)$ is four-dimensional, then lower bounds on the expansion of P can be derived. On the other hand, if the image has dimension at most three, then P must have one of the special forms $G(H(x) + K(y))$ or $G(H(x)K(y))$, for polynomials G, H, K (as in [1, 9]; see also the introduction). Tao proved this by observing that in this case the determinant of the Jacobian matrix of Ψ must vanish identically, leading to an identity for the partial derivatives of P , from which the special forms of P can be deduced.

Following Tao's general scheme, albeit in a different context, we define a variety

$$V := \{(x, x', y, y', z_1, z_2, z_3, z_4) \in \mathbb{C}^8 \mid F(x, y, z_1) = F(x, y', z_2) = F(x', y, z_3) = F(x', y', z_4) = 0\}. \quad (1)$$

Note that if we fix y, y' in V and eliminate x, x' , the range of the last four coordinates of V is $\gamma_{y, y'} \times \gamma_{y, y'}$ (up to the closure operation). Near most points $v \in V$, we use the implicit function theorem to represent V as the graph of a locally defined analytic function (which serves as a local analogue of the map Ψ above)

$$\Phi_v : (x, x', y, y') \mapsto (g_1(x, y), g_2(x, y'), g_3(x', y), g_4(x', y')).$$

If the determinant of the Jacobian of Φ_v vanishes at v , for all v in some relatively open subset of V , it leads to the special form of F . This derivation is similar to that of Tao, but our special form requires a somewhat different treatment.

The other side of our argument, when the determinant of the Jacobian is not identically zero, as above, is very different from that of Tao. Here we want to show that there are only finitely many popular curves. (The actual property that we show is somewhat different, but this is the spirit of our analysis.) We show that if γ is a popular curve (i.e., there are more than d^4 curves $\gamma_{y, y'} \in \Gamma$ that contain γ), then it is *infinitely* popular, in the sense that there is a one-dimensional curve γ^* of pairs $(y, y') \in \mathbb{C}^2$ for which $\gamma_{y, y'}$ contains γ . For V , this implies that if we restrict (y, y') to γ^* and project to the last four coordinates, then the image is contained in $\gamma \times \gamma$. In other words, the local map Φ_v sends an open subset of the three-dimensional variety $\mathbb{C}^2 \times \gamma^*$ to an open subset of the two-dimensional variety $\gamma \times \gamma$. The inverse mapping theorem now tells us that the determinant of the Jacobian of Φ_v vanishes on the three-dimensional variety $\mathbb{C}^2 \times \gamma^*$. Given that this determinant is not identically zero, its zero set is three-dimensional, so $\mathbb{C}^2 \times \gamma^*$ must be one of its $O_d(1)$ irreducible components. It follows that there are only $O_d(1)$ popular curves, which essentially establishes Proposition 2.3.

3.2 The varieties V , V_0 and W

Consider the variety $V \subset \mathbb{C}^8$ as defined in (1). V is not empty since, for any point $(x, y, z) \in Z(F)$, it contains (x, x, y, y, z, z, z, z) . It follows that V has dimension at least four; it can in fact be shown that V is four-dimensional. However, our analysis requires that the projection of V to the first four coordinates is four-dimensional, which does not follow directly. We show this in the following lemma. Throughout Section 3 we write $\pi_1 : \mathbb{C}^8 \rightarrow \mathbb{C}^4$ and $\pi_2 : \mathbb{C}^8 \rightarrow \mathbb{C}^4$ for the standard projections onto the first and the last four coordinates, respectively.

► **Lemma 3.1.** *We have $\text{Cl}(\pi_1(V)) = \mathbb{C}^4$.*

Proof. Let $(x_0, x'_0, y_0, y'_0) \in \mathbb{C}^4$. There exist $z_1, z_2, z_3, z_4 \in \mathbb{C}$ such that

$$F(x_0, y_0, z_1) = F(x_0, y'_0, z_2) = F(x'_0, y_0, z_3) = F(x'_0, y'_0, z_4) = 0,$$

unless we have $F(x_0, y_0, z) \equiv c$ for some nonzero $c \in \mathbb{C}$, or a similar identity holds for one of the pairs (x_0, y'_0) , (x'_0, y_0) , (x'_0, y'_0) . In other words, we have $(x_0, x'_0, y_0, y'_0) \in \pi_1(V)$ unless one of these exceptions holds.

Let $\sigma := \text{Cl}(\{(x_0, y_0) \in \mathbb{C}^2 \mid \exists c \text{ such that } F(x_0, y_0, z) \equiv c\})$ (note that here we include the case $c = 0$). We show (in the full version) that $\dim(\sigma) \leq 1$, so the set

$$\sigma' := \{(x, x', y, y') \mid \text{one of } (x, y), (x, y'), (x', y), (x', y') \text{ is in } \sigma\}$$

has dimension at most 3. By standard properties of the closure operation, we have $\text{Cl}(\mathbb{C}^4 \setminus \sigma') = \mathbb{C}^4$. As observed above, we have $\mathbb{C}^4 \setminus \sigma' \subset \pi_1(V)$, so we get $\mathbb{C}^4 = \text{Cl}(\mathbb{C}^4 \setminus \sigma') \subset \text{Cl}(\pi_1(V)) \subset \mathbb{C}^4$ and hence $\text{Cl}(\pi_1(V)) = \mathbb{C}^4$. ◀

We use the implicit function theorem to locally express each of the variables z_1, z_2, z_3, z_4 in terms of the corresponding pair of the first four variables x, x', y, y' . To facilitate this we first exclude the subvariety of V defined by $V_0 := V_1 \cup V_2 \cup V_3$, where

$$V_i := \{(x, x', y, y', z_1, z_2, z_3, z_4) \in V \mid F_i(x, y, z_1)F_i(x, y', z_2)F_i(x', y, z_3)F_i(x', y', z_4) = 0\},$$

and F_i stands for the derivative of F with respect to its i th variable, for $i = 1, 2, 3$.

The following lemma, whose proof we omit, asserts that $\text{Cl}(\pi_1(V_0))$ is a subvariety of V of dimension ≤ 3 . This property allows us to exclude $\text{Cl}(\pi_1(V_0))$ in most of our proof.

► **Lemma 3.2.** $\text{Cl}(\pi_1(V_0))$ has dimension at most three.

As explained in Section 3.1, we want to view V , around most of its points, as the graph of a locally defined mapping. We now define this mapping.

► **Lemma 3.3.** For each point $v \in V \setminus V_0$, there is an open neighborhood $N_v \subset \mathbb{C}^8$ of v such that $V_0 \cap N_v = \emptyset$, and an analytic mapping $\Phi_v : \pi_1(N_v) \rightarrow \pi_2(N_v)$, such that $V \cap N_v = \{(u, \Phi_v(u)) \mid u \in \pi_1(N_v)\}$.

Proof. Let $v = (a, a', b, b', c_1, c_2, c_3, c_4) \in V \setminus V_0$ be an arbitrary point. We apply the implicit function theorem (see [5]) to the equation $F(x, y, z_1) = 0$ at the point (a, b, c_1) . Since $v \notin V_0$, we have $F_3(a, b, c_1) \neq 0$. We thus obtain neighborhoods U of (a, b) in \mathbb{C}^2 and V of c_1 in \mathbb{C} , and an analytic mapping $g_1 : U \rightarrow V$ such that

$$\{(x, y, z_1) \in U \times V \mid F(x, y, z_1) = 0\} = \{(x, y, g_1(x, y)) \mid (x, y) \in U\}.$$

We can do the same at each of the points $(a, b', c_2), (a', b, c_3), (a', b', c_4)$, leading to analogous mappings g_2, g_3, g_4 . It follows that we can find neighborhoods N_1 of a , N_2 of a' , N_3 of b , and N_4 of b' , such that the mapping

$$\Phi_v : (x, x', y, y') \mapsto (g_1(x, y), g_2(x, y'), g_3(x', y), g_4(x', y'))$$

is defined and analytic over $N_1 \times N_2 \times N_3 \times N_4$. Then

$$N_v := (N_1 \times N_2 \times N_3 \times N_4) \times \Phi_v(N_1 \times N_2 \times N_3 \times N_4)$$

is a neighborhood of v in \mathbb{C}^8 satisfying the conclusion of the lemma. If needed, we can shrink it to be disjoint from V_0 . ◀

Let G be the polynomial in $\mathbb{C}[x, x', y, y', z_1, z_2, z_3, z_4]$ given by

$$G = F_2(x, y, z_1)F_1(x, y', z_2)F_1(x', y, z_3)F_2(x', y', z_4) - F_1(x, y, z_1)F_2(x, y', z_2)F_2(x', y, z_3)F_1(x', y', z_4).$$

Consider the subvariety $W := V \cap Z(G)$ of V . The significance of W (and of G) lies in the following lemma.

► **Lemma 3.4.** For $v \in V \setminus V_0$ we have $v \in W$ if and only if $\det(J_{\Phi_v}(\pi_1(v))) = 0$.

Proof. We write g_{ij} for the derivative of the function g_i (from the proof of Lemma 3.3), within its domain of definition, with respect to its j th variable, for $i = 1, 2, 3, 4$, and $j = 1, 2$. The Jacobian matrix of Φ_v , evaluated at $u = (x, x', y, y') \in \pi_1(N_v)$, where N_v is the neighborhood of v given in Lemma 3.3, equals

$$J_{\Phi_v}(u) = \begin{pmatrix} g_{11}(x, y) & g_{21}(x, y') & 0 & 0 \\ 0 & 0 & g_{31}(x', y) & g_{41}(x', y') \\ g_{12}(x, y) & 0 & g_{32}(x', y) & 0 \\ 0 & g_{22}(x, y') & 0 & g_{42}(x', y') \end{pmatrix}, \quad (2)$$

or, by implicit differentiation,

$$J_{\Phi_v}(u) = \begin{pmatrix} -\frac{F_1(x, y, z_1)}{F_3(x, y, z_1)} & -\frac{F_1(x, y', z_2)}{F_3(x, y', z_2)} & 0 & 0 \\ 0 & 0 & -\frac{F_1(x', y, z_3)}{F_3(x', y, z_3)} & -\frac{F_1(x', y', z_4)}{F_3(x', y', z_4)} \\ -\frac{F_2(x, y, z_1)}{F_3(x, y, z_1)} & 0 & -\frac{F_2(x', y, z_3)}{F_3(x', y, z_3)} & 0 \\ 0 & -\frac{F_2(x, y', z_2)}{F_3(x, y', z_2)} & 0 & -\frac{F_2(x', y', z_4)}{F_3(x', y', z_4)} \end{pmatrix},$$

for $z_1 = g_1(x, y)$, $z_2 = g_2(x, y')$, $z_3 = g_3(x', y)$, and $z_4 = g_4(x', y')$. Since $N_v \cap V_0 = \emptyset$, all the denominators are non-zero (and, for that matter, also all the numerators). Write $v = (a, a', b, b', c_1, c_2, c_3, c_4)$. Computing this determinant explicitly at the point $u = \pi_1(v) = (a, a', b, b')$, noticing that by construction $c_1 = g_1(a, b)$, $c_2 = g_2(a, b')$, $c_3 = g_3(a', b)$, and $c_4 = g_4(a', b')$, and clearing denominators, gives exactly $G(v)$, where G is the polynomial defining W . Thus, $\det J_{\Phi_v}(\pi_1(v)) = 0$ if and only if $G(v) = 0$. \blacktriangleleft

3.3 The varieties V_γ

We now make precise what it means for a popular curve to be infinitely popular.

► **Definition 3.5.** Let $\gamma \subset \mathbb{C}^2$ be an irreducible curve. An irreducible curve $\gamma^* \subset \mathbb{C}^2$ is an *associated curve* of γ if for all but finitely many $(y, y') \in \gamma^*$ we have $\gamma \subset \gamma_{y, y'}$.

Throughout this section, we will let γ denote a popular curve and γ^* an associated curve of γ . In Section 3.4, we will show that every γ has at least one associated curve. With each $\gamma \in \mathcal{C}$, we associate the variety

$$V_\gamma := \text{Cl}(V \cap (\mathbb{C}^2 \times \gamma_r^* \times \gamma_r \times \gamma_r)) \subset \mathbb{C}^8,$$

where γ^* is any curve associated to γ , and γ_r^* , γ_r denote the subsets of regular points of γ^* , γ , respectively. It easily follows from the definition of V that, for most regular points $(z_1, z_2), (z_3, z_4) \in \gamma_r$ and for most regular points $(y, y') \in \gamma^*$, there exist $x, x' \in \mathbb{C}$ such that $(x, x', y, y', z_1, z_2, z_3, z_4) \in V_\gamma$. We have the following key property.

► **Lemma 3.6.** For all $\gamma \in \mathcal{C}$ we have $V_\gamma \subset W \cup V_0$.

Proof. It is sufficient to show that

$$V'_\gamma := V \cap (\mathbb{C}^2 \times \gamma_r^* \times \gamma_r \times \gamma_r) \subset W \cup V_0.$$

For this, let $v \in V'_\gamma \setminus V_0$. Then Lemma 3.3 gives an open neighborhood N_v of v , disjoint from V_0 , so that $V \cap N_v$ is the graph of an analytic map $\Phi_v : B_1 \rightarrow B_2$, where $B_1 := \pi_1(N_v)$ and $B_2 := \pi_2(N_v)$.

Assume, for contradiction, that $v \notin W$. Then Lemma 3.4 gives $\det(J_{\Phi_v}(\pi_1(v))) \neq 0$. By the inverse mapping theorem (see [5]), Φ_v is bianalytic on a sufficiently small neighborhood of $\pi_1(v)$, which, by shrinking N_v if needed, we may assume to be B_1 .

Consider the mapping $\bar{\Phi}_v := \Phi_v \circ \pi_1$ restricted to $V \cap N_v$. Note that $\bar{\Phi}_v$ is bianalytic. Indeed, π_1 restricted to $V \cap N_v$ is clearly bianalytic (its inverse is $u \mapsto (u, \Phi_v(u))$), so $\bar{\Phi}_v$ is the composition of two bianalytic functions, hence itself bianalytic. By definition of V_γ we have $\bar{\Phi}_v(V_\gamma \cap N_v) \subset \gamma \times \gamma$.

Write $v = (a, a', b, b', c_1, c_2, c_3, c_4)$, and note that, by the definition of V'_γ , $(c_1, c_2), (c_3, c_4)$ are regular points of γ and (b, b') is a regular point of γ^* . We claim that there exists an open $N \subset N_v$ such that $V_\gamma \cap N$ is locally three-dimensional. Indeed, we may assume, without loss of generality, that none of the tangents to γ at $(c_1, c_2), (c_3, c_4)$, and to γ^* at (b, b') are vertical in the respective planes (otherwise, we simply switch the roles of the first and the second coordinate in the relevant copy of \mathbb{C}^2). Applying the implicit function theorem (see [5]) to γ and γ^* at these regular points, we may therefore write $z_2 = \rho_1(z_1), z_4 = \rho_2(z_3)$, and $y' = \rho_3(y)$ in sufficiently small neighborhoods of $(b, b'), (c_1, c_2), (c_3, c_4)$, along the respective curves, for analytic functions ρ_1, ρ_2, ρ_3 . Similarly, applying the implicit function theorem to $Z(F)$ in sufficiently small neighborhoods of $(a, b, c_1), (a', b, c_3)$ (which we may, since we are away from V_0), we may write $x = \sigma_1(y, z_1), x' = \sigma_2(y, z_3)$, for analytic functions σ_1, σ_2 . Combining the functions above, we obtain an open neighborhood N of v such that the map

$$(y, z_1, z_3) \mapsto (\sigma_1(y, z_1), \sigma_2(y, z_3), y, \rho_3(y), z_1, \rho_1(z_1), z_3, \rho_2(z_3))$$

is bianalytic from an open neighborhood of (b, c_1, c_3) to $V_\gamma \cap N$. This implies that $V_\gamma \cap N$ is locally three-dimensional. Since $\gamma \times \gamma$ has local dimension 2 at every pair of regular points, and $\bar{\Phi}_v$ preserves local dimension, since it is bianalytic, this yields a contradiction, which completes the proof of the lemma. \blacktriangleleft

► **Lemma 3.7.** *If $\gamma \in \mathcal{C}$ is not an axis-parallel line, then $\text{Cl}(\pi_1(V_\gamma)) = \mathbb{C}^2 \times \gamma^*$.*

Proof. We clearly have $\pi_1(V_\gamma) \subset \pi_1(\mathbb{C}^2 \times \gamma^* \times \gamma \times \gamma) = \mathbb{C}^2 \times \gamma^*$, so, since $\mathbb{C}^2 \times \gamma^*$ is a variety, we get $\text{Cl}(\pi_1(V_\gamma)) \subset \mathbb{C}^2 \times \gamma^*$.

By definition, there is a finite subset $S \subset \gamma^*$ such that, for all $(b, b') \in \gamma^* \setminus S$, $\gamma \subset \gamma_{b,b'}$; fix such a point (b, b') which is also a regular point of γ^* . Then, by definition of V , it is easily checked that

$$\pi_1(V_\gamma) \cap Z(y - b, y' - b') \supset \beta_{b,b'} \times \beta_{b,b'} \times \{(b, b')\},$$

where $\beta_{b,b'} := \{x \in \mathbb{C} \mid \exists (c_1, c_2) \in \gamma_r \text{ such that } F(x, b, c_1) = F(x, b', c_2) = 0\}$. Since γ is not a line parallel to any of the axes,¹ one can show (details in the full version) that $\text{Cl}(\beta_{b,b'}) = \mathbb{C}$. Hence

$$\begin{aligned} \text{Cl}(\pi_1(V_\gamma)) &\supset \text{Cl}\left(\bigcup_{(b,b') \in \gamma_r^* \setminus S} \beta_{b,b'} \times \beta_{b,b'} \times \{(b, b')\}\right) \supset \bigcup_{(b,b') \in \gamma_r^* \setminus S} \text{Cl}(\beta_{b,b'} \times \beta_{b,b'} \times \{(b, b')\}) \\ &= \mathbb{C}^2 \times \text{Cl}\left(\bigcup_{(b,b') \in \gamma_r^* \setminus S} \{(b, b')\}\right) = \mathbb{C}^2 \times \text{Cl}(\gamma_r^* \setminus S) = \mathbb{C}^2 \times \gamma^*, \end{aligned}$$

using that the closure of an infinite union *contains* the union of the closures, and that the closure of a product is the product of the closures. This completes the proof of the lemma. \blacktriangleleft

¹ If γ were such a line, one of the equations, say $F(x, b, c_1) = 0$ would have a fixed value of c_1 , and only $O_d(1)$ values of x .

3.4 The associated curves

In this section we show that if a curve γ is popular, then it has at least one associated curve, of the sort defined in Definition 3.5. First we need the following sharpened form of Bézout's inequality for many curves. A proof can be found in Tao [15].

► **Lemma 3.8** (Bézout for many curves). *If \mathcal{F} is a (possibly infinite) family of algebraic curves in \mathbb{C}^2 , each of degree at most δ , then $\deg(\bigcap_{C \in \mathcal{F}} C) \leq \delta^2$. In other words, either $\bigcap_{C \in \mathcal{F}} C$ is 0-dimensional and has cardinality at most δ^2 , or it has dimension 1 and degree at most δ^2 .*

Recall that \mathcal{C} is the set of popular curves, i.e., irreducible curves γ that are contained in $\gamma_{y,y'}$ for more than d^4 points $(y, y') \in \mathbb{C}^2 \setminus \mathcal{S}$ (where \mathcal{S} is the set constructed in Section 2). Lemma 3.9 strengthens this property, by showing that if γ is popular, then there is a 1-dimensional set of curves $\gamma_{y,y'}$ that contain γ .

► **Lemma 3.9.** *Every $\gamma \in \mathcal{C}$ has at least one associated curve. More precisely, for every $\gamma \in \mathcal{C}$ there exists an algebraic curve $\gamma^* \subset \mathbb{C}^2$ of degree at most d^2 such that for all but finitely many $(y, y') \in \gamma^*$ we have $\gamma \subset \gamma_{y,y'}$.*

Proof. By definition of \mathcal{C} , if $\gamma \in \mathcal{C}$, then there exists a set $I \subset \mathbb{C}^2 \setminus \mathcal{S}$ of size $|I| = d^4 + 1$ such that $\gamma \subset \gamma_{y,y'}$ for all $(y, y') \in I$. This means that for all $(y, y') \in I$ and for all but finitely many $(z, z') \in \gamma$, there is an $x \in \mathbb{C}$ such that $F(x, y, z) = F(x, y', z') = 0$, which implies that $(y, y') \in \gamma_{z,z'}^*$. Thus we have $I \subset \gamma_{z,z'}^*$ for all but finitely many $(z, z') \in \gamma$.

Let \mathcal{F} be the infinite family of curves $\gamma_{z,z'}^*$ over all $(z, z') \in \gamma$ satisfying $I \subset \gamma_{z,z'}^*$, and define $S_I := \bigcap_{\gamma_{z,z'}^* \in \mathcal{F}} \gamma_{z,z'}^*$. Then we have $I \subset S_I$. Since all the curves in \mathcal{F} have degree at most d^2 , Lemma 3.8 implies that S_I has degree at most d^4 . Since $|I| > d^4$, S_I must have dimension 1. Let γ^* be any irreducible component of S_I .

If $(y, y') \in \gamma^*$, then for all but finitely many $(z, z') \in \gamma$ we have $(y, y') \in \gamma_{z,z'}^*$. It follows that for all but finitely many $(y, y') \in \gamma^*$, and for all but finitely many $(z, z') \in \gamma$ (where the excluded points (z, z') depend on the choice of (y, y')), we have $(z, z') \in \gamma_{y,y'}$. Since both γ and $\gamma_{y,y'}$ are algebraic curves, and γ is irreducible, we have $\gamma \subset \gamma_{y,y'}$ for all but finitely many $(y, y') \in \gamma^*$. This means γ^* is an associated curve of γ . ◀

3.5 Case 1: $\dim \text{Cl}(\pi_1(W)) \leq 3$ implies few popular curves

Throughout this subsection we assume that $\dim \text{Cl}(\pi_1(W)) \leq 3$, and establish the existence of the set \mathcal{X} in Proposition 2.3(a).

As the statement of Lemma 3.7 suggests, popular curves that are axis-parallel lines require a different treatment, provided by the following simple lemma, whose proof we omit.

► **Lemma 3.10.** *There is a 1-dimensional variety $\mathcal{X}_1 \subset \mathbb{C}^2$ with $\deg(\mathcal{X}_1) = O(d^2)$ containing \mathcal{S} , such that, for every $(y_1, y_2) \in \mathbb{C}^2 \setminus \mathcal{X}_1$, the curve γ_{y_1, y_2} contains no axis-parallel line.*

We also need the following observation.

► **Lemma 3.11.** *An irreducible curve γ^* is associated to at most d^2 curves $\gamma \in \mathcal{C}$.*

Proof. Suppose there is a set \mathcal{C}' of $d^2 + 1$ distinct curves $\gamma \in \mathcal{C}$ that γ^* is associated to. For each $\gamma \in \mathcal{C}'$, we have that, for all but finitely many $(y, y') \in \gamma^*$, γ is contained in $\gamma_{y,y'}$. It follows that there is a point $(y, y') \in \gamma^*$ such that $\gamma \subset \gamma_{y,y'}$ for all $\gamma \in \mathcal{C}'$. This is a contradiction, because $\gamma_{y,y'}$ has at most d^2 irreducible components. ◀

We are now ready to prove the key fact that the number of popular curves is bounded.

► **Lemma 3.12.** *There are $O(d^7)$ distinct popular curves $\gamma \in \mathcal{C}$ that are not axis-parallel lines, and there are $O(d^5)$ distinct associated curves of popular curves that are not axis-parallel lines.*

Proof. Let $\gamma \in \mathcal{C}$, assume that it is not an axis-parallel line, and let γ^* be an associated curve of γ . Since γ^* is irreducible, $\mathbb{C}^2 \times \gamma^*$ is an irreducible variety. Using Lemma 3.7 and Lemma 3.6, we have

$$\mathbb{C}^2 \times \gamma^* = \text{Cl}(\pi_1(V_\gamma)) \subset \text{Cl}(\pi_1(W \cup V_0)) = X \cup Y,$$

for $X := \text{Cl}(\pi_1(W))$ and $Y := \text{Cl}(\pi_1(V_0))$. We have $\dim(X) \leq 3$ by the assumption in this subsection, and $\dim(Y) \leq 3$ by Lemma 3.2. We also have $\deg(X) = O(d^5)$ and $\deg(Y) = O(d^5)$, since both are unions of closures of projections of varieties defined by five polynomials, each of degree at most $O(d)$. Since $X \cup Y$ is at most 3-dimensional, and each $\mathbb{C}^2 \times \gamma^*$ is an irreducible 3-dimensional subvariety of $X \cup Y$, it follows that $\mathbb{C}^2 \times \gamma^*$ is one of the finitely many irreducible components of $X \cup Y$.

Let T be the set of all associated curves of all curves $\gamma \in \mathcal{C}$ (excluding γ that are axis-parallel lines). The preceding argument shows that T is a finite set. Moreover, we have

$$\sum_{\gamma^* \in T} \deg(\gamma^*) = \sum_{\gamma^* \in T} \deg(\mathbb{C}^2 \times \gamma^*) \leq \deg(X \cup Y) = O(d^5).$$

This implies that the total number of distinct associated curves is $O(d^5)$. Since by Lemma 3.9 each popular curve has at least one associated curve, and by Lemma 3.11 each associated curve is associated to at most d^2 popular curves, it follows that the number of popular curves is bounded by $O(d^7)$. ◀

Finally, we show that the union of all the associated curves (which are not axis-parallel lines) has bounded degree.

► **Lemma 3.13.** *Let $\mathcal{X}_2 := \text{Cl}(\{(y, y') \in \mathbb{C}^2 \mid \exists \gamma \in \mathcal{C}, \text{ not axis-parallel line, s.t. } \gamma \subset \gamma_{y,y'}\})$. Then \mathcal{X}_2 is 1-dimensional; its purely 1-dimensional component has degree $O(d^7)$, and the number of 0-dimensional components is $O(d^{11})$.*

Proof. Any 1-dimensional irreducible component of \mathcal{X}_2 is an associated curve. By Lemma 3.12, there are $O(d^5)$ associated curves γ^* , and by Lemma 3.9 each is of degree at most $O(d^2)$. This implies that union of the purely 1-dimensional components of \mathcal{X}_2 has degree $O(d^7)$.

We next bound the number of 0-dimensional components of \mathcal{X}_2 . By Lemmas 3.11 and 3.12, the number of popular curves $\gamma \in \mathcal{C}$ is at most $O(d^7)$. We show that, for each of them, the number of isolated points outside the associated curves is at most d^4 . Let $\gamma \in \mathcal{C}$ and let $I \subset \mathbb{C}^2 \setminus \mathcal{S}$ denote the set consisting of isolated points, such that $\gamma \subset \gamma_{y,y'}$ for all $(y, y') \in I$. Exactly as in the proof of Lemma 3.9, there is a set S_I , which is the intersection of an infinite family of curves $\gamma_{z,z'}$ containing I . Thus we have $I \subset S_I$. By Lemma 3.8, S_I has degree at most d^4 , and therefore contains at most d^4 isolated points. ◀

We put $\mathcal{X} := \mathcal{X}_1 \cup \mathcal{X}_2$. Combining Lemma 3.10 and Lemma 3.13, we get $\dim(\mathcal{X}) = 1$ and $\deg(\mathcal{X}) = O(d^{11})$. From the definitions of \mathcal{X}_1 and \mathcal{X}_2 it is clear that for $(y, y') \notin \mathcal{X}$, the curve $\gamma_{y,y'}$ does not contain any popular curve. This completes the proof of Proposition 2.3(a) in Case 1. Proposition 2.3(b) is proved in a fully symmetric manner.

3.6 Case 2: $\dim \text{Cl}(\pi_1(W)) = 4$ implies a special form of F

Throughout this subsection we assume that $\dim \text{Cl}(\pi_1(W)) = 4$. By definition, $W \subset V$, and we already know that $\dim V = 4$, so W must be four-dimensional too, which implies that there exists an irreducible component $V' \subset W$ such that $\dim V' = 4$ and $\text{Cl}(\pi_1(V')) = \mathbb{C}^4$. We will work only with V' in the rest of this subsection. We first show that most points of $Z(F)$, excluding only a lower-dimensional subset, can be extended to points of V' , in the following sense.

► **Lemma 3.14.** *There exists a one-dimensional subvariety $Z_0 \subset Z(F)$ such that, for every $(a, b, c_1) \in Z(F) \setminus Z_0$, there exist a', b', c_2, c_3, c_4 such that $(a, a', b, b', c_1, c_2, c_3, c_4)$ is a regular point of V' which is not in V_0 .*

Proof. Let $\rho : \mathbb{C}^8 \rightarrow \mathbb{C}^6$ be the (permuted) projection map $\rho : (x, x', y, y', z_1, z_2, z_3, z_4) \mapsto (x, y, z_1, x', y', z_4)$. We claim that $\text{Cl}(\rho(V')) = Z(F) \times Z(F)$. Since $Z(F) \times Z(F)$ is four-dimensional and irreducible, and since, by definition of V , $\rho(V') \subset Z(F) \times Z(F)$, it suffices to prove that $\text{Cl}(\rho(V'))$ is four-dimensional. We observe that $\sigma(\rho(V')) = \pi_1(V')$, where $\sigma : (x, y, z_1, x', y', z_4) \mapsto (x, x', y, y')$. Because projections cannot increase dimension, we have $\dim \text{Cl}(\rho(V')) \geq \dim \text{Cl}(\pi_1(V')) = 4$, proving our claim.

By the standard properties of the closure operation, $U_1 := \text{Cl}((Z(F) \times Z(F)) \setminus \rho(V')) = \text{Cl}(\text{Cl}(\rho(V')) \setminus \rho(V'))$ is at most three-dimensional, and $U_2 := \text{Cl}(\rho(V_0 \cap V'))$ is clearly also at most three-dimensional. Since V' is irreducible, the subvariety V'_s of singular points of V' is at most three-dimensional, so $U_3 := \text{Cl}(\rho(V'_s))$ is also at most three-dimensional. Hence, $U := U_1 \cup U_2 \cup U_3$ is a variety in \mathbb{C}^6 of dimension at most 3. We set

$$Z'_0 := \{p \in Z(F) \mid \dim((\{p\} \times Z(F)) \cap U) \geq 2\}.$$

In other words (using the fact that $\{p\} \times Z(F)$ is irreducible), $p \in Z'_0$ if and only if $\{p\} \times Z(F) \subset U$, so $Z'_0 \times Z(F) \subset U$. Since U is a variety, we have $\text{Cl}(Z'_0) \times Z(F) = \text{Cl}(Z'_0 \times Z(F)) \subset U$. Since U is at most three-dimensional and $Z(F)$ is two-dimensional, we must have that, for $Z_0 := \text{Cl}(Z'_0)$, $\dim Z_0 \leq 1$.

Finally, let $(a, b, c_1) \in Z(F) \setminus Z_0$. By definition of Z_0 , we have

$$\dim((\{(a, b, c_1)\} \times Z(F)) \cap U) \leq 1.$$

Thus there exists a point $(a, b, c_1, a', b', c_4) \in (Z(F) \times Z(F)) \setminus U$. By definition of U , this implies that $(a, b, c_1, a', b', c_4) \in \rho(V') \setminus U$, which in turn means that there exist $c_2, c_3 \in \mathbb{C}$ such that $(a, a', b, b', c_1, c_2, c_3, c_4) \in V' \setminus V_0$ is a regular point of V' , as asserted. ◀

Let Z_0 be the variety given by Lemma 3.14.

► **Lemma 3.15.** *Let $u = (a, b, c_1) \in Z(F) \setminus Z_0$. Then there exist open sets $D_i \subset \mathbb{C}$ and analytic functions $\varphi_i : D_i \rightarrow \mathbb{C}$, for $i = 1, 2, 3$, such that $(a, b, c_1) \in D_1 \times D_2 \times D_3$ and*

$$(x, y, z) \in Z(F) \text{ if and only if } \varphi_1(x) + \varphi_2(y) + \varphi_3(z) = 0,$$

for every $(x, y, z) \in D_1 \times D_2 \times D_3$.

Proof. By applying Lemma 3.14 to $u = (a, b, c_1)$, we obtain $a', b', c_2, c_3, c_4 \in \mathbb{C}$, such that $v := (a, a', b, b', c_1, c_2, c_3, c_4) \in V' \setminus V_0$ is a regular point of V' . By Lemma 3.3, there exist neighborhoods D_1 of a , D_2 of a' , E_1 of b , and E_2 of b' , and a mapping

$$\Phi_v : (x, x', y, y') \mapsto (g_1(x, y), g_2(x, y'), g_3(x', y), g_4(x', y')),$$

analytic over $D_1 \times D_2 \times E_1 \times E_2$, such that its graph is the intersection $V \cap N_v$, for some open neighborhood N_v of v in \mathbb{C}^8 (whose π_1 -projection is $D_1 \times D_2 \times E_1 \times E_2$). Note that, since v is a regular point of V' , $V' \cap N_v$ is necessarily four-dimensional, and so it must coincide with $V \cap N_v$. Thus, restricting the analysis to the neighborhood N_v , we may use V and V' interchangeably in what follows.

Since $V' \subset W$, we have, recalling the definition of the variety W , that

$$G(x, x', y, y', z_1, z_2, z_3, z_4) = 0,$$

for every $(x, x', y, y', z_1, z_2, z_3, z_4) \in V' \cap N_v$. By the implicit function theorem, the functions g_1, \dots, g_4 satisfy, in a suitable neighborhood of v , $g_{1i}(x, y) = -\frac{F_i(x, y, g_1(x, y))}{F_3(x, y, g_1(x, y))}$, and similarly for g_2, g_3, g_4 . By the definition of G , this is easily seen to imply that

$$g_{11}(x, y)g_{22}(x, y')g_{32}(x', y)g_{41}(x', y') = g_{12}(x, y)g_{21}(x, y')g_{31}(x', y)g_{42}(x', y'),$$

for every $(x, x', y, y') \in D_1 \times D_2 \times E_1 \times E_2$. In particular, fixing $x' = a'$ and $y' = b'$, there exists an open neighborhood $D_1 \times D_2$ of $(a, b) \in \mathbb{C}^2$, such that, for every $(x, y) \in D_1 \times D_2$,

$$g_{11}(x, y)g_{22}(x, b')g_{32}(a', y)g_{41}(a', b') = g_{12}(x, y)g_{21}(x, b')g_{31}(a', y)g_{42}(a', b'). \tag{3}$$

Because $v \notin V_0$, we have $g_{11}(a, b) = -\frac{F_1(a, b, c_1)}{F_3(a, b, c_1)} \neq 0$. Similarly, $g_{22}(a, b')$, $g_{32}(a', b)$, $g_{41}(a', b')$, $g_{12}(a, b)$, $g_{21}(a, b')$, $g_{31}(a', b)$, and $g_{42}(a', b')$ are all nonzero. The continuity of all the relevant functions implies that, by shrinking $D_1 \times D_2$ if needed, we may assume that neither side of (3) is zero for any $(x, y) \in D_1 \times D_2$. Thus we can rewrite (3) as

$$\frac{g_{11}(x, y)}{p(x)} = \frac{g_{12}(x, y)}{q(y)}, \tag{4}$$

where $p(x) = g_{21}(x, b')g_{42}(a', b')/g_{22}(x, b')$ is analytic and nonzero on D_1 and $q(y) = g_{32}(a', y)g_{41}(a', b')/g_{31}(a', y)$ is analytic and nonzero on D_2 . By Lang [7, Theorem III.6.1], there exist analytic primitives φ_1, φ_2 so that $\varphi_1'(x) = p(x)$ on D_1 and $\varphi_2'(y) = q(y)$ on D_2 .

We express the function $g_1(x, y)$ in terms of new coordinates (ξ, η) , given by

$$\xi = \varphi_1(x) + \varphi_2(y), \quad \eta = \varphi_1(x) - \varphi_2(y). \tag{5}$$

Since p, q are continuous and nonzero at a, b , respectively, it follows that φ_1, φ_2 are injections in suitable respective neighborhoods of a, b , so by shrinking D_1 and D_2 still further, if needed, we may assume that the system (5) is invertible in $D_1 \times D_2$.

Returning to the standard notation, denoting partial derivatives by variable subscripts, we have $\xi_x = \varphi_1'(x)$, $\xi_y = \varphi_2'(y)$, $\eta_x = \varphi_1'(x)$, and $\eta_y = -\varphi_2'(y)$. Using the chain rule, we obtain

$$g_{11} = g_{1\xi}\xi_x + g_{1\eta}\eta_x = \varphi_1'(x)(g_{1\xi} + g_{1\eta}) = p(x)(g_{1\xi} + g_{1\eta})$$

$$g_{12} = g_{1\xi}\xi_y + g_{1\eta}\eta_y = \varphi_2'(y)(g_{1\xi} - g_{1\eta}) = q(y)(g_{1\xi} - g_{1\eta}),$$

which gives $\frac{g_{11}(x, y)}{p(x)} - \frac{g_{12}(x, y)}{q(y)} \equiv 2g_{1\eta}(x, y)$, on $D_1 \times D_2$. Combining this with (4), we get $g_{1\eta}(x, y) \equiv 0$. This means that g_1 depends only on the variable ξ , so it has the form $g_1(x, y) = \psi(\varphi_1(x) + \varphi_2(y))$, for a suitable analytic function ψ . The analyticity of ψ is an easy consequence of the analyticity of φ_1, φ_2 , and g_1 , and the fact that $\varphi_1'(x)$ and $\varphi_2'(y)$ are nonzero, combined with repeated applications of the chain rule. Let $E := \{\varphi_1(x) + \varphi_2(y) \mid (x, y) \in D_1 \times D_2\}$ and $D_3 := \{\psi(z) \mid z \in E\}$. We observe that $g_{11}(x, y) = \psi'(\varphi_1(x) + \varphi_2(y)) \cdot p(x)$. As

argued above, we have $g_{11}(x, y) \neq 0$ for all $(x, y) \in D_1 \times D_2$, implying that $\psi'(\varphi_1(x) + \varphi_2(y))$ is nonzero for $(x, y) \in D_1 \times D_2$. Therefore, $\psi : E \rightarrow D_3$ is invertible by the inverse mapping theorem (see [5]).

Letting $\varphi_3(z) := -\psi^{-1}(z)$, we get for $(x, y, z) \in D_1 \times D_2 \times D_3$ that $\varphi_1(x) + \varphi_2(y) + \varphi_3(z) = 0$ if and only if $(x, y, z) \in Z(F) \cap (D_1 \times D_2 \times D_3)$. This completes the proof of the lemma. ◀

Finally, Lemma 3.15 has established that F satisfies property (ii) of the theorem, which completes the proof of Proposition 2.3 for this case. ◀

Acknowledgements. Part of this research was performed while the authors were visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation. The authors deeply appreciate the stimulating environment and facilities provided by IPAM, which have facilitated the intensive and productive collaboration that have lead to this paper. The authors would also like to thank Hong Wang, Kaloyan Slavov and József Solymosi for several helpful discussions. Some of the insights in our analysis were inspired by talks given by Terry Tao at IPAM about his work [14].

References

- 1 G. Elekes and L. Rónyai, A combinatorial problem on polynomials and rational functions, *J. Combinat. Theory Ser. A* 89 (2000), 1–20.
- 2 G. Elekes and E. Szabó, How to find groups? (And how to use them in Erdős geometry?), *Combinatorica* 32 (2012), 537–571.
- 3 G. Elekes and E. Szabó, On triple lines and cubic curves: The Orchard Problem revisited, in [arXiv:1302.5777](https://arxiv.org/abs/1302.5777) (2013).
- 4 G. Elekes, M. Simonovits, and E. Szabó, A combinatorial distinction between unit circles and straight lines: How many coincidences can they have?, *Combinat. Probab. Comput.* 18 (2009), 691–705.
- 5 K. Fritzsche and H. Grauert, *From Holomorphic Functions to Complex Manifolds*, Springer-Verlag, New York, 2002.
- 6 L. Guth and N. H. Katz, On the Erdős distinct distances problem in the plane, *Annals Math.* 18 (2015), 155–190.
- 7 S. Lang, *Complex Analysis*, Springer-Verlag, New York, 1999.
- 8 J. Pach and F. de Zeeuw, Distinct distances on algebraic curves in the plane, in [arXiv:1308.0177](https://arxiv.org/abs/1308.0177) (2013).
- 9 O. E. Raz, M. Sharir, and J. Solymosi, Polynomials vanishing on grids: The Elekes-Rónyai problem revisited, *Amer. J. Math.*, to appear. Also in *Proc. 30th Annu. Sympos. Comput. Geom.*, 2014, 251–260. Also in [arXiv:1401.7419](https://arxiv.org/abs/1401.7419) (2014).
- 10 O. E. Raz, M. Sharir, and J. Solymosi, On triple intersections of three families of unit circles, *Proc. 30th Annu. Sympos. Comput. Geom.*, 2014, 198–205. Also in [arXiv:1407.6625](https://arxiv.org/abs/1407.6625) (2014).
- 11 M. Sharir, A. Sheffer, and J. Solymosi, Distinct distances on two lines, *J. Combinat. Theory Ser. A* 120 (2013), 1732–1736.
- 12 M. Sharir and J. Solymosi, Distinct distances from three points, *Combinat. Probab. Comput.*, to appear. Also in [arXiv:1308.0814](https://arxiv.org/abs/1308.0814) (2013).
- 13 J. Solymosi and F. de Zeeuw, Incidence bounds for complex algebraic curves on Cartesian products, in [arXiv:1502.05304](https://arxiv.org/abs/1502.05304) (2015).
- 14 T. Tao, Expanding polynomials over finite fields of large characteristic, and a regularity lemma for definable sets, in [arXiv:1211.2894](https://arxiv.org/abs/1211.2894) (2012).
- 15 T. Tao, *Bézout's inequality*, <http://terrytao.wordpress.com/2011/03/23/bezouts-inequality>.

Bisector Energy and Few Distinct Distances*

Ben Lund¹, Adam Sheffer², and Frank de Zeeuw³

1 Rutgers University, USA, lund.ben@gmail.com

2 California Institute of Technology, USA, adamsh@gmail.com

3 École Polytechnique Fédérale de Lausanne, Switzerland, fdezeeuw@gmail.com

Abstract

We define the *bisector energy* $\mathcal{E}(\mathcal{P})$ of a set \mathcal{P} in \mathbb{R}^2 to be the number of quadruples $(a, b, c, d) \in \mathcal{P}^4$ such that a, b determine the same perpendicular bisector as c, d . Equivalently, $\mathcal{E}(\mathcal{P})$ is the number of *isosceles trapezoids* determined by \mathcal{P} . We prove that if no line or circle contains $M(n)$ points of an n -point set \mathcal{P} , then for any $\varepsilon > 0$ we have

$$\mathcal{E}(\mathcal{P}) = O\left(M(n)^{\frac{2}{5}} n^{\frac{12}{5} + \varepsilon} + M(n)n^2\right).$$

We derive the lower bound $\mathcal{E}(\mathcal{P}) = \Omega(M(n)n^2)$, matching our upper bound when $M(n)$ is large.

We use our upper bound on $\mathcal{E}(\mathcal{P})$ to obtain two rather different results:

- (i) If \mathcal{P} determines $O(n/\sqrt{\log n})$ distinct distances, then for any $0 < \alpha \leq 1/4$, there exists a line or circle that contains at least n^α points of \mathcal{P} , or there exist $\Omega(n^{8/5-12\alpha/5-\varepsilon})$ distinct lines that contain $\Omega(\sqrt{\log n})$ points of \mathcal{P} . This result provides new information towards a conjecture of Erdős [7] regarding the structure of point sets with few distinct distances.
- (ii) If no line or circle contains $M(n)$ points of \mathcal{P} , the number of distinct perpendicular bisectors determined by \mathcal{P} is $\Omega(\min\{M(n)^{-2/5}n^{8/5-\varepsilon}, M(n)^{-1}n^2\})$.

1998 ACM Subject Classification G.2 Discrete Mathematics

Keywords and phrases Combinatorial geometry, distinct distances, incidence geometry

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.537

1 Introduction

Guth and Katz [11] proved that every set of n points in \mathbb{R}^2 determines $\Omega(n/\log n)$ distinct distances. This almost completely settled a conjecture of Erdős [5], who observed that the $\sqrt{n} \times \sqrt{n}$ integer lattice determines $\Theta(n/\sqrt{\log n})$ distances, and conjectured that every set of n points determines at least this number of distances. Beyond the remaining $\sqrt{\log n}$ gap, this leaves open the question of which point sets determine few distances. Erdős [7] asked whether every set that determines $O(n/\sqrt{\log n})$ distances “has lattice structure”. He then wrote: “*The first step would be to decide if there always is a line which contains $cn^{1/2}$ of the points (and in fact n^ε would already be interesting).*”

Embarrassingly, almost three decades later the bound n^ε seems as distant as it ever was. The following bound is a consequence of an argument of Szemerédi, presented by Erdős [6].

* Part of this research was performed while the authors were visiting the Institute for Pure and Applied Mathematics (IPAM) in Los Angeles, which is supported by the National Science Foundation. Work on this paper by Frank de Zeeuw was partially supported by Swiss National Science Foundation Grants 200020-144531 and 200021-137574. Work on this paper by Ben Lund was supported by NSF grant CCF-1350572.



► **Theorem 1.1** (Szemerédi). *If a set \mathcal{P} of n points in \mathbb{R}^2 determines $O(n/\sqrt{\log n})$ distances, then there exists a line containing $\Omega(\sqrt{\log n})$ points of \mathcal{P} .*

Recently, it was noticed that this bound can be slightly improved to $\Omega(\log n)$ points on a line (see [19]). Assuming that no line contains an asymptotically larger number of points, one can deduce the existence of $\Omega(n/\log n)$ distinct lines that contain $\Omega(\log n)$ points of \mathcal{P} . By inspecting Szemerédi’s proof, it is also apparent that these lines are *perpendicular bisectors* of pairs of points of \mathcal{P} .

This problem was recently approached from the other direction in [15, 16, 20]. Combining the results of these three papers implies the following. If an n -point set $\mathcal{P} \subset \mathbb{R}^2$ determines $o(n)$ distances, then no line contains $\Omega(n^{43/52+\varepsilon})$ points of \mathcal{P} , no circle contains $\Omega(n^{5/6})$ points, and no other constant-degree irreducible algebraic curve contains $\Omega(n^{3/4})$ points.

In the current paper we study a different aspect of sets with few distinct distances. Our main tool is a bound on the *bisector energy* of the point set (see below for a formal definition). Using this tool, we prove that if a point set \mathcal{P} determines $O(n/\sqrt{\log n})$ distinct distances, then there exists a line or a circle with many points of \mathcal{P} , or the number of lines containing $\Omega(\sqrt{\log n})$ points must be significantly larger than implied by Theorem 1.1. As another application of bisector energy, we prove that if no line or circle contains many points of a point set \mathcal{P} , then \mathcal{P} determines a large number of distinct perpendicular bisectors. We will provide more background to both results after we have properly stated them.

2 Results

Bisector energy. Given two distinct points $a, b \in \mathbb{R}^2$, we denote by $\mathcal{B}(a, b)$ their *perpendicular bisector* (i.e., the line consisting of all points that are equidistant from a and b); for brevity, we usually refer to it as the *bisector* of a and b . We define the *bisector energy* of \mathcal{P} as

$$\mathcal{E}(\mathcal{P}) = p \left| \{ (a, b, c, d) \in \mathcal{P}^4 : a \neq b, c \neq d, \text{ and } \mathcal{B}(a, b) = \mathcal{B}(c, d) \} \right|.$$

Equivalently, $\mathcal{E}(\mathcal{P})$ is the number of *isosceles trapezoids* determined by \mathcal{P} (not counting isosceles triangles).¹ In Section 3, we prove the following upper bound on this quantity.

► **Theorem 2.1.** *Let $M(n)$ be an arbitrary function with positive values. For any n -point set $\mathcal{P} \subset \mathbb{R}^2$, such that no line or circle contains $M(n)$ points of \mathcal{P} , we have²*

$$\mathcal{E}(\mathcal{P}) = O \left(M(n)^{\frac{2}{5}} n^{\frac{12}{5}+\varepsilon} + M(n)n^2 \right).$$

The bound of Theorem 2.1 is dominated by its first term when $M(n) = O(n^{2/3+\varepsilon'})$. We note that one important ingredient of our proof is the result of Guth and Katz [11]; without it, we would obtain a weaker (although nontrivial) bound on the bisector energy (see the remark at the end of Section 3.3).

In Section 3.4, we derive a lower bound for the maximum bisector energy. It shows that Theorem 2.1 is tight when its second term dominates, i.e., when $M(n) = \Omega(n^{2/3+\varepsilon'})$.

► **Theorem 2.2.** *For any n and $M(n)$, there exists a set \mathcal{P} of n points in \mathbb{R}^2 such that no line or circle contains $M(n)$ points of \mathcal{P} , and $\mathcal{E}(\mathcal{P}) = \Omega(M(n)n^2)$.*

¹ Note that if each distinct pair of points of \mathcal{P} determines a distinct bisector, then $\mathcal{E}(\mathcal{P}) = 2n(n-1)$, since quadruples of the form (a, b, a, b) , (a, b, b, a) , (b, a, a, b) , and (b, a, b, a) , are counted for every $(a, b) \in \mathcal{P}^2$.

² Throughout this paper, when we state a bound involving an ε , we mean that this bound holds for every $\varepsilon > 0$, with the multiplicative constant of the $O(\cdot)$ -notation depending on ε .

We conjecture that $\mathcal{E}(\mathcal{P}) = O(M(n)n^2)$ is true for all $M(n)$. In parallel to our work, Hanson, Iosevich, Lund, and Roche-Newton [12] proved a variant of Theorem 2.1 in \mathbb{F}_q^2 .

Few distinct distances. Pach and Tardos [14] proved that an n -point set $\mathcal{P} \subset \mathbb{R}^2$ determines $O(n^{2.137})$ isosceles triangles. They also observed that this bound implies that \mathcal{P} contains a point from which there are $\Omega(n^{0.863})$ distinct distances (a result obtained earlier in [24] and improved slightly in [13]). Similarly, our upper bound on the number of isosceles trapezoids determined by a point set \mathcal{P} has implications concerning the distinct distances that are determined by \mathcal{P} .

We deduce the following theorem from Theorem 2.1. More precisely, it follows from the slightly more general Theorem 4.1 that we prove in Section 4.

► **Theorem 2.3.** *Let $\mathcal{P} \subset \mathbb{R}^2$ be a set of n points that spans $O(n/\sqrt{\log n})$ distinct distances. For any $0 < \alpha \leq 1/4$, at least one of the following holds (with constants independent of α).*

- (i) *There exists a line or a circle containing $\Omega(n^\alpha)$ points of \mathcal{P} .*
- (ii) *There are $\Omega(n^{\frac{8}{5} - \frac{12\alpha}{5} - \varepsilon})$ lines that contain $\Omega(\sqrt{\log n})$ points of \mathcal{P} .*

If our conjecture that $\mathcal{E}(\mathcal{P}) = O(M(n)n^2)$ is true, alternative (ii) in the conclusion of Theorem 2.3 improves to $\Omega(n^{2-3\alpha} \log n)$ lines that contain $\Omega(\sqrt{\log n})$ points of \mathcal{P} .

We believe that Theorem 2.3 is a step towards Erdős’s lattice conjecture. We mention several recent results and conjectures that together paint an interesting picture.

Green and Tao [10] proved that, given an n -point set in \mathbb{R}^2 such that more than $n^2/6 - O(n)$ lines contain at least three of the points, most of the points must lie on a cubic curve (an algebraic curve of degree at most three). Elekes and Szabó [4] stated the stronger conjecture that if an n -point set determines $\Omega(n^2)$ collinear triples, then many of the points lie on a cubic curve; unfortunately, at this point it is not even known whether there must be a cubic that contains *ten* points of the set. Erdős and Purdy [8] conjectured that if n points determine $\Omega(n^2)$ collinear *quadruples*, then there must be five points on a line. If the point set is already known to lie on a low-degree algebraic curve, then both conjectures hold [4, 18]. On the other hand, Solymosi and Stojaković [21] proved that for any constant k , there are point sets with $\Omega(n^{2-\varepsilon})$ lines with exactly k points, but no line with $k + 1$ points.

The philosophy of these statements is that if there are many lines containing many points, then most points must lie on some low-degree algebraic curve. Our result shows that for an n -point set with few distinct distances, there is a line or circle with very many points, or else there are many lines with many points. In particular, in the second case there would be many collinear triples (although not quite as many as $\Omega(n^2)$), and many lines with very many (more than a constant) points. This suggests that few distinct distances should imply some algebraic structure. Let us pose a specific question: Is there a $0 < \beta < 1$ such that if n points determine $\Omega(n^{1+\beta})$ lines with $\Omega(\sqrt{\log n})$ points, then many of the points must lie on a low-degree algebraic curve?

Distinct bisectors. Let $\mathcal{B}(\mathcal{P})$ be the set of those lines that are (distinct) perpendicular bisectors of \mathcal{P} . Since any point of \mathcal{P} determines $n - 1$ distinct bisectors with the other points of \mathcal{P} , we have a trivial lower bound $|\mathcal{B}(\mathcal{P})| \geq n - 1$. If \mathcal{P} is a set of equally spaced points on a circle, then $|\mathcal{B}(\mathcal{P})| = n$. Similarly, if \mathcal{P} is a set of n equally spaced points on a line, then $|\mathcal{B}(\mathcal{P})| = 2n - 3$. As we now show, forbidding many points on a line or circle forces $|\mathcal{B}(\mathcal{P})|$ to be significantly larger.

► **Theorem 2.4.** *If an n -point set $\mathcal{P} \subset \mathbb{R}^2$ has no $M(n)$ points on a line or circle, then*

$$|\mathcal{B}(\mathcal{P})| = \Omega\left(\min\left\{M(n)^{-\frac{2}{5}}n^{\frac{8}{5}-\varepsilon}, M(n)^{-1}n^2\right\}\right).$$

Proof. For any line $\ell \subset \mathbb{R}^2$, set $E_\ell = \{(a, b) \in \mathcal{P}^2 : a \neq b, \mathcal{B}(a, b) = \ell\}$. By the Cauchy-Schwarz inequality, we have

$$\mathcal{E}(\mathcal{P}) = \sum_{\ell \in \mathcal{B}(\mathcal{P})} |E_\ell|^2 \geq \frac{1}{|\mathcal{B}(\mathcal{P})|} \left(\sum_{\ell \in \mathcal{B}(\mathcal{P})} |E_\ell| \right)^2 = \Omega\left(\frac{n^4}{|\mathcal{B}(\mathcal{P})|}\right)$$

Combining this with the bound of Theorem 2.1 immediately implies the theorem. ◀

We are not aware of any previous bound on the minimum number of distinct bisectors.

Theorem 2.4 is related to a series of results initiated by Elekes and Rónyai [2], studying the expansion properties of polynomials and rational functions. For instance, in [17] it is proved that a polynomial function $F : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ takes $\Omega(n^{4/3})$ values on the n^2 pairs from a finite set in \mathbb{R} of size n , unless F has a special form. Elekes and Szabó [3] derived, among other things, the following two-dimensional generalization (rephrased for our convenience, and omitting some details). If $F : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ is a rational function that is not of a special form, and $\mathcal{P} \subset \mathbb{R}^2$ is an n -point set such that no low-degree curve contains many points of \mathcal{P} , then F takes $\Omega(n^{1+\varepsilon})$ values on $\mathcal{P} \times \mathcal{P}$.

Theorem 2.4 proves a better bound for the function \mathcal{B} , with a less restrictive condition on \mathcal{P} . If we view a line $y = sx + t$ as a point $(s, t) \in \mathbb{R}^2$, then (see the proof of Lemma 3.1)

$$\mathcal{B}(a_x, a_y, b_x, b_y) = \left(-\frac{a_x - b_x}{a_y - b_y}, \frac{(a_x^2 + a_y^2) - (b_x^2 + b_y^2)}{2(a_y - b_y)} \right)$$

is a rational function $\mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$. Theorem 2.4 says that \mathcal{B} takes many distinct values on $\mathcal{P} \times \mathcal{P}$ if \mathcal{P} has few points on a line or circle. So we have replaced the broad condition of [3] that not too many points lie on a low-degree curve, with the very specific condition that not too many points lie on a line or circle.

An incidence bound. To prove Theorem 2.1, we use the incidence bound below. It is a refined version of a theorem from Fox et al. [9], with explicit dependence on the parameter t , which we allow to depend on m and n . We reproduce the proof in Section 5 to determine this dependence. Given a set $\mathcal{P} \subset \mathbb{R}^d$ of points and a set $\mathcal{S} \subset \mathbb{R}^d$ of varieties, the *incidence graph* is a bipartite graph with vertex sets \mathcal{P} and \mathcal{S} , such that $(p, S) \in \mathcal{P} \times \mathcal{S}$ is an edge in the graph if $p \in S$. We write $I(\mathcal{P}, \mathcal{S})$ for the number of edges of this graph, or in other words, for the number of *incidences* between \mathcal{P} and \mathcal{S} . We denote the complete bipartite graph on s and t vertices by $K_{s,t}$ (in the incidence graph, such a subgraph corresponds to s points that are contained in t varieties). For the definitions of the algebraic terms in this statement we refer to [9].

► **Theorem 2.5.** *Let \mathcal{S} be a set of n constant-degree varieties and \mathcal{P} a set of m points, both in \mathbb{R}^d , such that the incidence graph of $\mathcal{P} \times \mathcal{S}$ contains no copy of $K_{s,t}$ (where s is a constant, but t may depend on m, n). Moreover, assume that $\mathcal{P} \subset V$, where V is an irreducible constant-degree variety of dimension e . Then*

$$I(\mathcal{P}, \mathcal{S}) = O\left(m^{\frac{s(e-1)}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1}} t^{\frac{e-1}{es-1}} + tm + n\right).$$

3 Proof of Theorem 2.1

In this section we prove Theorem 2.1 by relating the bisector energy to an incidence problem between points and algebraic surfaces in \mathbb{R}^4 . In Section 3.1 we define the surfaces, in Section 3.2 we analyze their intersection properties, and in Section 3.3 we apply the incidence bound of Theorem 2.5 to prove Theorem 2.1. Finally, in Section 3.4 we derive Theorem 2.2, which provides a lower bound for Theorem 2.1.

Throughout this section we assume that we have rotated \mathcal{P} so that no two points have the same x - or y -coordinate; in particular, we assume that no perpendicular bisector is horizontal or vertical.

3.1 Bisector surfaces

Recall that in Theorem 2.1 we consider an n -point set $\mathcal{P} \subset \mathbb{R}^2$. We define

$$\mathcal{P}^{2*} = \{(a, c) \in \mathcal{P}^2 : a \neq c\},$$

and similarly

$$\mathcal{P}^{4*} = \{(a, b, c, d) \in \mathcal{P}^4 : a \neq b, c \neq d; a \neq c, b \neq d\}.$$

Also recall that for distinct $a, b \in \mathcal{P}$, we denote by $\mathcal{B}(a, b)$ the perpendicular bisector of a and b . We define the *bisector surface* of a pair $(a, c) \in \mathcal{P}^{2*}$ as

$$S_{ac} = \{(b, d) \in \mathbb{R}^2 \times \mathbb{R}^2 : (a, b, c, d) \in \mathcal{P}^{4*}, \mathcal{B}(a, b) = \mathcal{B}(c, d)\},$$

and we set $\mathcal{S} = \{S_{ac} : (a, c) \in \mathcal{P}^{2*}\}$. The surface S_{ac} is not an algebraic variety (so we are using the word “surface” loosely), but the lemma below shows that S_{ac} is “close to” a variety $\overline{S_{ac}}$. That S_{ac} is contained in a constant-degree variety of the same dimension is no surprise (one can take the *Zariski closure*), but we need to analyze this variety in detail to establish the exact relationship.

We will work mostly with the surface S_{ac} in the rest of this proof, rather than with the variety $\overline{S_{ac}}$, because its definition is easier to handle. Then, when we apply our incidence bound, which holds only for varieties, we will switch to $\overline{S_{ac}}$. Fortunately, the lemma shows that this makes no difference in terms of the incidence graph.

► **Lemma 3.1.** *For distinct $a, c \in \mathcal{P}$, there exists a two-dimensional constant-degree algebraic variety $\overline{S_{ac}}$ such that $S_{ac} \subset \overline{S_{ac}}$. Moreover, if $(b, d) \in (\overline{S_{ac}} \setminus S_{ac}) \cap \mathcal{P}^{2*}$, then $(a, b, c, d) \notin \mathcal{P}^{4*}$.*

Proof. Consider a point $(b, d) \in S_{ac}$. Write the equation defining the perpendicular bisector $\mathcal{B}(a, b) = \mathcal{B}(c, d)$ as $y = sx + t$. The slope s satisfies

$$s = -\frac{a_x - b_x}{a_y - b_y} = -\frac{c_x - d_x}{c_y - d_y}. \tag{1}$$

By definition $\mathcal{B}(a, b)$ passes through the midpoint $((a_x + b_x)/2, (a_y + b_y)/2)$ of a and b , as well as through the midpoint $((c_x + d_x)/2, (c_y + d_y)/2)$ of c and d . We thus have

$$\frac{a_y + b_y}{2} - s \frac{a_x + b_x}{2} = t = \frac{c_y + d_y}{2} - s \frac{c_x + d_x}{2}. \tag{2}$$

By combining (1) and (2) we obtain

$$(a_y - b_y)(c_x^2 + c_y^2 - d_x^2 - d_y^2) = (c_y - d_y)(a_x^2 + a_y^2 - b_x^2 - b_y^2). \tag{3}$$

From (1) and (3) we see that $(b, d) = (x_1, x_2, x_3, x_4)$ satisfies

$$\begin{aligned} f_{ac}(x_1, x_2, x_3, x_4) &= (a_x - x_1)(c_y - x_4) - (a_y - x_2)(c_x - x_3) = 0, \\ g_{ac}(x_1, x_2, x_3, x_4) &= (a_y - x_2)(c_x^2 + c_y^2 - x_3^2 - x_4^2) - (c_y - x_4)(a_x^2 + a_y^2 - x_1^2 - x_2^2) = 0. \end{aligned}$$

Since any point $(b, d) \in S_{ac}$ satisfies these two equations, we have

$$S_{ac} \subset Z(f_{ac}, g_{ac}) = \overline{S_{ac}}.$$

By reexamining the above analysis, we see that if a point $(b, d) \in \overline{S_{ac}} \cap \mathcal{P}^{2*}$ is not in S_{ac} , we must have $a_y = b_y$ or $c_y = d_y$, since then (1) is not well defined. By the assumption that no two points of \mathcal{P} have the same y -coordinate, this implies $a = b$ or $c = d$, so $(a, b, c, d) \notin \mathcal{P}^{4*}$.

It remains to prove that $\overline{S_{ac}}$ is a constant-degree two-dimensional variety. The constant degree is immediate from f_{ac} and g_{ac} being polynomials of degree at most three. As just observed, a point $(b, d) \in \overline{S_{ac}} \setminus S_{ac}$ satisfies $a_y = b_y$ or $c_y = d_y$. If $a_y = b_y$, then for $f_{ac}(b, d) = g_{ac}(b, d) = 0$ to hold, we must have $a_x = b_x$ or $c_y = d_y$. Similarly, if $c_y = d_y$, then $c_x = d_x$ or $a_y = b_y$. We see that in each case we get two independent linear equations, which define a plane, so $\overline{S_{ac}} \setminus S_{ac}$ is the union of three two-dimensional planes. Thus, it suffices to prove that S_{ac} is two-dimensional. For this, we simply show that for any valid value of b there is at most one valid value of d . Let $C_{ac} \subset \mathbb{R}^2$ denote the circle that is centered at c and incident to a (here we use $a \neq c$). It is impossible for b to lie on C_{ac} , since this would imply that the bisector $\mathcal{B}(a, b)$ contains c , and thus that $\mathcal{B}(a, b) \neq \mathcal{B}(c, d)$. For any choice of $b \notin C_{ac}$, the bisector $\mathcal{B}(a, b)$ is well-defined and is not incident to c , so there is a unique $d \in \mathbb{R}^2$ with $\mathcal{B}(a, b) = \mathcal{B}(c, d)$ (i.e., so that $(b, d) \in S_{ac}$). ◀

3.2 Intersections of bisector surfaces

We denote by \mathbf{R}_{ab} the reflection of \mathbb{R}^2 across the line $\mathcal{B}(a, b)$. Observe that if $\mathcal{B}(a, b) = \mathcal{B}(c, d)$, then $\mathbf{R}_{ab} = \mathbf{R}_{cd}$, and this reflection maps a to b and c to d ; this in turn implies that $|ac| = |bd|$. That is, $(b, d) \in S_{ac}$ implies $|ac| = |bd|$. It follows that if $|ac| = \delta$, then the surface S_{ac} is contained in the hypersurface

$$H_\delta = \{(b, d) \in \mathbb{R}^2 \times \mathbb{R}^2 : |bd| = \delta\}.$$

We can thus partition \mathcal{S} into classes corresponding to the distances δ that are determined by pairs of points of \mathcal{P} . Each class consists of the surfaces S_{ac} with $|ac| = \delta$, all of which are fully contained in H_δ .

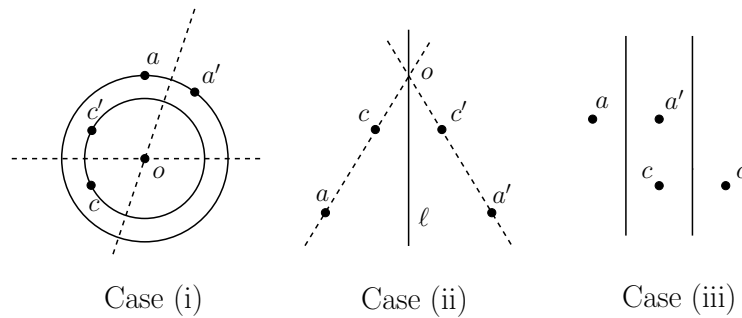
We now study the intersection of the surfaces contained in a common hypersurface H_δ .

► **Lemma 3.2.** *Let $(a, c) \neq (a', c')$ and $|ac| = |a'c'| = \delta \neq 0$. Then there exist curves $C_1, C_2 \subset \mathbb{R}^2$, which are either two concentric circles or two parallel lines, such that $a, a' \in C_1$, $c, c' \in C_2$, and $S_{ac} \cap S_{a'c'}$ is contained in the set*

$$H_\delta \cap (C_1 \times C_2) = \{(b, d) \in \mathbb{R}^2 \times \mathbb{R}^2 : b \in C_1, d \in C_2, |bd| = \delta\}.$$

Proof. We split the analysis into three cases: (i) $|\mathcal{B}(a, a') \cap \mathcal{B}(c, c')| = 1$, (ii) $\mathcal{B}(a, a') = \mathcal{B}(c, c')$, and (iii) $\mathcal{B}(a, a') \cap \mathcal{B}(c, c') = \emptyset$. The three cases are depicted in Figure 1.

Case (i). Let $o = \mathcal{B}(a, a') \cap \mathcal{B}(c, c')$. Then there exist two (not necessarily distinct) circles C_1, C_2 around o such that $a, a' \in C_1$ and $c, c' \in C_2$. If $(b, d) \in S_{ac} \cap S_{a'c'}$, then the reflection \mathbf{R}_{ab} maps a to b and c to d , and similarly, $\mathbf{R}_{a'b}$ maps b to a' and d to c' . We set



■ **Figure 1** The three cases in the analysis of Lemma 3.2.

$\mathbf{T} = \mathbf{R}_{a'b} \circ \mathbf{R}_{ab}$, and notice that this is a rotation whose center o^* is the intersection point of $\mathcal{B}(a, b) = \mathcal{B}(c, d)$ and $\mathcal{B}(a', b) = \mathcal{B}(c', d)$. Note that $\mathbf{T}(a) = a'$ and $\mathbf{T}(c) = c'$, so o^* lies on both $\mathcal{B}(a, a')$ and $\mathcal{B}(c, c')$. Since $o = \mathcal{B}(a, a') \cap \mathcal{B}(c, c')$, we obtain that $o = o^*$. Since $\mathcal{B}(a, b)$ passes through o , we have that b is incident to C_1 . Similarly, since $\mathcal{B}(c, d)$ passes through o , we have that d is incident to C_2 . This implies that (b, d) lies in $H_\delta \cap (C_1 \times C_2)$.

Case (ii). Let ℓ be the line $\mathcal{B}(a, a') = \mathcal{B}(c, c')$. The line segment ac is a reflection across ℓ of the line segment $a'c'$. Thus, the intersection point o of the lines that contains these two segments is incident to ℓ . Let C_1 be the circle centered at o that contains a and a' , and let C_2 be the circle centered at o that contains c and c' . With this definition of o , C_1 , and C_2 , we can repeat the analysis of case (i), obtaining the same conclusion.

Case (iii). In this case $\mathcal{B}(a, a')$ and $\mathcal{B}(c, c')$ are parallel. The analysis of this case is similar to that in case (i), but with lines instead of circles.

Let C_1 be the line that is incident to a and a' , and let C_2 be the line that is incident to c and c' . If $(b, d) \in S_{ac} \cap S_{a'c'}$, then, as before, \mathbf{R}_{ab} maps a to b and c to d , and $\mathbf{R}_{a'b}$ maps b to a' and d to c' . Since $\mathcal{B}(a', b)$ and $\mathcal{B}(a, b)$ are parallel, we have that $\mathbf{T} = \mathbf{R}_{a'b} \circ \mathbf{R}_{ab}$ is a translation in the direction orthogonal to these two lines. This implies that $b \in C_1$ and $d \in C_2$, which completes the analysis of this case. ◀

In Section 3.3, we will apply the incidence bound of Theorem 2.5 to the point set $\mathcal{P}^{2*} = \{(b, d) \in \mathcal{P}^2 : b \neq d\}$ and the set of surfaces \mathcal{S} . For this we need to show that the incidence graph contains no complete bipartite graph $K_{2,M}$; that is, that for any two points of \mathcal{P}^{2*} (where \mathcal{P}^{2*} is considered as a point set in \mathbb{R}^4) there is a bounded number of surfaces of \mathcal{S} that contain both points. In the following lemma we prove the more general statement that the incidence graph contains no copy of $K_{2,M}$ and no copy of $K_{M,2}$. Note that this is the only point in the proof of Theorem 2.1 where we use the condition that no M points are on a line or circle.

► **Corollary 3.3.** *If no line or circle contains M points of \mathcal{P} , then the incidence graph of \mathcal{P}^{2*} and \mathcal{S} contains neither a copy of $K_{2,M}$ nor a copy of $K_{M,2}$.*

Proof. Consider two distinct surfaces $S_{ac}, S_{a'c'} \in \mathcal{S}$ with $|ac| = |a'c'| = \delta$. Lemma 3.2 implies that there exist two lines or circles C_1, C_2 such that $(b, d) \in S_{ac} \cap S_{a'c'}$ only if $b \in C_1$ and $d \in C_2$. Since no line or circle contains M points of \mathcal{P} , we have $|C_1 \cap \mathcal{P}| < M$. Given $b \in (C_1 \cap \mathcal{P}) \setminus \{a\}$, there is at most one $d \in \mathcal{P}$ such that $\mathcal{B}(a, b) = \mathcal{B}(c, d)$, and thus at most one point $(b, d) \in S_{ac}$. (Notice that no points of the form $(a, d) \in \mathcal{P}^{2*}$ are in S_{ac} .) Thus

$$|(S_{ac} \cap S_{a'c'}) \cap \mathcal{P}^{2*}| < M.$$

That is, the incidence graph contains no copy of $K_{M,2}$.

We now define “dual” surfaces

$$S_{bd}^* = \{(a, c) \in \mathbb{R}^2 \times \mathbb{R}^2 : a \neq b, c \neq d, \mathcal{B}(a, b) = \mathcal{B}(c, d)\},$$

and set $\mathcal{S}^* = \{S_{bd}^* : (b, d) \in \mathcal{P}^{2*}\}$. By a symmetric argument, we get

$$|(S_{bd}^* \cap S_{b'd'}^*) \cap \mathcal{P}^{2*}| < M$$

for all $(b, d) \neq (b', d')$. Observe that $(a, c) \in S_{bd}^*$ if and only if $(b, d) \in S_{ac}$. Hence, having fewer than M points $(a, c) \in (S_{bd}^* \cap S_{b'd'}^*) \cap \mathcal{P}^{2*}$ is equivalent to having fewer than M surfaces S_{ac} that contain both (b, d) and (b', d') ; i.e., the incidence graph contains no $K_{2,M}$. ◀

3.3 Applying the incidence bound

We set

$$Q = \{(a, b, c, d) \in \mathcal{P}^{4*} : \mathcal{B}(a, b) = \mathcal{B}(c, d)\},$$

and note that $|Q| + \binom{n}{2} = \mathcal{E}(\mathcal{P})$, where the term $\binom{n}{2}$ accounts for the quadruples of the form (a, b, a, b) . As we saw in Section 3.2, every quadruple $(a, b, c, d) \in Q$ satisfies $|ac| = |bd|$.

Let $\delta_1, \dots, \delta_D$ denote the distinct distances that are determined by pairs of distinct points in \mathcal{P} . We partition \mathcal{P}^{2*} into the disjoint subsets Π_1, \dots, Π_D , where

$$\Pi_i = \{(u, v) \in \mathcal{P}^{2*} : |uv| = \delta_i\}.$$

We also partition \mathcal{S} into disjoint subsets $\mathcal{S}_1, \dots, \mathcal{S}_D$, defined by

$$\mathcal{S}_i = \{S_{ac} \in \mathcal{S} : |ac| = \delta_i\}.$$

Let m_i be the number of $(a, c) \in \mathcal{P}^{2*}$ such that $|ac| = \delta_i$. Note that $|\Pi_i| = |\mathcal{S}_i| = m_i$ and

$$\sum m_i = n(n-1).$$

A quadruple $(a, b, c, d) \in \mathcal{P}^{4*}$ is in Q if and only if the point (b, d) is incident to S_{ac} . Moreover, there exists a unique $1 \leq i \leq D$ such that $(b, d) \in \Pi_i$ and $S_{ac} \in \mathcal{S}_i$. Therefore, it suffices to study each Π_i and \mathcal{S}_i separately. That is, we have

$$|Q| = \sum_{i=1}^D I(\Pi_i, \mathcal{S}_i).$$

We apply our incidence bound to \mathcal{S}_i , or rather, to the corresponding set of varieties $\overline{\mathcal{S}}_i = \{\overline{S}_{ac} : S_{ac} \in \mathcal{S}_i\}$. By Lemma 3.1, the incidence graph of Π_i with $\overline{\mathcal{S}}_i$ is the same as with \mathcal{S}_i , hence also does not contain a copy of $K_{2,M}$ by Corollary 3.3. Observe that $\Pi_i \subset H_{\delta_i}$. The hypersurface H_{δ_i} is irreducible, three-dimensional, and of a constant degree, since it is defined by the irreducible polynomial $(x_1 - x_3)^2 + (x_2 - x_4)^2 - \delta_i$. Thus we can apply Theorem 2.5 to each $I(\Pi_i, \overline{\mathcal{S}}_i)$, with $m = n = m_i$, $V = H_{\delta_i}$, $d = 4$, $e = 3$, $s = 2$, and $t = M$. This implies that

$$I(\Pi_i, \mathcal{S}_i) = I(\Pi_i, \overline{\mathcal{S}}_i) = O\left(M^{\frac{2}{5}} m_i^{\frac{7}{5} + \varepsilon} + M m_i\right). \tag{4}$$

Let J be the set of indices $1 \leq j \leq D$ for which the bound in (4) is dominated by the term $M^{\frac{2}{5}} m_j^{\frac{7}{5} + \varepsilon}$. By recalling that $\sum_{j=1}^D m_j = n(n-1)$, we get

$$\sum_{j \notin J} I(\Pi_j, \mathcal{S}_j) = O(Mn^2).$$

Next we consider $\sum_{j \in J} I(\Pi_j, \mathcal{S}_j) = O(\sum_{j \in J} M^{2/5} m_j^{7/5+\epsilon})$. By [11, Proposition 2.2], we have

$$\sum m_j^2 = O(n^3 \log n).$$

This implies that the number of m_j for which $m_j \geq x$ is $O(n^3 \log n/x^2)$. By using a dyadic decomposition, we obtain

$$\begin{aligned} M^{-2/5} n^{-\epsilon} \sum_{j \in J} I(\Pi_j, \mathcal{S}_j) &= O\left(\sum_{m_j \leq \Delta} m_j^{7/5} + \sum_{k \geq 1} \sum_{2^{k-1} \Delta < m_j \leq 2^k \Delta} m_j^{7/5}\right) \\ &= O\left(\Delta^{7/5} \cdot \frac{n^2}{\Delta} + \sum_{k \geq 1} (2^k \Delta)^{7/5} \cdot \frac{n^3 \log n}{(2^k \Delta)^2}\right) \\ &= O\left(\Delta^{2/5} n^2 + \frac{n^3 \log n}{\Delta^{3/5}}\right). \end{aligned}$$

By setting $\Delta = n \log n$, we have

$$\sum_{j \in J} I(\Pi_j, \mathcal{S}_j) = O\left(M^{2/5} n^{12/5+\epsilon} \log^{2/5} n\right) = O\left(M^{2/5} n^{12/5+\epsilon'}\right).$$

In conclusion,

$$\mathcal{E}(\mathcal{P}) \leq |Q| + n^2 = \sum_{j \in J} I(\Pi_j, \mathcal{S}_j) + \sum_{j \notin J} I(\Pi_j, \mathcal{S}_j) + n^2 = O\left(M^{2/5} n^{12/5+\epsilon'} + Mn^2\right),$$

which completes the proof of Theorem 2.1.

► **Remark (Remark about the incidence bound).** Instead of partitioning the problem into D separate incidence problems, one can apply an incidence bound directly to the point set \mathcal{P}^{2*} and the surface set \mathcal{S} . Roughly speaking, the best known bounds for incidences with two-dimensional surfaces in \mathbb{R}^4 , whose incidence graph contains no $K_{2,M}$, are of the form $|\mathcal{P}^{2*}|^{2/3} |\mathcal{S}|^{2/3}$. Relying on such an incidence bound (and not using [11]) would yield a bound $|Q| = O(M^{1/3} n^{8/3} + Mn^2) = O(M^{1/3} n^{8/3})$, which is nontrivial but weaker than our bound.

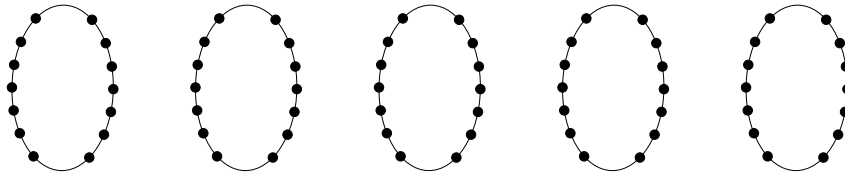
3.4 A lower bound for $\mathcal{E}(\mathcal{P})$

In this section we prove Theorem 2.2. In particular, for any n and $M(n) \geq 32$, we show that there exists a set \mathcal{P} of n points in \mathbb{R}^2 such that any line or circle contains at most $M(n)$ points of \mathcal{P} , and $\mathcal{E}(\mathcal{P}) = \Omega(M(n)n^2)$. Note that we can suppose $M(n) \geq 32$ without loss of generality since, if $M(n) < 32$, an arbitrary point set has $\mathcal{E}(\mathcal{P}) = \Omega(n^2) = \Omega(M(n)n^2)$.

For simplicity, we assume that $M(n)$ is a multiple of 8, and that n is divisible by $M(n)$. It is straightforward to extend the construction to values that do not satisfy these conditions.

Let C be an ellipse that is centered at the origin, has a major axis of length 2 that is parallel to the y -axis, and a minor axis of length 1 that is parallel to the x -axis. Let \mathcal{P}^+ be an arbitrary set of $4n/M(n)$ points on C , each having a strictly positive x -coordinate. Let \mathcal{P}^- be the reflection of \mathcal{P}^+ over the y -axis, and set $\mathcal{P}' = \mathcal{P}^+ \cup \mathcal{P}^-$. We denote by \mathcal{P}'_j the translate of \mathcal{P}' by $(4j, 0)$. Finally, we take $\mathcal{P} = \mathcal{P}'_0 \cup \mathcal{P}'_1 \cup \dots \cup \mathcal{P}'_{M(n)/8-1}$. An example is depicted in Figure 2.

Note that \mathcal{P} lies on the union of $M(n)/8$ ellipses. Since a line can intersect an ellipse in at most two points, and a circle can intersect an ellipse in at most four points, we indeed have that a line or circle contains at most $M(n)$ points of \mathcal{P} .



■ **Figure 2** The lower bound construction.

It remains to prove that $\mathcal{E}(\mathcal{P}) = \Omega(M(n)n^2)$. For every integer $M(n)/32 \leq j \leq M(n)/16$, we denote by ℓ_j the vertical line $x = 4j$. For every such j , there are $\Theta(n)$ points of \mathcal{P} that are to the left of ℓ_j , and such that the reflection of every such point across ℓ_j is another point of \mathcal{P} . That is, for every $M(n)/32 \leq j \leq M(n)/16$, the line ℓ_j is the perpendicular bisector of $\Theta(n)$ pairs of points of \mathcal{P} . The assertion of the theorem follows, since there are $\Theta(M(n))$ such lines, each contributing $\Theta(n^2)$ to $\mathcal{E}(\mathcal{P})$.

4 Proof of Theorem 2.3

In this section we prove that Theorem 2.3 follows from Theorem 2.1. In fact, we prove the following more general version of Theorem 2.3.

► **Theorem 4.1.** *Let $K(n)$ and $M(n)$ be two functions satisfying $K(n) = O(\log n)$ and $M(n) = O(n^{1/4})$. If an n -point set $\mathcal{P} \subset \mathbb{R}^2$ spans $D = O(n/K(n))$ distinct distances, then at least one of the following holds.*

- (i) *There exists a line or a circle containing $M(n)$ points of \mathcal{P} .*
- (ii) *There are $\Omega(M(n)^{-\frac{12}{5}} n^{\frac{8}{5}-\varepsilon})$ lines that contain $\Omega(K(n))$ points of \mathcal{P} .*

Since Guth and Katz [11] proved that any n -point set spans $\Omega(n/\log n)$ distinct distances, the assumption that $K = O(\log n)$ is not a real restriction. The original formulation of Theorem 2.3 is immediately obtained by setting $K(n) = \sqrt{\log n}$ and $M(n) = n^\alpha$.

Proof. For simplicity, we use the notation $K = K(n)$ and $M = M(n)$ throughout this proof. We assume that (i) does not hold, and prove that (ii) holds in this case.

Given a point set $\mathcal{P} \subset \mathbb{R}^2$, we denote by $\mathcal{B}^*(\mathcal{P})$ the *multiset* of bisectors that are spanned by ordered pairs of \mathcal{P}^{2*} . Recall that $\mathcal{B}(\mathcal{P})$ is the *set* of distinct lines of $\mathcal{B}^*(\mathcal{P})$. For every line $\ell \in \mathcal{B}(\mathcal{P})$, we denote by $\mu(\ell)$ its *multiplicity* in $\mathcal{B}^*(\mathcal{P})$ (i.e., the number of times it occurs in the multiset), and set $\rho(\ell) = |\ell \cap \mathcal{P}|$. We define

$$I(\mathcal{P}, \mathcal{B}^*(\mathcal{P})) = \sum_{\ell \in \mathcal{B}(\mathcal{P})} \mu(\ell)\rho(\ell);$$

that is, $I(\mathcal{P}, \mathcal{B}^*(\mathcal{P}))$ is the number incidences with respect to their multiplicities.

We derive a lower bound on $I(\mathcal{P}, \mathcal{B}^*(\mathcal{P}))$ by using an argument that is similar to the one in Szemerédi’s proof of Theorem 1.1. Let $T \subset \mathcal{P}^3$ be the set of triples (p, q, r) of distinct points of \mathcal{P} such that $|pq| = |pr|$. Note that a triple (p, q, r) is in T if and only if p is incident to $\mathcal{B}(q, r)$. That is,

$$I(\mathcal{P}, \mathcal{B}^*(\mathcal{P})) = |T|.$$

Denote the distances that are determined by pairs of \mathcal{P}^{2*} as $\delta_1, \dots, \delta_D$. For every point $p \in \mathcal{P}$ and $1 \leq i \leq D$, let $\Delta_{i,p}$ denote the number of points of \mathcal{P} that have distance δ_i from

p . Let $T_p \subset T$ denote the set of triples of T in which the first element is p . Applying the Cauchy-Schwarz inequality yields

$$|T_p| = \Omega \left(\sum_{i=1}^D \Delta_{i,p}^2 \right) = \Omega \left(\frac{1}{D} \left(\sum_{i=1}^D \Delta_{i,p} \right)^2 \right) = \Omega \left(\frac{n^2}{D} \right).$$

This in turn implies

$$I(\mathcal{P}, \mathcal{B}^*(\mathcal{P})) = |T| = \sum_{p \in \mathcal{P}} |T_p| = \Omega \left(\frac{n^3}{D} \right) = \Omega(Kn^2). \tag{5}$$

We remark that by the Szemerédi-Trotter theorem [23], the number of incidences between n points and n^2 distinct lines is $O(n^2)$. This does not contradict (5) since the lines in the multiset $\mathcal{B}^*(\mathcal{P})$ need not be distinct. A priori, it might be that $\mathcal{B}^*(\mathcal{P})$ consists of $\Theta(n)$ distinct lines, each with multiplicity $\Theta(n)$ and incident to $\Theta(K)$ points. However, our bound on the bisector energy excludes such cases.

Let c_t be the constant implicit in the lower bound on $|T|$; we have

$$|T| \geq c_t Kn^2.$$

Let L^+ be the subset of lines in $\mathcal{B}(\mathcal{P})$ that are each incident to at least $c_t K/2$ points. Then

$$\begin{aligned} c_t Kn^2 &\leq \sum_{\ell \in \mathcal{B}(\mathcal{P})} \mu(\ell)\rho(\ell) \\ &= \sum_{\ell \in L^+} \mu(\ell)\rho(\ell) + \sum_{\ell \in \mathcal{B}(\mathcal{P}) \setminus L^+} \mu(\ell)\rho(\ell) \\ &\leq \sum_{\ell \in L^+} \mu(\ell)\rho(\ell) + c_t Kn^2/2, \end{aligned}$$

where we use the fact that each ordered pair of points has a unique bisector, and hence contributes to $\sum_{\ell \in \mathcal{B}(\mathcal{P})} \mu(\ell)$ exactly once. Applying Cauchy-Schwarz, we get

$$c_t^2 K^2 n^4 / 4 \leq \sum_{\ell \in L^+} \mu(\ell)^2 \sum_{\ell \in L^+} \rho(\ell)^2.$$

Note that $\sum_{\ell \in \mathcal{B}(\mathcal{P})} \mu(\ell)^2 = \Theta(\mathcal{E}(\mathcal{P}))$. Since $M = O(n^{1/4}) = O(n^{2/3-\epsilon})$, Theorem 2.1 implies $\sum_{\ell \in \mathcal{B}(\mathcal{P})} \mu(\ell)^2 = O(M^{2/5} n^{12/5+\epsilon})$. We can bound $\sum \rho(\ell)^2$ using the assumption that no line contains more than M points, so

$$K^2 n^4 = O(M^{2/5} n^{12/5+\epsilon} \cdot M^2 |L^+|),$$

and hence

$$|L^+| = \Omega(K^2 n^{8/5-\epsilon} M^{-12/5}).$$

Since $K = O(\log n)$, it can be absorbed into the factor n^ϵ in the final bound. ◀

► Remark. Notice that the proof of Theorem 4.1 also applies when $M(n) = \Omega(n^{1/4})$. However, this would lead to a bound for the number of lines in (ii) that is weaker than the bound that is implied by Theorem 1.1.

5 Proof of Theorem 2.5

We now present the proof of the incidence bound that we use. As mentioned in the introduction, this proof is essentially from [9]; we reproduce it here to determine the dependence on the parameter t . We refer to [9] for the definitions used here. We prove a more general version than we need, since it seems to come at no extra cost, and may be useful elsewhere.

The proof uses the Kővári-Sós-Turán theorem (e.g., see Bollobás [1, Theorem IV.9]).

► **Lemma 5.1** (Kővári-Sós-Turán). *Let G be a bipartite graph with vertex set $A \cup B$. Let $s \leq t$. Suppose that G contains no $K_{s,t}$, i.e., for any s vertices in A , there are at most t vertices in B connected to both. Then*

$$|E(G)| = O(t^{1/s}|A||B|^{(s-1)/s} + |B|).$$

We amplify the weak bound of Lemma 5.1 by using *polynomial partitioning*. Given a polynomial $f \in \mathbb{R}[x_1, \dots, x_d]$, we write $Z(f) = \{p \in \mathbb{R}^d : f(p) = 0\}$. We say that $f \in \mathbb{R}[x_1, \dots, x_d]$ is an r -partitioning polynomial for a finite set $\mathcal{P} \subset \mathbb{R}^d$ if no connected component of $\mathbb{R}^d \setminus Z(f)$ contains more than $|\mathcal{P}|/r$ points of \mathcal{P} (notice that there is no restriction on the number of points of \mathcal{P} that are in $Z(f)$). Guth and Katz [11] introduced this notion and proved that for every $\mathcal{P} \subset \mathbb{R}^d$ and $1 \leq r \leq |\mathcal{P}|$, there exists an r -partitioning polynomial of degree $O(r^{1/d})$. In [9], the following generalization was proved.

► **Theorem 5.2** (Partitioning on a variety). *Let V be an irreducible variety in \mathbb{R}^d of dimension e and degree D . Then for every finite $\mathcal{P} \subset V$ there exists an r -partitioning polynomial f of degree $O(r^{1/e})$ such that $V \not\subset Z(f)$. The implicit constant depends only on d and D .*

We are now ready to prove our incidence bound. For the convenience of the reader, we first repeat the statement of the theorem.

Theorem 2.5. *Let \mathcal{S} be a set of n constant-degree varieties and let \mathcal{P} be a set of m points, both in \mathbb{R}^d , such that the incidence graph of $\mathcal{P} \times \mathcal{S}$ contains no copy of $K_{s,t}$ (where s is a constant, but t may depend on m, n). Moreover, assume that $\mathcal{P} \subset V$, where V is an irreducible constant-degree variety of dimension e . Then*

$$I(\mathcal{P}, \mathcal{S}) = O\left(m^{\frac{s(e-1)}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1}} t^{\frac{e-1}{es-1}} + tm + n\right).$$

Proof. We use induction on e and m , with the induction claim

$$I(\mathcal{P}, \mathcal{S}) \leq \alpha_{1,e} m^{\frac{s(e-1)}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1}} t^{\frac{e-1}{es-1}} + \alpha_{2,e}(tm + n), \tag{6}$$

for constants $\alpha_{1,e}, \alpha_{2,e}$ depending only on e . The base cases for the induction are simple. If m is sufficiently small, then (6) follows immediately by choosing sufficiently large values for $\alpha_{1,e}$ and $\alpha_{2,e}$. Similarly, when $e = 0$, we again obtain (6) when $\alpha_{1,e}$ and $\alpha_{2,e}$ are sufficiently large (as a function of d and the degree of V).

The constants $d, e, D, s, 1/\varepsilon$ are given and thus fixed. The other constants are to be chosen, and the dependencies between them are

$$C_{\text{weak}}, C_{\text{part}}, C_{\text{inter}} \ll C_{\text{cells}} \ll C_{\text{Höld}} \ll r \ll C_{\text{comps}} \ll \alpha_{2,e} \ll \alpha_{1,e},$$

where $C \ll C'$ means that C' is to be chosen sufficiently large compared to C ; in particular, C should be chosen before C' . Furthermore, the constants $\alpha_{1,e}, \alpha_{2,e}$ depend on $\alpha_{1,e-1}, \alpha_{2,e-1}$, which by the induction claim depend only on e .

By Lemma 5.1, there exists a constant C_{weak} depending on d, s such that

$$I(\mathcal{P}, \mathcal{S}) \leq C_{\text{weak}} \left(mn^{1-1/s}t^{1/s} + n \right).$$

When $m \leq (n/t)^{1/s}$, and $\alpha_{2,e}$ is sufficiently large, we have $I(\mathcal{P}, \mathcal{S}) \leq \alpha_{2,e}n$. Therefore, in the remainder of the proof we can assume that $n < m^s t$, which implies

$$n = n^{\frac{d-1}{ds-1}} n^{\frac{d(s-1)}{ds-1}} \leq m^{\frac{s(d-1)}{ds-1}} n^{\frac{d(s-1)}{ds-1}} t^{\frac{(d-1)}{ds-1}}. \tag{7}$$

Partitioning. By Theorem 5.2, there exists an r -partitioning polynomial f with respect to V of degree at most $C_{\text{part}} \cdot r^{1/e}$, for a constant C_{part} . Denote the cells of $V \setminus Z(f)$ as $\Omega_1, \dots, \Omega_N$. Since we are working over the reals, there exists a constant-degree polynomial g such that $Z(g) = V$. Then, by [22, Theorem A.2], the number of cells is bounded by $C_{\text{cells}} \cdot \deg(f)^{\dim V} = C_{\text{cells}} \cdot r$, for some constant C_{cells} depending on C_{part} .

We partition $I(\mathcal{P}, \mathcal{S})$ into the following three subsets:

- I_1 consists of the incidences $(p, S) \in \mathcal{P} \times \mathcal{S}$ such that $p \in V \cap Z(f)$, and some irreducible component of $V \cap Z(f)$ contains p and is fully contained in S .
- I_2 consists of the incidences $(p, S) \in \mathcal{P} \times \mathcal{S}$ such that $p \in V \cap Z(f)$, and no irreducible component of $V \cap Z(f)$ that contains p is contained in S .
- $I_3 = I(\mathcal{P}, \mathcal{S}) \setminus (I_1 \cup I_2)$, the set of incidences $(p, S) \in \mathcal{P} \times \mathcal{S}$ such that p is not contained in $V \cap Z(f)$.

Note that we indeed have $I(\mathcal{P}, \mathcal{S}) = I_1 + I_2 + I_3$.

Bounding I_1 . The points of $\mathcal{P} \subset \mathbb{R}^d$ that participate in incidences of I_1 are all contained in the variety $V_0 = V \cap Z(f)$. Set $\mathcal{P}_0 = \mathcal{P} \cap V_0$ and $m_0 = |\mathcal{P}_0|$. Since V is an irreducible variety and $V \not\subset Z(f)$, V_0 is a variety of dimension at most $e - 1$ and of degree that depends on r . By [22, Lemma 4.3], the intersection V_0 is a union of C_{comps} irreducible components, where C_{comps} is a constant depending on r and d .³ The degrees of these components also depend only on these values (for a proper definition of degrees and further discussion, e.g., see [9]).

Consider an irreducible component W of V_0 . If W contains at most $s - 1$ points of \mathcal{P}_0 , it yields at most $(s - 1)n$ incidences. Otherwise, since the incidence graph contains no $K_{s,t}$, there are at most $t - 1$ varieties of \mathcal{S} that fully contain W , yielding at most $(t - 1)m_0$ incidences. By summing up, choosing sufficiently large $\alpha_{1,e}, \alpha_{2,e}$, and applying (7), we have

$$I_1 \leq C_{\text{comps}}(sn + tm_0) < \frac{\alpha_{2,e}}{2}(n + tm_0) < \frac{\alpha_{1,e}}{4} m^{\frac{s(e-1)}{es-1}} n^{\frac{e(s-1)}{es-1}} t^{\frac{(e-1)}{es-1}} + \frac{\alpha_{2,e}}{2} tm_0. \tag{8}$$

Bounding I_2 . The points that participate in I_2 lie in $V_0 = V \cap Z(f)$, and the varieties that participate do not contain any component of V_0 . Because V_0 has dimension at most $e - 1$, and the participating varieties do not contain any component of V_0 , we can apply the induction claim on each irreducible component of V_0 . Since V_0 has C_{comps} irreducible components, we get

$$I_2 \leq C_{\text{comps}} \alpha_{1,e-1} m_0^{\frac{s(e-2)}{(e-1)s-1} + \epsilon} n^{\frac{(e-1)(s-1)}{(e-1)s-1}} t^{\frac{e-2}{(e-1)s-1}} + \alpha_{2,e-1}(tm_0 + n),$$

³ This lemma only applies to complex varieties. However, we can take the *complexification* of the real variety and apply the lemma to it (for the definition of a complexification, e.g., see [25, Section 10]). The number of irreducible components of the complexification cannot be smaller than number of irreducible components of the real variety (e.g., see [25, Lemma 7]).

with $\alpha_{1,e-1}$ and $\alpha_{2,e-1}$ depending on the degree of the irreducible component of V_0 , which in turn depends on r . The analysis that leads to (7) also yields the following bound.

$$m^{\frac{s(e-2)}{(e-1)s-1} + \varepsilon} n^{\frac{(e-1)(s-1)}{(e-1)s-1} \frac{e-2}{t^{(e-1)s-1}}} \leq m^{\frac{s(e-1)}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1} \frac{e-1}{t^{es-1}}}.$$

By applying (7) to remove the term $\alpha_{2,e-1}n$, and by choosing $\alpha_{1,e}$ and $\alpha_{2,e}$ sufficiently large as a function of $C_{\text{comps}}, \alpha_{1,e-1}, \alpha_{2,e-1}$, we obtain

$$I_2 \leq \frac{\alpha_{1,e}}{4} m^{\frac{s(e-1)}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1} \frac{e-1}{t^{es-1}}} + \frac{\alpha_{2,e}}{2} tm_0. \quad (9)$$

Bounding I_3 . For every $1 \leq i \leq N$, we set $\mathcal{P}_i = \mathcal{P} \cap \Omega_i$ and denote by \mathcal{S}_i the set of varieties of \mathcal{S} that intersect the cell Ω_i but do not contain it. We also set $m_i = |\mathcal{P}_i|$ and $n_i = |\mathcal{S}_i|$. Since f is an r -partitioning polynomial, we have $m_i \leq m/r$.

We have $\sum_{i=1}^N m_i = m - m_0$. By [22, Theorem A.2], there exists a constant C_{inter} such that the following holds for every $S \in \mathcal{S}$. The subvariety $S \cap V$ of V , which must have dimension at most $e-1$, intersects at most $C_{\text{inter}} \cdot \deg(f)^{\dim(S \cap V)} = C_{\text{inter}} \cdot r^{(e-1)/e}$ cells. This implies that

$$\sum_{i=1}^N n_i \leq C_{\text{inter}} \cdot r^{(e-1)/e} \cdot n.$$

By Hölder's inequality we have

$$\begin{aligned} \sum_{i=1}^N n_i^{\frac{e(s-1)}{es-1}} &\leq \left(\sum_{i=1}^N n_i \right)^{\frac{e(s-1)}{es-1}} \left(\sum_{i=1}^N 1 \right)^{\frac{e-1}{es-1}} \\ &\leq \left(C_{\text{inter}} r^{(e-1)/e} n \right)^{\frac{e(s-1)}{es-1}} (C_{\text{cells}} r)^{\frac{e-1}{es-1}} \\ &\leq C_{\text{Höld}} r^{\frac{(e-1)s}{es-1}} n^{\frac{e(s-1)}{es-1}}, \end{aligned}$$

where $C_{\text{Höld}}$ depends on $C_{\text{inter}}, C_{\text{cells}}$. Using the induction hypothesis, we obtain

$$\begin{aligned} \sum_{i=1}^N I(\mathcal{P}_i, \mathcal{S}_i) &\leq \sum_{i=1}^N \left(\alpha_{1,e} m_i^{\frac{(e-1)s}{es-1} + \varepsilon} n_i^{\frac{e(s-1)}{es-1} \frac{e-1}{t^{es-1}}} + \alpha_{2,e} (tm_i + n_i) \right) \\ &\leq \alpha_{1,e} \frac{m^{\frac{(e-1)s}{es-1} + \varepsilon} t^{\frac{(e-1)}{es-1}}}{r^{\frac{(e-1)s}{es-1} + \varepsilon}} \sum_{i=1}^N n_i^{\frac{e(s-1)}{es-1}} + \sum_{i=1}^N \alpha_{2,e} (tm_i + n_i) \\ &\leq \alpha_{1,e} C_{\text{Höld}} \frac{m^{\frac{(e-1)s}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1} \frac{e-1}{t^{es-1}}}}{r^\varepsilon} + \alpha_{2,e} \left(t(m - m_0) + C_{\text{inter}} r^{\frac{e-1}{e}} n \right). \end{aligned}$$

By choosing $\alpha_{1,e}$ sufficiently large with respect to $C_{\text{inter}}, r, \alpha_{2,e}$, and using (7), we get

$$\sum_{i=1}^N I(\mathcal{P}_i, \mathcal{S}_i) \leq 2\alpha_{1,e} C_{\text{Höld}} \frac{m^{\frac{(e-1)s}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1} \frac{e-1}{t^{es-1}}}}{r^\varepsilon} + \alpha_{2,e} t(m - m_0).$$

Finally, choosing r sufficiently large with respect to $C_{\text{Höld}}$ gives

$$I_3 = \sum_{i=1}^N I(\mathcal{P}_i, \mathcal{S}_i) \leq \frac{\alpha_{1,e}}{2} m^{\frac{(e-1)s}{es-1} + \varepsilon} n^{\frac{e(s-1)}{es-1} \frac{e-1}{t^{es-1}}} + \alpha_{2,e} t(m - m_0). \quad (10)$$

Summing up. By combining $I(\mathcal{P}, \mathcal{S}) = I_1 + I_2 + I_3$ with (8), (9), and (10), we obtain

$$I(\mathcal{P}, \mathcal{S}) \leq \alpha_{1,\varepsilon} m^{\frac{s(\varepsilon-1)}{\varepsilon s-1} + \varepsilon} n^{\frac{\varepsilon(s-1)}{\varepsilon s-1}} t^{\frac{(\varepsilon-1)}{\varepsilon s-1}} + \alpha_{2,\varepsilon}(tm + n),$$

which completes the induction step and the proof of the theorem. \blacktriangleleft

References

- 1 B. Bollobás, *Graph Theory: An Introductory Course*, Springer-Verlag, 1979.
- 2 G. Elekes and L. Rónyai, A combinatorial problem on polynomials and rational functions, *J. Combin. Theory Ser. A* **89** (2000), 1–20.
- 3 G. Elekes and E. Szabó, How to find groups? (And how to use them in Erdős geometry?), *Combinatorica* **32** (2012), 537–571.
- 4 G. Elekes and E. Szabó, On triple lines and cubic curves: The Orchard Problem revisited, in [arXiv:1302.5777](https://arxiv.org/abs/1302.5777) (2013).
- 5 P. Erdős, On sets of distances of n points, *Amer. Math. Monthly* **53** (1946), 248–250.
- 6 P. Erdős, On some problems of elementary and combinatorial geometry, *Ann. Mat. Pura Appl.* **103** (1975), 99–108.
- 7 P. Erdős, On some metric and combinatorial geometric problems, *Discrete Math.* **60** (1986), 147–153.
- 8 P. Erdős and G. Purdy, Some extremal problems in geometry IV, *Proc. 7th Southeastern Conference on Combinatorics, Graph Theory, and Computing* (1976), 307–322.
- 9 J. Fox, J. Pach, A. Sheffer, A. Suk, and J. Zahl, A semi-algebraic version of Zarankiewicz’s problem, in [arXiv:1407.5705](https://arxiv.org/abs/1407.5705) (2014).
- 10 B. Green and T. Tao, On sets defining few ordinary lines, *Disc. Comput. Geom.* **50** (2013), 409–468.
- 11 L. Guth and N.H. Katz, On the Erdős distinct distances problem in the plane, *Annals Math.* **181** (2015), 155–190.
- 12 B. Hanson, A. Iosevich, B. Lund, and O. Roche-Newton, On distinct perpendicular bisectors and pinned distances in finite fields, in [arXiv:1412.1611](https://arxiv.org/abs/1412.1611) (2014).
- 13 N.H. Katz and G. Tardos, A new entropy inequality for the Erdős distance problem, *Towards a Theory of Geometric Graphs* (J. Pach, ed.), Contemporary Mathematics **342** (2004), 119–126.
- 14 J. Pach and G. Tardos, Isosceles Triangles Determined by a Planar Point Set, *Graphs and Combinatorics* **18** (2002), 769–779.
- 15 J. Pach and F. de Zeeuw, Distinct distances on algebraic curves in the plane, *Proc. 30th Ann. Symp. on Comp. Geometry* (2014), 549–557.
- 16 O.E. Raz, O. Roche-Newton, and M. Sharir, Sets with few distinct distances do not have heavy lines, in [arXiv:1410.1654](https://arxiv.org/abs/1410.1654) (2014).
- 17 O.E. Raz, M. Sharir, and J. Solymosi, Polynomials vanishing on grids: The Elekes-Rónyai problem revisited, *Proc. 30th Ann. Symp. on Comp. Geometry* (2014), 251–260.
- 18 O.E. Raz, M. Sharir, and F. de Zeeuw, Polynomials vanishing on Cartesian products: The Elekes-Szabó theorem revisited, *Proc. 31st Ann. Symp. on Comp. Geometry* (2015).
- 19 A. Sheffer, Few Distinct Distances Implies Many Points on a Line, blog post, 2014.
- 20 A. Sheffer, J. Zahl, and F. de Zeeuw, Few distinct distances implies no heavy lines or circles, *Combinatorica*, to appear.
- 21 J. Solymosi and M. Stojaković, Many collinear k -tuples with no $k + 1$ collinear points, *Discrete Comput. Geom.* **50** (2013), 811–820.
- 22 J. Solymosi and T. Tao, An incidence theorem in higher dimensions, *Discrete Comput. Geom.* **48** (2012), 255–280.

- 23 E. Szemerédi and W. Trotter, Extremal problems in discrete geometry, *Combinatorica* **3** (1983), 381–392.
- 24 G. Tardos, On distinct sums and distinct distances, *Adv. Math.* **180** (2003), 275–289.
- 25 H. Whitney, Elementary structure of real algebraic varieties, *Annals Math.* **66** (1957), 545–556.

Incidences between Points and Lines in Three Dimensions*

Micha Sharir and Noam Solomon

School of Computer Science, Tel Aviv University, Tel Aviv, Israel
michas@post.tau.ac.il, noam.solom@gmail.com

Abstract

We give a fairly elementary and simple proof that shows that the number of incidences between m points and n lines in \mathbb{R}^3 , so that no plane contains more than s lines, is

$$O(m^{1/2}n^{3/4} + m^{2/3}n^{1/3}s^{1/3} + m + n)$$

(in the precise statement, the constant of proportionality of the first and third terms depends, in a rather weak manner, on the relation between m and n).

This bound, originally obtained by Guth and Katz [8] as a major step in their solution of Erdős's distinct distances problem, is also a major new result in incidence geometry, an area that has picked up considerable momentum in the past six years. Its original proof uses fairly involved machinery from algebraic and differential geometry, so it is highly desirable to simplify the proof, in the interest of better understanding the geometric structure of the problem, and providing new tools for tackling similar problems. This has recently been undertaken by Guth [6]. The present paper presents a different and simpler derivation, with better bounds than those in [6], and without the restrictive assumptions made there. Our result has a potential for applications to other incidence problems in higher dimensions.

1998 ACM Subject Classification G.2.1 Combinatorics

Keywords and phrases Combinatorial Geometry, Algebraic Geometry, Incidences, The Polynomial Method

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.553

1 Introduction

Let P be a set of m distinct points in \mathbb{R}^3 and let L be a set of n distinct lines in \mathbb{R}^3 . Let $I(P, L)$ denote the number of incidences between the points of P and the lines of L ; that is, the number of pairs (p, ℓ) with $p \in P$, $\ell \in L$, and $p \in \ell$. If all the points of P and all the lines of L lie in a common plane, then the classical Szemerédi–Trotter theorem [24] yields the worst-case tight bound

$$I(P, L) = O(m^{2/3}n^{2/3} + m + n). \quad (1)$$

This bound clearly also holds in three dimensions, by projecting the given lines and points onto some generic plane. Moreover, the bound will continue to be worst-case tight by placing

* Work on this paper by Noam Solomon and Micha Sharir was supported by Grant 892/13 from the Israel Science Foundation. Work by Micha Sharir was also supported by Grant 2012/229 from the U.S.–Israel Binational Science Foundation, by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11), and by the Hermann Minkowski-MINERVA Center for Geometry at Tel Aviv University.



all the points and lines in a common plane, in a configuration that yields the planar lower bound.

In the 2010 groundbreaking paper of Guth and Katz [8], an improved bound has been derived for $I(P, L)$, for a set P of m points and a set L of n lines in \mathbb{R}^3 , provided that not too many lines of L lie in a common plane. Specifically, they showed:¹

► **Theorem 1** (Guth and Katz [8]). *Let P be a set of m distinct points and L a set of n distinct lines in \mathbb{R}^3 , and let $s \leq n$ be a parameter, such that no plane contains more than s lines of L . Then*

$$I(P, L) = O\left(m^{1/2}n^{3/4} + m^{2/3}n^{1/3}s^{1/3} + m + n\right).$$

This bound was a major step in the derivation of the main result of [8], which was to prove an almost-linear lower bound on the number of distinct distances determined by any finite set of points in the plane, a classical problem posed by Erdős in 1946 [5]. Their proof uses several nontrivial tools from algebraic and differential geometry, most notably the Cayley–Salmon theorem on osculating lines to algebraic surfaces in \mathbb{R}^3 , and additional properties of ruled surfaces. All this machinery comes on top of the main innovation of Guth and Katz, the introduction of the *polynomial partitioning technique*; see below.

In this paper, we provide a simple derivation of this bound, which bypasses most of the techniques from algebraic geometry that are used in the original proof. A recent related study by Guth [6] provides another simpler derivation of a similar bound, but (a) the bound obtained in [6] is slightly worse, involving extra factors of the form m^ε , for any $\varepsilon > 0$, and (b) the assumptions there are stronger, namely that no algebraic surface of degree at most c_ε , a (potentially large) constant that depends on ε , contains more than s lines of L (in fact, Guth considers in [6] only the case $s = \sqrt{n}$). It should be noted, though, that Guth also manages to derive a (slightly weaker but still) near-linear lower bound on the number of distinct distances.

As in the classical work of Guth and Katz [8], and in the follow-up study of Guth [6], here too we use the polynomial partitioning method, as pioneered in [8]. The main difference between our approach and those of [6, 8] is the choice of the degree of the partitioning polynomial. Whereas Guth and Katz [8] choose a large degree, and Guth [6] chooses a constant degree, we choose an intermediate degree. This reaps many benefits from both the high-degree and the constant-degree approaches, and pays a small price in the bound (albeit much better than in [6]). Specifically, our main result is a simple and fairly elementary derivation of the following result.

► **Theorem 2.** *Let P be a set of m distinct points and L a set of n distinct lines in \mathbb{R}^3 , and let $s \leq n$ be a parameter, such that no plane contains more than s lines of L . Then*

$$I(P, L) \leq A_{m,n} \left(m^{1/2}n^{3/4} + m\right) + B \left(m^{2/3}n^{1/3}s^{1/3} + n\right), \quad (2)$$

where B is an absolute constant, and, for another suitable absolute constant $b > 1$,

$$A_{m,n} = O\left(\frac{\log(m^2n)}{b^{\log(n^3/m^2)}}\right), \quad \text{for } m \leq n^{3/2}, \quad \text{and} \quad O\left(\frac{\log(m^3/n^4)}{b^{\log(m^2/n^3)}}\right), \quad \text{for } m \geq n^{3/2}. \quad (3)$$

¹ We skip over certain subtleties in their bound: They also assume that no *regulus* contains more than s input lines, but then they are able also to bound the number of intersection points of the lines. Moreover, if one also assumes that each point is incident to at least three lines then the term m in the bound can be dropped.

- **Remarks. 1.** Only the range $\sqrt{n} \leq m \leq n^2$ is of interest; outside this range, regardless of the dimension of the ambient space, we have the well known and trivial upper bound $O(m + n)$.
2. The term $m^{2/3}n^{1/3}s^{1/3}$ comes from the planar Szemerédi–Trotter bound (1), and is unavoidable, as it can be attained if we densely “pack” points and lines into planes, in patterns that realize the bound in (1).
 3. Ignoring this term and the term n (needed only to cater to the case $m \ll n^{1/2}$), the two terms $m^{1/2}n^{3/4}$ and m “compete” for dominance; the former dominates when $m \leq n^{3/2}$ and the latter when $m \geq n^{3/2}$. Thus the bound in (2) is qualitatively different within these two ranges.
 4. The threshold $m = n^{3/2}$ also arises in the related problem of *joints* (points incident to at least three non-coplanar lines) in a set of n lines in 3-space; see [7].

A concise rephrasing of the bound in (2) and (3) is as follows. We partition each of the ranges $m \leq n^{3/2}$, $m > n^{3/2}$ into a sequence of subranges $n^{\alpha_{j-1}} < m \leq n^{\alpha_j}$, $j = 0, 1, \dots$ (for $m \leq n^{3/2}$), or $n^{\alpha_{j-1}} > m \geq n^{\alpha_j}$, $j = 0, 1, \dots$ (for $m \geq n^{3/2}$), so that within each range the bound asserted in the theorem holds for some fixed constant of proportionality (denoted as $A_{m,n}$ in the bound), where these constants grow, exponentially in j , as prescribed in (3), as m approaches $n^{3/2}$ (from either side). Informally, if we keep m “sufficiently away” from $n^{3/2}$, the bound in (2) holds with a fixed constant of proportionality. Handling the “border range” $m \approx n^{3/2}$ is also fairly straightforward, although, to bypass the exponential growth of the constant of proportionality, it results in a slightly different bound; see below for details.

Our proof is elementary to the extent that, among other things, it avoids any explicit handling of *singular* and *flat* points on the zero set of the partitioning polynomial. While these notions are relatively easy to handle in three dimensions (see, e.g., [4, 7]), they become more complex notions in higher dimensions (as witnessed, for example, in our companion work on the four-dimensional setting [19]), making proofs based on them harder to extend.

Additional merits and features of our analysis are discussed in detail in the concluding section. In a nutshell, the main merits are:

- (i) We use two separate partitioning polynomials. The first one is of “high” degree, and is used to prune away some points and lines, and to establish useful properties of the surviving points and lines. The second partitioning step, using a polynomial of “low” degree, is then applied, from scratch, to the surviving input, exploiting the properties established in the first step. This idea seems to have a potential for further applications (as in [19]).
- (ii) Because of the way we use the polynomial partitioning technique, we need induction to handle incidences within the cells of the second partition. One of the nontrivial achievements of our technique is the ability to retain the “planar” term $O(m^{2/3}n^{1/3}s^{1/3})$ in the bound in (2) through the inductive process. Without such care, this term does not “pass well” through the induction, which has been a sore issue in several recent works on related problems (see [16, 17, 18]). This is one of the main reasons for using two separate partitioning steps.

Background

Incidence problems have been a major topic in combinatorial and computational geometry for the past thirty years, starting with the aforementioned Szemerédi–Trotter bound [24] back in 1983. Several techniques, interesting in their own right, have been developed, or adapted, for the analysis of incidences, including the crossing-lemma technique of Székely [23], and

the use of cuttings as a divide-and-conquer mechanism (e.g., see [2]). Connections with range searching and related algorithmic problems in computational geometry have also been noted, and studies of the Kakeya problem (see, e.g., [25]) indicate the connection between this problem and incidence problems. See Pach and Sharir [13] for a comprehensive (albeit a bit outdated) survey of the topic.

The landscape of incidence geometry has dramatically changed in the past six years, due to the infusion, in two groundbreaking papers by Guth and Katz [7, 8], of new tools and techniques drawn from algebraic geometry. Although their two direct goals have been to obtain a tight upper bound on the number of joints in a set of lines in three dimensions [7], and a near-linear lower bound for the classical distinct distances problem of Erdős [8], the new tools have quickly been recognized as useful for incidence bounds. See [4, 9, 10, 17, 22, 28, 29] for a sample of recent works on incidence problems that use the new algebraic machinery.

The simplest instances of incidence problems involve points and lines, tackled by Szemerédi and Trotter in the plane [24], and by Guth and Katz in three dimensions [8]. Other recent studies on incidence problems include incidences between points and lines in four dimensions (Sharir and Solomon [18, 19]), and incidences between points and circles in three dimensions (Sharir, Sheffer and Zahl [17]), not to mention incidences with higher-dimensional surfaces, such as in [1, 9, 22, 28, 29]. In a companion paper (with Sheffer) [16], we study the general case of incidences between points and curves in any dimension, and derive reasonably sharp bounds (albeit weaker in several respects than the one derived here).

The fact that tools from algebraic geometry form the major key for successful solution of difficult problems in combinatorial geometry, has led to intensive research of the new tools, aiming to extend them and to find new applications. A major purpose of this study, as well as of Guth [6], is to show that one can still tackle successfully the problems using less heavy algebraic machinery. This offers a new, simplified, and more elementary approach, which we expect to prove potent for other applications too, such as those just mentioned. Looking for simpler, yet effective techniques that would be easier to extend to more involved contexts (such as incidences in higher dimensions) has been our main motivation for this study. See the concluding section for further discussion.

2 Proof of Theorem 2

The proof proceeds by induction on m . As already mentioned, the bound in (2) is qualitatively different in the two ranges $m \leq n^{3/2}$ and $m \geq n^{3/2}$. The analysis bifurcates accordingly. While the general flow is fairly similar in both cases, there are many differences too.

The case $m < n^{3/2}$

We partition this range into a sequence of ranges $m \leq n^{\alpha_0}$, $n^{\alpha_0} < m \leq n^{\alpha_1}, \dots$, where $\alpha_0 = 1/2$ and the sequence $\{\alpha_j\}_{j \geq 0}$ is increasing and converges to $3/2$. More precisely, as our analysis will show, we can take $\alpha_j = \frac{3}{2} - \frac{2}{j+2}$, for $j \geq 0$. The induction is actually on the index j of the range $n^{\alpha_{j-1}} < m \leq n^{\alpha_j}$, and establishes (2) for m in this range, with a coefficient A_j (written in (2, 3) as $A_{m,n}$) that increases with j . This paradigm has already been used in Sharir et al. [17] and in Zahl [29], for related incidence problems, albeit in a somewhat less effective manner; see the discussion at the end of the paper.

The base range of the induction is $m \leq \sqrt{n}$, where the trivial general upper bound on point-line incidences, in any dimension, yields $I = O(m^2 + n) = O(n)$, so (2) holds for a sufficiently large choice of the initial constant A_0 .

Assume then that (2) holds for all $m \leq n^{\alpha_{j-1}}$ for some $j \geq 1$, and consider an instance of the problem with $n^{\alpha_{j-1}} < m \leq n^{3/2}$ (the analysis will force us to constrain this upper bound in order to complete the induction step, thereby obtaining the next exponent α_j).

Fix a parameter r , whose precise value will be chosen later (in fact, and this is a major novelty of our approach, there will be two different choices for r —see below), and apply the polynomial partitioning theorem of Guth and Katz (see [8] and [10, Theorem 2.6]), to obtain an r -partitioning trivariate (real) polynomial f of degree $D = O(r^{1/3})$. That is, every connected component of $\mathbb{R}^3 \setminus Z(f)$ contains at most m/r points of P , where $Z(f)$ denotes the zero set of f . By Warren’s theorem [27] (see also [10]), the number of components of $\mathbb{R}^3 \setminus Z(f)$ is $O(D^3) = O(r)$.

Set $P_1 := P \cap Z(f)$ and $P'_1 := P \setminus P_1$. A major recurring theme in this approach is that, although the points of P'_1 are more or less evenly partitioned among the cells of the partition, no nontrivial bound can be provided for the size of P_1 ; in the worst case, all the points of P could lie in $Z(f)$. Each line $\ell \in L$ is either fully contained in $Z(f)$ or intersects it in at most D points (since the restriction of f to ℓ is a univariate polynomial of degree at most D). Let L_1 denote the subset of lines of L that are fully contained in $Z(f)$ and put $L'_1 = L \setminus L_1$. We then have

$$I(P, L) = I(P_1, L_1) + I(P_1, L'_1) + I(P'_1, L'_1).$$

We first bound $I(P_1, L'_1)$ and $I(P'_1, L'_1)$. As already observed, we have

$$I(P_1, L'_1) \leq |L'_1| \cdot D \leq nD.$$

We estimate $I(P'_1, L'_1)$ as follows. For each (open) cell τ of $\mathbb{R}^3 \setminus Z(f)$, put $P_\tau = P \cap \tau$ (that is, $P'_1 \cap \tau$), and let L_τ denote the set of the lines of L'_1 that cross τ ; put $m_\tau = |P_\tau| \leq m/r$, and $n_\tau = |L_\tau|$. Since every line $\ell \in L'_1$ crosses at most $1 + D$ components of $\mathbb{R}^3 \setminus Z(f)$, we have

$$\sum_\tau n_\tau \leq n(1 + D), \quad \text{and} \quad I(P'_1, L'_1) = \sum_\tau I(P_\tau, L_\tau).$$

For each τ we use the trivial bound $I(P_\tau, L_\tau) = O(m_\tau^2 + n_\tau)$. Summing over the cells, we get

$$I(P'_1, L'_1) = \sum_\tau I(P_\tau, L_\tau) = O\left(r \cdot (m/r)^2 + \sum_\tau n_\tau\right) = O(m^2/r + nD) = O(m^2/D^3 + nD).$$

For the initial value of D , we take $D = m^{1/2}/n^{1/4}$ (which we get from a suitable value of $r = \Theta(D^3)$), note that $1 \leq D \leq m^{1/3}$, and get the bound

$$I(P'_1, L'_1) + I(P_1, L'_1) = O(m^{1/2}n^{3/4}).$$

This choice of D is the one made in [8]. It is sufficiently large to control the situation in the cells, by the bound just obtained, but requires heavy-duty machinery from algebraic geometry to handle the situation on $Z(f)$.

We now turn to $Z(f)$, where we need to estimate $I(P_1, L_1)$. Since all the incidences involving any point in P'_1 and/or any line in L'_1 have already been accounted for, we discard these sets, and remain with P_1 and L_1 only. We “forget” the preceding polynomial partitioning step, and start afresh, applying a new polynomial partitioning to P_1 with a polynomial g of degree E , which will typically be much smaller than D , but still non-constant.

Before doing this, we note that the set of lines L_1 has a special structure, because all its lines lie on the algebraic surface $Z(f)$, which has degree D . We exploit this to derive the following lemmas. We emphasize, since this will be important later on in the analysis, that Lemmas 3–7 hold for any choice of (r and) D .

We note that in general the partitioning polynomial f may be reducible, and apply some of the following arguments to each irreducible factor separately. Clearly, there are at most D such factors.

► **Lemma 3.** *Let π be a plane which is not a component of $Z(f)$. Then π contains at most D lines of L_1 .*

Proof. Suppose to the contrary that π contains at least $D + 1$ lines of L . Every generic line λ in π intersects these lines in at least $D + 1$ distinct points, all belonging to $Z(f)$. Hence f must vanish identically on λ , and it follows that $f \equiv 0$ on π , so π is a component of $Z(f)$, contrary to assumption. ◀

► **Lemma 4.** *The number of incidences between the points of P_1 that lie in the planar components of $Z(f)$ and the lines of L_1 , is $O(m^{2/3}n^{1/3}s^{1/3} + nD)$.*

Proof. Clearly, f can have at most D linear factors, and thus $Z(f)$ can contain at most D planar components. Enumerate them as π_1, \dots, π_k , where $k \leq D$. Let \tilde{P}_1 denote the subset of the points of P_1 that lie in these planar components. Assign each point of \tilde{P}_1 to the first plane π_i , in this order, that contains it, and assign each line of L_1 to the first plane that fully contains it; some lines might not be assigned at all in this manner. For $i = 1, \dots, k$, let \tilde{P}_i denote the set of points assigned to π_i , and let \tilde{L}_i denote the set of lines assigned to π_i . Put $m_i = |\tilde{P}_i|$ and $n_i = |\tilde{L}_i|$. Then $\sum_i m_i \leq m$ and $\sum_i n_i \leq n$; by assumption, we also have $n_i \leq s$ for each i . Then

$$I(\tilde{P}_i, \tilde{L}_i) = O(m_i^{2/3}n_i^{2/3} + m_i + n_i) = O(m_i^{2/3}n_i^{1/3}s^{1/3} + m_i + n_i).$$

Summing over the k planes, we get, using Hölder's inequality,

$$\sum_i I(\tilde{P}_i, \tilde{L}_i) = \sum_i O(m_i^{2/3}n_i^{1/3}s^{1/3} + m_i + n_i) = O\left(m^{2/3}n^{1/3}s^{1/3} + m + n\right).$$

We also need to include incidences between points $p \in \tilde{P}_1$ and lines $\ell \in L_1$ not assigned to the same plane as p (or not assigned to any plane at all). Any such incidence (p, ℓ) can be charged (uniquely) to the intersection point of ℓ with the plane π_i to which p has been assigned. The number of such intersections is $O(nD)$, and the lemma follows. ◀

► **Lemma 5.** *Each point $p \in Z(f)$ is incident to at most D^2 lines of L_1 , unless $Z(f)$ has an irreducible component that is either a plane containing p or a cone with apex p .*

Proof. Fix any line ℓ that passes through p , and write its parametric equation as $\{p + tv \mid t \in \mathbb{R}\}$, where v is the direction of ℓ . Consider the Taylor expansion of f at p along ℓ , namely $f(p + tv) = \sum_{i=1}^D \frac{1}{i!} F_i(p; v) t^i$, where $F_i(p; v)$ is the i -th order derivative of f at p in direction v ; it is a homogeneous polynomial in v (p is considered fixed) of degree i , for $i = 1, \dots, D$. For each line $\ell \in L_1$ that passes through p , f vanishes identically on ℓ , so we have $F_i(p; v) = 0$ for each i . Assuming that p is incident to more than D^2 lines of L_1 , we conclude that the homogeneous system

$$F_1(p; v) = F_2(p; v) = \dots = F_D(p; v) = 0 \tag{4}$$

has more than D^2 (projectively distinct) roots. The classical Bézout's theorem, applied in the projective plane where the directions v are represented (e.g., see [3]), asserts that, since

all these polynomials are of degree at most D , each pair of polynomials $F_i(p; v), F_j(p; v)$ must have a common factor. The following slightly more involved inductive argument shows that in fact all these polynomials must have a common factor.²

► **Lemma 6.** *Let $f_1, \dots, f_n \in \mathbb{C}[x, y, z]$ be n homogeneous polynomials of degree at most D . If $|Z(f_1, \dots, f_n)| > D^2$, then all the f_i 's have a nontrivial common factor.*

Proof. The proof is via induction on n , and is omitted in this version. ◀

Continuing with the proof of Lemma 5, there is an infinity of directions v that satisfy (4), so there is an infinity of lines passing through v and contained in $Z(f)$. The union of these lines can be shown to be a two-dimensional algebraic variety,³ contained in $Z(f)$, so $Z(f)$ has an irreducible component that is either a plane through p or a cone with apex p , as claimed. ◀

► **Lemma 7.** *The number of incidences between the points of P_1 that lie in the (non-planar) conic components of $Z(f)$, and the lines of L_1 , is $O(m + nD)$.*

Proof. Let σ be such an (irreducible) conic component of $Z(f)$ and let p be its apex. We observe that σ cannot contain any line that is not incident to p , because such a line would span with p a plane contained in σ , contradicting the assumption that σ is irreducible and non-planar. It follows that the number of incidences between $P_\sigma := P_1 \cap \sigma$ and L_σ , consisting of the lines of L_1 contained in σ , is thus $O(|P_\sigma| + |L_\sigma|)$ (p contributes $|L_\sigma|$ incidences, and every other point at most one incidence). Applying a similar “first-come-first-serve” assignment of points and lines to the conic components of $Z(f)$, as we did for the planar components in the proof of lemma 4, and adding the bound $O(nD)$ on the number of incidences between points and lines not assigned to the same component, we obtain the bound asserted in the lemma. ◀

► **Remark.** Note that in both Lemma 4 and Lemma 7, we bound the number of incidences between points on planar or conic components of $Z(f)$ and *all* the lines of L_1 .

Pruning. To continue, we remove all the points of P_1 that lie in some planar or conic component of $Z(f)$, and all the lines of L_1 that are fully contained in such components. With the choice of $D = m^{1/2}/n^{1/4}$, we lose in the process

$$O(m^{2/3}n^{1/3}s^{1/3} + m + nD) = O(m^{1/2}n^{3/4} + m^{2/3}n^{1/3}s^{1/3})$$

incidences (recall that the term m is subsumed by the term $m^{1/2}n^{3/4}$ for $m < n^{3/2}$). Continue, for simplicity of notation, to denote the sets of remaining points and lines as P_1 and L_1 , respectively, and their sizes as m and n . Now each point is incident to at most D^2 lines (a fact that we will not use for this value of D), and no plane contains more than D lines of L_1 , a crucial property for the next steps of the analysis. That is, this allows us to replace the input parameter s , bounding the maximum number of coplanar lines, by D ; this is a key step that makes the induction work.

² See also [14] for a similar observation.

³ It is simply the variety given by the equations (4), rewritten as $F_1(p; x - p) = F_2(p; x - p) = \dots = F_D(p; x - p) = 0$. It is two-dimensional because it is contained in $Z(f)$, hence at most two-dimensional, and it cannot be one-dimensional since it would then consist of only finitely many lines (see, e.g., [19, Lemma 2.3]).

A new polynomial partitioning

We now return to the promised step of constructing a new polynomial partitioning. We adapt the preceding notation, with a few modifications. We choose a degree E , typically much smaller than D , and construct a partitioning polynomial g of degree E for P_1 . With an appropriate value of $r = \Theta(E^3)$, we obtain $O(r)$ open cells, each containing at most m/r points of P_1 , and each line of L_1 either crosses at most $E + 1$ cells, or is fully contained in $Z(g)$.

Set $P_2 := P_1 \cap Z(g)$ and $P'_2 := P_1 \setminus P_2$. Similarly, denote by L_2 the set of lines of L_1 that are fully contained in $Z(g)$, and put $L'_2 := L_1 \setminus L_2$. We first dispose of incidences involving the lines of L_2 . (That is, now we first focus on incidences within $Z(g)$, and only then turn to look at the cells.) By Lemma 4 and Lemma 7, the number of incidences involving points P_2 that lie in some planar or conic component of $Z(g)$, and all the lines of L_2 , is

$$O(m^{2/3}n^{1/3}s^{1/3} + m + nE) = O(m^{1/2}n^{3/4} + m^{2/3}n^{1/3}s^{1/3} + n).$$

(For $E \ll D$, this might be a gross overestimation, but we do not care.) We remove these points from P_2 , and remove all the lines of L_2 that are contained in such components; continue to denote the sets of remaining points and lines as P_2 and L_2 . Now each point is incident to at most E^2 lines of L_2 (Lemma 5), so the number of remaining incidences involving points of P_2 is $O(mE^2)$; for E suitably small, this bound will be subsumed by $O(m^{1/2}n^{3/4})$.

Unlike the case of a “large” D , namely, $D = m^{1/2}/n^{1/4}$, here the difficult part is to treat incidences within the cells of the partition. Since $E \ll D$, we cannot use the naive bound $O(n^2 + m)$ within each cell, because that would make the overall bound too large. Therefore, to control the incidence bound within the cells, we proceed in the following inductive manner.

For each cell τ of $\mathbb{R}^3 \setminus Z(g)$, put $P_\tau := P'_2 \cap \tau$, and let L_τ denote the set of the lines of L'_2 that cross τ ; put $m_\tau = |P_\tau| \leq m/r$, and $n_\tau = |L_\tau|$. Since every line $\ell \in L_1$ (that is, of L'_2) crosses at most $1 + E$ components of $\mathbb{R}^3 \setminus Z(g)$, we have $\sum_\tau n_\tau \leq n(1 + E)$.

It is important to note that at this point of the analysis the sizes of P_1 and of L_1 might be smaller than the original respective values m and n . In particular, we may no longer assume that $|P_1| > |L_1|^{\alpha_{j-1}}$, as we did assume for m and n . Nevertheless, in what follows m and n will denote the original values, which serve as upper bounds for the respective actual sizes of P_1 and L_1 , and the induction will work correctly with these values; see below for details.

In order to apply the induction hypothesis within the cells of the partition, we want to assume that $m_\tau \leq n_\tau^{\alpha_{j-1}}$ for each τ . To ensure that, we require that the number of lines of L'_2 that cross a cell be at most n/E^2 . Cells τ that are crossed by $\kappa n/E^2$ lines, for $\kappa > 1$, are treated as if they occur $\lceil \kappa \rceil$ times, where each incarnation involves all the points of P_τ , and at most n/E^2 lines of L_τ . The number of subproblems remains $O(E^3)$. Arguing similarly, we may also assume that $m_\tau \leq m/E^3$ for each cell τ (by “duplicating” each cell into a constant number of subproblems, if needed).

We therefore require that $\frac{m}{E^3} \leq \left(\frac{n}{E^2}\right)^{\alpha_{j-1}}$. (Note that, as already commented above, these are only upper bounds on the actual sizes of these subsets, but this will have no real effect on the induction process.) That is, we require

$$E \geq \left(\frac{m}{n^{\alpha_{j-1}}}\right)^{1/(3-2\alpha_{j-1})}. \tag{5}$$

With these preparations, we apply the induction hypothesis within each cell τ , recalling

that no plane contains more than D lines⁴ of $L'_2 \subseteq L_1$, and get

$$\begin{aligned} I(P_\tau, L_\tau) &\leq A_{j-1} \left(m_\tau^{1/2} n_\tau^{3/4} + m_\tau \right) + B \left(m_\tau^{2/3} n_\tau^{1/3} D^{1/3} + n_\tau \right) \\ &\leq A_{j-1} \left((m/E^3)^{1/2} (n/E^2)^{3/4} + m/E^3 \right) + B \left((m/E^3)^{2/3} (n/E^2)^{1/3} D^{1/3} + n/E^2 \right). \end{aligned}$$

Summing these bounds over the cells τ , that is, multiplying them by $O(E^3)$, we get, for a suitable absolute constant b ,

$$I(P'_2, L'_2) = \sum_\tau I(P_\tau, L_\tau) \leq bA_{j-1} \left(m^{1/2} n^{3/4} + m \right) + B \left(m^{2/3} n^{1/3} E^{1/3} D^{1/3} + nE \right).$$

We now require that $E = O(D)$. Then the last term satisfies $nE = O(nD) = O(m^{1/2} n^{3/4})$, and, as already remarked, the preceding term m is also subsumed by the first term. The second term, after substituting $D = O(m^{1/2}/n^{1/4})$, becomes $O(m^{5/6} n^{1/4} E^{1/3})$. Hence, with a slightly larger b , we have

$$I(P'_2, L'_2) \leq bA_{j-1} m^{1/2} n^{3/4} + bBm^{5/6} n^{1/4} E^{1/3}.$$

Adding up all the bounds, including those for the portions of P and L that were discarded during the first partitioning step, we obtain, for a suitable constant c ,

$$I(P, L) \leq c \left(m^{1/2} n^{3/4} + m^{2/3} n^{1/3} s^{1/3} + n + mE^2 \right) + bA_{j-1} m^{1/2} n^{3/4} + bBm^{5/6} n^{1/4} E^{1/3}.$$

We choose E to ensure that the two E -dependent terms are dominated by the term $m^{1/2} n^{3/4}$. That is,

$$\begin{aligned} m^{5/6} n^{1/4} E^{1/3} &\leq m^{1/2} n^{3/4}, \quad \text{or} \quad E \leq n^{3/2}/m, \\ \text{and} \quad mE^2 &\leq m^{1/2} n^{3/4}, \quad \text{or} \quad E \leq n^{3/8}/m^{1/4}. \end{aligned}$$

Since $n^{3/2}/m = (n^{3/8}/m^{1/4})^4$, and both sides are ≥ 1 , the latter condition is stricter, and we ignore the former. As already noted, we also require that $E = O(D)$; specifically, we require that $E \leq m^{1/2}/n^{1/4}$.

In conclusion, recalling (5), the two constraints on the choice of E are

$$\left(\frac{m}{n^{\alpha_{j-1}}} \right)^{1/(3-2\alpha_{j-1})} \leq E \leq \min \left\{ \frac{n^{3/8}}{m^{1/4}}, \frac{m^{1/2}}{n^{1/4}} \right\}, \tag{6}$$

and, for these constraints to be compatible, we require that $\left(\frac{m}{n^{\alpha_{j-1}}} \right)^{1/(3-2\alpha_{j-1})} \leq \frac{n^{3/8}}{m^{1/4}}$, or $m \leq n^{\frac{9+2\alpha_{j-1}}{2(7-2\alpha_{j-1})}}$, and that $\left(\frac{m}{n^{\alpha_{j-1}}} \right)^{1/(3-2\alpha_{j-1})} \leq \frac{m^{1/2}}{n^{1/4}}$, which fortunately always holds, as is easily checked, since $m \leq n^{3/2}$ and $\alpha_{j-1} \geq 1/2$. Note that we have not explicitly stated any concrete choice of E ; any value satisfying (6) will do. We put

$$\alpha_j := \frac{9 + 2\alpha_{j-1}}{2(7 - 2\alpha_{j-1})},$$

and conclude that if $m \leq n^{\alpha_j}$ then the bound asserted in the theorem holds, with $A_j = bA_{j-1} + c$ and $B = c$. This completes the induction step. Note that the recurrence $A_j = bA_{j-1} + c$ solves to $A_j = O(b^j)$.

⁴ This was the main reason for carrying out the first partitioning step, as already noted.

It remains to argue that the induction covers the entire range $m = O(n^{3/2})$. Using the above recurrence for the α_j 's, with $\alpha_0 = 1/2$, it easily follows that $\alpha_j = \frac{3}{2} - \frac{2}{j+2}$, for each $j \geq 0$, showing that α_j converges to $3/2$, implying that the entire range $m = O(n^{3/2})$ is covered by the induction.

To calibrate the dependence of the constant of proportionality on m and n , we note that, for $n^{\alpha_{j-1}} \leq m < n^{\alpha_j}$, the constant is $O(b^j)$. We have

$$\frac{3}{2} - \frac{2}{j+1} = \alpha_{j-1} \leq \frac{\log m}{\log n}, \quad \text{or} \quad j \leq \frac{\frac{1}{2} + \frac{\log m}{\log n}}{\frac{3}{2} - \frac{\log m}{\log n}} = \frac{\log(m^2 n)}{\log(n^3/m^2)}.$$

This establishes the expression for $A_{m,n}$ given in the statement of the theorem.

Handling the middle ground $m \approx n^{3/2}$. Some care is needed when m approaches $n^{3/2}$, because of the potentially unbounded growth of the constant A_j . We show, in the full version, that

$$I(P, L) = O\left(2^c \sqrt{\log n} \left(m^{1/2} n^{3/4} + m^{2/3} n^{1/3} s^{1/3} + m + n\right)\right), \quad (7)$$

for a suitable absolute constant c . In other words, the bound in (2) and (3) holds for any $m \leq n^{3/2}$, but, for $m \geq n^{\alpha_{j_0}}$ one should use instead the bound in (7), which controls the exponential growth of the constants of proportionality within this range.

The case $m > n^{3/2}$

The analysis of this case is, in a sense, a mirror image of the preceding analysis, except for a new key lemma (Lemma 8). Due to lack of space, most details are omitted, and can be found in the full version [20].

We partition this range into a sequence of ranges $m \geq n^{\alpha_0}$, $n^{\alpha_1} \leq m < n^{\alpha_0}$, \dots , where $\alpha_0 = 2$ and the sequence $\{\alpha_j\}_{j \geq 0}$ is decreasing and converges to $3/2$. The induction is on the index j of the range $n^{\alpha_j} \leq m < n^{\alpha_{j-1}}$, and establishes (2) for m in this range, with a coefficient A_j (written in (2,3) as $A_{m,n}$) that increases with j .

The base range of the induction is $m \geq n^2$, where we have the general bound $I = O(n^2 + m) = O(m)$, so (2) holds for a sufficiently large choice of the initial constant A_0 . Assume then that (2) holds for all $m \geq n^{\alpha_{j-1}}$ for some $j \geq 1$, and consider an instance of the problem with $n^{3/2} \leq m < n^{\alpha_{j-1}}$.

For a parameter r , to be specified later, apply the polynomial partition theorem to obtain an r -partitioning trivariate (real) polynomial f of degree $D = O(r^{1/3})$. That is, every connected component of $\mathbb{R}^3 \setminus Z(f)$ contains at most m/r points of P , and the number of components of $\mathbb{R}^3 \setminus Z(f)$ is $O(D^3) = O(r)$.

Set $P_1 := P \cap Z(f)$ and $P'_1 := P \setminus P_1$. Each line $\ell \in L$ is either fully contained in $Z(f)$ or intersects it in at most D points. Let L_1 denote the subset of lines of L that are fully contained in $Z(f)$ and put $L'_1 = L \setminus L_1$. As before, we have $I(P, L) = I(P_1, L_1) + I(P_1, L'_1) + I(P'_1, L'_1)$, and $I(P_1, L'_1) \leq |L'_1| \cdot D \leq nD$. We estimate $I(P'_1, L'_1)$ as in the preceding case, where, for the initial value of D , we take $D = n^2/m$, noting that $1 \leq D^3 \leq m$ because $n^{3/2} \leq m \leq n^2$, and get the bound

$$I(P'_1, L'_1) + I(P_1, L'_1) = O(n^2/D + m + nD) = O(m + n^3/m) = O(m),$$

where the latter bound follows since $m \geq n^{3/2}$.

To estimate $I(P_1, L_1)$, we discard all other lines and points, forget the preceding polynomial partitioning step, and start afresh, applying a new polynomial partitioning to P_1 with a polynomial g of degree E , which will typically be much smaller than D , but still non-constant.

For this case we need the following lemma, which can be regarded, in some sense, as a dual (albeit somewhat more involved) version of Lemma 5. Unlike the rest of the analysis, the best way to prove this lemma is by switching to the complex projective setting. This is needed for one key step in the proof, where we need the property that the projection of a complex projective variety is a variety. Once this is done, we can switch back to the real affine case, and complete the proof.

We say that a point $p \in P_1$ is *1-poor* (resp., *2-rich*) if it is incident to at most one line (resp., to at least two lines) of L_1 . We also recall that a *regulus* is a doubly-ruled surface in \mathbb{R}^3 or in \mathbb{C}^3 . It is the union of all lines that pass through three fixed pairwise skew lines; it is a quadric, which is either a hyperbolic paraboloid or a one-sheeted hyperboloid.

► **Lemma 8.** *Let f be an irreducible polynomial in $\mathbb{C}[x, y, z]$, such that $Z(f)$ is not a complex plane nor a complex regulus, and let L_1 be a finite set of lines fully contained in $Z(f)$. Then, with the possible exception of at most two lines, each line $\ell \in L_1$ is incident to at most $O(D^3)$ 2-rich points.*

Proof. The strategy of the proof is to charge each incidence of ℓ with some 2-rich point p to an intersection of ℓ with another line of L_1 that passes through p , and to argue that, in general, there can be only $O(D^3)$ such other lines. This in turn will be shown by arguing that the union of all the lines that are fully contained in $Z(f)$ and pass through ℓ is a one-dimensional variety, of degree $O(D^3)$, from which the claim will follow. As we will show, this will indeed be the case except when ℓ is one of at most two “exceptional” lines on $Z(f)$.

Fix a line ℓ as in the lemma, assume for simplicity that it passes through the origin, and write it as $\{tv_0 \mid t \in \mathbb{C}\}$; since ℓ is a real line, v_0 can be assumed to be real. Consider the union $V(\ell)$ of all the lines that are fully contained in $Z(f)$ and are incident to ℓ ; that is, $V(\ell)$ is the union of ℓ with the set of all points $p \in Z(f) \setminus \ell$ for which there exists $t \in \mathbb{C}$ such that the line connecting p to $tv_0 \in \ell$ is fully contained in $Z(f)$. In other words, for such a t and for each $s \in \mathbb{C}$, we have $f((1-s)p + stv_0) = 0$. Regarding the left-hand side as a polynomial in s , we can write it as $\sum_{i=0}^D G_i(p; t)s^i \equiv 0$, for suitable (complex) polynomials $G_i(p; t)$ in p and t , each of total degree at most D . In other words, p and t have to satisfy the system

$$G_0(p; t) = G_1(p; t) = \dots = G_D(p; t) = 0, \tag{8}$$

which defines an algebraic variety $\sigma(\ell)$ in $\mathbb{P}^4(\mathbb{C})$. Note that, substituting $s = 0$, we have $G_0(p; t) \equiv f(p)$, and that the limit points (tv_0, t) (corresponding to points on ℓ) also satisfy this system, since in this case $f((1-s)tv_0 + stv_0) = f(tv_0) = 0$ for all s .

In other words, $V(\ell)$ is the projection of $\sigma(\ell)$ into $\mathbb{P}^3(\mathbb{C})$, given by $(p, t) \mapsto p$. For each $p \in Z(f) \setminus \ell$ this system has only finitely many solutions in t , for otherwise the plane spanned by p and ℓ_0 would be fully contained in $Z(f)$, contrary to our assumption.

By the projective extension theorem (see, e.g., [3, Theorem 8.6]), the projection of $\sigma(\ell)$ into $\mathbb{P}^3(\mathbb{C})$, in which t is discarded, is an algebraic variety $\tau(\ell)$. We observe that $\tau(\ell)$ is contained in $Z(f)$, and is therefore of dimension at most two.

Assume first that $\tau(\ell)$ is two-dimensional. As f is irreducible over \mathbb{C} , we must have $\tau(\ell) = Z(f)$. This implies that each point $p \in Z(f) \setminus \ell$ is incident to a (complex) line that is fully contained in $Z(f)$ and is incident to ℓ . In particular, $Z(f)$ is ruled by complex lines.

By assumption, $Z(f)$ is neither a complex plane nor a complex regulus. We may also assume that $Z(f)$ is not a complex cone, for then each line in L_1 is incident to at most one 2-rich point (namely, the apex of $Z(f)$), making the assertion of the lemma trivial. It then follows that $Z(f)$ is an irreducible singly ruled (complex) surface. As argued in Guth and Katz [8] (see also our companion paper [21] for an independent analysis of this situation, which caters more explicitly to the complex setting too), $Z(f)$ can contain at most two lines ℓ with this property.

Excluding these (at most) two exceptional lines ℓ , we may thus assume that $\tau(\ell)$ is (at most) a one-dimensional curve.

Clearly, by definition, each point $(p, t) \in \sigma(\ell)$, except for $p \in \ell$, defines a line λ , in the original 3-space, that connects p to tv_0 , and each point $q \in \lambda$ satisfies $(q, t) \in \sigma(\ell)$. Hence, the line $\{(q, t) \mid q \in \lambda\}$ is fully contained in $\sigma(\ell)$, and therefore the line λ is fully contained in $\tau(\ell)$. Since $\tau(\ell)$ is one-dimensional, this in turn implies (see, e.g., [19, Lemma 2.3]) that $\tau(\ell)$ is a *finite* union of (complex) lines, whose number is at most $\deg(\tau(\ell))$. This also implies that $\sigma(\ell)$ is the union of the same number of lines, and in particular $\sigma(\ell)$ is also one-dimensional, and the number of lines that it contains is at most $\deg(\sigma(\ell))$.

We claim that this latter degree is at most $O(D^3)$. This follows from a well-known result in algebra (see, e.g., Schmid [15, Lemma 2.2]), that asserts that, since $\sigma(\ell)$ is a one-dimensional curve in $\mathbb{P}^4(\mathbb{C})$, and is the common zero set of polynomials, each of degree $O(D)$, its degree is $O(D^3)$.

This completes the proof of the lemma. (The passage from the complex projective setting back to the real affine one is trivial for this property.) \blacktriangleleft

► Corollary 9. *Let f be a real or complex trivariate polynomial of degree D , such that (the complexification of) $Z(f)$ does not contain any complex plane nor any complex regulus. Let L_1 be a set of n lines fully contained in $Z(f)$, and let P_1 be a set of m points contained in $Z(f)$. Then $I(P_1, L_1) = O(m + nD^3)$.*

Proof. Write $f = \prod_{i=1}^s f_i$ for its decomposition into irreducible factors, for $s \leq D$. We apply Lemma 8 to each complex factor f_i of the f . By the observation preceding Lemma 8, some of these factors might be complex (non-real) polynomials, even when f is real. That is, regardless of whether the original f is real or not, we carry out the analysis in the complex projective space $\mathbb{P}^3(\mathbb{C})$, and regard $Z(f_i)$ as a variety in that space.

Note also that, by focussing on the single irreducible component $Z(f_i)$ of $Z(f)$, we consider only points and lines that are fully contained in $Z(f_i)$. We thus shrink P_1 and L_1 accordingly, and note that the notions of being 2-rich or 1-poor are now redefined with respect to the reduced sets. All of this will be rectified at the end of the proof.

Assign each line $\ell \in L_1$ to the first component $Z(f_i)$, in the above order, that fully contains ℓ , and assign each point $p \in P_1$ to the first component that contains it. If a point p and a line ℓ are incident, then either they are both assigned to the same component $Z(f_i)$, or p is assigned to some component $Z(f_i)$ and ℓ , which is assigned to a later component, is not contained in $Z(f_i)$. Each incidence of the latter kind can be charged to a crossing between ℓ and $Z(f_i)$, and the total number of these crossings is $O(nD)$. It therefore suffices to consider incidences between points and lines assigned to the same component. Moreover, if a point p is 2-rich with respect to the entire collection L_1 but is 1-poor with respect to the lines assigned to its component, then all of its incidences except one are accounted by the preceding term $O(nD)$, which thus takes care also of the single incidence within $Z(f_i)$.

By Lemma 8, for each f_i , excluding at most two exceptional lines, the number of incidences between a line assigned to (and contained in) $Z(f_i)$ and the points assigned to $Z(f_i)$ that

are still 2-rich within $Z(f_i)$, is $O(\deg(f_i)^3) = O(D^3)$. Summing over all relevant lines, we get the bound $O(nD^3)$.

Finally, each irreducible component $Z(f_i)$ can contain at most two exceptional lines, for a total of at most $2D$ such lines. The number of 2-rich points on each such line ℓ is at most n , since each such point is incident to another line, so the total number of corresponding incidences is at most $O(nD)$, which is subsumed by the preceding bound $O(nD^3)$. The number of incidences with 1-poor points is, trivially, at most m . This completes the proof of the corollary. \square

We next bound the number of incidences between points and lines on planar and reguli components of $Z(f)$, discard the relevant points and lines, and note that no plane contains more than $O(D)$ of the surviving lines, as argued in Lemma 3.

We then construct a new partitioning polynomial g , of degree E much smaller than D , and rerun the analysis for g and E , as in the case of small m , where we use induction to bound the number of incidences within the partition cells. The reasoning is similar, but the calculations are different due to the different range of m . Omitting further details (for which see [20]), we show that the induction step carries out if we choose

$$\alpha_j = \frac{3}{2} + \frac{1}{4j-2},$$

for $j \geq 3$ (the treatment of the first two values of α_j is different for certain technical reasons). This sequence does indeed converge to $3/2$ as $j \rightarrow \infty$, implying that the entire range $m = \Omega(n^{3/2})$ is covered by the induction. \blacktriangleleft

3 Discussion

In this paper we derived an asymptotically tight bound for the number of incidences between a set P of points and a set L of lines in \mathbb{R}^3 . This bound has already been established by Guth and Katz [8], where the main tool was the use of partitioning polynomials. As already mentioned, the main novelty here is to use two separate partitioning polynomials of different degrees; the one with the higher degree is used as a pruning mechanism, after which the maximum number of coplanar lines of L can be better controlled (by the degree D of the polynomial), which is a key ingredient in making the inductive argument work.

The second main tool of Guth and Katz was the Cayley–Salmon theorem. This theorem says that a surface in \mathbb{R}^3 of degree D cannot contain more than $11D^2 - 24D$ lines, unless it is *ruled by lines*. This is an “ancient” theorem, from the 19th century, combining algebraic and differential geometry, and its re-emergence in recent years has kindled the interest of the combinatorial geometry community in classical (and modern) algebraic geometry. New proofs of the theorem were obtained (see, e.g., Terry Tao’s blog [26]), and generalizations to higher dimensions have also been developed (see Landsberg [12]). However, the theorem only holds over the complex field, and using it over the reals requires some care.

There is also an alternative way to bound the number of point-line incidences using flat and singular points. However, as already remarked, these two, as well as the Cayley–Salmon machinery, are non-trivial constructs, especially in higher dimensions, and their generalization to other problems in combinatorial geometry (even incidence problems with curves other than lines or incidences with lines in higher dimensions) seem quite difficult (and are mostly open). It is therefore of considerable interest to develop alternative, more elementary interfaces between algebraic and combinatorial geometry, which is a primary goal of the present paper (as well as of Guth’s recent work [6]).

In this regard, one could perhaps view Lemma 5 and Corollary 9 as certain weaker analogs of the Cayley–Salmon theorem, which are nevertheless easier to derive, without having to use differential geometry. Some of the tools in Guth’s paper [6] might also be interpreted as such weaker variants of the Cayley–Salmon theory. It would be interesting to see suitable extensions of these tools to higher dimensions.

Besides the intrinsic interest in simplifying the Guth–Katz analysis, the present work has been motivated by our study of incidences between points and lines in four dimensions. This has begun in a year-old companion paper [18], where we have used the polynomial partitioning method, with a polynomial of constant degree. This, similarly to Guth’s work in three dimensions [6], has resulted in a slightly weaker bound and considerably stricter assumptions concerning the input set of lines. In a more involved follow-up study [19], we have managed to improve the bound, and to get rid of the restrictive assumptions, using two partitioning steps, with polynomials of non-constant degrees, as in the present paper. However, the analysis in [19] is not as simple as in the present paper, because, even though there are generalizations of the Cayley–Salmon theorem to higher dimensions (due to Landsberg, as mentioned above), it turns out that a thorough investigation of the variety of lines fully contained in a given hypersurface of non-constant degree, is a fairly intricate and challenging problem, raising many deep questions in algebraic geometry, some of which are still unresolved.

One potential application of the techniques used in this paper, mainly the interplay between partitioning polynomials of different degrees, is to the problem, recently studied by Sharir, Sheffer and Zahl [17], of bounding the number of incidences between points and circles in \mathbb{R}^3 . That paper uses a partitioning polynomial of constant degree, and, as a result, the term that caters to incidences within lower-dimensional spaces (such as our term $m^{2/3}n^{1/3}s^{1/3}$) does not go well through the induction mechanism, and consequently the bound derived in [17] was weaker. We believe that our technique can improve the bound of [17] in terms of this “lower-dimensional” term.

A substantial part of the present paper (half of the proof of the theorem) was devoted to the treatment of the case $m > n^{3/2}$. However, under the appropriate assumptions, the number of points incident to at least two lines was shown by Guth and Katz [8] to be bounded by $O(n^{3/2})$. A recent note by Kollár [11] gives a simplified proof, including an explicit multiplicative constant. In his work, Kollár does not use partitioning polynomials, but employs more advanced algebraic geometric tools, like the *arithmetic genus* of a curve, which serves as an upper bound for the number of singular points. If we accept (pedagogically) the upper bound $O(n^{3/2})$ for the number of 2-rich points as a “black box”, the regime in which $m > n^{3/2}$ becomes irrelevant, and can be discarded from the analysis, thus greatly simplifying the paper.

A challenging problem is thus to find an elementary proof that the number of points incident to at least two lines is $O(n^{3/2})$ (e.g., without the use of the Cayley–Salmon theorem or the tools used by Kollár). Another challenging (and probably harder) problem is to improve the bound of Guth and Katz when the bound s on the maximum number of mutually coplanar lines is $\ll n^{1/2}$: In their original derivation, Guth and Katz [8] consider mainly the case $s = n^{1/2}$, and the lower bound construction in [8] also has $s = n^{1/2}$. Another natural further research direction is to find further applications of partitioning polynomials of intermediate degrees.

References

- 1 S. Basu and M. Sombra, Polynomial partitioning on varieties and point-hypersurface incidences in four dimensions, in arXiv:1406.2144.
- 2 K. Clarkson, H. Edelsbrunner, L. Guibas, M. Sharir and E. Welzl, Combinatorial complexity bounds for arrangements of curves and spheres, *Discrete Comput. Geom.* 5 (1990), 99–160.
- 3 D. Cox, J. Little and D. O’Shea, *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*, Springer Verlag, Heidelberg, 2007.
- 4 G. Elekes, H. Kaplan and M. Sharir, On lines, joints, and incidences in three dimensions, *J. Combinat. Theory, Ser. A* 118 (2011), 962–977. Also in arXiv:0905.1583.
- 5 P. Erdős, On sets of distances of n points, *Amer. Math. Monthly* 53 (1946), 248–250.
- 6 L. Guth, Distinct distance estimates and low-degree polynomial partitioning, in arXiv:1404.2321.
- 7 L. Guth and N. H. Katz, Algebraic methods in discrete analogs of the Kakeya problem, *Advances Math.* 225 (2010), 2828–2839. Also in arXiv:0812.1043v1.
- 8 L. Guth and N. H. Katz, On the Erdős distinct distances problem in the plane, *Annals Math.* 181 (2015), 155–190. Also in arXiv:1011.4105.
- 9 H. Kaplan, J. Matoušek, Z. Safernová and M. Sharir, Unit distances in three dimensions, *Combinat. Probab. Comput.* 21 (2012), 597–610. Also in arXiv:1107.1077.
- 10 H. Kaplan, J. Matoušek and M. Sharir, Simple proofs of classical theorems in discrete geometry via the Guth–Katz polynomial partitioning technique, *Discrete Comput. Geom.* 48 (2012), 499–517. Also in arXiv:1102.5391.
- 11 J. Kollár, Szemerédi–Trotter-type theorems in dimension 3, in arXiv:1405.2243.
- 12 J. M. Landsberg, is a linear space contained in a submanifold? On the number of derivatives needed to tell, *J. Reine Angew. Math.* 508 (1999), 53–60.
- 13 J. Pach and M. Sharir, Geometric incidences, in *Towards a Theory of Geometric Graphs* (J. Pach, ed.), *Contemporary Mathematics*, Vol. 342, Amer. Math. Soc., Providence, RI, 2004, pp. 185–223.
- 14 O. Raz, M. Sharir, and F. De Zeeuw, Polynomials vanishing on Cartesian products: The Elekes–Szabó Theorem revisited, These proceedings.
- 15 J. Schmid, On the affine Bézout inequality, *Manuscripta Mathematica* 88(1) (1995), 225–232.
- 16 M. Sharir, A. Sheffer, and N. Solomon, Incidences with curves in \mathbb{R}^d , manuscript, 2014.
- 17 M. Sharir, A. Sheffer, and J. Zahl, Improved bounds for incidences between points and circles, *Combinat. Probab. Comput.*, in press. Also in *Proc. 29th ACM Symp. on Computational Geometry* (2013), 97–106, and in arXiv:1208.0053.
- 18 M. Sharir and N. Solomon, Incidences between points and lines in \mathbb{R}^4 , *Proc. 30th Annu. ACM Sympos. Comput. Geom.*, 2014, 189–197.
- 19 M. Sharir and N. Solomon, Incidences between points and lines in four dimensions, in arXiv:1411.0777.
- 20 M. Sharir and N. Solomon, Incidences between points and lines in three dimensions, in arXiv:1501.02544.
- 21 M. Sharir and N. Solomon, Incidences between points and lines on a two-dimensional variety, manuscript, 2014. in arXiv:1501.01670.
- 22 J. Solymosi and T. Tao, An incidence theorem in higher dimensions, *Discrete Comput. Geom.* 48 (2012), 255–280.
- 23 L. Székely, Crossing numbers and hard Erdős problems in discrete geometry, *Combinat. Probab. Comput.* 6 (1997), 353–358.
- 24 E. Szemerédi and W. T. Trotter, Extremal problems in discrete geometry, *Combinatorica* 3 (1983), 381–392.

- 25 T. Tao, From rotating needles to stability of waves: Emerging connections between combinatorics, analysis, and PDE, *Notices AMS* 48(3) (2001), 294–303.
- 26 T. Tao, The Cayley–Salmon theorem via classical differential geometry, <http://terrytao.wordpress.com>, March 2014.
- 27 H. E. Warren, Lower bound for approximation by nonlinear manifolds, *Trans. Amer. Math. Soc.* 133 (1968), 167–178.
- 28 J. Zahl, An improved bound on the number of point-surface incidences in three dimensions, *Contrib. Discrete Math.* 8(1) (2013). Also in arXiv:1104.4987.
- 29 J. Zahl, A Szemerédi-Trotter type theorem in \mathbb{R}^4 , in arXiv:1203.4600.

The Number of Unit-Area Triangles in the Plane: Theme and Variations*

Orit E. Raz and Micha Sharir

School of Computer Science, Tel Aviv University
Tel Aviv 69978, Israel
{oritraz,michas}@post.tau.ac.il

Abstract

We show that the number of unit-area triangles determined by a set S of n points in the plane is $O(n^{20/9})$, improving the earlier bound $O(n^{9/4})$ of Apfelbaum and Sharir [2]. We also consider two special cases of this problem: (i) We show, using a somewhat subtle construction, that if S consists of points on three lines, the number of unit-area triangles that S spans can be $\Omega(n^2)$, for any triple of lines (it is always $O(n^2)$ in this case). (ii) We show that if S is a *convex grid* of the form $A \times B$, where A, B are *convex* sets of $n^{1/2}$ real numbers each (i.e., the sequences of differences of consecutive elements of A and of B are both strictly increasing), then S determines $O(n^{31/14})$ unit-area triangles.

1998 ACM Subject Classification G.2 Discrete Mathematics

Keywords and phrases Combinatorial geometry, incidences, repeated configurations

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.569

1 Introduction

In 1967, Oppenheim (see [9]) asked the following question: Given n points in the plane and $A > 0$, how many triangles spanned by the points can have area A ? By applying a scaling transformation, one may assume $A = 1$ and count the triangles of *unit* area. Erdős and Purdy [8] showed that a $\sqrt{\log n} \times (n/\sqrt{\log n})$ section of the integer lattice determines $\Omega(n^2 \log \log n)$ triangles of the same area. They also showed that the maximum number of such triangles is at most $O(n^{5/2})$. In 1992, Pach and Sharir [10] improved the bound to $O(n^{7/3})$, using the Szemerédi-Trotter theorem [16] (see below) on the number of point-line incidences. More recently, Dumitrescu et al. [4] have further improved the upper bound to $O(n^{44/19}) = O(n^{2.3158})$, by estimating the number of incidences between the given points and a 4-parameter family of quadratic curves. In a subsequent improvement, Apfelbaum and Sharir [2] have obtained the upper bound $O(n^{9/4+\epsilon})$, for any $\epsilon > 0$, which has been slightly improved to $O(n^{9/4})$ in Apfelbaum [1]. This has been the best known upper bound so far.

In this paper we further improve the bound to $O(n^{20/9})$. Our proof uses a different reduction of the problem to an incidence problem, this time to incidences between points and two-dimensional algebraic surfaces in \mathbb{R}^4 . A very recent result of Solymosi and De Zeeuw [15] provides a sharp upper bound for the number of such incidences, similar to the

* Work on this paper by Orit E. Raz and Micha Sharir was supported by Grant 892/13 from the Israel Science Foundation and by the Israeli Centers of Research Excellence (I-CORE) program (Center No. 4/11). Work by Micha Sharir was also supported by Grant 2012/229 from the U.S.–Israel Binational Science Foundation and by the Hermann Minkowski-MINERVA Center for Geometry at Tel Aviv University. Part of this research was performed while the authors were visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation.



Szemerédi–Trotter bound, provided that the points, surfaces, and incidences satisfy certain fairly restrictive assumptions. The main novel features of our analysis are thus (a) the reduction of the problem to this specific type of incidence counting, and (b) showing that the assumptions of [15] are satisfied in our context.

After establishing this main result, we consider two variations, in which better bounds can be obtained.

We first consider the case where the input points lie on three arbitrary lines. It is easily checked that in this case there are at most $O(n^2)$ unit-area triangles. We show, in Section 3, that this bound is tight, and can be attained for any triple of lines. Rather than just presenting the construction, we spend some time showing its connection to a more general problem studied by Elekes and Rónyai [6] (see also the recent developments in [7, 11, 12]), involving the zero set of a trivariate polynomial within a triple Cartesian product. Skipping over the details, which are spelled out in Section 3, it turns out that the case of unit-area triangles determined by points lying on three lines is an exceptional case in the theory of Elekes and Rónyai [6], which then leads to a construction with $\Theta(n^2)$ unit-area triangles.

Another variation that we consider concerns unit-area triangles spanned by points in a *convex grid*. That is, the input set is of the form $A \times B$, where A and B are convex sets of $n^{1/2}$ real numbers each; a set of real numbers is called *convex* if the differences between consecutive elements form a strictly increasing sequence. We show that in this case $A \times B$ determine $O(n^{31/14})$ unit-area triangles. The main technical tool used in our analysis is a result of Schoen and Shkredov [13] on difference sets involving convex sets.¹

2 Unit-area triangles in the plane

► **Theorem 2.1.** *The number of unit-area triangles spanned by n points in the plane is $O(n^{20/9})$.*

We first recall the Szemerédi–Trotter theorem [16] on point-line incidences in the plane.

► **Theorem 2.2** (Szemerédi and Trotter [16]).

- (i) *The number of incidences between M distinct points and N distinct lines in the plane is $O(M^{2/3}N^{2/3} + M + N)$.*
- (ii) *Given M distinct points in the plane and a parameter $k \leq M$, the number of lines incident to at least k of the points is $O(M^2/k^3 + M/k)$. Both bounds are tight in the worst case.*

Proof of Theorem 2.1. Let S be a set of n points in the plane, and let U denote the set of unit-area triangles spanned by S . For any pair of distinct points, $p \neq q \in S$, let ℓ_{pq} denote the line through p and q . The points r for which the triangle pqr has unit area lie on two lines ℓ_{pq}^-, ℓ_{pq}^+ parallel to ℓ_{pq} and at distance $2/|pq|$ from ℓ_{pq} on either side. We let $\ell'_{pq} \in \{\ell_{pq}^-, \ell_{pq}^+\}$ be the line that lies to the left of the vector \vec{pq} . We then have²

$$|U| = \frac{1}{3} \sum_{(p,q) \in S \times S} |\ell'_{pq} \cap S|.$$

It suffices to consider only triangles pqr of U , that have the property that at least one of the three lines $\ell_{pq}, \ell_{pr}, \ell_{qr}$ is incident to at most $n^{1/2}$ points of S , because the number of

¹ Very recently, in work in progress, jointly with I. Shkredov, the bound is further improved in this case.

² In this sum, as well in similar sums in the sequel, we only consider pairs of distinct points in $S \times S$.

triangles in U that do not have this property is $O(n^{3/2})$. Indeed, by Theorem 2.2(ii), there exist at most $O(n^{1/2})$ lines in \mathbb{R}^2 , such that each contains at least $n^{1/2}$ points of S . Since every triple of those lines supports (the edges of) at most one triangle (some of the lines might be mutually parallel, and some triples might intersect at points that do not belong to S), these lines support in total at most $O(n^{3/2})$ triangles, and, in particular, at most $O(n^{3/2})$ triangles of U . Since this number is subsumed in the asserted bound on $|U|$, we can therefore ignore such triangles in our analysis. In what follows, U denotes the set of the remaining unit-area triangles.

We charge each of the surviving unit-area triangles pqr to one of its sides, say pq , such that ℓ_{pq} contains at most $n^{1/2}$ points of S . That is, we have $|U| \leq \sum_{(p,q) \in (S \times S)^*} |\ell'_{pq} \cap S|$,

where $(S \times S)^*$ denotes the subset of pairs $(p, q) \in S \times S$, such that $p \neq q$, and the line ℓ_{pq} is incident to at most $n^{1/2}$ points of S .

A major problem in estimating $|U|$ is that the lines ℓ'_{pq} , for $p, q \in S$, are not necessarily distinct, and the analysis has to take into account the (possibly large) multiplicity of these lines. (If the lines were distinct then $|U|$ would be bounded by the number of incidences between $n(n-1)$ lines and n points, which is $O(n^2)$ — see Theorem 2.2(i).) Let L denote the collection of lines $\{\ell'_{pq} \mid (p, q) \in (S \times S)^*\}$ (without multiplicity). For $\ell \in L$, we define $(S \times S)_\ell$ to be the set of all pairs $(p, q) \in (S \times S)^*$, for which $\ell'_{pq} = \ell$. We then have

$$|U| \leq \sum_{\ell \in L} |\ell \cap S| |(S \times S)_\ell|.$$

Fix some integer parameter $k \leq n^{1/2}$, to be set later, and partition L into the sets

$$L^- = \{\ell \in L \mid |\ell \cap S| < k\}, \quad L^+ = \{\ell \in L \mid k \leq |\ell \cap S| \leq n/k\}, \quad L^{++} = \{\ell \in L \mid |\ell \cap S| > n/k\}.$$

We have

$$|U| \leq \sum_{\ell \in L^-} |\ell \cap S| |(S \times S)_\ell| + \sum_{\ell \in L^+} |\ell \cap S| |(S \times S)_\ell| + \sum_{\ell \in L^{++}} |\ell \cap S| |(S \times S)_\ell|.$$

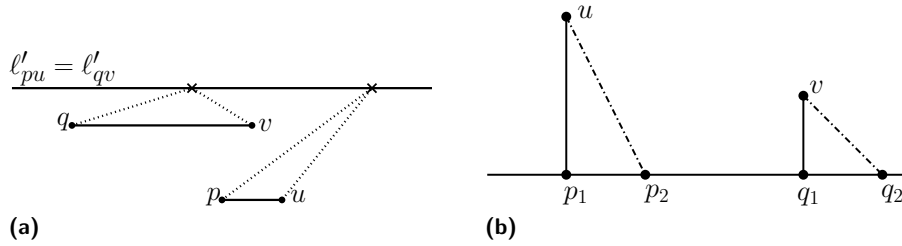
The first sum is at most $k \sum_{\ell \in L^-} |(S \times S)_\ell| \leq kn^2$, because $\sum_{\ell \in L^-} |(S \times S)_\ell|$ is at most $|(S \times S)^*| \leq |S \times S| = n^2$. The same (asymptotic) bound also holds for the the third sum. Indeed, since $n/k \geq n^{1/2}$, the number of lines in L^{++} is at most $O(k)$, as follows from Theorem 2.2(ii), and, for each $\ell \in L^{++}$, we have $|\ell \cap S| \leq n$ and $|(S \times S)_\ell| \leq n$ (for any $p \in S$, $\ell \in L$, there exists at most one point $q \in S$, such that $\ell'_{pq} = \ell$). This yields a total of at most $O(n^2k)$ unit-area triangles. It therefore remains to bound the second sum, over L^+ .

Applying the Cauchy-Schwarz inequality to the second sum, it follows that

$$|U| \leq O(n^2k) + \left(\sum_{\ell \in L^+} |\ell \cap S|^2 \right)^{1/2} \left(\sum_{\ell \in L^+} |(S \times S)_\ell|^2 \right)^{1/2}.$$

Let N_j (resp., $N_{\geq j}$), for $k \leq j \leq n/k$, denote the number of lines $\ell \in L^+$ for which $|\ell \cap S| = j$ (resp., $|\ell \cap S| \geq j$). By Theorem 2.2(ii), $N_{\geq j} = O(n^2/j^3 + n/j)$. Hence

$$\begin{aligned} \sum_{\ell \in L^+} |\ell \cap S|^2 &= \sum_{j=k}^{n/k} j^2 N_j \leq k^2 N_{\geq k} + \sum_{j=k+1}^{n/k} (2j-1) N_{\geq j} \\ &= O\left(\frac{n^2}{k} + nk + \sum_{j=k+1}^{n/k} \left(\frac{n^2}{j^2} + n\right)\right) = O\left(\frac{n^2}{k}\right) \end{aligned}$$



■ **Figure 1** (a) A quadruple (p, u, q, v) in Q . (b) If p_1, q_1, p_2, q_2 are collinear and $|p_1 p_2| = |q_1 q_2|$ then $\ell_{p_2 u}, \ell_{q_2 v}$ are not parallel to one another, for every $(u, v) \in \sigma_{p_1 q_1} \setminus \ell_{p_1 q_1}$. Thus, in particular, $(u, v) \notin \sigma_{p_2 q_2}$.

(where we used the fact that $k \leq n^{1/2}$). It follows that

$$|U| = O\left(n^2 k + \frac{n}{k^{1/2}} \left(\sum_{\ell \in L^+} |(S \times S)_\ell|^2\right)^{1/2}\right).$$

To estimate the remaining sum, put

$$Q := \{(p, u, q, v) \in S^4 \mid (p, u), (q, v) \in (S \times S)_\ell, \text{ for some } \ell \in L^+\}.$$

That is, Q consists of all quadruples (p, u, q, v) such that $\ell'_{pu} = \ell'_{qv} \in L^+$, and each of ℓ_{pu}, ℓ_{qv} contains at most $n^{1/2}$ points of S . See Figure 1(a) for an illustration.

The above bound on $|U|$ can then be written as

$$|U| = O\left(n^2 k + \frac{n|Q|^{1/2}}{k^{1/2}}\right). \tag{1}$$

The main step of the analysis is to establish the following upper bound on $|Q|$.

► **Proposition 2.3.** *Let Q be as above. Then $|Q| = O(n^{8/3})$.*

The proposition, combined with (1), implies that $|U| = O(n^2 k + n^{7/3}/k^{1/2})$, which, if we choose $k = n^{2/9}$, becomes $|U| = O(n^{20/9})$. Since the number of triangles that we have discarded is only $O(n^{3/2})$, Theorem 2.1 follows. ◀

Proof of Proposition 2.3. Consider first quadruples $(p, u, q, v) \in Q$, with all four points p, u, q, v collinear. As is easily checked, in this case (p, u, q, v) must also satisfy $|pu| = |qv|$. It follows that a line ℓ in the plane, which is incident to at most j points of S , can support at most j^3 such quadruples. By definition, $(S \times S)_\ell \subset (S \times S)^*$ for each $\ell \in L^+$, so the line $\ell_{pu} = \ell_{qv}$ is incident to at most $n^{1/2}$ points of S , and it suffices to consider only lines ℓ with this property. Using the preceding notations $N_j, N_{\geq j}$, the number of quadruples under consideration is

$$O\left(\sum_{j \leq n^{1/2}} j^3 N_j\right) = O\left(\sum_{j \leq n^{1/2}} j^2 N_{\geq j}\right) = O\left(\sum_{j \leq n^{1/2}} j^2 \cdot \frac{n^2}{j^3}\right) = O(n^2 \log n).$$

This is subsumed by the asserted bound on $|Q|$, so, in what follows we only consider quadruples $(p, u, q, v) \in Q$, such that p, u, q, v are not collinear.

For convenience, we assume that no pair of points of S share the same x - or y -coordinate; this can always be enforced by a suitable rotation of the coordinate frame. The property that two pairs of $S \times S$ are associated with a common line of L can then be expressed in the following algebraic manner.

► **Lemma 2.4.** Let $(p, u, q, v) \in S^4$, and represent $p = (a, b), u = (x, y), q = (c, d)$, and $v = (z, w)$, by their coordinates in \mathbb{R}^2 . Then $\ell'_{pu} = \ell'_{qv}$ if and only if

$$\frac{y - b}{x - a} = \frac{w - d}{z - c} \quad \text{and} \quad \frac{bx - ay + 2}{x - a} = \frac{dz - cw + 2}{z - c}. \tag{2}$$

Proof. Let $\alpha, \beta \in \mathbb{R}$ be such that $\ell'_{(a,b)(x,y)} = \{(t, \alpha t + \beta) \mid t \in \mathbb{R}\}$. Then, by the definition of $\ell'_{(a,b)(x,y)}$, we have

$$\frac{1}{2} \begin{vmatrix} a & x & t \\ b & y & \alpha t + \beta \\ 1 & 1 & 1 \end{vmatrix} = 1, \quad \text{or} \quad (b - y - \alpha(a - x))t - \beta(a - x) + ay - bx = 2,$$

for all $t \in \mathbb{R}$. Thus, $\alpha = \alpha(a, b, x, y) = \frac{y-b}{x-a}$, $\beta = \beta(a, b, x, y) = \frac{bx-ay+2}{x-a}$. Then the constraint $\ell'_{(a,b)(x,y)} \equiv \ell'_{(c,d)(z,w)}$ can be written as $\alpha(a, b, x, y) = \alpha(c, d, z, w)$, $\beta(a, b, x, y) = \beta(c, d, z, w)$, which is (2). ◀

We next transform the problem of estimating $|Q|$ into an incidence problem. With each pair $(p = (a, b), q = (c, d)) \in S \times S$, we associate the two-dimensional surface $\sigma_{pq} \subset \mathbb{R}^4$ which is the locus of all points $(x, y, z, w) \in \mathbb{R}^4$ that satisfy the system (2). The degree of σ_{pq} is at most 4, being the intersection of two quadratic hypersurfaces. We let Σ denote the set of surfaces

$$\Sigma := \{\sigma_{pq} \mid (p, q) \in S \times S, p \neq q\}.$$

For $(p_1, q_1) \neq (p_2, q_2)$, the corresponding surfaces $\sigma_{p_1q_1}, \sigma_{p_2q_2}$ are distinct; the proof of this fact is omitted here. We also consider the set $\Pi := S \times S$, regarded as a point set in \mathbb{R}^4 (identifying $\mathbb{R}^2 \times \mathbb{R}^2 \simeq \mathbb{R}^4$). We have $|\Pi| = |\Sigma| = O(n^2)$. The set $I(\Pi, \Sigma)$, the set of incidences between Π and Σ , is naturally defined as

$$I(\Pi, \Sigma) := \{(\pi, \sigma) \in \Pi \times \Sigma \mid \pi \in \sigma\}.$$

By Lemma 2.4, we have $(x, y, z, w) \in \sigma_{pq}$ if and only if $\ell'_{pu} = \ell'_{qv}$, where $u := (x, y)$ and $v := (z, w)$. This implies that $|Q| \leq |I(\Pi, \Sigma)|$.

Consider the subcollection \mathcal{I} of incidences $((x, y, z, w), \sigma_{pq}) \in I(\Pi, \Sigma)$, such that $p, q, u := (x, y), v := (z, w)$ are non-collinear (as points in \mathbb{R}^2). As already argued, the number of collinear quadruples in Q is $O(n^2 \log n)$, and hence $|Q| \leq |\mathcal{I}| + O(n^2 \log n)$. So to bound $|Q|$ it suffices to obtain an upper bound on $|\mathcal{I}|$.

For this we use the following recent result of Solymosi and De Zeeuw [15] (see also the related results in [14, 17]). To state it we need the following definition.

► **Definition 2.5.** A two-dimensional constant-degree surface σ in \mathbb{R}^4 is said to be *slanted* (the original term used in [15] is *good*), if, for every $p \in \mathbb{R}^2$, $\rho_i^{-1}(p) \cap \sigma$ is finite, for $i = 1, 2$, where ρ_1 and ρ_2 are the projections of \mathbb{R}^4 onto its first and last two coordinates, respectively.

► **Theorem 2.6** (Solymosi and De Zeeuw [15]). Let S be a subset of \mathbb{R}^2 , and let Γ be a finite set of two-dimensional constant-degree slanted surfaces. Set $\Pi := S \times S$, and let $\mathcal{I} \subset I(\Pi, \Gamma)$. Assume that for every pair of distinct points $\pi_1, \pi_2 \in \Pi$ there are at most $O(1)$ surfaces $\sigma \in \Sigma$ such that both pairs $(\pi_1, \sigma), (\pi_2, \sigma)$ are in \mathcal{I} . Then

$$|\mathcal{I}| = O\left(|\Pi|^{2/3}|\Sigma|^{2/3} + |\Pi| + |\Sigma|\right).$$

To apply Theorem 2.6, we need the following key technical proposition, whose proof is given in the next subsection.

► **Proposition 2.7.** *Let Π , Σ , and \mathcal{I} be the sets that arise in our setting, as specified above. Then, (a) the surfaces of Σ are all slanted, and (b) for every pair of distinct points $\pi_1, \pi_2 \in \Pi$, there are at most three surfaces $\sigma \in \Sigma$ such that both pairs $(\pi_1, \sigma), (\pi_2, \sigma)$ are in \mathcal{I} .*

We have $|\Pi|, |\Sigma| = O(n^2)$. Therefore, Theorem 2.6 implies that $|\mathcal{I}| = O(n^{8/3})$, which completes the proof of Proposition 2.3 (and, consequently, of Theorem 2.1). ◀

2.1 Proof of Proposition 2.7

We start by eliminating z and w from (2). An easy calculation shows that

$$\begin{aligned} z &= \frac{2(x-a)}{(b-d)(x-a) + (c-a)(y-b) + 2} + c, \\ w &= \frac{2(y-b)}{(b-d)(x-a) + (c-a)(y-b) + 2} + d. \end{aligned} \quad (3)$$

This expresses σ_{pq} as the graph of a linear rational function from \mathbb{R}^2 to \mathbb{R}^2 (which is undefined on the line at which the denominator vanishes). Passing to homogeneous coordinates, replacing (x, y) by (x_0, x_1, x_2) and (z, w) by (z_0, z_1, z_2) , we can re-interpret σ_{pq} as the graph of a projective transformation $T_{pq} : \mathbb{RP}^2 \rightarrow \mathbb{RP}^2$, given by

$$\begin{pmatrix} z_0 \\ z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} ad - bc + 2 & b - d & c - a \\ c(ad - bc) + 2(c - a) & c(b - d) + 2 & c(c - a) \\ d(ad - bc) + 2(d - b) & d(b - d) & d(c - a) + 2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix}.$$

The representation (3) implies that every (x, y) defines at most one pair (z, w) such that $(x, y, z, w) \in \sigma_{pq}$. By the symmetry of the definition of σ_{pq} , every pair (z, w) also determines at most one pair (x, y) such that $(x, y, z, w) \in \sigma_{pq}$. This shows that, for any $p \neq q \in \mathbb{R}^2$, the surface σ_{pq} is slanted, which proves Proposition 2.7(a).

For Proposition 2.7(b), it is equivalent, by the symmetry of the setup, to prove the following dual statement: For any $p_1 \neq q_1, p_2 \neq q_2 \in S$, such that $(p_1, q_1) \neq (p_2, q_2)$, we have $|\sigma_{p_1q_1} \cap \sigma_{p_2q_2} \cap \mathcal{I}| \leq 3$.

Let $p_1, q_1, p_2, q_2 \in S$ be as above, and assume that $|\sigma_{p_1q_1} \cap \sigma_{p_2q_2} \cap \mathcal{I}| \geq 4$. Note that this means that the two projective transformations $T_{p_1q_1}, T_{p_2q_2}$ agree in at least four distinct points of the projective plane. We claim that in this case $\sigma_{p_1q_1}$ and $\sigma_{p_2q_2}$, regarded as graphs of functions on the affine xy -plane, must coincide on some line in that plane.

This is certainly the case if $\sigma_{p_1q_1}$ and $\sigma_{p_2q_2}$ coincide,³ so we may assume that these surfaces are distinct, which implies that $T_{p_1q_1}$ and $T_{p_2q_2}$ are distinct projective transformations.

As is well known, two distinct projective transformations of the plane cannot agree at four distinct points so that no three of them are collinear. Hence, out of the four points at which $T_{p_1q_1}$ and $T_{p_2q_2}$ agree, three must be collinear. Denote this triple of points (in the projective xy -plane) as u_1, u_2, u_3 , and their respective images (in the projective zw -planes) as $v_i = T_{p_1q_1}(u_i) = T_{p_2q_2}(u_i)$, for $i = 1, 2, 3$. Then the line λ that contains u_1, u_2, u_3 is mapped by both $T_{p_1q_1}$ and $T_{p_2q_2}$ to a line λ^* , and, as a matter of fact, both transformations coincide on λ .

Passing back to the affine setting, let then λ, λ^* be a pair of lines in the xy -plane and the zw -plane, respectively, such that, for every $(x, y) \in \lambda$ (other than the point at which the

³ One can show that this cannot happen, but it has no effect on our analysis.

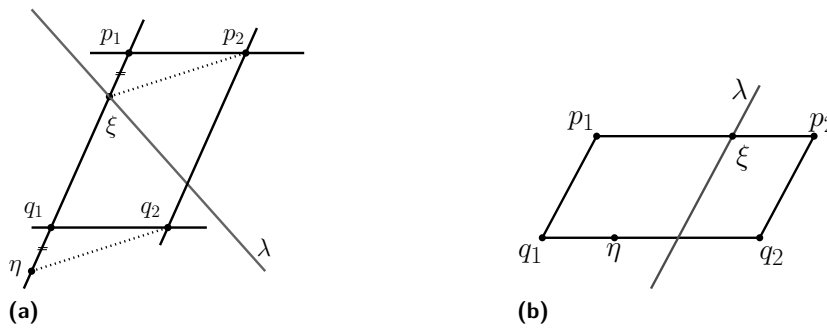


Figure 2 (a) The properties $|p_1\xi| = |q_1\eta|$, $\ell_{p_1p_2} \parallel \ell_{q_1q_2}$, and $(\xi, \eta) \in \sigma_{p_2q_2}$ imply that the triangles $p_1\xi p_2$, $q_1\eta q_2$ are congruent, and therefore $\ell_{p_1q_1}$, $\ell_{p_2q_2}$ must be parallel to one another. (b) $p_1q_1q_2p_2$ is a parallelogram and λ is parallel to $\ell_{p_1q_1}$ and $\ell_{p_2q_2}$.

denominator in (3) vanishes) there exists $(z, w) \in \lambda^*$, satisfying $(x, y, z, w) \in \sigma_{p_1q_1} \cap \sigma_{p_2q_2}$. We show that in this case p_1, q_1, p_2, q_2 are all collinear and $|p_1p_2| = |q_1q_2|$.

We first observe that $\ell_{p_1p_2} \parallel \ell_{q_1q_2}$. Indeed, if each of $\lambda \cap \ell_{p_1p_2}$ and $\lambda \cap \ell_{q_1q_2}$ is either empty or infinite, then we must have $\ell_{p_1p_2} \parallel \ell_{q_1q_2}$ (since both are parallel to λ). Otherwise, assume without loss of generality that $|\ell_{p_1p_2} \cap \lambda| = 1$, and let ξ denote the unique point in this intersection. Let η be the point such that (ξ, η) satisfies (3) with respect to both surfaces $\sigma_{p_1q_1}, \sigma_{p_2q_2}$ (the same point arises for both surfaces because $\xi \in \lambda$). That is, $\ell'_{p_1\xi} = \ell'_{q_1\eta}$, and $\ell'_{p_2\xi} = \ell'_{q_2\eta}$. In particular, $\ell_{p_1\xi} \parallel \ell_{q_1\eta}$, and $\ell_{p_2\xi} \parallel \ell_{q_2\eta}$. Since, by construction, $\xi \in \ell_{p_1p_2}$, we have $\ell_{p_1\xi} \equiv \ell_{p_2\xi}$, which yields that also $\ell_{q_1\eta} \parallel \ell_{q_2\eta}$. Thus necessarily q_1, q_2, η are collinear, and $\ell_{q_1q_2} \parallel \ell_{p_1p_2}$, as claimed.

Assume that at least one of $\ell_{p_1q_1}, \ell_{p_2q_2}$ intersects λ in exactly one point; say, without loss of generality, it is $\ell_{p_1q_1}$, and let ξ denote the unique point in this intersection. Similar to the argument just made, let η be the point such that (ξ, η) satisfies (3) with respect to both surfaces $\sigma_{p_1q_1}, \sigma_{p_2q_2}$. Note that since $\xi \in \ell_{p_1q_1}$, we must have $\eta \in \ell_{p_1q_1}$ too, and $|p_1\xi| = |q_1\eta|$. In particular, since $p_1 \neq q_2$, by assumption, we also have $\xi \neq \eta$. Using the properties $\ell_{p_1p_2} \parallel \ell_{q_1q_2}$ and $(\xi, \eta) \in \sigma_{p_2q_2}$, it follows that the triangles $p_1\xi p_2, q_1\eta q_2$ are congruent; see Figure 2(a). Thus, in particular, $|p_2\xi| = |q_2\eta|$. Since, by construction, also $\ell'_{p_2\xi} \equiv \ell'_{q_2\eta}$, it follows that $p_2, q_2 \in \ell_{\xi\eta}$. We conclude that in this case p_1, q_1, p_2, q_2 are collinear and $|p_1p_2| = |q_1q_2|$.

We are therefore left only with the case where each of $\lambda \cap \ell_{p_1q_1}$ and $\lambda \cap \ell_{p_2q_2}$ is either empty or infinite. That is, we have $\ell_{p_1q_1} \parallel \ell_{p_2q_2}$ (since both are parallel to λ). As has already been argued, we also have $\ell_{p_1p_2} \parallel \ell_{q_1q_2}$, and thus $p_1q_1q_2p_2$ is a parallelogram; see Figure 2(b). In particular, $|p_1p_2| = |q_1q_2|$. Let ξ be the intersection point of $\ell_{p_1p_2}$ with λ , and let η be the point such that (ξ, η) satisfies (3) with respect to both surfaces $\sigma_{p_1q_1}, \sigma_{p_2q_2}$. By construction $\ell_{p_1\xi} \parallel \ell_{q_1\eta}$ and $\ell_{p_2\xi} (= \ell_{p_1\xi}) \parallel \ell_{q_2\eta}$. Hence η must lie on $\ell_{q_1q_2}$. It is now easily checked that the only way in which (ξ, η) can lie on both surfaces $\sigma_{p_1q_1}$ and $\sigma_{p_2q_2}$ is when p_1, q_1, p_2, q_2 are all collinear; see Figure 2(b).

To recap, so far we have shown that for p_1, q_1, p_2 , and q_2 as above, either $|\sigma_{p_1q_1} \cap \sigma_{p_2q_2}| \leq 3$, or p_1, q_1, p_2 , and q_2 are collinear with $|p_1p_2| = |q_1q_2|$. It can then be shown that, in the latter case, any point $(u, v) \in \sigma_{p_1q_1} \cap \sigma_{p_2q_2}$ must satisfy $u, v \in \ell_{p_1q_1}$; see Figure 1(b). Thus, for a point $\pi \in \mathbb{R}^4$ incident to each of $\sigma_{p_1q_1}, \sigma_{p_2q_2}$, neither of $(\pi, \sigma_{p_1q_1}), (\pi, \sigma_{p_2q_2})$ is in \mathcal{I} . In other words, $\sigma_{p_1q_1} \cap \sigma_{p_2q_2} \cap \mathcal{I} = \emptyset$ in this case. This contradiction completes the proof of Proposition 2.7. ◀

3 Unit-area triangles spanned by points on three lines

In this section we consider the special case where S is contained in the union of three distinct lines l_1, l_2, l_3 . More precisely, we write $S = S_1 \cup S_2 \cup S_3$, with $S_i \subset l_i$, for $i = 1, 2, 3$, and we are only interested in the number of unit-area triangles spanned by triples of points in $S_1 \times S_2 \times S_3$. It is easy to see that in this case the number of unit-area triangles of this kind is $O(n^2)$. Indeed, for any pair of points $p, q \in S_1 \times S_2$, the line ℓ'_{pq} intersects l_3 in at most one point, unless ℓ'_{pq} coincides with l_3 . Ignoring situation of the latter kind, we get a total of $O(n^2)$ unit-area triangles. If no two lines among l_1, l_2, l_3 are parallel to one another, it can be checked that the number of pairs (p, q) such that $\ell'_{pq} = l_3$ is at most a constant, thus contributing a total of at most $O(n)$ unit-area triangles. For the case where two (or more) lines among l_1, l_2, l_3 are parallel, the number of unit-area triangles is easily seen to be $O(n^2)$.

In this section we present a rather subtle construction that shows that this bound is tight in the worst case, for any triple of distinct lines. Instead of just presenting the construction, we spend some time showing its connection to a more general setup considered by Elekes and Rónyai [6] (and also, in more generality, by Elekes and Szabó [7]).

Specifically, the main result of this section is the following.

► **Theorem 3.1.** *For any triple of distinct lines l_1, l_2, l_3 in \mathbb{R}^2 , and for any integer n , there exist subsets $S_1 \subset l_1, S_2 \subset l_2, S_3 \subset l_3$, each of cardinality $\Theta(n)$, such that $S_1 \times S_2 \times S_3$ spans $\Theta(n^2)$ unit-area triangles.*

Proof. The upper bound has already been established (for any choice of S_1, S_2, S_3), so we focus on the lower bound. We recall that by the area formula for triangles in the plane, if

$$\frac{1}{2} \begin{vmatrix} p_x & q_x & r_x \\ p_y & q_y & r_y \\ 1 & 1 & 1 \end{vmatrix} = 1, \quad (4)$$

then the points $p = (p_x, p_y)$, $q = (q_x, q_y)$ and $r = (r_x, r_y)$ form the vertices of a positively oriented unit-area triangle in \mathbb{R}^2 . (Conversely, if Δpqr has area 1 then the left-hand side of (4) has value ± 1 , depending on the orientation of (p, q, r) .)

To establish the lower bound, we distinguish between three cases, depending on the number of pairs of parallel lines among l_1, l_2, l_3 .

The three lines l_1, l_2, l_3 are mutually parallel. In this case we may assume without loss of generality that they are of the form

$$l_1 = \{(t, 0) \mid t \in \mathbb{R}\}, \quad l_2 = \{(t, 1) \mid t \in \mathbb{R}\}, \quad l_3 = \{(t, \alpha) \mid t \in \mathbb{R}\},$$

for some $1 < \alpha \in \mathbb{R}$. (We translate and rotate the coordinate frame so as to place l_1 at the x -axis and then apply an area-preserving linear transformation that scales the x - and y -axes by reciprocal values.) Then, as easily verified, the sets

$$\begin{aligned} S_1 &:= \{(x_i := \frac{i}{1-\alpha}, 0) \mid i = 1, \dots, n\} \subset l_1, \\ S_2 &:= \{(y_j := \frac{j}{\alpha}, 1) \mid j = 1, \dots, n\} \subset l_2, \\ S_3 &:= \{(z_{ij} := i + j - 2, \alpha) \mid i, j = 1, \dots, n\} \subset l_3 \end{aligned}$$

are such that $S_1 \times S_2 \times S_3$ spans $\Omega(n^2)$ unit-area triangles.

There is exactly one pair of parallel lines among l_1, l_2, l_3 . Using an area-preserving affine transformation of \mathbb{R}^2 (and possibly re-indexing the lines), we may assume that

$$l_1 = \{(t, 0) \mid t \in \mathbb{R}\}, \quad l_2 = \{(t, 1) \mid t \in \mathbb{R}\}, \quad l_3 = \{(0, t) \mid t \in \mathbb{R}\}.$$

Using (4), it is easily checked that the sets

$$\begin{aligned} S_1 &:= \{(x_i := 2^i + 2, 0) \mid i = 1, \dots, n\} \subset l_1, \\ S_2 &:= \{(y_j := 2^j + 2, 1) \mid j = 1, \dots, n\} \subset l_2, \\ S_3 &:= \{(0, z_{ij} := \frac{1}{1-2^{j-i}}) \mid i, j = 1, \dots, n, \quad i \neq j\} \subset l_3, \end{aligned}$$

span $\Omega(n^2)$ unit-area triangles.

No pair of lines among l_1, l_2, l_3 are parallel. This is the most involved case. Using an area-preserving affine transformation of \mathbb{R}^2 (that is, a linear map with determinant ± 1 and a translation), we may assume that the lines are given by

$$l_1 = \{(t, 0) \mid t \in \mathbb{R}\}, \quad l_2 = \{(0, t) \mid t \in \mathbb{R}\}, \quad l_3 = \{(t, -t + \alpha) \mid t \in \mathbb{R}\},$$

for some $\alpha \in \mathbb{R}$. By (4), the points $(x, 0) \in l_1$, $(0, y) \in l_2$, and $(z, -z + \alpha) \in l_3$ span a unit-area triangle if

$$\frac{1}{2} \begin{vmatrix} x & 0 & z \\ 0 & y & -z + \alpha \\ 1 & 1 & 1 \end{vmatrix} = 1, \quad \text{or} \quad z = f(x, y) := \frac{xy - \alpha x - 2}{y - x}.$$

Thus it suffices to find sets $X, Y, Z \subset \mathbb{R}$, each of cardinality $\Theta(n)$, such that

$$|\{(x, y, z) \in X \times Y \times Z \mid z = f(x, y)\}| = \Omega(n^2);$$

then the sets

$$S_1 := \{(x, 0) \mid x \in X\} \subset l_1, \quad S_2 := \{(0, y) \mid y \in Y\} \subset l_2, \quad S_3 := \{(z, -z + \alpha) \mid z \in Z\} \subset l_3,$$

are such that $S_1 \times S_2 \times S_3$ spans $\Omega(n^2)$ unit-area triangles.

The construction of S_1, S_2, S_3 : General context. As mentioned at the beginning of this section, rather than stating what S_1, S_2, S_3 are, we present the machinery that we have used for their construction, thereby demonstrating that this problem is a special case of the theory of Elekes and Rónyai [6]; we also refer the reader to the more recent related studies [7, 11, 12].

One of the main results of Elekes and Rónyai is the following. (Note that the bound in (i) has recently been improved to $O(n^{11/6})$ in [11, 12].)

► **Theorem 3.2** (Elekes and Rónyai [6]). *Let $f(x, y)$ be a bivariate real rational function. Then one of the following holds.*

(i) *For any triple of sets $A, B, C \subset \mathbb{R}$, each of size n ,*

$$|\{(x, y, z) \in A \times B \times C \mid z = f(x, y)\}| = o(n^2).$$

(ii) *There exist univariate real rational functions h, φ, ψ , such that f has one of the forms*

$$f(x, y) = h(\varphi(x) + \psi(y)), \quad f(x, y) = h(\varphi(x)\psi(y)), \quad f(x, y) = h\left(\frac{\varphi(x) + \psi(y)}{1 - \varphi(x)\psi(y)}\right).$$

Our problem is thus a special instance of the context in Theorem 3.2. Specifically, we claim that $f(x, y) = \frac{xy - \alpha x - 2}{y - x}$ satisfies condition (ii) of the theorem, which in turn will lead to the (natural) construction of the desired sets S_1, S_2, S_3 (see below for details).

So we set the task of describing a necessary and sufficient condition that a real bivariate (twice differentiable) function $F(x, y)$ is locally⁴ of the form $F(x, y) = h(\varphi(x) + \psi(y))$, for suitable univariate twice differentiable functions h, φ, ψ (not necessarily rational functions). This condition is presented in [6] where its (rather straightforward) necessity is argued. It is mentioned in [6] that the sufficiency of this test was observed by A. Jarai Jr. (apparently in an unpublished communication). Since no proof is provided in [6], we present in the full version a proof, for the sake of completeness.

► **Lemma 3.3.** *Let $F(x, y)$ be a bivariate twice-differentiable real function, and assume that neither of F_x, F_y is identically zero. Let $D(F) \subset \mathbb{R}^2$ denote the domain of definition of F , and let U be a connected component of the relatively open set $D(F) \setminus (\{F_y = 0\} \cup \{F_x = 0\}) \subset \mathbb{R}^2$. We let $q(x, y) := F_x/F_y$, which is defined, with a constant sign, over U . Then $\frac{\partial^2(\log |q(x, y)|)}{\partial x \partial y} \equiv 0$ over U if and only if F , restricted to U , is of the form $F(x, y) = h(\varphi(x) + \psi(y))$, for some (twice-differentiable) univariate real functions φ, ψ , and h .*

Proof. The proof shows that if $\frac{\partial^2(\log |q(x, y)|)}{\partial x \partial y} \equiv 0$ then

$$F_x/F_y = \varphi'(x)/\psi'(y), \quad (5)$$

for suitable twice differentiable strictly monotone functions φ and ψ , and then shows that this implies that F is of the form $F(x, y) = h(\varphi(x) + \psi(y))$, as claimed. ◀

The construction of S_1, S_2, S_3 : Specifics. We next apply Lemma 3.3 to our specific function $f(x, y) = \frac{xy - \alpha x - 2}{y - x}$. In what follows we fix a connected open set $U \subset D(f) \setminus (\{f_x = 0\} \cup \{f_y = 0\})$, and restrict the analysis only to points $(x, y) \in U$. We have

$$f_x = \frac{y^2 - \alpha y - 2}{(y - x)^2}, \quad \text{and} \quad f_y = \frac{-x^2 + \alpha x + 2}{(y - x)^2}.$$

By assumption, the numerators are nonzero and of constant signs, and the denominator is nonzero, over U . In particular, we have $\frac{f_x}{f_y} = \frac{(-x^2 + \alpha x + 2)^{-1}}{(y^2 - \alpha y - 2)^{-1}}$. That is, without explicitly testing that the condition in Lemma 3.3 holds, we see that f_x/f_y has the form in (5). Hence Lemma 3.3 implies that $f(x, y)$ can be written as $f(x, y) = h(\varphi(x) + \psi(y))$, for suitable twice-differentiable univariate functions φ, ψ , and h , where φ and ψ are given (up to additive constants) by $\varphi'(x) = -\frac{1}{x^2 - \alpha x - 2}$, $\psi'(y) = \frac{1}{y^2 - \alpha y - 2}$. As explained above, this already implies that f satisfies property (ii) of Theorem 3.2.

⁴ Note that such a local representation of F allows one to construct sets A, B, C showing that property (i) of Theorem 3.2 does not hold for F , i.e., sets such that there are $\Theta(n^2)$ solutions of $z = F(x, y)$ in $A \times B \times C$. This, using Theorem 3.2, implies the validity of property (ii) (globally, and with rational functions).

Straightforward integration of these expressions yields that, up to a common multiplicative factor, which can be dropped, we have⁵ $\varphi(x) = \ln \left| \frac{x - s_2}{x - s_1} \right|$, $\psi(y) = \ln \left| \frac{y - s_1}{y - s_2} \right|$, where s_1, s_2 are the two real roots of $s^2 - \alpha s - 2 = 0$.

We conclude that $f(x, y) = \frac{xy - \alpha x - 2}{y - x}$ is a function of

$$\varphi(x) + \psi(y) = \ln \left| \frac{x - s_2}{x - s_1} \right| + \ln \left| \frac{y - s_1}{y - s_2} \right| = \ln \left| \frac{x - s_2}{x - s_1} \right| \cdot \left| \frac{y - s_1}{y - s_2} \right|,$$

or, rather, a function of $u = \frac{x - s_2}{x - s_1} \cdot \frac{y - s_1}{y - s_2}$. A tedious calculation, which we omit, shows

that $f(x, y) = \frac{s_2 - s_1 u}{1 - u}$, confirming that f does indeed have one of the special forms in Theorem 3.2 above. That is, $f(x, y) = h(\varphi(x)\psi(y))$, where h, φ, ψ are the rational functions $h(u) = \frac{s_2 - s_1 u}{1 - u}$, $\varphi(x) = \frac{x - s_2}{x - s_1}$, $\psi(y) = \frac{y - s_1}{y - s_2}$ (these are not the φ, ψ in the derivation above).

We then choose points $x_1, \dots, x_n, y_1, \dots, y_n \in \mathbb{R}$ such that

$$\frac{x_i - s_2}{x_i - s_1} = \frac{y_i - s_2}{y_i - s_1} = 2^i, \quad \text{or} \quad x_i = y_i = \frac{2^i s_1 - s_2}{2^i - 1},$$

for $i = 1, \dots, n$, and let $X := \{x_1, \dots, x_n\}$ and $Y := \{y_1, \dots, y_n\}$. For $x = x_i, y = y_j$, the corresponding value of u is 2^{i-j} . Hence, setting

$$Z := \{f(x_i, y_j) \mid 1 \leq i, j \leq n\} = \left\{ \frac{s_2 - s_1 \cdot 2^{i-j}}{1 - 2^{i-j}} \mid 1 \leq i, j \leq n \right\},$$

which is clearly also of size $\Theta(n)$, completes the proof. ◀

4 Unit-area triangles in convex grids

A set $X = \{x_1, \dots, x_n\}$, with $x_1 < x_2 < \dots < x_n$, of real numbers is said to be *convex* if $x_{i+1} - x_i > x_i - x_{i-1}$, for every $i = 2, \dots, n - 1$. See [5, 13] for more details and properties of convex sets.

In this section we establish the following improvement of Theorem 2.1 for convex grids.

► **Theorem 4.1.** *Let $S = A \times B$, where $A, B \subset \mathbb{R}$ are convex sets of size $n^{1/2}$ each. Then the number of unit-area triangles spanned by the points of S is $O(n^{31/14})$.*

Proof. With each point $p = (a, b, c) \in A^3$ we associate a plane $h(p)$ in \mathbb{R}^3 , given by

$$\frac{1}{2} \begin{vmatrix} a & b & c \\ x & y & z \\ 1 & 1 & 1 \end{vmatrix} = 1, \quad \text{or equivalently by} \quad (c - b)x + (a - c)y + (b - a)z = 2. \quad (6)$$

We put $H := \{h(p) \mid p \in A^3\}$.

A triangle with vertices $(a_1, x_1), (a_2, x_2), (a_3, x_3)$ has unit area if and only if the left-hand side of (6) has absolute value 1, so for half of the permutations (i_1, i_2, i_3) of $(1, 2, 3)$, we have

⁵ Note also that f is defined over $y \neq x$, whereas in our derivation we also had to exclude $\{f_x = 0\} \cup \{f_y = 0\}$, i.e. $\{x = s_1\} \cup \{x = s_2\} \cup \{y = s_1\} \cup \{y = s_2\}$. Nevertheless, the final expression coincides with f also over these excluded lines.

$(x_{i_1}, x_{i_2}, x_{i_3}) \in h(a_{i_1}, a_{i_2}, a_{i_3})$. In other words, the number of unit-area triangles is at most one third of the number of incidences between the points of B^3 and the planes of H . In addition to the usual problematic issue that arise in point-plane incidence problems, where many planes can pass through a line that contains many points (see, e.g., [3]), we need to face here the issue that the planes of H are in general not distinct, and may arise with large multiplicity. Denote by $w(h)$ the *multiplicity* of a plane $h \in H$, that is, $w(h)$ is the number of points $p \in A^3$ for which $h(p) = h$. Observe that, for $p, p' \in A^3$,

$$h(p) \equiv h(p') \text{ if and only if } p' \in p + (1, 1, 1)\mathbb{R}. \quad (7)$$

We can transport this notion to points of A^3 , by defining the *multiplicity* $w(p)$ of a point $p \in A^3$ by

$$w(p) := |(p + (1, 1, 1)\mathbb{R}) \cap A^3|.$$

Then we clearly have $w(h(p)) = w(p)$ for each $p \in A^3$. Similarly, for $q \in B^3$, we put, by a slight abuse of notation,

$$w(q) := |(q + (1, 1, 1)\mathbb{R}) \cap B^3|,$$

and refer to it as the *multiplicity* of q . (Clearly, the points of B^3 are all distinct, but the notion of their “multiplicity” will become handy in one of the steps of the analysis — see below.)

Fix a parameter $k \in \mathbb{N}$, whose specific value will be chosen later. We say that $h \in H$ (resp., $p \in A^3$, $q \in B^3$) is *k-rich*, if its multiplicity is at least k ; otherwise we say that it is *k-poor*. For a unit-area triangle T , with vertices (a, x) , (b, y) , (c, z) , we say that T is *rich-rich* (resp., *rich-poor*, *poor-rich*, *poor-poor*) if $(a, b, c) \in A^3$ is *k-rich* (resp., rich, poor, poor), and $(x, y, z) \in B^3$ is *k-rich* (resp., poor, rich, poor). (These notions depend on the parameter k , which is fixed throughout this section.)

Next, we show that our assumption that A and B are convex allows us to have some control on the multiplicity of the points and the planes, which we need for the proof.

For two given subsets $X, Y \subset \mathbb{R}$, and for any $s \in \mathbb{R}$, denote by $\delta_{X,Y}(s)$ the number of representations of s in the form $x - y$, with $x \in X$, $y \in Y$. The following lemma is taken from Schoen and Shkredov.

► **Lemma 4.2** (Schoen and Shkredov [13]). *Let $X, Y \subset \mathbb{R}$, with X convex. Then, for any $\tau \geq 1$, we have $|\{s \in X - Y \mid \delta_{X,Y}(s) \geq \tau\}| = O\left(\frac{|X||Y|^2}{\tau^3}\right)$.*

Lemma 4.2 implies that the number of points $(a, b) \in A^2$, for which the line $(a, b) + (1, 1)\mathbb{R}$ contains at least k points of A^2 , is $O(n^{3/2}/k^2)$. Indeed, the number of differences $s \in A - A$ with $\delta_{A,A}(s) \geq \tau$ is $O(n^{3/2}/\tau^3)$. Each difference s determines, in a 1-1 manner, a line in \mathbb{R}^2 with orientation $(1, 1)$ that contains the $\delta_{A,A}(s)$ pairs $(a, b) \in A^2$ with $b - a = s$. Let M_τ (resp., $M_{\geq \tau}$) denote the number of differences $s \in A - A$ with $\delta_{A,A}(s) = \tau$ (resp., $\delta_{A,A}(s) \geq \tau$). Then the desired number of points is

$$\sum_{\tau \geq k} \tau M_\tau = k M_{\geq k} + \sum_{\tau > k} M_{\geq \tau} = O(n^{3/2}/k^2) + \sum_{\tau > k} O(n^{3/2}/\tau^3) = O(n^{3/2}/k^2).$$

► **Lemma 4.3.** *The number of k -rich points in A^3 and in B^3 is $O(n^2/k^2)$.*

Proof. Let $(a, b, c) \in A^3$ be *k-rich*. Then, by definition, the line $l := (a, b, c) + (1, 1, 1)\mathbb{R}$ contains at least k points of A^3 . We consider the line $l' := (a, b) + (1, 1)\mathbb{R}$, which is the (orthogonal) projection of l onto the xy -plane, which we identify with \mathbb{R}^2 . Note that the projection of the points of $l \cap A^3$ onto \mathbb{R}^2 is injective and its image is equal to $l' \cap A^2$. In

particular, l' contains at least k points of A^2 . As just argued, the total number of such points in A^2 (lying on some line of the form l' , that contains at least k points of A^2) is $O(n^{3/2}/k^2)$. Each such point is the projection of at most $n^{1/2}$ k -rich points of A^3 (this is the maximum number of lines of the form $(a, b, c) + (1, 1, 1)\mathbb{R}$ that project onto the same line l'). Thus, the number of k -rich points in A^3 is $O(\frac{n^{3/2}}{k^2} \cdot n^{1/2}) = O(n^2/k^2)$. The same bound applies to the number of k -rich points in B^3 , by a symmetric argument. ◀

► **Remark.** The proof of Lemma 4.3 shows, in particular, that the images of the sets of k -rich points of A^3 and of B^3 , under the projection map onto the xy -plane, are of cardinality $O(n^{3/2}/k^2)$.]

In what follows, we bound separately the number of unit-area triangles that are rich-rich, poor-rich (and, symmetrically, rich-poor), and poor-poor.

Rich-rich triangles. Note that for $((a, b, c), (\xi, \eta)) \in A^3 \times B^2$, with $a \neq b$, there exists at most one point $\zeta \in B$ such that $T((a, \xi), (b, \eta), (c, \zeta))$ has unit area. Indeed, the point (c, ζ) must lie on a certain line $l((a, \xi), (b, \eta))$ parallel to $(a, \xi) - (b, \eta)$. This line intersects $x = c$ in exactly one point (because $a \neq b$), which determines the potential value of ζ . Thus, since we are now concerned with the number of rich-rich triangles (and focusing at the moment on the case where $a \neq b$), it suffices to bound the number of such pairs $((a, b, c), (\xi, \eta))$, with $(a, b, c) \in A^3$ being rich, and $(\xi, \eta) \in B^2$ being the projection of a rich point of B^3 , which is $O((n^2/k^2) \cdot (n^{3/2}/k^2)) = O(n^{7/2}/k^4)$, using Lemma 4.3 and the Remark following it.

It is easy to check that the number of unit-area triangles $T(p, q, r)$, where $p, q, r \in P$ and p, q share the same abscissa (i.e., A -component), is $O(n^2)$. Indeed, there are $\Theta(n^{3/2})$ such pairs (p, q) , and for each of them there exist at most $n^{1/2}$ points $r \in P$, such that $T(p, q, r)$ has unit area (because the third vertex r must lie on a certain line $l(p, q)$, which passes through at most this number of points of P); here we do not use the fact that we are interested only in rich-rich triangles. We thus obtain the following lemma.

► **Lemma 4.4.** *The number of rich-rich triangles spanned by P is $O\left(\frac{n^{7/2}}{k^4} + n^2\right)$.*

Poor-rich and rich-poor triangles. Without loss of generality, it suffices to consider only poor-rich triangles. Put

$$H_i := \{h \in H \mid 2^{i-1} \leq w(h) < 2^i\}, \quad \text{for } i = 1, \dots, \log k, \text{ and}$$

$$S_{\geq k} := \{q \in B^3 \mid w(q) \geq k\}.$$

That is, by definition, $\bigcup_i H_i$ is the collection of k -poor planes of H , and $S_{\geq k}$ is the set of k -rich points of B^3 . Since each element of H_i has multiplicity at least 2^{i-1} , we have the trivial bound $|H_i| \leq n^{3/2}/2^{i-1}$.

Consider the family of horizontal planes $\mathcal{F} := \{\xi_z\}_{z \in B}$, where $\xi_{z_0} := \{z = z_0\}$. Our strategy is to restrict $S_{\geq k}$ and H_i , for $i = 1, \dots, \log k$, to the planes $\xi \in \mathcal{F}$, and apply the Szemerédi–Trotter incidence bound (see Theorem 2.2) to the resulting collections of points and intersection lines, on each such ξ . Note that two distinct planes $h_1, h_2 \in H$ restricted to ξ , become two distinct lines in ξ . Indeed, each plane of H contains a line parallel to $(1, 1, 1)$, and two such planes, that additionally share a horizontal line within ξ , must be identical. Using the Remark following Lemma 4.3, we have that the number of rich points $(x, y, z_0) \in S_{\geq k}$, with z_0 fixed, is $O(n^{3/2}/k^2)$; that is, $|S_{\geq k} \cap \xi_{z_0}| = O(n^{3/2}/k^2)$ for every fixed z_0 .

The number of incidences between the points of $S_{\geq k}$ and the poor planes of H , counted with multiplicity (of the planes) is at most $\sum_{z \in B} \sum_{i=1}^{\log k} 2^i \cdot \mathcal{I}(S_{\geq k} \cap \xi_z, H_{iz})$, where $H_{iz} := \{h \cap \xi_z \mid h \in H_i\}$. By Theorem 2.2, this is at most

$$\begin{aligned} & \sum_{z \in B} \sum_{i=1}^{\log k} 2^i \cdot O\left(\left(\frac{n^{3/2}}{k^2}\right)^{2/3} \left(\frac{n^{3/2}}{2^{i-1}}\right)^{2/3} + \frac{n^{3/2}}{k^2} + \frac{n^{3/2}}{2^{i-1}}\right) \\ &= \sum_{z \in B} O\left(\frac{n^2}{k^{4/3}} \sum_{i=1}^{\log k} 2^{i/3} + \frac{n^{3/2}}{k^2} \sum_{i=1}^{\log k} 2^i + n^{3/2} \log k\right) \\ &= \sum_{z \in B} O\left(\frac{n^2}{k} + \frac{n^{3/2}}{k} + n^{3/2} \log k\right) = O\left(\frac{n^{5/2}}{k} + n^2 \log k\right). \end{aligned}$$

This bounds the number of poor-rich triangles spanned by P . Clearly, using a symmetric argument, this bound also applies to the number of rich-poor triangles spanned by P . We thus obtain the following lemma.

► **Lemma 4.5.** *The number of poor-rich triangles and of rich-poor triangles spanned by P is $O\left(\frac{n^{5/2}}{k} + n^2 \log k\right)$.*

Poor-poor triangles. Again we are going to use Theorem 2.2. For $i = 1, \dots, \log k$, put

$$S_i := \{q \in B^3 \mid 2^{i-1} \leq w(q) < 2^i\},$$

and let S'_i, H'_i be the respective (orthogonal) projections of S_i, H_i to the plane $\eta := \{x + y + z = 1\}$. Note that H'_i is a collection of lines in η . Moreover, arguing as above, two distinct planes of H_i project to two distinct lines of H'_i , and thus the multiplicity of the lines is the same as the multiplicity of the original planes of H_i . Similarly, a point $q \in S_i$ with multiplicity t projects to a point $q' \in S'_i$ with multiplicity t (by construction, there are exactly t points of S_i that project to q'). These observations allow us to use here too the trivial bounds $|S'_i| \leq n^{3/2}/2^{i-1}$, $|H'_i| \leq n^{3/2}/2^{i-1}$, for $i = 1, \dots, \log k$.

Applying Theorem 2.2 to the collections S'_i, H'_j in η , for $i, j = 1, \dots, \log k$, taking under account the multiplicity of the points and of the lines in these collections, we obtain that the number of incidences between the poor points and the poor planes, counted with the appropriate multiplicity, is at most

$$\begin{aligned} \sum_{i,j=1}^{\log k} 2^{i+j} \cdot \mathcal{I}(S'_i, H'_j) &= \sum_{i,j=1}^{\log k} 2^{i+j} \cdot O\left(\left(\frac{n^{3/2}}{2^{i-1}}\right)^{2/3} \left(\frac{n^{3/2}}{2^{j-1}}\right)^{2/3} + \frac{n^{3/2}}{2^{i-1}} + \frac{n^{3/2}}{2^{j-1}}\right) \\ &= O\left(n^2 \sum_{i,j=1}^{\log k} 2^{(i+j)/3} + n^{3/2} \sum_{i,j=1}^{\log k} (2^i + 2^j)\right) = O\left(n^2 k^{2/3} + n^{3/2} k \log k\right). \end{aligned}$$

Thus, we obtain the following lemma.

► **Lemma 4.6.** *The number of poor-poor triangles spanned by P is $O\left(n^2 k^{2/3} + n^{3/2} k \log k\right)$.*

In summary, the number of unit-area triangles spanned by P is

$$O\left(\frac{n^{7/2}}{k^4} + \frac{n^{5/2}}{k} + n^2 k^{2/3} + n^{3/2} k \log k\right). \quad (8)$$

Setting $k = n^{9/28}$ makes this bound $O(n^{31/14})$, and Theorem 4.1 follows. ◀

Acknowledgment. We are grateful to Frank de Zeeuw for several very helpful comments that simplified some parts of the analysis.

References

- 1 R. Apfelbaum, *Geometric Incidences and Repeated Configurations*, Ph.D. Dissertation, School of Computer Science, Tel Aviv University, 2013.
- 2 R. Apfelbaum and M. Sharir, An improved bound on the number of unit-area triangles, *Discrete Comput. Geom.* 44 (2010), 753–761.
- 3 R. Apfelbaum and M. Sharir, Large bipartite graphs in incidence graphs of points and hyperplanes, *SIAM J. Discrete Math.* 21 (2007), 707–725.
- 4 A. Dumitrescu, M. Sharir and Cs. D. Tóth, Extremal problems on triangle areas in two and three dimensions, *J. Combinat. Theory, Ser. A* 116 (2009), 1177–1198.
- 5 G. Elekes, M. Nathanson, and I. Ruzsa, Convexity and sumsets, *J. Number Theory* 83(2) (2000), 194–201.
- 6 G. Elekes and L. Rónyai, A combinatorial problem on polynomials and rational functions, *J. Combinat. Theory Ser. A* 89 (2000), 1–20.
- 7 G. Elekes and E. Szabó, How to find groups? (And how to use them in Erdős geometry?), *Combinatorica* 32 (2012), 537–571.
- 8 P. Erdős and G. Purdy, Some extremal problems in geometry, *J. Combinat. Theory* 10 (1971), 246–252.
- 9 P. Erdős and G. Purdy, Extremal problems in combinatorial geometry. in *Handbook of Combinatorics* (R. Graham, M. Grötschel and L. Lovász, editors), Vol. 1, 809–874, Elsevier, Amsterdam, 1995.
- 10 J. Pach and M. Sharir, Repeated angles in the plane and related problems, *J. Combinat. Theory Ser. A* 59 (1992), 12–22.
- 11 O. E. Raz, M. Sharir, and J. Solymosi, Polynomials vanishing on grids: The Elekes-Rónyai problem revisited, *Amer. J. Math.*, to appear. Also in *Proc. 30th Annu. ACM Sympos. Comput. Geom.*, 2014, 251–260.
- 12 O. E. Raz, M. Sharir, and F. de Zeeuw, Polynomials vanishing on Cartesian products: The Elekes-Szabó Theorem revisited. This proceedings.
- 13 T. Schoen and I. D. Shkredov, On sumsets of convex sets, *Combinat. Probab. Comput.* 20 (2011), 793–798.
- 14 J. Solymosi and T. Tao, An incidence theorem in higher dimensions, *Discrete Comput. Geom.* 48 (2012), 255–280.
- 15 J. Solymosi and F. de Zeeuw, Incidence bounds on Cartesian products, manuscript, 2014.
- 16 E. Szemerédi and W. T. Trotter, Extremal problems in discrete geometry, *Combinatorica* 3 (1983), 381–392.
- 17 J. Zahl, A Szemerédi-Trotter type theorem in \mathbb{R}^4 , in arXiv:1203.4600.

On the Number of Rich Lines in Truly High Dimensional Sets

Zeev Dvir¹ and Sivakanth Gopi²

- 1 Department of Computer Science and Department of Mathematics, Princeton University
35 Olden Street, Princeton, NJ 08540-5233, USA
zeev.dvir@gmail.com
- 2 Department of Computer Science, Princeton University
35 Olden Street, Princeton, NJ 08540-5233, USA
sgopi@cs.princeton.edu

Abstract

We prove a new upper bound on the number of r -rich lines (lines with at least r points) in a ‘truly’ d -dimensional configuration of points $v_1, \dots, v_n \in \mathbb{C}^d$. More formally, we show that, if the number of r -rich lines is significantly larger than n^2/r^d then there must exist a large subset of the points contained in a hyperplane. We conjecture that the factor r^d can be replaced with a tight r^{d+1} . If true, this would generalize the classic Szemerédi-Trotter theorem which gives a bound of n^2/r^3 on the number of r -rich lines in a planar configuration. This conjecture was shown to hold in \mathbb{R}^3 in the seminal work of Guth and Katz [7] and was also recently proved over \mathbb{R}^4 (under some additional restrictions) [14]. For the special case of arithmetic progressions (r collinear points that are evenly distanced) we give a bound that is tight up to lower order terms, showing that a d -dimensional grid achieves the largest number of r -term progressions.

The main ingredient in the proof is a new method to find a low degree polynomial that vanishes on many of the rich lines. Unlike previous applications of the polynomial method, we do not find this polynomial by interpolation. The starting observation is that the degree $r - 2$ Veronese embedding takes r -collinear points to r linearly dependent images. Hence, each collinear r -tuple of points, gives us a dependent r -tuple of images. We then use the design-matrix method of [1] to convert these ‘local’ linear dependencies into a global one, showing that all the images lie in a hyperplane. This then translates into a low degree polynomial vanishing on the original set.

1998 ACM Subject Classification G.2.1 Combinatorics: Counting Problems

Keywords and phrases Incidences, Combinatorial Geometry, Designs, Polynomial Method, Additive Combinatorics

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.584

1 Introduction

The Szemerédi-Trotter theorem gives a tight upper bound on the number of incidences between a collection of points and lines in the real plane. We write $A \lesssim B$ to denote $A \leq C \cdot B$ for some absolute constant C and $A \approx B$ if we have both $A \lesssim B$ and $B \lesssim A$. We use $A \gg B$ to mean $A \geq C \cdot B$ for some sufficiently large constant C and we sometimes use a subscript d to mean that the constant C in the inequalities can depend on d .



© Zeev Dvir and Sivakanth Gopi;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG’15).

Editors: Lars Arge and János Pach; pp. 584–598



Leibniz International Proceedings in Informatics

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

► **Theorem 1** ([17]). *Given a set of points V and a set of lines \mathcal{L} in \mathbb{R}^2 , let $I(V, \mathcal{L})$ be the set of incidences between V and \mathcal{L} . Then,*

$$I(V, \mathcal{L}) \lesssim |V|^{2/3} |\mathcal{L}|^{2/3} + |V| + |\mathcal{L}|.$$

This fundamental theorem has found many applications in various areas (see [4] for some examples) and is known to also hold in the complex plane \mathbb{C}^2 [18, 19]. In recent years there has been a growing interest in high dimensional variants of line-point incidence bounds [13, 9, 11, 14, 16, 3]. This is largely due to the breakthrough results of Guth and Katz [7] who proved the Erdős distinct distances conjecture. One of the main ingredients in their proof was an incidence theorem for configurations of lines in \mathbb{R}^3 satisfying some ‘truly 3 dimensional’ condition (e.g, not too many lines in a plane). The intuition is that, in high dimensions, it is ‘harder’ to create many incidences between points and lines. This intuition is of course false if our configuration happens to lie in some low-dimensional space. In this work we prove stronger line-point incidence bounds for sets of points that do not contain a large low-dimensional subset.

To state our main theorem we first restate the Szemerédi-Trotter theorem as a bound on the number of r -rich lines (lines containing at least r points) in a given set of points. Since our results will hold over the complex numbers we will switch now from \mathbb{R} to \mathbb{C} . The complex version of Szemerédi-Trotter was first proved by Tóth [18] and then proved using different methods by Zahl [19]. For a finite set of points V , we denote by $\mathcal{L}_r(V)$ the set of r -rich lines in V . The following is equivalent to Theorem-1 (but stated over \mathbb{C}).

► **Theorem 2** ([18, 19]). *Given a set V of n points in \mathbb{C}^2 , for $r \geq 2$,*

$$|\mathcal{L}_r(V)| \lesssim \frac{n^2}{r^3} + \frac{n}{r}.$$

Theorem 2 is tight since a two dimensional square grid of n points contains $\gtrsim n^2/r^3$ lines that are r -rich. We might then ask whether a d -dimensional grid $G_d = \{1, 2, \dots, h\}^d$, with $h \approx n^{1/d}$, has asymptotically the maximal number of r -rich lines among all n -point configurations that do not have a large low-dimensional subset. In [15], it was shown that for $r \ll_d n^{1/d}$,

$$|\mathcal{L}_r(G_d)| \approx_d \frac{n^2}{r^{d+1}}.$$

Clearly, we can obtain a larger number of rich lines in \mathbb{C}^d if V is a union of several low-dimensional grids. For example, for some $\alpha \gg_d 1$ and $d > \ell > 1$, we can take a disjoint union of $r^{d-\ell}/\alpha$ ℓ -dimensional grids G_ℓ of size $\alpha n/r^{d-\ell}$ each. Each of these grids will have $\gtrsim_d \alpha^2 n^2 / r^{2d-\ell+1}$ r -rich lines and so, together we will get $\gtrsim_d \alpha n^2 / r^{d+1}$ rich lines. We can also take a union of n/r lines containing r points each, to get more r -rich lines than in the d -dimensional grid G_d when $r \gg_d n^{1/d}$. We thus arrive at the following conjecture which, if true, would mean that the best one can do is to paste together a number of grids as above.

► **Conjecture 3.** *For $r \geq 2$, suppose $V \subset \mathbb{C}^d$ is a set of n points with*

$$|\mathcal{L}_r(V)| \gg_d \frac{n^2}{r^{d+1}} + \frac{n}{r}.$$

Then there exists $1 < \ell < d$ and a subset $V' \subset V$ of size $\gtrsim_d n/r^{d-\ell}$ which is contained in an ℓ -flat (i.e. an ℓ -dimensional affine subspace).

This conjecture holds in \mathbb{R}^3 [7] and, in a slightly weaker form, in \mathbb{R}^4 [14]. We compare these two results with ours later in the introduction. Our main result makes a step in the direction of this conjecture. First of all, our bound is off by a factor of r from the optimal bound (i.e. with n^2/r^d instead of n^2/r^{d+1}). Secondly, we are only able to detect a $(d-1)$ -dimensional subset (instead of finding the correct ℓ which may be smaller).

► **Theorem 4.** *For all $d \geq 1$ there exists constants C_d, C'_d such that the following holds. Let $V \subset \mathbb{C}^d$ be a set of n points and let $r \geq 2$ be an integer. Suppose that for some $\alpha \geq 1$,*

$$|\mathcal{L}_r(V)| \geq C_d \cdot \alpha \cdot \frac{n^2}{r^d}.$$

Then, there exists a subset $\tilde{V} \subset V$ of size at least $C'_d \cdot \alpha \cdot \frac{n}{r^{d-2}}$ contained in a $(d-1)$ -flat. We can take the constants C_d, C'_d to be $d^{cd}, d^{c'd}$ for absolute constants $c, c' > 0$.

Notice that the theorem is only meaningful when $r \gg d^c$ for some constant c (otherwise the factor r^d in the assumption will be swallowed by the constant C_d). On the other hand, if $r \gg n^{1/(d-1)}$ then the conclusion always holds. Hence, the theorem is meaningful when r is in a ‘middle’ range. Notice also that for $d = 2, 3$ and r sufficiently small, the condition of the theorem also cannot hold, by the Szemerédi-Trotter theorem. However, when d becomes larger, our theorem gives nontrivial results (and becomes closer to optimal for large d). The proof of Theorem 4 actually shows (Lemma 19) that, under the same hypothesis, most of the rich lines must be contained in a hypersurface of degree smaller than r . This in itself can be very useful, as we will see in the proof of Theorem 9 which uses this fact to prove certain sum-product estimates. The existence of such a low-degree hypersurface containing most of the curves can also be obtained when there are many r -rich curves of bounded degree with ‘two degrees of freedom’, i.e. through every pair of points there are at most $O(1)$ curves (see Remark 22).

Counting arithmetic progressions

An r -term arithmetic progression in \mathbb{C}^d is simply a set of r points of the form $\{y, y+x, y+2x, \dots, y+(r-1)x\}$ with $x, y \in \mathbb{C}^d$. This is a special case of r collinear points and, for this case, we can derive a tighter bound than for the general case. In a nutshell, we can show that a d -dimensional grid contains the largest number of r -term progressions, among all sets that do not contain a large $d-1$ dimensional subset. The main extra property of arithmetic progressions we use in the proof is that they behave well under products. That is, if we take a Cartesian product of V with itself, the number of progressions of length r squares.

For a finite set $V \subset \mathbb{C}^d$, let us denote the number of r -term arithmetic progressions contained in V by $\mathbf{AP}_r(V)$. We first observe that, for all sufficiently small r , the grid G_d (defined above) contains at least $\gtrsim_d n^2/r^d$ r -term progressions. To see where the extra factor of r comes from, notice that the $2r$ -rich lines in G_d will contain r arithmetic progressions of length r each. Our main theorem shows that this is optimal, as long as there is no large low-dimensional set.

► **Theorem 5.** *Let $0 < \epsilon < 1$ and $V \subset \mathbb{C}^d$ be a set of size n and suppose that for some $r \geq 4$ we have*

$$\mathbf{AP}_r(V) \gg_{d,\epsilon} \frac{n^2}{r^{d-\epsilon}}.$$

Then, there exists a subset $\tilde{V} \subset V$ of size $\gtrsim_{d,\epsilon} \frac{n}{r^{2d/\epsilon-1}}$ contained in a hyperplane.

1.1 Related Work

To make the comparison with prior work easier, Theorem 4 can be stated equivalently as follows:

► **Theorem 6** (Equiv. to Theorem 4). *Given a set V of n points in \mathbb{C}^d , let s_{d-1} denote the maximum number of points of V contained in a hyperplane. Then for $r \geq 2$,*

$$|\mathcal{L}_r(V)| \lesssim_d \frac{n^2}{r^d} + \frac{ns_{d-1}}{r^2}.$$

Using the incidence bound between points and lines in \mathbb{R}^3 proved by Guth and Katz [7], one can prove the following theorem from which Conjecture 3 in \mathbb{R}^3 trivially follows.

► **Theorem 7** (Guth and Katz [7]). *Given a set V of n points in \mathbb{R}^3 , let s_2 denote the maximum number of points of V contained in a 2-flat. Then for $r \geq 2$,*

$$|\mathcal{L}_r(V)| \lesssim \frac{n^2}{r^4} + \frac{ns_2}{r^3} + \frac{n}{r}.$$

Similarly, using the results of Sharir and Solomon [14], we can prove the following theorem from which a slightly weaker version of Conjecture 3 in \mathbb{R}^4 trivially follows.

► **Theorem 8** (Sharir and Solomon [14]). *Given a set V of n points in \mathbb{R}^4 , let s_2 denote the maximum number of points of V contained in a 2-flat and s'_3 denote the maximum number of points of V contained in a quadric hypersurface or a hyperplane. Then there is an absolute constant $c > 0$ such that for $r \geq 2$,*

$$|\mathcal{L}_r(V)| \lesssim 2^{c\sqrt{\log n}} \cdot \left(\frac{n^2}{r^5} + \frac{ns'_3}{r^4} + \frac{ns_2}{r^3} + \frac{n}{r} \right).$$

We are not aware of any examples where points arranged on a quadric hypersurface in \mathbb{R}^4 result in significantly more rich lines than in a four dimensional grid. It is, however, possible that one needs to weaken Conjecture 3 so that for some $1 < \ell < d$, an ℓ -dimensional hypersurface of constant degree (possibly depending on ℓ) contains $\gtrsim_d n/r^{d-\ell}$ points.

In [15], it was shown that $|\mathcal{L}_r(V)| \lesssim_d \frac{n^2}{r^{d+1}}$ when $V \subset \mathbb{R}^d$ is a *homogeneous* set. This roughly means that the point set is a perturbation of the grid G_d . In [10], the result was extended for pseudolines and homogeneous sets in \mathbb{R}^d where pseudolines are a generalization of lines which include constant degree irreducible algebraic curves. Adding the homogeneous condition on a set is a much stronger condition (for sufficiently small r) than requiring that no large subset belongs to a hyperplane (however, we cannot derive these results from ours since our dependence on d is suboptimal).

1.1.1 Subsequent Work

Subsequent to our work, Hablicsek and Scherr [8] improved Theorem 4 in the case of $V \subset \mathbb{R}^d$. It was shown that if $\mathcal{L}_r(V) \gg_d \frac{n^2}{r^{d+1}}$, then $\gtrsim_d \frac{n}{r^{d-1}}$ points are contained in a $(d-1)$ -flat. In a further improvement, Zahl [20] extended this result to $V \subset \mathbb{C}^d$ though with an ϵ loss in the exponent of n , i.e. if $\mathcal{L}_r(V) \gg_{d,\epsilon} \frac{n^{2+\epsilon}}{r^{d+1}}$ then $\gtrsim_{d,\epsilon} \frac{n^{1+\epsilon}}{r^{d-1}}$ points are contained in a $(d-1)$ -flat. This brings us closer to Conjecture 3, although the conclusion about a large low-dimensional subset is still very weak.

1.2 Overview of the proof

The main tool used in the proof of Theorem 4 is a rank bound for design matrices. A *design matrix* is a matrix with entries in \mathbb{C} and whose support (set of non-zero entries) forms a specific pattern. Namely, the supports of different columns have small intersections, the columns have large support and rows are sparse (see Definition 11). Design matrices were introduced in [1, 5] to study quantitative variants of the Sylvester-Gallai theorem. These works prove certain lower bounds on the rank of such matrices, depending only on the combinatorial properties of their support (see Section 2.1). Such rank bounds can be used to give *upper bounds* on the dimension of point configurations in which there are many ‘local’ linear dependencies. This is done by using the local dependencies to construct rows of a design matrix M , showing that its rank is high and then arguing that the dimension of the original set is small since it must lie in the kernel of M .

Suppose we have a configuration of points with many r -rich lines. Clearly, $r \geq 3$ collinear points are also linearly dependent. However, this conclusion does not use the fact that r may be larger than 3. To use this information, we observe that a certain map, called the Veronese embedding, takes r -collinear points to r linearly dependent points in a larger dimensional space (see Section 2.2). Thus we can create a design matrix using these linear dependencies similarly to the constructions of [2, 5] to get an upper bound on the dimension of the *image* of the original set, under the Veronese embedding. We use this upper bound to conclude that there is a polynomial of degree $r - 2$ which contains all the points in our original configuration. We then proceed in a way similar to the proof of the Joints conjecture by Guth and Katz [6] to conclude that there is a hyperplane which contains many points of the configuration (by finding a ‘flat’ point of the surface).

1.3 Application: Sum-product estimates

Here, we show a simple application of our techniques to prove sum product estimates over \mathbb{C} . Though we can get slightly better estimates (i.e. without the log factor) using the Szemerédi-Trotter theorem in the complex plane, we include them only as an example of how to use a higher-dimensional theorem in this setting. We hope that future progress on proving Conjecture 3 will result in progress on sum product problems.

We begin with some notation. For two sets $A, B \subset \mathbb{C}$ we denote by $A + B = \{a + b \mid a, b \in A\}$ the sum set of A and B . For a set $A \subset \mathbb{C}$ and a complex number $t \in \mathbb{C}$ we denote by $tA = \{ta \mid a \in A\}$ the dilate of A by t . Hence we have that $A + tA = \{a + ta' \mid a, a' \in A\}$.

► **Theorem 9.** *Let $A \subset \mathbb{C}$ be a set of N complex numbers and let $1 \ll C \ll \sqrt{N}$. Define the set*

$$T_C = \left\{ t \in \mathbb{C} \mid |A + tA| \leq \frac{N^{1.5}}{C\sqrt{\log N}} \right\}.$$

Then, $|T_C| \lesssim \frac{N}{C^2}$.

By taking C to be a large constant, an immediate corollary is:

► **Corollary 10.** *Let $A \subset \mathbb{C}$ be a finite set. Then*

$$|A + A \cdot A| = |\{a + bc \mid a, b, c \in A\}| \gtrsim \frac{|A|^{1.5}}{\sqrt{\log |A|}}.$$

1.4 Organization

In Section 2 we give some preliminaries, including on design matrices and the Veronese embedding. In Section 3 we prove Theorem 4. In Section 4 we prove Theorem 5. In Section 5 we prove Theorem 9.

2 Preliminaries

We begin with some notation. For a vector $v \in \mathbb{C}^n$ and a set $I \subset [n]$ we denote by $v_I \subset \mathbb{C}^I$ the restriction of v to indices in I . We denote the *support* of a vector $v \in \mathbb{C}^d$ by $\text{supp}(v) = \{i \in [d] \mid v_i \neq 0\}$ (this notation is extended to matrices as well). For a set of n points $V \subset \mathbb{C}^d$ and an integer ℓ , we denote by $V^\ell \subset \mathbb{C}^{d\ell}$ its ℓ -fold Cartesian product i.e. $V^\ell = V \times V \times \dots \times V$ (ℓ times) where we naturally identify $\mathbb{C}^d \times \mathbb{C}^d \times \dots \times \mathbb{C}^d$ (ℓ times) with $\mathbb{C}^{d\ell}$.

2.1 Design matrices

Design matrices, defined in [1], are matrices that satisfy a certain condition on their support.

► **Definition 11** (Design matrix). Let A be an $m \times n$ matrix over a field \mathbb{F} . Let $R_1, \dots, R_m \in \mathbb{F}^n$ be the rows of A and let $C_1, \dots, C_n \in \mathbb{F}^m$ be the columns of A . We say that A is a (q, k, t) -design matrix if

1. For all $i \in [m]$, $|\text{supp}(R_i)| \leq q$.
2. For all $j \in [n]$, $|\text{supp}(C_j)| \geq k$.
3. For all $j_1 \neq j_2 \in [n]$, $|\text{supp}(C_{j_1}) \cap \text{supp}(C_{j_2})| \leq t$.

Surprisingly, one can derive a general bound on the rank of complex design matrices, despite having no information on the values present at the non-zero positions of the matrix. The first bound of this form was given in [1] which was improved in [5].

► **Theorem 12** ([5]). Let A be an $m \times n$ matrix with entries in \mathbb{C} . If A is a (q, k, t) -design matrix then the following bounds hold:

$$\text{rank}(A) \geq n - \frac{ntq^2}{k}, \tag{1}$$

$$\text{rank}(A) \geq n - \frac{mtq^2}{k^2}. \tag{2}$$

2.2 The Veronese embedding

We denote by

$$\mathbf{m}(d, r) = \binom{d+r}{d}$$

the number of monomials of degree at most r in d variables. We will often use the lower bound $\mathbf{m}(d, r) \geq (r/d)^d$. The Veronese embedding $\phi_{d,r} : \mathbb{C}^d \mapsto \mathbb{C}^{\mathbf{m}(d,r)}$ sends a point $a = (a_1, \dots, a_d) \in \mathbb{C}^d$ to the vector of evaluations of all monomials of degree at most r at the point a . For example, the map $\phi_{2,2}$ sends (a_1, a_2) to $(1, a_1, a_2, a_1^2, a_1a_2, a_2^2)$. We can identify each point $w \in \mathbb{C}^{\mathbf{m}(d,r)}$ with a polynomial $f_w \in \mathbb{C}[x_1, \dots, x_d]$ of degree at most r in an obvious manner so that the value $f_w(a)$ at a point $a \in \mathbb{C}^d$ is given by the standard inner product $\langle w, \phi_{d,r}(a) \rangle$. We will use the following two easy claims.

► **Claim 13.** *Let $V \subset \mathbb{C}^d$ and let $U = \phi_{d,r}(V) \subset \mathbb{C}^{\mathbf{m}(d,r)}$. Then U is contained in a hyperplane iff there is a non-zero polynomial $f \in \mathbb{C}[x_1, \dots, x_d]$ of degree at most r that vanishes on all points of V .*

Proof. Each hyperplane in $\mathbb{C}^{\mathbf{m}(d,r)}$ is given as the set of points having inner product zero with some $w \in \mathbb{C}^{\mathbf{m}(d,r)}$. If we take the corresponding polynomial $f_w \in \mathbb{C}[x_1, \dots, x_d]$ we get that it vanishes on V iff $\phi_{d,r}(V)$ is contained in the hyperplane defined by w . ◀

► **Claim 14.** *Suppose the $r+2$ points $v_1, \dots, v_{r+2} \in \mathbb{C}^d$ are collinear and let $\phi = \phi_{d,r} : \mathbb{C}^d \mapsto \mathbb{C}^{\mathbf{m}(d,r)}$. Then, the points $\phi(v_1), \dots, \phi(v_{r+2})$ are linearly dependent. Moreover, every $r+1$ of the points $\phi(v_1), \dots, \phi(v_{r+2})$ are linearly independent.*

Proof. Denote $u_i = \phi(v_i)$ for $i = 1, \dots, r+2$. To show that the u_i 's are linearly dependent it is enough to show that, for any $w \in \mathbb{C}^{\mathbf{m}(d,r)}$, if all the $r+1$ inner products $\langle w, u_1 \rangle, \dots, \langle w, u_{r+1} \rangle$ are zero, then the inner product $\langle w, u_{r+2} \rangle$ must also be zero. Suppose this is the case, and let $f_w \in \mathbb{C}[x_1, \dots, x_d]$ be the polynomial of degree at most r associated with the point w so that $\langle w, u_i \rangle = f_w(v_i)$ for all $1 \leq i \leq r+1$. Since the points v_1, \dots, v_{r+2} are on a single line $L \subset \mathbb{C}^d$, and since the polynomial f_w vanishes on $r+1$ of them, we have that f_w must vanish identically on the line L and so $f_w(v_{r+2}) = \langle w, u_{r+2} \rangle = 0$ as well.

To show the ‘moreover’ part, suppose in contradiction that u_{r+1} is in the span of u_1, \dots, u_r . We can find, by interpolation, a non-zero polynomial $f \in \mathbb{C}[x_1, \dots, x_d]$ of degree at most r such that $f(v_1) = \dots = f(v_r) = 0$ and $f(v_{r+1}) = 1$. More formally, we can transform the line containing the $r+1$ points to the x_1 -axis by a linear transformation, and then interpolate a degree r polynomial in x_1 with the required properties using the invertibility of the Vandermonde matrix. Now, let $w \in \mathbb{C}^{\mathbf{m}(d,r)}$ be the point such that $f = f_w$. We know that $\langle w, u_i \rangle = 0$ for $i = 1 \dots r$ and thus, since u_{r+1} is in the span of u_1, \dots, u_r , we get that $f(v_{r+1}) = \langle w, u_{r+1} \rangle = 0$ in contradiction. This completes the proof. ◀

2.3 Polynomials vanishing on grids

We recall the Schwartz-Zippel lemma.

► **Lemma 15** ([12, 21]). *Let $S \subset \mathbb{F}$ be a finite subset of an arbitrary field \mathbb{F} and let $f \in \mathbb{F}[x_1, \dots, x_d]$ be a non-zero polynomial of degree at most r . Then*

$$|\{(a_1, \dots, a_d) \in S^d \subset \mathbb{F}^d \mid f(a_1, \dots, a_d) = 0\}| \leq r \cdot |S|^{d-1}.$$

An easy corollary is the following claim about homogeneous polynomials.

► **Lemma 16.** *Let $S \subset \mathbb{F}$ be a finite subset of an arbitrary field \mathbb{F} and let $f \in \mathbb{F}[x_1, \dots, x_d]$ be a non-zero homogeneous polynomial of degree at most r . Then*

$$|\{(1, a_2, \dots, a_d) \in \{1\} \times S^{d-1} \mid f(1, a_2, \dots, a_d) = 0\}| \leq r \cdot |S|^{d-2}.$$

Proof. Let $g(x_2, \dots, x_d) = f(1, x_2, \dots, x_d)$ be the polynomial one obtains from fixing $x_1 = 1$ in f . Then g is a polynomial of degree at most r in $d-1$ variables. If g was the zero polynomial then f would have been divisible by $1-x_1$ which is impossible for a homogeneous polynomial. Hence, we can use Lemma 15 to bound the number of zeros of g in the set S^{d-1} by $r \cdot |S|^{d-2}$. This completes the proof. ◀

Another useful claim says that if a degree one polynomial (i.e. the equation of a hyperplane) vanishes on a large subset of the product set V^ℓ , then there is another degree one polynomial that vanishes on a large subset of V .

► **Lemma 17.** *Let $V \subset \mathbb{C}^d$ be a set of n points and let $V^\ell \subset \mathbb{C}^{d\ell}$ be its ℓ -fold Cartesian product. Let $H \subset \mathbb{C}^{d\ell}$ be an affine hyperplane such that $|H \cap V^\ell| \geq \delta \cdot n^\ell$. Then, there exists an affine hyperplane $H' \subset \mathbb{C}^d$ such that $|H' \cap V| \geq \delta \cdot n$.*

Proof. Let $h \in \mathbb{C}^{d\ell}$ be the vector perpendicular to H so that $x \in H$ iff $\langle x, h \rangle = b$ for some $b \in \mathbb{C}$. Observing the product structure of $\mathbb{C}^{d\ell} = (\mathbb{C}^d)^\ell$ we can write $h = (h_1, \dots, h_\ell)$ with each $h_i \in \mathbb{C}^d$. W.l.o.g suppose that $h_1 \neq 0$. For each $a = (a_2, \dots, a_\ell) \in V^{\ell-1}$ let $V_a^\ell = V \times \{a_2\} \times \dots \times \{a_\ell\}$. Since there are $n^{\ell-1}$ different choices for $a \in V^{\ell-1}$, and since

$$|V^\ell \cap H| = \sum_{a \in V^{\ell-1}} |V_a^\ell \cap H|,$$

there must be some a with $|V_a^\ell \cap H| \geq \delta \cdot n$. Let $H' \subset \mathbb{C}^d$ be the hyperplane defined by the equation

$$x \in H' \text{ iff } \langle x, h_1 \rangle + \langle a_2, h_2 \rangle + \dots + \langle a_\ell, h_\ell \rangle = b.$$

Then, $H' \cap V$ is in one-to-one correspondence with the set $V_a^\ell \cap H$ and so has the same size. ◀

2.4 A graph refinement lemma

We will need the following simple lemma, showing that any bipartite graph can be refined so that both vertex sets have high minimum degree (relative the to the original edge density).

► **Lemma 18.** *Let $G = (A \sqcup B, E)$ be a bipartite graph with $E \subset A \times B$ and edge set $E \neq \emptyset$. Then there exists non-empty sets $A' \subset A$ and $B' \subset B$ such that if we consider the induced subgraph $G' = (A' \sqcup B', E')$ then*

1. *The minimum degree in A' is at least $\frac{|E|}{4|A|}$*
2. *The minimum degree in B' is at least $\frac{|E|}{4|B|}$*
3. $|E'| \geq |E|/2$.

Proof. We will construct A' and B' using an iterative procedure. Initially let $A' = A$ and $B' = B$. Let $G' = (A' \sqcup B', E')$ be the induced subgraph of G . If there is a vertex in A' with degree (in the induced subgraph G') less than $\frac{|E|}{4|A|}$, remove it from A' . If there is a vertex in B' with degree (in the induced subgraph G') less than $\frac{|E|}{4|B|}$, remove it from B' . At the end of this procedure, we are left with sets A', B' with the required min-degrees. We can count the number of edges lost as we remove vertices in the procedure. Whenever a vertex in A' is removed we lose at most $\frac{|E|}{4|A|}$ edges and whenever a vertex from B' is removed we lose at most $\frac{|E|}{4|B|}$ edges. So

$$|E'| \geq |E| - |A| \frac{|E|}{4|A|} - |B| \frac{|E|}{4|B|} \geq |E|/2. \quad \blacktriangleleft$$

3 Proof of Theorem 4

The main technical tool will be the following lemma, which shows that one can find a vanishing polynomial of low degree, assuming each point is in many rich lines.

► **Lemma 19.** *For each $d \geq 1$ there is a constant $K_d \leq 32(2d)^d$ such that the following holds. Let $V \subset \mathbb{C}^d$ be a set of n points and let $r \geq 4$ be an integer. Suppose that, through each point $v \in V$, there are at least k r -rich lines where*

$$k \geq K_d \cdot \frac{n}{r^{d-2}}.$$

Then, there exists a non-zero polynomial $f \in \mathbb{C}[x_1, \dots, x_d]$ of degree at most $r - 2$ such that $f(v) = 0$ for all $v \in V$.

If we have the stronger condition that the number of r -rich lines through each point of V is between k and $8k$ then we can get the same conclusion (vanishing f of degree $r - 2$) under the weaker inequality

$$k \geq K_d \cdot \frac{n}{r^{d-1}}.$$

Proof. Let $V = \{v_1, \dots, v_n\}$ and let $\phi = \phi_{d,r-2} : \mathbb{C}^d \mapsto \mathbb{C}^{\mathbf{m}(d,r-2)}$ be the Veronese embedding with degree bound $r - 2$. Let us denote $U = \{u_1, \dots, u_n\} \subset \mathbb{C}^{\mathbf{m}(d,r-2)}$ with $u_i = \phi(v_i)$ for all $i \in [n]$.

We will prove the lemma by showing that U is contained in a hyperplane and then using Claim 13 to deduce the existence of the vanishing polynomial. Let M be an $n \times \mathbf{m}(d, r - 2)$ matrix whose i 'th row is $u_i = \phi(v_i)$. To show that U is contained in a hyperplane, it is enough to show that $\text{rank}(M) < \mathbf{m}(d, r - 2)$. This will imply that the columns of M are linearly dependent, which means that all the rows lie in some hyperplane.

We will now construct a design matrix A such that $A \cdot M = 0$. Since $\text{rank}(A) + \text{rank}(M) \leq n$, we will be able to translate a lower bound on the rank of A (which will be given by Theorem 12) to the required upper bound on the rank of M . Each row in A will correspond to some collinear r -tuple in V . We will construct A in several stages. First, for each r -rich line $L \in \mathcal{L}_r(V)$ we will construct a set of r -tuples $R_L \subset \binom{V}{r}$ such that

1. Each r -tuple in R_L is contained in $L \cap V$.
2. Each point $v \in L \cap V$ is in at least one and at most two r -tuples from R_L .

If $|L \cap V|$ is a multiple of r , we can construct such a set R_L easily by taking a disjoint cover of r -tuples. If $|L \cap V|$ is not a multiple of r (but is still of size at least r) we can take a maximal set of disjoint r -tuples inside it and then add to it one more r -tuple that will cover the remaining elements and will otherwise intersect only one other r -tuple. This will guarantee that each point in $L \cap V$ is in at most two r -tuples from R_L . We define $R \subset \binom{V}{r}$ to be the union of all sets R_L over all r -rich lines L . We can now prove:

► **Claim 20.** *The set $R \subset \binom{V}{r}$ defined above has the following three properties.*

1. *Each point $v \in V$ is contained in at least k r -tuples from R .*
2. *Every pair of distinct points $u, v \in V$ is contained together in at most two r -tuples from R .*
3. *Let $(v_{i_1}, \dots, v_{i_r}) \in R$. Then there exists r non-zero coefficients $\alpha_1, \dots, \alpha_r \in \mathbb{C}$ so that $\alpha_1 \cdot u_{i_1} + \dots + \alpha_r \cdot u_{i_r} = 0$.*

If, in addition, we know that each point belongs to at most $8k$ rich lines (as in the second part of the lemma) then we also have that $|R| \leq 16nk/r$.

Proof. The first property follows from the fact that each v is in at least k r -rich lines and that each R_L with $v \in L$ has at least one r -tuple containing v . The second property follows from the fact that each pair u, v can belong to at most one r -rich line L and that each R_L can contain at most two r -tuples with both u and v . The fact that the r -tuple of point u_{i_1}, \dots, u_{i_r} is linearly dependent follows from Claim 14. The fact that all the coefficients α_j are non-zero holds since no proper subset of that r -tuple is linearly dependent (by the 'moreover' part of Claim 14). If each point is in at most $8k$ lines then each point is in at most $16k$ r -tuples (at most two on each line). This means that there could be at most $16nk/r$ tuples in R since otherwise, some point would be in too many tuples. ◀

We now construct the matrix A of size $m \times n$ where $m = |R|$. For each r -tuple $(v_{i_1}, \dots, v_{i_r}) \in R$ we add a row to A (the order of the rows does not matter) that has zeros in all positions except i_1, \dots, i_r and has values $\alpha_1, \dots, \alpha_r$ given by Claim 20 in those positions. Since the rows of M are the points u_1, \dots, u_n , the third item of Claim 20 guarantees that $A \cdot M = 0$ as we wanted. The next claim asserts that A is a design matrix.

► **Claim 21.** *The matrix A constructed above is a $(r, k, 2)$ -design matrix.*

Proof. Clearly, each row of A contains at most r non-zero coordinates. Since each point $v \in V$ is in at least k r -tuples from R we have that each column of A contains at least k non-zero coordinates. The size of the intersection of the supports of two distinct columns in A is at most two by item (2) of Claim 20. ◀

We now use Eq. (1) from Theorem 12 to get

$$\text{rank}(A) \geq n - \frac{2nr^2}{k}.$$

This implies (using $r \geq 4$) that

$$\text{rank}(M) \leq \frac{2nr^2}{k} \leq \left(\frac{r-2}{d}\right)^d < \mathbf{m}(d, r-2),$$

if

$$k \geq 2(2d)^d \cdot \frac{n}{r^{d-2}}.$$

If we have the additional assumption that each point is in at most $8k$ lines then, using the bound $m = |R| \leq 16nk/r$ in Eq. (2) of Theorem 12. We get

$$\text{rank}(A) \geq n - \frac{2mr^2}{k^2} \geq n - \frac{32nr}{k}$$

which gives

$$\text{rank}(M) \leq \frac{32nr}{k} < \mathbf{m}(d, r-2)$$

for

$$k \geq 32(2d)^d \frac{n}{r^{d-1}}.$$

Hence, the rows of M lie in a hyperplane. This completes the proof of the lemma. ◀

► **Remark 22.** Lemma 19 can be extended to the case where we have r -rich curves of bounded degree $D = O(1)$ with ‘two degrees of freedom’, i.e. through every pair of points there can be at most $C = O(1)$ distinct curves (e.g. unit circles). Under the Veronese embedding $\phi_{d, \lfloor \frac{r-2}{D} \rfloor}$, the images of r points on a degree D curve are linearly dependent. So we can still construct a design matrix as in the above proof where the design parameters depend on D, C . Once we get a hypersurface of degree $\lfloor \frac{r-2}{D} \rfloor$ vanishing on all the points, the hypersurface should also contain all the degree D r -rich curves.

We will now use Lemma 19 to prove Theorem 4. The reduction uses Lemma 18 to reduce to the case where each point has many rich lines through it. Once we find a vanishing low degree polynomial we analyze its singularities to find a point such that all lines through it are in some hyperplane.

Proof of Theorem 4. Since $\mathcal{L}_r(V) \leq n^2$ for all $r \geq 2$, by choosing $C_d > R_d^d$ we can assume that $r \geq R_d$ for any large constant R_d depending only on d .

Let $\mathcal{L} = \mathcal{L}_r(V)$ be the set of r -rich lines in V and let $I = I(\mathcal{L}, V)$ be the set of incidences between \mathcal{L} and V . By the conditions of the theorem we have

$$|I| \geq r|\mathcal{L}| \geq C_d \cdot \frac{\alpha n^2}{r^{d-1}}. \tag{3}$$

Applying Lemma 18 to the incidence graph between V and \mathcal{L} , we obtain non-empty subsets $V' \subset V$ and $\mathcal{L}' \subset \mathcal{L}$ such that each $v \in V'$ is in at least $k = \frac{|I|}{4n}$ lines from \mathcal{L}' and such that each line in \mathcal{L}' is $r/4$ -rich w.r.t to the set V' and

$$|I'| = |I(\mathcal{L}', V')| \geq |I|/2.$$

We would like to apply Lemma 19 with the stronger condition that each point is incident on approximately the same number of lines (which gives better dependence on r). To achieve this, we will further refine our set of points using dyadic pigeonholing.

Let $V' = V'_1 \sqcup V'_2 \sqcup \dots$ be a partition of V' into disjoint subsets where V'_j is the set of points incident to at least $k_j = 2^{j-1}k$ and less than $2^j k$ lines from \mathcal{L}' . Let $I'_j = I(\mathcal{L}', V'_j)$, so that

$$\sum_{j \geq 1} |I'_j| = |I'| \geq |I|/2.$$

Since $\sum_{j \geq 1} \frac{1}{2^{j^2}} < 1$, there exists j such that $|I'_j| \geq \frac{|I|}{4j^2}$. Let us fix j to this value for the rest of the proof.

We will first upper bound j . Since $|I'_j| > 0$, V'_j is non-empty and let $p \in V'_j$. There are at least k_j ($r/4$)-rich lines through p and by choosing $R_d \geq 8$, there are at least $r/4 - 1 \geq r/8$ points other than p on each of these lines and they are all distinct. So,

$$n = |V| \geq 2^{j-1}k \cdot \frac{r}{8} = \frac{2^{j-6}r|I|}{n} \geq C_d \frac{2^{j-6}\alpha n}{r^{d-2}} \geq \frac{2^{j-6}n}{r^{d-2}}.$$

This implies $j \lesssim d \log r$ where we assumed above that $C_d \geq 1$.

Since the lines in \mathcal{L}' need not be $r/4$ -rich w.r.t V'_j , we need further refinement. Apply Lemma 18 again on the incidence graph $I'_j = I(\mathcal{L}', V'_j)$ to get non-empty $V'' \subset V'_j$ and $\mathcal{L}'' \subset \mathcal{L}'$ and

$$|I''| = |I(\mathcal{L}'', V'')| \geq \frac{|I'_j|}{2} \geq \frac{|I|}{8j^2} \geq \frac{r|\mathcal{L}|}{8j^2}.$$

Each line in \mathcal{L}'' is incident to at least

$$\frac{|I'_j|}{4|\mathcal{L}'|} \geq \frac{r}{16j^2} = r_0$$

points from V'' and so \mathcal{L}'' is r_0 -rich w.r.t V'' . And each point in V'' is incident to at least

$$\frac{|I'_j|}{4|V'_j|} \geq \frac{k_j}{4} = 2^{j-3}k = k_0$$

and at most $2^j k = 8k_0$ lines from \mathcal{L}'' . Since $j \lesssim d \log r$, we can assume $r_0 = \frac{r}{16j^2} \geq 4$ by choosing $R_d \gg d^3$.

The following claim shows that we can apply Lemma 19 to V'' and \mathcal{L}'' .

► **Claim 23.** $k_0 \geq K_d \cdot \frac{|V''|}{r_0^{d-1}}$ where K_d is the constant in Lemma 19.

Proof. We have

$$|V''| \leq |V'_j| \leq \frac{|I|}{2^{j-1}k} = \frac{n}{2^{j-3}}.$$

So it is enough to show that

$$k_0 \geq K_d \cdot \frac{n}{2^{j-3}r_0^{d-1}}.$$

Substituting the bounds we have for k_0 and r_0 , this will follow from

$$|I| \geq 16K_d \cdot 2^{4d} \cdot \left(\frac{j^{2(d-1)}}{2^{2j}}\right) \frac{n^2}{r^{d-1}}$$

which follows from Eq. (3) by choosing $C_d > 16K_d \cdot 2^{4d} \cdot \max_j \left(\frac{j^{2(d-1)}}{2^{2j}}\right)$. ◀

Hence, by Lemma 19, there exists a non-zero polynomial $f \in \mathbb{C}[x_1, \dots, x_d]$ of degree at most $r_0 - 2$, vanishing at all points of V'' . W.l.o.g suppose f has minimal total degree among all polynomials vanishing on V'' . Since f has degree at most $r_0 - 2$ it must vanish identically on all lines in \mathcal{L}'' .

We say that a point $v \in V''$ is ‘flat’ if the set of lines from \mathcal{L}'' passing through v are contained in some affine hyperplane through v . Otherwise, we call the point v a ‘joint’. We will show that there is at least one flat point in V'' . Suppose towards a contradiction that all points in V'' are joints. Let $v \in V''$ be some point and let $\nabla f(v)$ be the gradient of f at v . Since f vanishes identically on all lines in \mathcal{L}'' we get that $\nabla f(v) = 0$ (v is a singular point of the hypersurface defined by f). We now get a contradiction since one of the coordinates of ∇f is a non-zero polynomial of degree smaller than the degree of f that vanishes on the entire set V'' .

Hence, there exists a point $v \in V''$ and an affine hyperplane H passing through v such that all r_0 -rich lines in \mathcal{L}'' passing through v are contained in H . Since there are at least k_0 such lines, and each line contain at least $r_0 - 1$ points in addition to v , we get that H contains at least

$$(r_0 - 1)k_0 \geq \frac{r}{32j^2} \cdot 2^{j-3} \frac{|I|}{4n} \geq C_d \left(\frac{2^{j-10}}{j^2}\right) \frac{\alpha n}{r^{d-2}} \geq C'_d \frac{\alpha n}{r^{d-2}}$$

points from V where $C'_d = C_d \cdot \min_j \left(\frac{2^{j-10}}{j^2}\right)$. We can take the constants to be $C_d = d^{\Theta(d)}$ and $C'_d = \frac{C_d}{2^{11}}$. ◀

► **Remark 24.** Observe that, we can take \mathcal{L} to be any subset of $\mathcal{L}_r(V)$ of size $\geq C_d \frac{\alpha n^2}{r^d}$ and obtain the same conclusion. Moreover, the hyperplane H that we obtain at the end contains $k_0 \gtrsim \frac{\alpha n}{r^d}$ lines of \mathcal{L} .

4 Proof of Theorem 5

We will reduce the problem of bounding r -term arithmetic progressions to that of bounding r -rich lines using the following claim:

► **Claim 25.** Let $V \subset \mathbb{C}^d$ then $\mathbf{AP}_r(V) \leq |\mathcal{L}_r([r] \times V)|$ where $[r] = \{0, 1, \dots, r - 1\}$

Proof. For $u, w \in \mathbb{C}^d, w \neq 0$, let $(u, u + w, \dots, u + (r - 1)w)$ be an r -term arithmetic progression in V . Then the line $\{(0, u) + z(1, w)\}_{z \in \mathbb{C}}$ is r -rich w.r.t the point set $[r] \times V \subset \mathbb{C}^{1+d}$; moreover this mapping is injective. ◀

We need the following claim regarding arithmetic progressions in product sets.

► **Claim 26.** *Let $V \subset \mathbb{C}^d$ be a set of n points and let $\ell \geq 1$ be an integer. Then, for all $r \geq 1$, the product set $V^\ell \subset \mathbb{C}^{d\ell}$ satisfies*

$$\mathbf{AP}_r(V^\ell) \geq \mathbf{AP}_r(V)^\ell.$$

Proof. Let $P(V)$ be the set of r -term arithmetic progressions in V and let $P(V^\ell)$ be the set of r -term progressions in V^ℓ . We will describe an injective mapping from $P(V)^\ell$ into $P(V^\ell)$. For $u, w \in \mathbb{C}^d$ let $L_{u,w} = \{u, u + w, \dots, u + (r - 1)w\}$ be the r -term progression starting at u with difference w . Let $u_1, \dots, u_\ell, w_1, \dots, w_\ell \in \mathbb{C}^d$ such that $L_{u_i, w_i} \in P(V)$ for each $i \in [\ell]$. We map them into the arithmetic progression $L_{u,w} \in P(V^\ell)$ with $u = (u_1, \dots, u_\ell)$ and $w = (w_1, \dots, w_\ell)$. Clearly, this map is injective (care should be taken to assign each progression a unique difference since these are determined up to a sign). ◀

Proof of Theorem 5. Let us assume $\mathbf{AP}_r(V) \gg_{d,\epsilon} \frac{n^2}{r^{d-\epsilon}}$. Let $\ell = \lceil \frac{1}{\epsilon} \rceil$. By Claim 26, $\mathbf{AP}_r(V^\ell) \geq \mathbf{AP}_r(V)^\ell$. Let \mathcal{L} be the collection of r -rich lines w.r.t $[r] \times V^\ell \subset \mathbb{C}^{1+d\ell}$ corresponding to nontrivial r -term arithmetic progressions in V^ℓ , as given by Claim 25. So

$$|\mathcal{L}_r([r] \times V^\ell)| \geq |\mathcal{L}| = \mathbf{AP}_r(V^\ell) \geq \mathbf{AP}_r(V)^\ell \gg_{d,\epsilon} \frac{n^{2\ell}}{r^{d\ell-\epsilon\ell}} \geq \frac{n^{2\ell}}{r^{d\ell-1}} = \frac{(n^\ell r)^2}{r^{d\ell+1}}.$$

By Theorem 4 (choosing the constants appropriately), there is a hyperplane H in $\mathbb{C}^{1+d\ell}$ which contains $\gtrsim_{d,\epsilon} \frac{n^\ell r}{r^{d\ell-1}}$ points of $[r] \times V^\ell$. Moreover, by Remark 24, H contains some of the lines of \mathcal{L} . So H cannot be one of the hyperplanes $\{z_1 = i\}_{i \in [r]}$ because they do not contain any lines of \mathcal{L} . So the intersection of H with one of the r hyperplanes $\{z_1 = i\}_{i \in [r]}$ (say j) gives a $(d\ell - 1)$ -flat which contains $\gtrsim_{d,\epsilon} \frac{n^\ell}{r^{d\ell-1}}$ points of $V^\ell \times \{j\}$. This gives a hyperplane H' in $\mathbb{C}^{d\ell}$ which contains $\gtrsim_{d,\epsilon} \frac{n^\ell}{r^{d\ell-1}}$ points of V^ℓ . Now by Lemma 17, we can conclude that there is a hyperplane in \mathbb{C}^d which contains $\gtrsim_{d,\epsilon} \frac{n}{r^{d/\epsilon-1}} \geq \frac{n}{r^{2d/\epsilon-1}}$ points of V . ◀

5 Proof of Theorem 9

Suppose in contradiction that $|T_C| > \lambda N/C^2$ for some large absolute constant λ which we will choose later. Let $Q \subset T_C$ be a set of size $|Q| = \lceil \lambda N/C^2 \rceil$ containing the zero element $0 \in Q$ (we have $0 \in T_C$ since the sum-set $|A + 0A| = |A|$ is small). Let

$$r = |Q|, m = \frac{N^{1.5}}{C\sqrt{\log N}}, d = \lceil 100 \log N \rceil.$$

We will use our assumption on the size of Q to construct a configuration of points $V \subset \mathbb{C}^d$ with many r -rich lines. Then we will use Lemma 19 to derive a contradiction. The set V will be a union of the sets

$$V_t = \{t\} \times (A + tA)^{d-1} = \{(t, a_2 + tb_2, \dots, a_d + tb_d) \mid a_i, b_j \in A\}$$

over all $t \in Q$, i.e. $V = \bigcup_{t \in Q} V_t$. Notice the special structure of the set $V_0 = \{0\} \times A^{d-1}$. We denote by

$$n = |V| \leq r \cdot m^{d-1} \tag{4}$$

Notice that, by construction, for every $a = (0, a_2, \dots, a_d)$ and every $b = (1, b_2, \dots, b_d)$ (with all the a_i, b_j in A), the line through the point $a \in V_0$ in direction b is r -rich w.r.t V .

Let us denote by $\mathcal{L} \subset \mathcal{L}_r(V)$ the set of all lines of this form. We thus have $|\mathcal{L}| = N^{2d-2}$. Let $I = I(V, \mathcal{L})$, then $|I| \geq r|\mathcal{L}|$. We now use Lemma 18 to find subsets $V' \subset V$ and $\mathcal{L}' \subset \mathcal{L}$ such that each point in V' is in at least

$$k = \frac{rN^{2d-2}}{4n}$$

lines from \mathcal{L}' , each line in \mathcal{L}' is $r_0 = r/4$ -rich w.r.t to the set V' and

$$|I(V', \mathcal{L}')| \geq |I|/2.$$

Observe that, since each line in \mathcal{L}' contains at most r points from V' , we have

$$|\mathcal{L}'| \geq |I(V', \mathcal{L}')|/r \geq |\mathcal{L}|/2.$$

The following claim shows that we can apply Lemma 19 on the set V' .

► **Claim 27.**

$$k \geq K_d \frac{n}{r_0^{d-2}}.$$

where $K_d = 32(2d)^d$ is the constant in Lemma 19.

Proof. Plugging in the value of k, r_0 and using bound Eq. 4 to bound n , we need to show that

$$r^{d-3} \geq \frac{32(8d)^d N^{d-1}}{(C^2)^{d-1} (\log N)^{d-1}}.$$

We now raise both sides to the power $1/(d-3)$ and use the fact that, for $\ell > \log X$, we have $1 \leq X^{1/\ell} \leq 2$. Thus it is enough to show

$$r \geq \frac{K'dN}{C^2 \log N} = \frac{K'N \lceil 100 \log N \rceil}{C^2}$$

for some absolute constant K' which holds by choosing $\lambda = 100K'$. ◀

Since $C \ll \sqrt{N}$, $r_0 \geq 4$. Applying Lemma 19, we get a non-zero polynomial $f \in \mathbb{C}[x_1, \dots, x_d]$ of degree at most $r_0 - 2$ that vanishes on all points in V' . This means that f must also vanish identically on all lines in \mathcal{L}' (since these are all r_0 -rich w.r.t V'). Since each line in \mathcal{L}' intersects V_0 exactly once, and since $|V_0| = N^{d-1}$, we get that there must be at least one point $v \in V_0$ that is contained in at least $|\mathcal{L}'|/N^{d-1} \geq \frac{1}{2}N^{d-1}$ lines (in different directions) from \mathcal{L}' . Let \tilde{f} denote the homogeneous part of f of highest degree. If f vanishes identically on a line in direction $b \in \mathbb{C}^d$, this implies that $\tilde{f}(b) = 0$ (to see this notice that the leading coefficient of $g(t) = f(a + tb)$ is $\tilde{f}(b)$). Hence, since all the directions of lines in \mathcal{L}' are from the set $\{1\} \times A^{d-1}$, we get that \tilde{f} has at least $\frac{1}{2}N^{d-1}$ zeros in the set $\{1\} \times A^{d-1}$. This contradicts Lemma 16 since the degree of \tilde{f} is at most $r_0 - 2 = r/4 - 2 < N/2$ (since $r = \lceil \lambda N/C^2 \rceil$ and $C \gg 1$). This completes the proof of Theorem 9. ◀

Acknowledgements. We thank Ben Green and Noam Solomon for helpful comments. Research supported by NSF grant CCF-1217416 and by the Sloan fellowship. Some of the work on the paper was carried out during the special semester on ‘Algebraic Techniques for Combinatorial and Computational Geometry’, held at the Institute for Pure and Applied Mathematics (IPAM) during Spring 2014.

References

- 1 B. Barak, Z. Dvir, A. Wigderson, and A. Yehudayoff. Fractional Sylvester-Gallai theorems. *Proceedings of the National Academy of Sciences*, 2012.
- 2 B. Barak, Z. Dvir, A. Yehudayoff, and A. Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC '11, pages 519–528, New York, NY, USA, 2011. ACM.
- 3 Saugata Basu and Martin Sombra. Polynomial partitioning on varieties and point-hypersurface incidences in four dimensions. *arXiv preprint arXiv:1406.2144*, 2014.
- 4 Z. Dvir. Incidence Theorems and Their Applications. *Foundations and Trends in Theoretical Computer Science*, 6(4):257–393, 2012.
- 5 Z. Dvir, S. Saraf, and A. Wigderson. Improved rank bounds for design matrices and a new proof of Kelly’s theorem. *Forum of Mathematics, Sigma*, 2, 10 2014.
- 6 L. Guth and N. Katz. Algebraic methods in discrete analogs of the Kakeya problem. *Advances in Mathematics*, 225(5):2828 – 2839, 2010.
- 7 Larry Guth and Nets Hawk Katz. On the Erdős distinct distances problem in the plane. *Annals of Mathematics*, 181(1):155–190, 2015.
- 8 Marton Hablicsek and Zachary Scherr. On the number of rich lines in high dimensional real vector spaces. *arXiv preprint arXiv:1412.7025*, 2014.
- 9 J. Kollar. Szemerédi–Trotter-type theorems in dimension 3. *arXiv:1405.2243*, 2014.
- 10 Izabella Laba and József Solymosi. Incidence theorems for pseudoflats. *Discrete & Computational Geometry*, 37(2):163–174, 2007.
- 11 M. Rudnev. On the number of incidences between planes and points in three dimensions. *arXiv:1407.0426v2*, 2014.
- 12 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.
- 13 Micha Sharir, Adam Sheffer, and Joshua Zahl. Improved bounds for incidences between points and circles. In *Proceedings of the Twenty-ninth Annual Symposium on Computational Geometry*, SoCG '13, pages 97–106, New York, NY, USA, 2013. ACM.
- 14 Micha Sharir and Noam Solomon. Incidences between points and lines in \mathbb{R}^4 . *arXiv preprint arXiv:1411.0777*, 2014.
- 15 József Solymosi and VH Vu. Distinct distances in high dimensional homogeneous sets. *Contemporary Mathematics*, 342:259–268, 2004.
- 16 József Solymosi and Terence Tao. An incidence theorem in higher dimensions. *Discrete and Computational Geometry*, 48(2):255–280, 2012.
- 17 E. Szemerédi and W. T. Trotter. Extremal problems in discrete geometry. *Combinatorica*, 3(3):381–392, 1983.
- 18 C. Toth. The Szemerédi-Trotter theorem in the complex plane. *arXiv:math/0305283v4*, 2003.
- 19 Joshua Zahl. A Szemerédi-Trotter type theorem in \mathbb{R}^4 . *CoRR*, abs/1203.4600, 2012.
- 20 Joshua Zahl. A note on rich lines in truly high dimensional sets. *arXiv preprint arXiv:1503.01729*, 2015.
- 21 R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 216–226. Springer-Verlag, 1979.

Realization Spaces of Arrangements of Convex Bodies*

Michael Gene Dobbins¹, Andreas Holmsen², and Alfredo Hubard³

1 GAIA, POSTECH
South Korea
dobbins@postech.ac.kr

2 Department of Mathematical Sciences, KAIST
South Korea
andreash@kaist.edu

3 GEOMETRICA, INRIA Sophia-Antipolis
France
alfredo.hubard@inria.fr

Abstract

We introduce *combinatorial types* of arrangements of convex bodies, extending order types of point sets to arrangements of convex bodies, and study their realization spaces. Our main results witness a trade-off between the combinatorial complexity of the bodies and the topological complexity of their realization space. On one hand, we show that every combinatorial type can be realized by an arrangement of convex bodies and (under mild assumptions) its realization space is contractible. On the other hand, we prove a universality theorem that says that the restriction of the realization space to arrangements of convex polygons with a bounded number of vertices can have the homotopy type of any primary semialgebraic set.

1998 ACM Subject Classification G.2.1 Combinatorics , F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Oriented matroids, Convex sets, Realization spaces, Mnev’s universality theorem

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.599

1 Introduction

We introduce a generalization of order types that provides a framework to study arrangements of convex sets and their convex dependencies. The notion we introduce is closely related to wiring diagrams [7] or primitive sorting networks [18]. It is also related to double pseudoline arrangements introduced by Pocchiola and Habert [14] and double allowable sequences introduced by Goodman and Pollack [11]. These related notions have applications in the study of visibility, transversal, and separation properties of convex sets [2, 23, 22, 16]. The generalization of order type studied here was fundamental to the authors’ work on

* M. G. Dobbins was supported by NRF grant 2011-0030044 (SRC-GAIA) funded by the government of South Korea. A. Holmsen was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (NRF-2010-0021048). A. Hubard was supported by Fondation Sciences Mathématiques de Paris and by the Advanced Grant of the European Research Council GUDHI (Geometric Understanding in Higher Dimensions).



generalizations of the Erdős-Szekeres theorem to arrangements of convex sets in the plane [4, 3]. In this paper, we address the relevant realizability questions.¹

Two indexed point sets $P = \{p_1, p_2 \dots p_n\}$ and $Q = \{q_1, q_2 \dots q_n\}$ in the plane are said to have the same order type when for every triple (i, j, k) the orientation of the triples p_i, p_j, p_k and q_i, q_j, q_k coincides. Equivalently by projective duality, a point set P corresponds to a dual family P^* of oriented great circles in the sphere, and point sets P and Q have the same order type when the families P^* and Q^* subdivide the sphere in the same way. That is, when there is a self-homeomorphism of the sphere that sends each cell of P^* to a corresponding cell of Q^* and preserves orientations. A number of geometric properties of point sets which are important in combinatorial convexity, incidence geometry and algorithms depend solely on the order type of the point sets and not on actual coordinates of the points.

More generally, we say that a sign function $\chi : \mathfrak{L}^3 \rightarrow \{+, 0, -\}$ is an **order type** when it satisfies the axioms of rank 3 acyclic chirotopes [1, page 126] [18, Chapter 10]. Specifically, χ is alternating, satisfies the Grassman-Plücker Relations, is acyclic (a restatement of Radon's partition theorem in terms of orientations), and is not identically zero. Order types that satisfy $\chi(i, j, k) \neq 0$ for any i, j, k distinct are called **simple**, and are equivalent to Donald Knuth's CC-systems [18].

Like simple order types, combinatorial types are finite combinatorial objects that can be associated to families of geometric objects, namely arrangements of convex bodies, which are assumed to satisfy certain genericity conditions. These will be defined precisely in Section 2, but for now we describe the equivalence relation that combinatorial types induce on arrangements of convex bodies. To do so, we define a duality for convex bodies that is analogous to projective duality for points in the plane. The **dual support curve** A^* of a convex body A in the plane, is the graph of its support function $h_A : \mathbb{S}^1 \rightarrow \mathbb{R}^1$, $h_A(\theta) := \max_{p \in A} \langle \theta, p \rangle$ on the cylinder $\mathbb{S}^1 \times \mathbb{R}^1$, where \mathbb{S}^1 is the unit circle and $\langle \cdot, \cdot \rangle$ is the standard inner product. In this way, every arrangement $\mathcal{A} = \{A_1, \dots, A_n\}$ corresponds to the **dual support system** $\mathcal{A}^* = \{A_1^*, \dots, A_n^*\}$ of curves on the cylinder $\mathbb{S}^1 \times \mathbb{R}^1$ given by the graphs of the functions $\{h_{A_1}, h_{A_2}, \dots, h_{A_n}\}$. In the other direction, not all functions $h : \mathbb{S}^1 \rightarrow \mathbb{R}^1$ are support functions, but we have the following sufficient conditions.

► **Remark.** Blaschke showed that if $h : \mathbb{S}^1 \rightarrow \mathbb{R}^1$ is C^2 -smooth and $h + h'' > 0$, then h is the support function of a planar curve with curvature bounded by $\frac{1}{h+h''}$ [12, Lemma 2.2.3]. Hence, by adding a sufficiently large constant to a family of smooth functions, we can ensure the family is the dual support system of an arrangement of convex bodies.

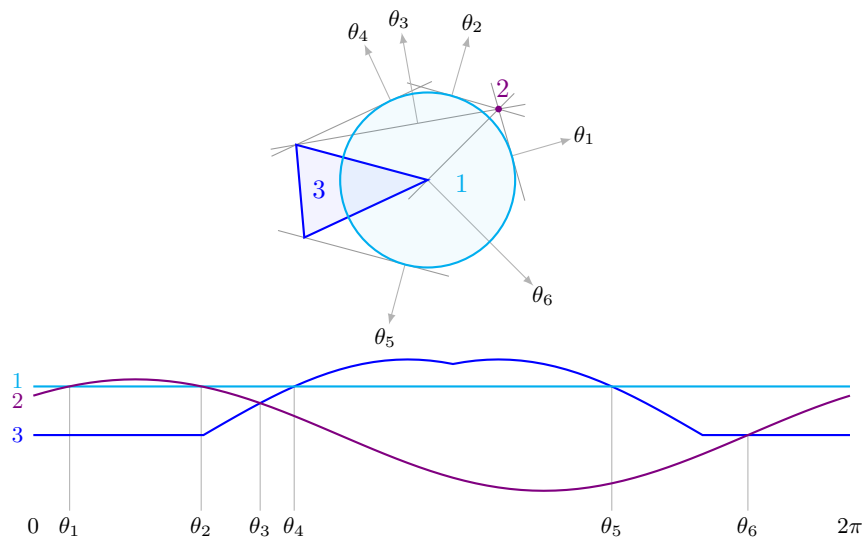
The combinatorial type of an arrangement of bodies $\text{ct}(\mathcal{A})$ is, essentially, a combinatorial encoding of the subdivision of the cylinder $\mathbb{S}^1 \times \mathbb{R}^1$ by the dual support curves \mathcal{A}^* . For now, we take the following theorem as an alternative topological definition.

► **Theorem 1.** *Two arrangements of convex bodies \mathcal{A} and \mathcal{B} have the same combinatorial type if and only if their dual systems \mathcal{A}^* and \mathcal{B}^* are related by a self-homeomorphism of the cylinder that preserves orientation and $+\infty$.*

Here, we say that a self-homeomorphism $\phi : \mathbb{S}^1 \times \mathbb{R}^1 \rightarrow \mathbb{S}^1 \times \mathbb{R}^1$ preserves $+\infty$ when for y sufficiently large the second coordinate of $\phi(\theta, y)$ is positive for all θ . Equivalently, ϕ preserves the counter-clockwise orientations of the support curves.

In the case of points, the duality that we defined through support functions is the usual projective duality renormalized to be on the cylinder. Consequently, two generic point sets

¹ The full details of the proofs of our results will appear in the journal version.



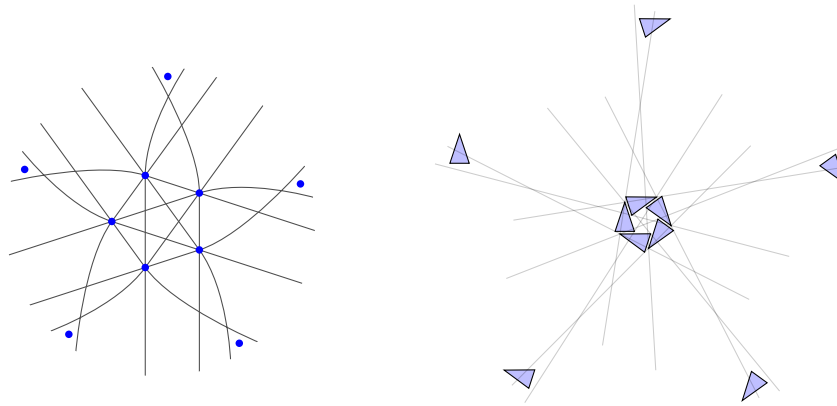
■ **Figure 1 Top:** An arrangement \mathcal{A} and its common supporting tangents. **Bottom:** Its dual support system \mathcal{A}^* .

have the same order type if and only if they have the same combinatorial type. Specifically, a point in the plane can be represented in homogeneous coordinates by a line in \mathbb{R}^3 , and its dual support curve is the intersection of the orthogonal complement of this line with the cylinder embedded in \mathbb{R}^3 . The same relationship holds between a body in the plane represented by a cone in \mathbb{R}^3 and the body’s dual support curve represented by its polar cone.

Although combinatorial types of arrangements are more general objects than simple order types, we associate an order type to the following class of arrangements. We say a triple of bodies is *orientable* when it has the combinatorial type of three generic points, and we say an arrangement is orientable when it consists of at least three bodies and every triple is orientable. In this case, every triple $\{A_i, A_j, A_k\} \subset \mathcal{A}$ contributes a single connected arc to the boundary of $\text{conv}(A_i, A_j, A_k)$, and we define the *orientation* of an ordered triple (A_i, A_j, A_k) to be positive when the bodies appear counter-clockwise in the given order on the boundary, and to be negative otherwise. Grünbaum implicitly observes that the cyclic orderings of the triples of an orientable arrangement form an order type in his discussion on planar arrangements of simple closed curves [13, Section 3.3].

1.1 Realizing order types

Not every order type can be realized by a point set. In fact, most order types are not realizable, and it is NP-hard to decide if a given order type is realizable [28]. Having a notion of combinatorial type allows us to approach questions regarding realizability by bodies rather than points [15]. The smallest *non-realizable* order type is the Non-Pappus Configuration, a configuration of 9 elements that violates Pappus’s Theorem [19, 27]. Pach and Tóth showed that the Non-Pappus Configuration can be realized by an arrangement of segments in the plane [24]. Figure 2 shows a non-realizable order type that can be realized by triangles, Goodman-Pollack’s “Bad Pentagon” [8], and the authors conjecture that this order type cannot be realized by segments. In contrast to point sets, we show that any order type, and in fact any combinatorial type, can be realized by an arrangement of bodies.



■ **Figure 2** Two realizations of the Bad Pentagon. **Left:** a realization in a topological plane [8]. **Right:** a realization by convex sets in the Euclidean plane.

► **Theorem 2.** *The orientations of the triples of any orientable arrangement is a simple order type. Two orientable arrangements have the same order type if and only if they have the same combinatorial type. And, every simple order type can be realized by an orientable arrangement.*

We informally describe how to construct an arrangement of a given simple order type.

Proof Idea. The Folkman-Lawrence representation theorem says that any rank 3 chirotope can be realized by a (symmetric) pseudocircle arrangement; that is, by a family of simple closed curves on the sphere such that each curve is preserved by the antipodal map ($x \mapsto -x$) and each pair of distinct curves intersect at exactly 2 points [6]. In the case of order types, there is some pair of points respectively to the left of each curve (labeled $+\infty$) and to the right of each curve (labeled $-\infty$). Pseudocircle arrangements can be swept, meaning a path from the point $-\infty$ to the point $+\infty$ can be continuously deformed while keeping its end-points fixed such that it passes through all other points on the sphere exactly once returning to its original position and it always intersects each pseudocircle at exactly one point [9, Theorem 2.9]. This defines a homeomorphism from the sphere with points $+\infty$ and $-\infty$ removed to the cylinder such that the image of each pseudocircle is the graph of a function $h_i : \mathbb{S}^1 \rightarrow \mathbb{R}^1$. Furthermore, this homeomorphism can be chosen so that each function h_i is the support function of a convex body. These convex bodies then form an orientable arrangement of the given order type.

Alternatively, such an arrangement can be constructed in the primal. A simple order type can be encoded by a sequence of permutations given by the order that a path intersects each pseudocircle as it sweeps the sphere. Consider a family of closed curves in the plane that each wind once around the origin and cross according to this sequence of permutations. By drawing these curves sufficiently close to the unit circle so that each curve is the boundary of a convex body, we obtain an orientable arrangement of the given order type. ◀

If we bound the complexity of the bodies, then some simple order types can no longer be realized. Indeed, we show that simple order types can always be realized by k -gons, but may require k to be arbitrarily large.

► **Theorem 3.** *Let k_n be the smallest integer for which every simple order type on n elements can be realized by an arrangement of k_n -gons. There are constants $c_1, c_2 > 0$ such that*

$$c_1 \frac{n}{\log n} \leq k_n \leq c_2 n^2.$$

Proof Idea. The primal construction of Theorem 2 that realizes a simple order type by convex bodies can be done so the resulting bodies are polygons, and this gives the upper bound on k_n . We get the lower bound on k_n by the following counting argument. Fix an integer k . The combinatorial type of an arrangement of n k -gons is determined by the order type of all kn vertices of the arrangement. Therefore, the number of combinatorial types that can be realized by n k -gons is at most the number of order types that can be realized by kn points $2^{O(kn(\log(n)+\log(k)))}$, which grows more slowly than the number of simple order types on n elements $2^{\Theta(n^2)}$ [10, 5]. Thus, for n sufficiently large, some simple order type cannot be realized by k -gons, so $k \leq k_n$. ◀

1.2 Realization spaces

An old conjecture of Ringel claimed that given two point sets with the same order type, one point set can be continuously deformed to the other while maintaining the order type [27]. This naturally leads to the study of *realization spaces* of order types, the set of all families of points with a fixed order type modulo projectivities. The conjecture can then be restated as, any non-empty realization space is connected. Ringel's conjecture was disproved in the early eighties, and the strongest result in this direction is Mnev's Universality Theorem [20, 21], which states that for any primary semialgebraic set \mathcal{Z} , there exists an order type whose realization space is homotopy equivalent to \mathcal{Z} . Recall that a primary semialgebraic set is the set of common solutions to some finite list of polynomial equations and strict inequalities in several real variables. This has led to a growing body of work [1, 17, 25, 26, 29, 30].

The main objective of this paper is to extend the study of realization spaces to arrangements of bodies of a fixed combinatorial type and exhibit a trade-off between the combinatorial complexity of the bodies and the topological complexity of their realization space. The first indication of this trade-off may be observed from Theorems 2 and 3, which imply that for general convex bodies the realization space of any simple order type is non-empty, but this fails for k -gons. We prove two contrasting results. First, we show in Theorem 4 that Ringel's intuition is correct in this generalized context: the realization space of any combinatorial type satisfying some mild assumptions is contractible; that is, it is non-empty and has no holes of any dimension. In particular, the set of arrangements (modulo planar rotations) of convex bodies realizing any fixed simple order type is contractible, and therefore connected. Second, we show in Theorem 5 that if the bodies are restricted to polygons with at most k vertices, then Mnev's Theorem generalizes.² Specifically, we show that for every k and every primary semialgebraic set \mathcal{Z} there is a combinatorial type whose k -gon realization space is homotopy equivalent to \mathcal{Z} . The main ideas of the proof of Theorem 4 are given in Section 3 and the construction for Theorem 5 is given in Section 4.

1.3 Relationship to double pseudoline arrangements

Pocchiola and Habert introduced an extension of chirotopes to arrangements of convex sets based on a similar notion of duality to what is presented here, called double pseudoline

² Note that Mnev's Theorem is more specific as it deals with stable equivalence.

arrangements [14]. The essential difference is that the dual double pseudoline of a convex set is defined as the quotient of the dual support curve by the \mathbb{Z}_2 action on the cylinder $(\theta, y) \sim (\theta, -y)$. Instead of a curve that wraps monotonically once around the cylinder, the dual double pseudoline is a curve that wraps monotonically twice around the Möbius strip. This leads to an extended notation of chirotopes that provides information about arrangements of convex sets which combinatorial types do not distinguish, such as disjointness and visibility. On the other hand, combinatorial types distinguish convex position of subarrangements and are simpler in certain respects that are crucial to the analysis in [4, 3] and the results of this paper.

2 Preliminaries and main theorems

In this section we state the main theorems and introduce definitions to be used throughout the paper. An *arrangement* we always mean a *finite indexed* non-empty collection of compact convex sets, which we call bodies.

2.1 Genericity assumptions

A *common supporting tangent* of a pair of bodies is a *directed line* tangent to each body such that both bodies are on its *left* side. In the dual, this corresponds to an intersection between two support curves. We say that a pair of bodies intersect *transversally* when no point of intersection is contained in a common supporting tangent. In the dual this corresponds to a pair of curves in the cylinder that cross at each point of intersection; that is, for a pair of curves that are respectively the graphs of functions f_1, f_2 , the function $f_1 - f_2$ has only isolated zeros and changes sign at each zero. An arrangement is called *generic* when it satisfies the following conditions:

- Each pair of bodies intersect transversally.
- No three bodies share a common supporting tangent.
- There are finitely many common supporting tangents.

A system is called *generic* when it satisfies the following conditions:

- Each pair of curves cross at each point of intersection.
- No three curves share a common point of intersection.
- There are finitely many crossings.

Every time we refer to an arrangement or a system, it is assumed to be generic. We will use non-generic point sets and their non-simple order types, but we do not refer to them as arrangements.

2.2 Combinatorial type

Let \mathfrak{S}_m be the symmetric group on m elements and $[m] = \{1, \dots, m\}$. Given $i \in [m-1]$, the *adjacent transposition* $\tau_i \in \mathfrak{S}_m$ is the permutation interchanging the i 'th and $i+1$ 'st entries,

$$\tau_i(x_1, \dots, x_m) = (x_1, \dots, x_{i-1}, x_{i+1}, x_i, x_{i+2}, \dots, x_m).$$

Let $H(\tau_i) = i$ denote the height of an adjacent transposition. A *swap sequence* $\sigma : [N] \rightarrow \mathfrak{S}_m$ is any sequence of adjacent transpositions such that $\sigma_N \circ \dots \circ \sigma_1$ is the identity permutation.

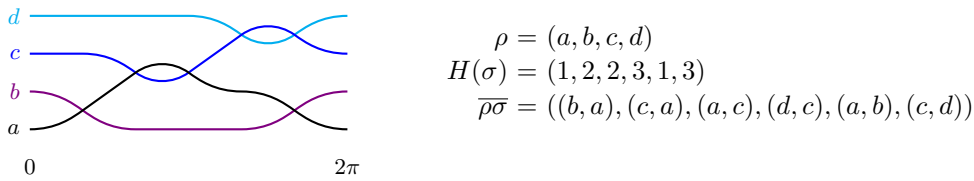


Figure 3 A system with its swap data (ρ, σ) and its incidence sequence $\overline{\rho\sigma}$. Note that systems are drawn as viewed from outside the cylinder, so counter-clockwise is to the right.

Fix an index set \mathcal{L} of size n . A *swap pair* (ρ, σ) on \mathcal{L} is a bijection $\rho : [n] \rightarrow \mathcal{L}$ together with a swap sequence $\sigma : [N] \rightarrow \mathfrak{S}_n$. We define an equivalence relation (\simeq^{swap}) on swap pairs as follows. Let $(\rho', \sigma') \simeq^{\text{swap}} (\rho, \sigma)$ when (ρ', σ') can be obtained from (ρ, σ) by performing any sequence of the following two *elementary operations*

- a *cyclic shift*

$$\rho' = \tau_{\sigma_1}(\rho), \quad \sigma'_i = \sigma_{(i+1 \bmod N)}$$

- an *elementary transposition*

$$\rho' = \rho, \quad \sigma' = \tau_i(\sigma) \quad \text{provided} \quad |H(\sigma_i) - H(\sigma_{i+1})| > 1.$$

A *combinatorial type* Ω on \mathcal{L} is the equivalence class $\Omega = \{(\rho', \sigma') : (\rho', \sigma') \simeq^{\text{swap}} (\rho, \sigma)\}$ of a swap pair (ρ, σ) .

To define the combinatorial type of a system \mathcal{S} , we order the crossings of \mathcal{S} lexicographically in $\mathbb{S}^1 \times \mathbb{R}^1$ where \mathbb{S}^1 is ordered according to the standard parametrization by the half-open interval $(0, 2\pi]$. Let ρ be the order of the indices of each curve from bottom to top before the first crossing of the system. Let σ be the swap sequence corresponding to each crossing. That is, let $H(\sigma_i)$ be 1 plus the number of curves below the i 'th crossing of \mathcal{S} . Observe that the sequence $\sigma_i \circ \dots \circ \sigma_1(\rho)$ for $i = 0, 1, \dots, N$ records the order of the curves in a sweep of the cylinder. The *combinatorial type* $\text{ct}(\mathcal{S})$ of a system \mathcal{S} is the equivalence class of its swap pair (ρ, σ) . The combinatorial type of an arrangement \mathcal{A} is that of its dual system, and by slight abuse of notation, we write $\text{ct}(\mathcal{A}) = \text{ct}(\mathcal{A}^*)$.

The *incidence sequence* $\overline{\rho\sigma} : [N] \rightarrow \mathcal{L}^2$ of a swap pair (ρ, σ) records the ordered pair of indices transposed by the action of each swap,

$$\overline{\rho\sigma}_i = (x_{H(\sigma_i)+1}, x_{H(\sigma_i)}) \quad \text{where} \quad x = \sigma_{i-1} \circ \dots \circ \sigma_1(\rho).$$

Note that the incidence sequence of equivalent swap pairs have the same multi-set of entries.

The *layers* of a system are the connected components of the union of curves of the system. Analogously, the layers of a combinatorial type are the connected components of the graph on \mathcal{L} defined by its incidence sequence. The *depth* of a combinatorial type is the number of layers excluding isolated vertices, and the depth 1 case is called *non-layered*.

2.3 Realization spaces

The *full realization space* $\mathcal{R}(\Omega)$ of a combinatorial type Ω is defined by

$$\mathcal{R}(\Omega) := \{\mathcal{A} \in \mathcal{U}_{\mathcal{L}} : \text{ct}(\mathcal{A}) = \Omega\}$$

where $\mathcal{U}_{\mathcal{L}}$ is the set of all arrangements of bodies indexed by \mathcal{L} . The Hausdorff metric d_H on compact subsets of \mathbb{R}^2 induces a metric on $\mathcal{R}(\Omega)$ by taking the maximum distance between

bodies having the same index. That is, for $\mathcal{A} = \{A_i\}_{i \in \mathcal{L}}$ and $\mathcal{B} = \{B_i\}_{i \in \mathcal{L}}$,

$$d(\mathcal{A}, \mathcal{B}) = \max_{i \in \mathcal{L}} d_H(A_i, B_i)$$

► **Remark.** The map that takes a convex body to its support function is an isometry from the space of convex bodies with the Hausdorff metric to the space of support functions on \mathbb{S}^1 with the supremum metric.

Depending on context, it may be convenient to regard realizations of a fixed combinatorial type as “the same” when they are related by a projective transformation. Let $\mathcal{A} \overset{\text{proj}}{\sim} \mathcal{B}$ when they are related by an admissible projectivity; that is, an invertible projective transformation π such that $\pi(A_i) = B_i$ for all $i \in \mathcal{L}$ and π is bounded and preserves orientation on the convex hull of $\bigcup \mathcal{A}$. The *(projective) realization space*, which we may simply call the “realization space”, is the quotient space

$$\tilde{\mathcal{R}}(\Omega) := \mathcal{R}(\Omega) / \overset{\text{proj}}{\sim}.$$

By a k -gon we mean a convex polygon with *at most* k vertices. The *full k -gon realization space* is given by

$$\mathcal{R}_k(\Omega) := \{\mathcal{A} \in \mathcal{R}(\Omega) : \forall i \in \mathcal{L}. A_i \text{ is a } k\text{-gon}\}$$

Similarly, we have the *(projective) k -gon realization space* $\tilde{\mathcal{R}}_k(\Omega) := \mathcal{R}_k(\Omega) / \overset{\text{proj}}{\sim}$. Let $\mathbb{T}^d = \mathbb{S}^1 \times \dots \times \mathbb{S}^1$ denote the d -torus, the d -fold product of \mathbb{S}^1 .

► **Theorem 4.** *The realization space $\tilde{\mathcal{R}}(\Omega)$ of any non-layered combinatorial type Ω is contractible. Moreover, if Ω has depth $d > 1$, then $\tilde{\mathcal{R}}(\Omega)$ is homotopic to a $(d-1)$ -torus.*

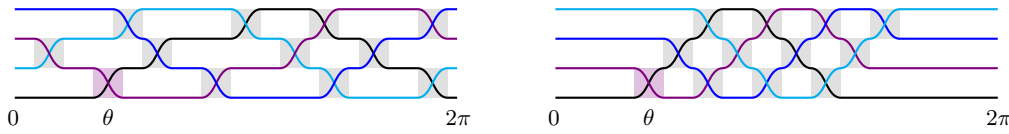
► **Remark.** Orientable combinatorial types are always non-layered.

► **Theorem 5.** *For every primary semialgebraic set \mathcal{Z} and every positive integer k , there exists a combinatorial type Ω such that $\tilde{\mathcal{R}}_k(\Omega)$ is homotopic to \mathcal{Z} .*

3 Contractibility

To show contractibility, we construct a standard arrangement of convex bodies for each combinatorial type by defining its dual support system. We then show that the full realization space $\mathcal{R}(\Omega)$ is equivariantly homotopic to a circle \mathbb{S}^1 by defining a deformation retraction to the subspace of rotated copies of the standard arrangement. By *equivariantly* homotopic we mean that the corresponding homotopy maps commute with $SO(2)$. We then pass to the (projective) realization space $\tilde{\mathcal{R}}(\Omega)$ by identifying arrangements related by admissible projectivities. Since rotations are admissible projectivities, this defines a deformation retraction from $\tilde{\mathcal{R}}(\Omega)$ to a point.

The deformation retraction from $\mathcal{R}(\Omega)$ to a circle will proceed in two steps. First in Lemma 6, we deform the support system of a given arrangement to a system of the same combinatorial type that depends only on the (angular) position of each crossing; see Figure 4 Left. We can then consider just the positions of the crossings and ignore the rest of the geometry of the system. Second in Lemma 7, we move the crossings to a set of standard positions that depend only on: the given combinatorial type and the position of a certain crossing that we fix; see Figure 4 Right. The set of possible standard systems we get in the end is parametrized by the position of this fixed crossing, which defines an embedding of the circle in $\mathcal{R}(\Omega)$. The first deformation retraction depends on the following remark.



■ **Figure 4 Left:** The system $\alpha^*(V)$ depending on the angular positions of the crossings as given by the support configuration $V \in \mathcal{V}(\Omega)$. **Right:** The system $W(\theta, \delta)$ with the marked crossing fixed at angle θ obtained by rotating all other crossings of $\alpha^*(V)$ clockwise.

► **Remark.** For any pair of convex bodies A and B , $(A + B)^* = A^* + B^*$ with Minkowski addition on the left and addition of the support functions defining the curves on the right. And, for $t \geq 0$, $(tA)^* = t(A^*)$. Hence, the set of all support functions is a convex cone. That is, if g and h are support functions, then so is $sg + th$ for $s, t \geq 0$. Note however, that the set of dual support systems of a fixed combinatorial type is not a convex set.

3.1 Support configurations

The **support configuration** of an arrangement \mathcal{A} indexed by \mathfrak{L} is a labeled vector configuration $\text{sc}(\mathcal{A}) \subset \mathfrak{L}^2 \times \mathbb{S}^1$ which contains a triple (i, j, θ) if bodies A_i, A_j have a common supporting tangent line ℓ that first meets A_i and then meets A_j and has outward normal vector θ . We say labels $(i, j), (i', j')$ are **disjoint** when $\{i, j\} \cap \{i', j'\} = \emptyset$. Note that a unit vector θ may appear in multiple elements of $\text{sc}(\mathcal{A})$ with disjoint labels. Dually, $\text{sc}(\mathcal{A})$ corresponds to the crossings of \mathcal{A}^* . Specifically, $(i, j, \theta) \in \text{sc}(\mathcal{A})$ when curves A_i^* and A_j^* cross at θ with A_i^* crossing downward and A_j^* crossing upward. That is, the respective support functions f_i, f_j of A_i, A_j are equal at θ and $f_j - f_i$ is increasing at θ .

Observe that the support configuration of an arrangement determines the combinatorial type of that arrangement. For a given combinatorial type Ω , we will define its **support configuration space** $\mathcal{V}(\Omega)$, which will turn out to be the set of support configurations of all arrangements realizing Ω . We first define the set of labeled configurations $\mathcal{V}(\rho, \sigma)$ corresponding to a given swap pair (ρ, σ) . Recall that $\overline{\rho\sigma}$ records the ordered pairs of indices transposed by σ acting sequentially on ρ . Observe that if (ρ, σ) is the swap pair of a system, then $\overline{\rho\sigma}_i$ for $i \in [N]$ is the labeling of the i -th crossing of the system. Recall also that we order \mathbb{S}^1 by the parametrization by $(0, 2\pi]$. Let

$$\mathcal{V}(\rho, \sigma) := \left\{ \{(\overline{\rho\sigma}_i, \theta_i) : i \in [N]\} : \begin{array}{l} \theta_i \in \mathbb{S}^1, \theta_i \leq \theta_{i+1}, \\ \theta_i = \theta_{i+1} \Rightarrow |H(\sigma_i) - H(\sigma_{i+1})| > 1 \end{array} \right\},$$

$$\mathcal{V}(\Omega) := \bigcup_{(\rho, \sigma) \in \Omega} \mathcal{V}(\rho, \sigma).$$

Note that a vector $\theta \in \mathbb{S}^1$ might appear multiple times in $\mathcal{V}(\Omega)$ with different labels provided the pairs of indices in the labels are disjoint.

We define a metric on $\mathcal{V}(\Omega)$ as follows. For a given support configuration X and a given ordered pair of indices $(i, j) \in \binom{\mathfrak{L}}{2}$, let $X_{(i,j)} := \{\theta \in \mathbb{S}^1 : (i, j, \theta) \in X\}$. For two support configurations, $X, Y \subset \mathfrak{L}^2 \times \mathbb{S}^1$,

$$d(X, Y) = \max_{(i,j) \in \binom{\mathfrak{L}}{2}} d_H(X_{(i,j)}, Y_{(i,j)})$$

where the distance between two direction vectors is given by their angle and d_H is the corresponding Hausdorff metric on sets.

► **Lemma 6.** *For any combinatorial type Ω , the full realization space $\mathcal{R}(\Omega)$ is non-empty and equivariantly homotopic to the support configuration space $\mathcal{V}(\Omega)$.*

Proof Idea. For $\mathcal{A} \in \mathcal{R}(\Omega)$ with swap pair (ρ, σ) , we have $\text{sc}(\mathcal{A}) \in \mathcal{V}(\rho, \sigma) \subset \mathcal{V}(\Omega)$, so assigning each arrangement to its support configuration defines a map $\text{sc}: \mathcal{R}(\Omega) \rightarrow \mathcal{V}(\Omega)$, which will be one direction of the homotopy equivalence.

For the other direction, we define an embedding $\alpha: \mathcal{V}(\Omega) \rightarrow \mathcal{R}(\Omega)$. For each labeled configuration $V \in \mathcal{V}(\Omega)$, we construct a system of curves $\alpha^*(V) = \{A_i^* : i \in \mathcal{L}\}$ where $A_i^* = f_i(\mathbb{S}^1)$, $f_i: \mathbb{S}^1 \rightarrow \mathbb{R}^1$, and show that $\alpha^*(V)$ is the dual support system of an arrangement $\alpha(V)$ that has support configuration V . The system $\alpha^*(V)$ that we construct may be regarded as a smooth analog of Goodman's wiring diagram [7].

Fix $V \in \mathcal{V}(\Omega)$, let $V_i \subset \mathbb{S}^1$ denote the vectors of V with labels involving i , and let δ be the minimum angular distance between any two vectors of V with non-disjoint labels. For $v = (i, j, \theta) \in V$ define the open arc $\Theta(v) := (\theta - \delta/2, \theta + \delta/2) \subset \mathbb{S}^1$. Now define f_i to be constant on the complement of the arcs $\Theta(V_i)$, and to smoothly increase or decrease by ± 1 symmetrically about θ in each arc $\Theta(v)$ according to the label on $v \in V_i$; that is, f_i increases on $\Theta(v)$ if $(j, i, \theta) \in V$ and decreases if $(i, j, \theta) \in V$ for some j .³ We can additionally require each pair f_i, f_j to coincide on $V_i \cap V_j$, and this determines each subfamily of $\alpha^*(V)$ corresponding to a layer of Ω up to a common additive constant. A proof of this is given in the journal version. To fix this additive constant in the case of one layer, let

$$\min_{(i, \theta) \in \mathcal{L} \times \mathbb{S}^1} (f_i(\theta) + f_i''(\theta)) = 1,$$

and in the case of multiple layers, let the minimum of each successively higher layer be greater than the maximum of the layer immediately below by 1. Now the system $\alpha^*(V)$ defined by the functions f_i is the dual support system of an arrangement $\alpha(V) \in \mathcal{R}(\Omega)$ that is uniquely and continuously determined by $V \in \mathcal{V}(\Omega)$, and $\text{sc}(\alpha(V)) = V$. This gives us a subspace $\alpha(\Omega) := \{\alpha(V) : V \in \mathcal{V}(\Omega)\} \subset \mathcal{R}(\Omega)$ that is homeomorphic to $\mathcal{V}(\Omega) = \text{sc}(\mathcal{R}(\Omega))$. For $\mathcal{A} \in \mathcal{R}(\Omega)$ define $\mathcal{A}_t := t\alpha(\text{sc}(\mathcal{A})) + (1-t)\mathcal{A}$ for $0 \leq t \leq 1$ by Minkowski addition on each body of the arrangement. Since $\text{sc}(\mathcal{A}) = \text{sc}(\alpha(\text{sc}(\mathcal{A})))$ and, as we linearly interpolate between two systems with the same crossings, the crossings remain fixed, $\text{sc}(\mathcal{A}_t)$ is constant for all $t \in [0, 1]$. Thus, $\alpha(\Omega)$ is an equivariant deformation retract of $\mathcal{R}(\Omega)$. ◀

3.2 Local sequences and standard configurations

We define a deformation retraction from the support configuration space $\mathcal{V}(\Omega)$ to a subspace of standard configurations $\mathcal{W}(\Omega) \subset \mathcal{V}(\Omega)$, which is homeomorphic to a torus. The standard configuration we choose is similar to the “compressed form” given in [18, page 31]. For this, we introduce an encoding of combinatorial type extending the local sequences of a point set. The **local sequence** $\lambda_i = (\lambda_{i,1}, \dots, \lambda_{i,n_i})$ of $i \in \mathcal{L}$ for a system \mathcal{S} lists the indices of the curves that S_i crosses in order according to the parametrization by $(0, 2\pi]$. Similarly, the local sequence λ_i for a swap pair (ρ, σ) is the sequence of indices $\lambda_{i,j}$ appearing together with i as part of a pair $(\lambda_{i,j}, i)$ or $(i, \lambda_{i,j})$ in the incidence sequence $\overline{\rho\sigma}$. Let Λ denote the tableau that has $\lambda_{\rho(i)}$ as its i th row. We say Λ is a **tableau representation** of the combinatorial type $\Omega \ni (\rho, \sigma)$. We say Ω is **periodic** when for some $p > 1$, some tableau representation Λ

³ The definition of f_i on $\Theta(v)$ is irrelevant as long as f_i is C^2 -smooth, monotonic, symmetric about θ , and varies continuously with respect to V . A cubic spline would suffice for this.

is the row-wise concatenation of p copies of some other tableau Λ' . We say a pair $j, k \in \mathcal{L}$ are **adjacent** in a tableau Λ with rows λ_i when

$$\lambda_j = (k, \lambda_{j,2}, \dots, \lambda_{j,n_j}) \quad \text{and} \quad \lambda_k = (j, \lambda_{k,2}, \dots, \lambda_{k,n_k}).$$

► **Lemma 7.** *For any non-layered combinatorial type Ω , the support configuration space $\mathcal{V}(\Omega)$ is equivariantly homotopic to the circle \mathbb{S}^1 .*

Proof Idea. Assume Ω is non-periodic. The periodic case is similar, and is dealt with in the journal version. We first construct a labeled vector configuration $W(\theta, \delta)$ for $\theta \in \mathbb{S}^1$ and $\delta > 0$ sufficiently small as follows. Let Λ_{\min} be the lexicographically minimal tableau representation of Ω for which there exists exactly one adjacent pair. Note that it is always possible to find a tableau representation of a non-layered combinatorial type with exactly one adjacent pair. We will define a sequence of configurations W_t recursively starting from $t = 0$. To start, set $\Lambda_0 = \Lambda_{\min}$, $\theta_0 = \theta$, $W_0 = \emptyset$. Let $\{(i_{t,1}, j_{t,1}), \dots, (i_{t,m_t}, j_{t,m_t})\}$ be the set of all adjacent pairs in Λ_t where $(i_{t,m}, j_{t,m})$ is ordered according to the row order of Λ_t . Let

$$W_{t+1} = W_t \cup \{(i_{t,1}, j_{t,1}, \theta_t), \dots, (i_{t,m_t}, j_{t,m_t}, \theta_t)\},$$

$\theta_{t+1} = \theta_t + \delta$, and let Λ_{t+1} be the tableau obtained from Λ_t by interchanging the corresponding pairs of rows and deleting the first entry from each of these rows. Eventually, $\Lambda_T = \emptyset$ for some minimal T , and we obtain a support configuration $W(\theta, \delta)$ of combinatorial type Ω where the minimum angular distance between vectors with non-disjoint labels is δ . Finally, let $\mathcal{W}(\Omega) = \{W(\theta, \delta_0) : \theta \in \mathbb{S}^1\}$ where $\delta_0 = 2\pi/N$.

The unique adjacent pair of Λ_{\min} corresponds to a specific labeled vector $\tilde{v} = (i, j, \theta)$ in each configuration $V \in \mathcal{V}(\Omega)$. To define a deformation retraction sending V to a configuration $W \in \mathcal{W}(\Omega)$, first fix \tilde{v} and rotate each of the other vectors clockwise as much as possible without decreasing the minimum distance δ between vectors with non-disjoint labels. That is, rotated each vector $x \neq \tilde{v}$ clockwise until there is a vector y at angular distance δ in the clockwise direction from x such that y has already stopped rotating and x and y have non-disjoint labels. Once all vectors have stopped rotating, we will have arrived at the configuration $W(\theta, \delta)$. Then, continuously rescale δ to $2\pi/N$ keeping \tilde{v} fixed. This gives a deformation retraction from $\mathcal{V}(\Omega)$ to $\mathcal{W}(\Omega) \simeq \mathbb{S}^1$. ◀

3.3 Proof of contractibility

Proof of Theorem 4. In the depth 1 case, the full realization space $\mathcal{R}(\Omega)$ is homotopic to the support configuration space $\mathcal{V}(\Omega)$ by Lemma 6, which is homotopic to \mathbb{S}^1 by Lemma 7. Since these homotopies are equivariant and rotations are included among admissible projective transformations, the realization space $\tilde{\mathcal{R}}(\Omega)$ is contractible.

In the depth $d > 1$ case, partition Ω into layers $\Omega = \Omega_1 \cup \dots \cup \Omega_d$. If we restrict a support configuration of Ω to those vectors with labels in a layer Ω_i , then we obtain a support configuration of Ω_i . Hence, $\mathcal{V}(\Omega) \subset \mathcal{V}(\Omega_1) \times \dots \times \mathcal{V}(\Omega_d)$. In the other direction, if we are given support configurations $V_i \in \mathcal{V}(\Omega_i)$, then $\bigcup_{i \in [d]} V_i \in \mathcal{V}(\Omega)$. Hence $\mathcal{V}(\Omega) = \mathcal{V}(\Omega_1) \times \dots \times \mathcal{V}(\Omega_d)$, and therefore by Lemmas 6 and 7, $\mathcal{R}(\Omega)$ is homotopic to \mathbb{T}^d , so the realization space $\tilde{\mathcal{R}}(\Omega)$ is homotopic to a $(d-1)$ -torus \mathbb{T}^{d-1} . ◀

4 Universality

We prove the following slightly more specific result.

► **Lemma 8.** *For any k order types χ_1, \dots, χ_k on $[n]$, where at least two are distinct, there is a combinatorial type Ω on $[n]$ such that its k -gon realization space $\tilde{\mathcal{R}}_k(\Omega)$ is homotopy equivalent to $\tilde{\mathcal{R}}_1(\chi_1) \times \dots \times \tilde{\mathcal{R}}_1(\chi_k)$.*

Proof of Theorem 5. Fix a primary semialgebraic set \mathcal{Z} and $k > 1$. Let χ_1 be the order type of the Mnev point set with point realization space homotopic to \mathcal{Z} . Let χ_2, \dots, χ_k all be the order type of n points in convex position. Note that the point realization space of n points in convex position is contractible. With this, the k -gon realization space of Ω from Lemma 8 is also homotopic to \mathcal{Z} . ◀

To show Lemma 8, we construct a combinatorial type Ω such that for every realization \mathcal{A} of Ω by k -gons, the vertices of each k -gon can be labeled. That is, each vertex can be uniquely identified using only information encoded in the combinatorial type. Note that this is not possible in general, as combinatorial type does not provide information about individual vertices directly. Furthermore, we construct Ω so that the order type of the vertices of \mathcal{A} is the same in every realization and each χ_i appears as a subset of the vertices.

We define Ω in two ways: in the primal we construct an arrangement of k -gons, then in the dual we construct a system of curves. First index the order types χ_i so that the cyclic ordering $\chi_1, \chi_2, \dots, \chi_k, \chi_1, \dots$ is not periodic with period smaller than k . This is possible by the assumption that there are at least two distinct order types.

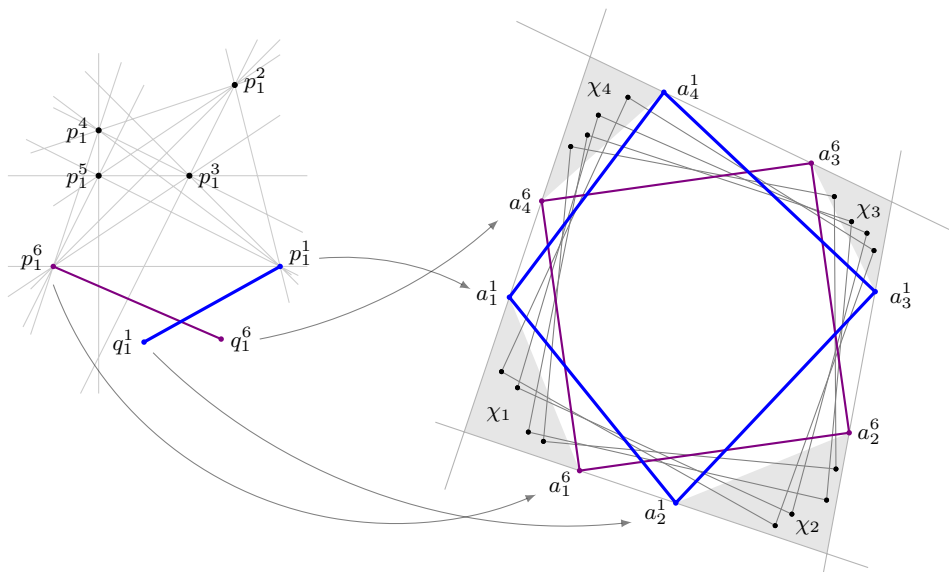
4.1 The primal construction

The primal construction $\mathcal{A} = \{A^1, \dots, A^n\}$ depends on certain arbitrary choices that will not affect the combinatorial type. Assume for the primal construction that each χ_i is realizable; the non-realizable case is defined by the dual construction only.

Let \mathcal{A}_0 be an arrangement of $2k$ points in convex position denoted by $a_1^1, a_1^n, a_2^1, a_2^n, \dots, a_k^1, a_k^n$ in counter-clockwise order, such that the lines ℓ_i spanning a_i^n and a_{i+1}^1 bound a convex k -gon B .⁴ Observe that $B \setminus \text{conv}(\mathcal{A}_0)$ consists of k triangular regions. We construct \mathcal{A} by placing a point set realizing one of the χ_i in each of these triangular regions, then we define the k -gons A^s to have vertices consisting of one point from each region; see Figure 5 for an example with $n = 6, k = 4$.

Let χ_i be defined on the index set $\{\binom{1}{i}, \dots, \binom{n}{i}\}$, and let $\mathcal{P}_i = \{p_i^1, \dots, p_i^n\}$ be a realization of χ_i . Furthermore, let χ_i indexed so that p_i^1 and p_i^2 appear on the boundary of the convex hull of \mathcal{P}_i and the local sequence of p_i^1 is $p_i^2, p_i^3, \dots, p_i^n$. That is, the angles θ_i^s at p_i^1 from the semiline through p_i^2 to the semiline through p_i^s are increasing in the counter-clockwise direction, $0 = \theta_i^2 < \theta_i^3 < \dots < \theta_i^n < \pi$. Note that this implies p_i^n is also on the boundary of the convex hull of \mathcal{P}_i , which we will call the **convex boundary** for short. Now augment \mathcal{P}_i by two points as follows. Let $\mathcal{Q}_i = \mathcal{P}_i \cup \{q_i^1, q_i^n\}$ such that $p_i^n, q_i^1, q_i^n, p_i^1$ appear consecutively in counter-clockwise order on the convex boundary of \mathcal{Q}_i and no line through any two points of \mathcal{P}_i separates the points q_i^1, q_i^n, p_i^1 . Note that this uniquely determines the order type of \mathcal{Q}_i ; see Figure 5 (left). Now define projective transformations ϕ_i such that $\phi_i(q_i^n) = a_{i-1}^n, \phi_i(p_i^1) = a_i^1, \phi_i(p_i^n) = a_i^n, \phi_i(q_i^1) = a_{i+1}^1$, and let $\mathcal{P} = \{a_1^1, a_1^2, \dots, a_2^1, \dots, a_n^1, \dots, a_n^n\}$ where $a_i^s = \phi_i(p_i^s)$. Finally, let $\mathcal{A} = \{A^1, \dots, A^n\}$ where $A^s = \text{conv}(\{a_1^s, a_2^s, \dots, a_k^s\})$, and let Ω denote the combinatorial type of \mathcal{A} .

⁴ Here subscripts are indices over \mathbb{Z}_k , so in particular ℓ_k is the line spanning a_k^n and a_1^1 .



■ **Figure 5** The point set \mathcal{P}_1 on the left is mapped to points on the right by the projective transformation determined by $p_1^1 \mapsto a_1^1$, $q_1^1 \mapsto a_2^1$, $p_1^6 \mapsto a_1^6$, $q_1^6 \mapsto a_4^6$

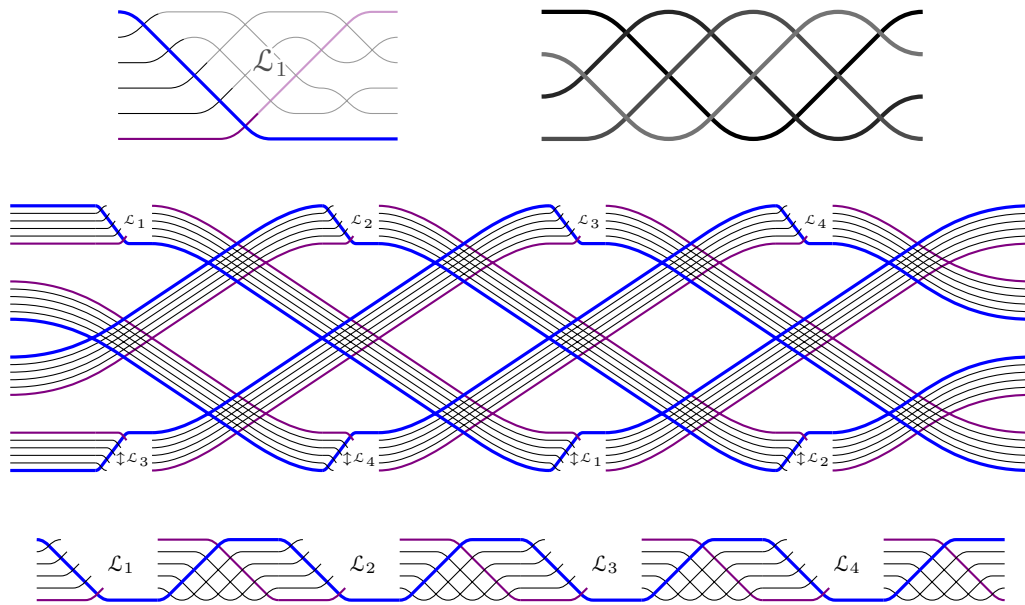
4.2 Path systems

We call the graph of some indexed family of functions defined over an interval a *path system*. We say two path systems are *equivalent* when they are related by an orientation preserving self-homeomorphism of the plane that also preserves indices and the orientations of the paths. We will always assume that the end-points of a path system are all distinct, and that the paths satisfy the same genericity conditions given in Subsection 2.1 for systems of curves. For path systems $\mathcal{L}_1, \mathcal{L}_2$ over intervals $I_1, I_2 \subset \mathbb{R}$ with the same number of paths, the concatenation $\mathcal{L}_1 \cdot \mathcal{L}_2$ is the path system obtained by identifying the right edge of $I_1 \times \mathbb{R}$ with the left edge of $I_2 \times \mathbb{R}$ by a homeomorphism sending the right end-points of \mathcal{L}_1 to the left end-points of \mathcal{L}_2 . Here indices must be dealt with appropriately. If the left end-points of \mathcal{L}_1 appear in the same order as the right end-points, then we may form a system of curves $\circlearrowleft \mathcal{L}_1$ by identifying the left and right edges of $I_1 \times \mathbb{R}$ by a homeomorphism that identifies the left and right end-points of each path in \mathcal{L}_1 . Let $\updownarrow \mathcal{L}_1$ denote the path system obtained by flipping \mathcal{L}_1 vertically by the map $(x, y) \mapsto (x, -y)$. Given an order type χ , we say a path system \mathcal{L} is a *pseudoline representation* of χ when $\mathcal{S} = \circlearrowleft(\mathcal{L} \cdot \updownarrow \mathcal{L})$ is an orientable system with order type χ as in Theorem 2. We say an element i is on the *convex boundary* of χ when the corresponding curve S_i appears on the upper envelope of a corresponding system \mathcal{S} .

► **Remark.** For each element i on the convex boundary of an order type χ , there is a unique class of equivalent pseudoline representations \mathcal{L} where L_i starts as the top most path and crosses all other paths, thereby going to the bottom, before any other crossings occur.

4.3 The dual construction

Let χ_i be an order type on elements $\{\binom{1}{i}, \dots, \binom{n}{i}\}$ indexed as in the primal construction, and let \mathcal{L}_i be a pseudoline representation of χ_i with paths L_i^1, \dots, L_i^n such that L_i^1 starts at the top and crosses all other paths first as in the above remark. Let $\mathcal{C} = \{C_1, \dots, C_k\}$ be the dual system of k points in convex position indexed in counter-clockwise order, and observe that each curve C_i appears exactly once on the upper envelope and once on the



■ **Figure 6** **Top left:** The pseudoline representation \mathcal{L}_1 of χ_1 . **Top right:** The system \mathcal{C} . **Center:** The system \mathcal{S} . **Bottom:** The system \mathcal{T} of combinatorial type Ω .

lower envelope of \mathcal{C} . Let \mathcal{S} be a system of curves where each curve $C_i \in \mathcal{C}$ is replaced by n curves $\{S_i^1, \dots, S_i^n\}$ in a small tubular neighborhood about C_i crossing to form a copy of \mathcal{L}_i above all other curves of \mathcal{S} and a copy of $\downarrow \mathcal{L}_i$ below all other curves of \mathcal{S} . Let T^s be the upper envelope of the curves S_1^s, \dots, S_k^s , and let $\mathcal{T} = \{T^1, \dots, T^n\}$. Equivalently, let \mathcal{U} be the path system of size n where each path from bottom to top crosses all paths below itself (beginning with the bottom path crossing no other paths and ending with the top path crossing all other paths), and let $\mathcal{T} = \circ(\mathcal{L}_1 \cdot \mathcal{U} \cdot \mathcal{L}_2 \cdot \mathcal{U} \cdots \mathcal{L}_k \cdot \mathcal{U})$. Finally, define Ω to be the combinatorial type of \mathcal{T} . See Figure 6 for an example with $n = 6$, $k = 4$.

Proof Idea of Lemma 8. Each body A^t for $t \in \{2, \dots, n\}$ appears k times on the convex boundary of $\{A^1, A^t\}$, so A^1 and A^t must each have exactly k vertices. In this way, the vertices of \mathcal{A} can be partitioned into k parts consisting of one vertex from each A^s for $s \in [n]$, and these parts realize each χ_i in the given cyclic order. Since the sequence of χ_i is not periodic with period smaller than k , each part is associated to each χ_i in a unique way. This defines a map $\varphi : \tilde{\mathcal{R}}_k(\Omega) \rightarrow \tilde{\mathcal{R}}_1(\chi_i)$ such that $\varphi_1 \times \cdots \times \varphi_k : \tilde{\mathcal{R}}_k(\Omega) \rightarrow \tilde{\mathcal{R}}_1(\chi_1) \times \cdots \times \tilde{\mathcal{R}}_1(\chi_k)$ is a fiber bundle with contractible fiber, so $\tilde{\mathcal{R}}_k(\Omega)$ is homotopic to $\tilde{\mathcal{R}}_1(\chi_1) \times \cdots \times \tilde{\mathcal{R}}_1(\chi_k)$. ◀

References

- 1 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, 1999.
- 2 Raghavan Dhandapani, Jacob E. Goodman, Andreas Holmsen, and Richard Pollack. Interval sequences and the combinatorial encoding of planar families of pairwise disjoint convex sets. *Rev. Roum. Math. Pures Appl.*, 50(5-6):537–553, 2005.
- 3 Michael Gene Dobbins, Andreas Holmsen, and Alfredo Hubbard. Regular systems of paths and families of convex sets in convex position. To appear in *Transactions of the AMS*.

- 4 Michael Gene Dobbins, Andreas Holmsen, and Alfredo Hubard. The Erdős-Szekeres problem for non-crossing convex sets. *Mathematika*, 60(2):463–484, 2014.
- 5 Stefan Felsner and Pavel Valtr. Coding and counting arrangements of pseudolines. *Discrete & Computational Geometry*, 46(3):405–416, 2011.
- 6 Jon Folkman and Jim Lawrence. Oriented matroids. *Journal of Combinatorial Theory, Series B*, 25(2):199–236, 1978.
- 7 Jacob E. Goodman. Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Mathematics*, 32(1):27–35, 1980.
- 8 Jacob E. Goodman and Richard Pollack. On the combinatorial classification of nondegenerate configurations in the plane. *Journal of Combinatorial Theory, Series A*, 29(2):220–235, 1980.
- 9 Jacob E. Goodman and Richard Pollack. Semispaces of configurations, cell complexes of arrangements. *Journal of Combinatorial Theory, Series A*, 37(3):257–293, 1984.
- 10 Jacob E. Goodman and Richard Pollack. Upper bounds for configurations and polytopes in \mathbb{R}^d . *Discrete & Computational Geometry*, 1(1):219–227, 1986.
- 11 Jacob E. Goodman and Richard Pollack. The combinatorial encoding of disjoint convex sets in the plane. *Combinatorica*, 28(1):69–81, 2008.
- 12 Helmut Groemer. *Geometric applications of Fourier series and spherical harmonics*, volume 61 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1996.
- 13 Branko Grünbaum. *Arrangements and spreads*. American Mathematical Society, 1972.
- 14 Luc Habert and Michel Pocchiola. Arrangements of double pseudolines. In *Proceedings of the 25th Annual Symposium on Computational Geometry*, pages 314–323. ACM, 2009.
- 15 Alfredo Hubard. Erdős-Szekeres para convexos. Bachelor’s thesis, UNAM, 2005.
- 16 Alfredo Hubard, Luis Montejano, Emiliano Mora, and Andrew Suk. Order types of convex bodies. *Order*, 28(1):121–130, 2011.
- 17 Michael Kapovich and John J. Millson. Universality theorems for configuration spaces of planar linkages. *Topology*, 41(6):1051–1107, 2002.
- 18 Donald E. Knuth. *Axioms and hulls*, volume 606 of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.
- 19 Friedrich Levi. Die teilung der projektiven ebene durch gerade oder pseudogerade. *Ber. Math.-Phys. Kl. Sächs. Akad. Wiss.*, 78:256–267, 1926.
- 20 Nicolai E. Mnev. Varieties of combinatorial types of projective configurations and convex polytopes. *Doklady Akademii Nauk SSSR*, 283(6):1312–1314, 1985.
- 21 Nicolai E. Mnev. 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.
- 22 Mordechai Novick. Allowable interval sequences and line transversals in the plane. *Discrete & Computational Geometry*, 48(4):1058–1073, 2012.
- 23 Mordechai Novick. Allowable interval sequences and separating convex sets in the plane. *Discrete & Computational Geometry*, 47(2):378–392, 2012.
- 24 János Pach and Géza Tóth. Families of convex sets not representable by points. In *Algorithms, architectures and information systems security*, volume 3, page 43. World Scientific, 2008.
- 25 Arnau Padrol and Louis Theran. Delaunay triangulations with disconnected realization spaces. In *Proceedings of the 30th Annual Symposium on Computational Geometry*, pages 163–170. ACM, 2014.
- 26 Jürgen Richter-Gebert. *Realization spaces of polytopes*, volume 1643 of *Lecture Notes in Mathematics*. Springer Verlag, 1996.

- 27 Gerhard Ringel. Teilungen der ebene durch geraden oder topologische geraden. *Mathematische Zeitschrift*, 64(1):79–102, 1956.
- 28 Peter W. Shor. Stretchability of pseudolines is NP-hard. In *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4, pages 531–554. American Mathematical Society, 1991.
- 29 Yasuyuki Tsukamoto. New examples of oriented matroids with disconnected realization spaces. *Discrete & Computational Geometry*, 49(2):287–295, 2013.
- 30 Ravi Vakil. Murphy’s law in algebraic geometry: badly-behaved deformation spaces. *Inventiones Mathematicae*, 164(3):569–590, 2006.

Computing Teichmüller Maps between Polygons

Mayank Goswami¹, Xianfeng Gu², Vamsi P. Pingali³, and Gaurish Telang⁴

- 1 Algorithms and Complexity, Max-Planck Institute for Informatics
Saarbrücken 66123, Germany
gmayank@mpi-inf.mpg.de
- 2 Department of Computer Science, Stony Brook University
Stony Brook, NY 11794-4400, USA
gu@cs.stonybrook.edu
- 3 Department of Mathematics, Johns Hopkins University
Baltimore, MD - 21218, USA
vpingali@math.jhu.edu
- 4 Department of Applied Mathematics and Statistics, Stony Brook University
Stony Brook, NY 11794-3600, USA
gaurish.telang@stonybrook.edu

Abstract

By the Riemann mapping theorem, one can bijectively map the *interior* of an n -gon P to that of another n -gon Q conformally (i.e., in an angle preserving manner). However, when this map is extended to the boundary it need not necessarily map the *vertices* of P to those of Q . For many applications it is important to find the “best” vertex-preserving mapping between two polygons, i.e., one that minimizes the maximum angle distortion (the so-called dilatation). Such maps exist, are unique, and are known as extremal quasiconformal maps or Teichmüller maps.

There are many efficient ways to approximate conformal maps, and the recent breakthrough result by Bishop computes a $(1 + \epsilon)$ -approximation of the Riemann map in linear time. However, only heuristics have been studied in the case of Teichmüller maps.

We present two results in this paper. One studies the problem in the continuous setting and another in the discrete setting.

In the continuous setting, we solve the problem of finding a finite time procedure for approximating Teichmüller maps. Our construction is via an iterative procedure that is proven to converge in $O(\text{poly}(1/\epsilon))$ iterations to a $(1 + \epsilon)$ -approximation of the Teichmüller map. Our method uses a reduction of the polygon mapping problem to the marked sphere problem, thus solving a more general problem.

In the discrete setting, we reduce the problem of finding an approximation algorithm for computing Teichmüller maps to two basic subroutines, namely, computing discrete 1) compositions and 2) inverses of discretely represented quasiconformal maps. Assuming finite-time solvers for these subroutines we provide a $(1 + \epsilon)$ -approximation algorithm.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Teichmüller maps, Surface registration, Extremal Quasiconformal maps, Computer vision

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.615

1 Introduction

A foundational result in complex analysis, the Riemann mapping theorem, implies that the interiors of two n -gons P and Q can be mapped bijectively and conformally (i.e., in an angle



© Mayank Goswami, Vamsi P. Pingali, Xianfeng Gu, and Gaurish Telang;
licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 615–629



Leibniz International Proceedings in Informatics

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

preserving way¹) to one another. By a result of Caratheodory [4], such a map $f : P \rightarrow Q$ extends continuously to the boundary of P (the edges). Generally the vertices of P do not map to the vertices of Q under this extended mapping.

Consider the collection of functions f that map P to Q , and take the vertices of P to the vertices of Q . In general such an f is bound to stretch angles, and a classical way to measure this angle stretch by f at a point $p \in P$ is by means of a complex-valued function $\mu_f(p)$ called the Beltrami coefficient² of f . The Beltrami coefficient satisfies $\|\mu_f\|_\infty < 1$. If μ_f is identically zero, then f is conformal. The problem we consider is computing the “best” such map f_* in the above class, i.e., an f_* such that the norm of its Beltrami coefficient $\|\mu_{f_*}\|_\infty$ is the smallest amongst all (uncountably many) maps satisfying the above conditions. Bijective maps that stretch angles but by a bounded amount are called quasiconformal homeomorphisms (q.c.h.), and the best q.c.h. f_* is called the extremal q.c. map, or the Teichmüller map.

As an example consider two rectangles $R_i = [0, a_i] \times [0, b_i] (i = 1, 2)$ in the plane. Consider the space of all q.c.h. $f : R_1 \rightarrow R_2$ such that f takes the vertices to the vertices. It was shown by Grötzsch [12] that the affine map $f_*(x, y) = (a_2x/a_1, b_2y/b_1)$ with $\mu_*(x, y) = (1 - r)/(1 + r), r = b_2a_1/a_2b_1$ is the unique extremal q.c. map; any other map f would stretch angles at some point $p \in R_1$ more than g (i.e., $\exists p \in R_1 : |\mu_f(p)| > |\mu_*(p)|$). For the general n -gon case mentioned above, such a nice formula does not exist for the extremal map. However, the extremal map exists and is unique. These are the famous theorems of Teichmüller [22, 23], proven rigorously later by Ahlfors [1].

Algorithms for computing the Riemann map from a polygon to the disc [8, 7, 2] have gathered a lot of attention and found many applications. However, no such algorithm that approximates the extremal map is known. In contrast to the Riemann mapping theorem, where a constructive proof is known, the proof by Teichmüller/Ahlfors is an existence result only. In fact, to the authors’ knowledge there does not exist a method that, given a starting f between P and Q , computes a g with $\|\mu_g\|_\infty < \|\mu_f\|_\infty$ if one exists. We are motivated by the following question.

Question: Does there exist a finite-time approximation algorithm for computing the Teichmüller map between two n -gons?

We give the first results for theoretically constructing and algorithmically computing Teichmüller maps for the polygon problem above. Our procedure is iterative; we start with a q.c.h. that sends the vertices of P to those of Q in the prescribed order, improve on it, and then recurse on the improved map.

The need for an algorithm. Conformal geometry has found many applications in the fields of computer graphics [14], computer vision [24] and medical imaging [25, 13]. Computing Teichmüller maps generalizes almost all of these applications as q.c. maps allow boundary values to be prescribed. In [26], it was concluded that extremal q.c. maps have almost all the properties desired from an ideal surface registration algorithm, one of the biggest problems in computer vision.

¹ A homeomorphism f is angle preserving if it preserves oriented angles between curves: For any two curves γ_1 and γ_2 through a point p and oriented angle θ between them, $f(\gamma_1)$ and $f(\gamma_2)$ intersect at $f(p)$ at angle θ .

² For a function f between open sets in the complex plane \mathbb{C} , $\mu_f = \frac{f_{\bar{z}}}{f_z} = \frac{f_x + if_y}{f_x - if_y}$, where f_x and f_y denote partials w.r.t x and y , respectively.

An algorithm for computing Teichmüller maps would be a step forward in examining various questions in pure mathematics too. In [3] the author proposes how an algorithm for our problem would help us attack one of the most famous conjectures in geometric function theory – Brennan’s conjecture. Teichmüller theory is an active area of research in mathematics, and it has connections to topology³, dynamics, algebraic geometry, and number theory [15]. An algorithm for our problem helps one compute and visualize geodesics in the so-called Teichmüller space (w.r.t. Teichmüller’s metric), which may be of independent interest.

Related work. Almost all algorithms in computational q.c. geometry have appeared mainly in graphics or vision venues. In many works (e.g. [19]) a q.c.h. is represented by its Beltrami coefficient, and softwares implementing basic subroutines (e.g. solving the Beltrami equation) in computational q.c. geometry have existed for some time.

The first paper addressing the problem of computing extremal q.c. maps was [26]. The authors propose an interesting heuristic based on Teichmüller’s characterization; they formulate an energy function and minimize it using an alternate-descent method. Simulations showed that if the initial map is chosen correctly, the algorithm converges in many instances. Unfortunately, the energy obtained is “highly nonlinear” and non-convex. Even in the absence of numerical errors due to discretization, it is not known whether the minimization procedure converges to an approximation of the extremal map.

In [17] another heuristic was proposed using the connection to the theory of harmonic maps. This was simulated on a variety of examples and in many instances ended up with a good answer. However, no convergence proofs (continuous or discrete settings) were provided. Recently, in [18] it is argued that a procedure similar to that in [17] converges in the limit if certain parameters are chosen carefully manually. However, there are no bounds on the progress made in a step, and therefore it is not known if the procedure (even in the continuous setting) ends with an approximation in finite time.

Results. In comparison to all the previous work, we take a theoretical approach to constructing an algorithm for Teichmüller maps. In the continuous setting we have a procedure (Theorem 8) that converges in the limit to the exact extremal map and we also give bounds on the progress made in each step. Using this we can show that our procedure always, no matter what the starting map, gives an arbitrarily good approximation of the desired map in a finite number of iterations. A salient feature of our analysis is that we do not use an energy-based approach and work directly with the dilatation (the maximum angle stretch).

In the discrete setting, we state precisely all the subroutines needed for our algorithm and provide approximation guarantees. We present a novel subroutine *INF-EXT* that produces a type of Beltrami coefficient fundamental in the study of extremal maps, and prove (Theorem 9) that it produces an arbitrarily good approximation. We give error bounds on the discrete algorithm we propose, and show that (Theorem 10) modulo two basic subroutines⁴, our algorithm produces a $(1 + \epsilon)$ -approximation of the extremal map.

³ It had been used by Lipman Bers to give a simpler proof of Thurston’s classification theorem for surface homeomorphisms.

⁴ It is indeed surprising that tasks as basic as composing two q.c.h. (specified by their piecewise constant Beltrami coefficient), or finding the inverse of one, cannot be accomplished correctly yet. These two subroutines have been implemented in the past various times without error bounds, and as of now no approximation algorithms exist for them.

Because of space constraints in this extended abstract, all of our complete proofs can be found in the full version [11].

2 Informal discussion of results and techniques

As mentioned in the introduction, the aim is to compute the extremal q.c.h between two polygons. Intuitively, if μ_f is the Beltrami coefficient of f , f maps an infinitesimally small circle around p to something that roughly looks like a small ellipse at $f(p)$, with $(1 + |\mu_f(p)|)/(1 - |\mu_f(p)|)$ as the ratio between its major and minor axes.

Our strategy to tackle the polygon mapping problem is to first reduce it to the marked sphere problem. The marked sphere problem is: Given a q.c.h. f_0 from the sphere to itself taking a collection of given points (z_k) to another collection (w_k) , compute the unique extremal q.c.h. f_* that not only takes z_k to w_k (for all k) but is also isotopic to f_0 (i.e. it can be “continuously deformed” to f_0 after pinning the values at z_k). We first prove that a solution to the marked sphere problem gives a solution to the polygon mapping problem (Theorem 7). For future reference we also note that the complex plane can be thought of as the sphere minus the north pole (the point at infinity).

Representation and complexity. In theory, a normalized q.c.h. f can be specified by specifying μ_f . For computational purposes, unless a closed form expression for f_* or μ_* is available, the best one can do is to evaluate f_* or μ_* on a dense mesh of points inside the domain. Our goal can be stated as follows.

Goal: Given a $\delta > 0$, compute the values of f_* on a given set of points inside the base polygon P , where the Beltrami coefficient μ_f of f satisfies $\|\mu_f\|_\infty < \|\mu_*\|_\infty + \delta$.

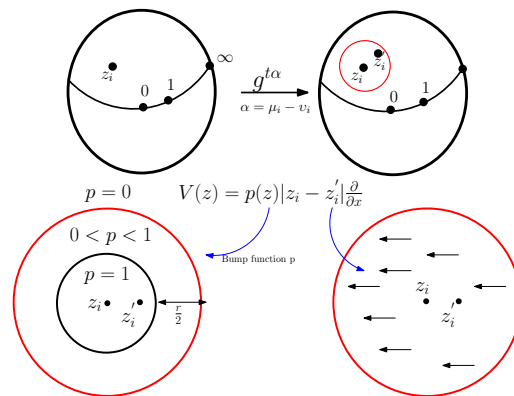
Even if the polygons P and Q have rational coordinates, there is no known way to represent the extremal map with finite precision (for all we know, all representations may consist of transcendental numbers). In fact, we have found examples where this is true even for the Schwarz-Christoffel mapping⁵. Thus, it is not known whether the problem is in NP or not. We therefore straightaway aim towards an approximation algorithm. The model we consider is a real RAM model, where we are allowed to do exact basic arithmetic operations and take logarithms of complex numbers in constant time.

2.1 Continuous construction

One of our main results is constructing a sequence of q.c.h. f_i (which can all be continuously deformed to the starting map f_0) that converge to the desired extremal q.c.h. quickly (to get within ϵ of the extremal one we need $O(1/\epsilon^4)$ iterations). The map f_{i+1} is obtained from the previous one f_i by a composition $f_{i+1} = f_i \circ h_{i+1}$ where h_{i+1} fixes all the z_k and is obtained from f_i by convex optimization and solving (partial and ordinary) differential equations.

The main innovation in our approach is to “search” for the “best” map indirectly in a sense. One important result [1] in Teichmüller theory is the following : Given a complex-valued function $\mu_f(p)$ such that $\|\mu_f\|_\infty < 1$ there is an essentially unique q.c.h. f such that $\mu_f(p)$ is its “angle-stretch”. In other words, the q.c.h. are “indexed” by their Beltrami coefficients.

⁵ The Schwarz-Christoffel mapping is the “explicit” formula for the conformal map from the upper half plane \mathbb{H} to a polygon, and, by composition, a formula for the conformal map between two polygons



■ **Figure 1** Construction of the self map h_i . Left: The map $g^{t\alpha}$ moves z_i to a point z_i' within $O(t^2)$. Right: The disk of radius $r = O(t^2)$ enlarged, showing the bump function p and the direction of the flow of the vector field.

One recovers f from μ_f by solving a partial differential equation called the *Beltrami equation*, $f_{\bar{z}}/f_z = \mu$.

Given the Beltrami coefficient μ_i of f_i , we search for the best (least L^∞ norm) Beltrami coefficient v_i satisfying a certain technical condition called “infinitesimal equivalence” (Definition 3). This essentially boils down to a convex optimization problem. For a small $t > 0$, the q.c.h. g_i corresponding to the Beltrami coefficient $t(\mu_i - v_i)$ almost fixes the (z_k) . It moves them only slightly (Figure 1 left). Then we correct for this motion by flowing the images $z_k' = g_i(z_k)$ back to (z_k) by solving a system of ordinary differential equations using a vector field, shown in the right side of Figure 1. We then compose these two maps.

This way, we get a map h_{i+1} which fixes the points (z_k) . Moreover, we can prove that $f_{i+1} = f_i \circ h_{i+1}$ has a smaller maximum angle-stretch (i.e., smaller dilatation) than f_i . We iterate this process to converge to an arbitrarily good approximation of the desired extremal q.c.h. relatively quickly.

2.2 Approximation algorithm

We discretize the continuous construction given above in order to come up with an approximation algorithm **EXTREMAL** modulo two basic subroutines. Along the way we come up with a subroutine (which we call **INF-EXT**) that finds the best piecewise constant Beltrami coefficient v that is infinitesimally equivalent to a given one μ . We believe that this is an interesting technical result in its own right.

Our input is a mesh of sample points on the sphere, a triangulation of the sphere, a piecewise constant Beltrami coefficient (corresponding to the starting map f_0), and an error tolerance δ . The desired output is the collection of images of these sample points under the extremal q.c.h. within the error tolerance.

We follow the same steps as in the continuous construction. There is a small technicality in that we need a special kind of triangulation, and might need to make this triangulation smaller each time we use any of our subroutines to control the errors. To this end, we use a subroutine **TRIANG** which is constructed using the Delaunay refinement algorithm. We take the piecewise constant Beltrami coefficient μ_i , feed it into **INF-EXT** and obtain a piecewise constant Beltrami v_i . Just as in the continuous construction, we choose an appropriate $t > 0$ and find the q.c.h. g_i corresponding to the Beltrami coefficient $t(\mu_i - v_i)$. To obtain g_i we solve the Beltrami equation using a subroutine **BELTRAMI**.

The q.c.h. g_i moves the points (z_k) a bit. We remedy this by using the vector field method through a subroutine **VECT-FIELD**. The subroutines **BELTRAMI** and **VECT-FIELD** are standard. Then we compose the maps akin to the continuous construction. Here is where we need to assume the existence of two technical, basic subroutines **PIECEWISE-COMP** and **PIECEWISE-INV**. Once this composition is performed, we obtain a map f_{i+1} which has a smaller dilatation than f_i . We set f_{i+1} as the starting map and iterate; the algorithm terminates by producing an approximation of the desired extremal map f_* .

The issue with the two subroutines **PIECEWISE-COMP** and **PIECEWISE-INV** is as follows: Given piecewise constant Beltrami coefficients α and β (whose corresponding q.c.h. are F and G respectively) we want to compute a good piecewise constant approximation of the Beltrami coefficient corresponding to F^{-1} and to $F \circ G$. Any algorithm in computational q.c. geometry may require these subroutines. There are good candidates for such subroutines but the problem is to prove their correctness. We did not perform any complexity analysis of our algorithm simply because we do not know the complexity of the conjectural subroutines **PIECEWISE-COMP** and **PIECEWISE-INV**. But we expect our algorithm **EXTREMAL** (including the assumed subroutines) to run in polynomial time.

3 Preliminaries

In this section we present the main players from q.c. theory involved in our construction. Various eminent mathematicians (Teichmüller, Ahlfors, Bers, Reich, Strebel, Krushkal, Hamilton, etc.) have contributed to Teichmüller theory. We refer the reader to [10] and [15] for some excellent introductions to Teichmüller theory.

Quasiconformal maps and Beltrami coefficients/differentials. For a function f between two open sets in the complex plane, define partials $f_z = f_x - if_y$ and $f_{\bar{z}} = f_x + if_y$, where f_x and f_y are the partials with respect to (Euclidean coordinates) x and y . Let $\hat{\mathbb{C}}$ denote the Riemann sphere (\mathbb{C} union the point at infinity). A homeomorphism $f: \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ is quasiconformal provided that it satisfies the *Beltrami equation* $f_{\bar{z}} = \mu(z)f_z$ for some complex-valued function μ satisfying $\|\mu\|_\infty < 1$. μ is called the *Beltrami coefficient*, and is a measure of the non-conformality of f . In particular, the map f is conformal if μ is identically 0. The following theorem makes the notion of the Beltrami coefficients indexing the corresponding q.c.h. precise.

► **Theorem 1.** *The Beltrami equation gives a one to one correspondence between the set of quasiconformal homeomorphisms of $\hat{\mathbb{C}}$ that fix the points 0, 1 and ∞ and the set of measurable complex-valued functions μ on $\hat{\mathbb{C}}$ for which $\|\mu\|_\infty < 1$. Furthermore, the normalized solution f^μ of the Beltrami equation of μ depends holomorphically on μ and for any $r > 0$ there exists $\delta > 0$ and $C(r) > 0$ such that*

$$|f^{t\mu}(z) - z - tV(z)| \leq C(r)t^2 \text{ for } |z| < r \text{ and } |t| < \delta, \tag{1}$$

where $V(z) = -\frac{z(z-1)}{\pi} \int \int_{\mathbb{C}} \frac{\mu(\zeta)d\xi d\eta}{\zeta(\zeta-1)(\zeta-z)}$, and $\zeta = \xi + i\eta$.

We need some more definitions and concepts. They are summarized here:

Composition formula. Let μ, σ and τ be the Beltrami coefficients of quasiconformal maps f^μ, f^σ and f^τ with $f^\tau = f^\sigma \circ (f^\mu)^{-1}$. Then

$$\tau = \left(\frac{\sigma - \mu}{1 - \bar{\mu}\sigma} \frac{1}{\theta} \right) \circ (f^\mu)^{-1}, \text{ where } p = \frac{\partial}{\partial z} f^\mu(z) \text{ and } \theta = \frac{\bar{p}}{p}. \tag{2}$$

Quadratic differentials. For $R = \hat{\mathbb{C}}_{\{0,1,\infty,z_1,\dots,z_{n-3}\}}$ (the Riemann sphere with n marked points, three of which are normalized to be 0, 1 and ∞), the complex vector space formed by the linear span of the $n - 3$ functions

$$\phi_k(z) = \frac{1}{z(z-1)(z-z_k)}, \quad 1 \leq k \leq n-3, \tag{3}$$

is called the space of holomorphic quadratic differentials on R , denoted by $A(R)$.

Equivalence relations on Beltrami coefficients. Let $B(R)$ denote the set of all complex-valued measurable functions on R . Let $B_1(R) = \{\mu \in B(R) : \|\mu\|_\infty < 1\}$. Given two coefficients μ and ν in $B_1(R)$, denote the solution to their respective normalized⁶ Beltrami equations as f^μ and f^ν . Let R_0 and R_1 denote two marked spheres. The following definition concerns maps from R_0 to R_1 .

► **Definition 2 (Global equivalence).** μ and ν are called globally equivalent ($\mu \sim_g \nu$) if:

1. $f^\mu(z_i) = f^\nu(z_i) \forall i$.
2. The identity map from R_1 to R_1 is homotopic to $f^\nu \circ (f^\mu)^{-1}$ via a homotopy consisting of quasiconformal homeomorphisms.

A Beltrami coefficient ν is called trivial if it is globally equivalent to 0. A Beltrami coefficient with the least L_∞ norm in its global class is called globally extremal. *In other words, the marked sphere problem specifies as input a Beltrami coefficient μ , and asks to output the extremal Beltrami coefficient μ_* that is globally equivalent to μ .*

► **Definition 3 (Infinitesimal equivalence).** μ and ν are infinitesimally equivalent (written $\mu \sim_i \nu$) if $\int_R \mu \phi = \int_R \nu \phi$ for all $\phi \in A(R)$, with $\|\phi\| = 1$. A Beltrami coefficient ν is called infinitesimally trivial if it is infinitesimally equivalent to 0.

► **Definition 4 (Infinitesimally extremal).** A Beltrami coefficient ν is called infinitesimally extremal if $\|\nu\|_\infty \leq \|\mu\|_\infty$ for all $\mu \sim_i \nu$.

Optimality condition. The importance of the infinitesimally extremal Beltrami coefficients is conveyed by the celebrated Hamilton-Krushkal, Reich-Strebel, necessary and sufficient condition for optimality. Informally, this theorem states that a Beltrami coefficient μ_* is globally extremal if and only if it is infinitesimally extremal and the corresponding q.c.h. takes the domain to the desired target. See [10] for a precise statement.

Another important fact is that for all the cases we are interested in, any globally extremal Beltrami coefficient is of Teichmüller form – it can be written as $\mu_* = k_* \bar{\phi}/|\phi|$, for a unique constant $k_* < 1$ and a unique quadratic differential $\phi \in A(R)$.

An important remark on the optimality condition. Note that given a starting μ , the ν that is extremal in the infinitesimal class of μ will be of Teichmüller form. However, *it will generally not be globally equivalent to μ* . This is why we have an iterative procedure – if ν was also globally equivalent to μ we would be done in one step. We use ν and μ to obtain μ_1 , and inductively ν_1 to obtain μ_2 and so on, to get to the globally extremal μ^* which is in the same global class as μ and is infinitesimally extremal in its class, and hence is of Teichmüller form.

⁶ Fixing the points 0,1 and ∞ . Hence the freedom of Möbius transformation is accounted for.

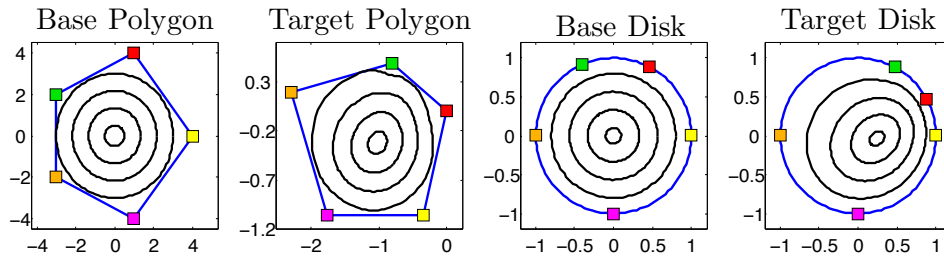


Figure 2 An example of a Teichmüller map between pentagons. If ϕ_1 and ϕ_2 are a basis of the space of quadratic differentials, the above map corresponds to the solution to the Beltrami equation of $\mu = \frac{\phi}{8\phi}$, where $\phi = \frac{1}{3}\phi_1 + \frac{2}{3}\phi_2$. On the right is the same map when pulled to the unit disks via the Riemann mapping.

4 Problem statement and main theorems

In this section we first describe the polygon mapping and the marked sphere problems, and prove that the marked sphere problem is more general. We will then proceed to state our main results.

4.1 Problem statements and reduction

Let P and Q be two n -gons⁷ in the plane. Let $\{v_i\}_{i=1}^n$ and $\{v'_i\}_{i=1}^n$ be an ordering of the vertices of P and Q , respectively. The fact that the polygons are conformally equivalent to the upper half plane \mathbb{H} , and that composition by conformal maps does not change the dilatation imply that an n -gon is essentially the same as \mathbb{H} with n marked points on the boundary $\partial\mathbb{H} = \mathbb{R}$.

► **Problem 5** (Polygon mapping problem). *Given $\{z_1, \dots, z_n, w_1, \dots, w_n\} \in \partial\overline{\mathbb{H}}$, find $\tilde{f}_* : \overline{\mathbb{H}} \rightarrow \overline{\mathbb{H}}$ (with Beltrami coefficient μ_*) satisfying:*

1. \tilde{f}_* is a quasiconformal homeomorphism of $\overline{\mathbb{H}}$ to itself.
2. $\tilde{f}_*(z_i) = w_i, i \in \{1, \dots, n\}$
3. $\|\tilde{\mu}_*\|_\infty \leq \|\mu_f\|_\infty$ for all f satisfying (1) and (2) above.

Note that by Teichmüller’s theorems the above \tilde{f}_* exists and is unique. We state the marked sphere problem next, and show that it is in fact a generalization of the polygon mapping problem.

► **Problem 6** (Marked sphere problem). *Given $\{z_1, \dots, z_{n-3}, z_{n-2} = 0, z_{n-1} = 1, z_n = \infty\}$, $\{w_1, \dots, w_{n-3}, w_{n-2} = 0, w_{n-1} = 1, w_n = \infty\}$, and $f_0 : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ such that $f_0(z_i) = w_i$, find $f_* : \hat{\mathbb{C}} \rightarrow \hat{\mathbb{C}}$ satisfying:*

1. f_* is a quasiconformal homeomorphism of $\hat{\mathbb{C}}$ to itself.
2. f_* is isotopic to f_0 relative to the points $\{0, 1, \infty, z_1, \dots, z_{n-3}\}$, i.e. $f_*(z_i) = w_i$.
3. $\|\mu_*\|_\infty \leq \|\mu_f\|_\infty$ for all f satisfying (1) and (2) above.

We call the base z_i -marked sphere R and the target w_i -marked sphere S from now on. The reason why the marked sphere problem requires a starting map f_0 as input is that by Teichmüller’s theorem, the extremal map exists and is unique within each isotopy class. The following theorem shows that Problem 6 is indeed general.

⁷ We allow for ∞ to be a vertex of the polygon.

► **Theorem 7** (Reduction). *An algorithm for Problem 6 can be used to give a solution to Problem 5.*

Proof sketch. Consider an instance of the polygon mapping problem, and map the polygons conformally in linear time using [2] to the upper-half plane such that the vertices go to points on the real line. Then, using a piecewise affine function f_0 map the corresponding upper half-planes to one another taking the vertices to the vertices. Since f_0 is real on \mathbb{R} , we extend it by symmetry to the entire Riemann sphere. Call this extended map f . This then provides us a special instance of the marked sphere problem, where all the marked points are on the real line. We then prove that the extremal map f_* homotopic to f is symmetric, and that the restriction of f_* to the upper half plane solves the original polygon mapping problem. Full proof in [11]. ◀

4.2 Results

Denote the Beltrami coefficient of f_0 by μ_0 . We want to obtain μ_* that is globally equivalent (Definition 2) to μ_0 and has the smallest L_∞ norm in this global class. We will obtain a sequence of q.c.h. f_i (and their Beltrami coefficients μ_i) that in the limit converge to the unique extremal map f_* (and the dilatations of μ_i will converge to the dilatation of μ_*). All the μ_i lie in the same global class – that of μ_0 . The main difficulty we overcome is that since the global class of μ_0 does not have a “nice” structure (e.g. it is not convex in the generic case; in fact the only way to know whether two Beltrami coefficients μ_1 and μ_2 are in the same global class is to solve their Beltrami equations). To overcome this, we break up this minimization over the global class of μ_0 into a sequence of minimizations over the infinitesimal classes (Definition 3) of μ_i (that are convex domains) and solutions of differential equations.

We will first present our main theorem in the continuous setting. By the “continuous setting” we mean that we assume the existence of black boxes that solve all the sub-problems involved exactly; e.g. given a Beltrami coefficient μ , we can get $f^\mu(z)$ for any z exactly.

► **Theorem 8** (Limiting procedure for Marked Sphere Problem). *There exists a sequence of q.c.h. f_i s.t.:*

1. **Isotopic:** f_i is isotopic to f_0 , and $f_i(z_j) = w_j$, for all i and j .
2. **“Explicit” construction:** Let v_i be the extremal coefficient in the infinitesimal class of μ_i . Then μ_{i+1} is an “explicit function” of μ_i and v_i in that it can be obtained by solving two differential equations depending only on μ_i and v_i .
3. **Uniform Convergence:** $f_i \rightarrow f_*$ uniformly and $\|\mu_i\|_\infty \rightarrow \|\mu_*\|_\infty$ as $i \rightarrow \infty$.
4. **Fast convergence:** There exist constants $C > 0$ and $\delta_0 > 0$ such that for all $\delta < \delta_0$ and for all $i \geq C/(\delta^4(1 - \|\mu_0\|_\infty)^2)$ we have $\|\mu_i\|_\infty - k_* < \delta$.

Basically, getting v_i from μ_i is the convex optimization part, and getting μ_{i+1} from μ_i and v_i requires solving differential equations.

Now we proceed to the discrete implementation of our procedure. We represent all Beltrami coefficients as piecewise constant coefficients⁸ on a fine mesh. Every step of the continuous procedure mentioned above is shown to have a discrete analogue. The mesh we

⁸ In fact, the existence of the solution to the Beltrami equation of an arbitrary $\mu \in L^\infty$ with $\|\mu\|_\infty < 1$ was shown by 1) first showing the existence of the solution to a piecewise constant μ' , 2) sewing the individual piecewise q.c. maps along the boundary, and 3) taking a limit of such piecewise constant coefficients $\mu'_n \rightarrow \mu$ and showing that the maps converge.

will be working on depends on the error tolerance δ required. The first theorem tells us how to discretise the convex optimization part.

► **Theorem 9** (Discrete infinitesimally extremal). *Given an error tolerance $0 < \delta < 1$, a collection of n marked points z_1, z_2, \dots, z_n , a triangulation Δ_ϵ and a piecewise constant Beltrami coefficient μ (where $|\mu| < 1$ on every triangle), there exists an algorithm INF-EXT that computes a piecewise constant Beltrami coefficient \hat{v} such that $|\hat{v}| - |\mu|_\infty < \delta$ everywhere.*

Now we proceed towards the other steps. Computational quasiconformal theory is a field still in its infancy, and very few error estimates on these widely-used discretizations are known. We introduce two subroutines PIECEWISE-COMP and PIECEWISE-INV (their precise definitions are in section 6) that concern the discretization of compositions and inverses of quasiconformal maps. Assuming the existence of the subroutines PIECEWISE-COMP and PIECEWISE-INV we construct an approximation algorithm for the Teichmüller map.

► **Theorem 10** (Teichmüller Map Algorithm). *Assume the existence of the aforementioned subroutines. Given a triangulation T_0 that includes n marked points z_1, \dots, z_n , a mesh of sample points S , an error tolerance δ , and a piecewise constant Beltrami coefficient μ_0 whose corresponding q.c.h. f_0 satisfies $f_0(z_j) = w_j$, there exists an algorithm EXTREMAL that computes Δ_ϵ , and the images of S up to an error of δ under a q.c.h. F having a piecewise constant (in the computed triangulation) Beltrami coefficient μ_F such that*

1. $\|\mu_F\|_\infty - \|\mu_*\|_\infty < \delta$ where μ_* is the Beltrami coefficient of the extremal quasiconformal map on the marked sphere in the isotopy class of f_0 .
2. $|F(z_i) - w_i| = O(\delta)$.

Thus our main result in the discrete case is a reduction of this approximation problem to two basic subroutines. We do not address the complexity of our approximation algorithm and expect that (along with the two conjectural subroutines) our algorithm runs in polynomial time.

5 The continuous construction

We first summarize our construction of the sequence $\{f_i\}$ of q.c.h. that converge to the extremal map. At step i , given the q.c.h. f_i with Beltrami coefficient μ_i , let v_i denote the infinitesimally extremal Beltrami coefficient in the infinitesimal class of μ_i . Let $k_i = \|\mu_i\|_\infty$ and $k_i^0 = \|v_i\|_\infty$. Observe that $\mu_i - v_i$ is infinitesimally trivial (Definition 3).

1. Choose t such that

$$t = \min \left(\frac{3}{4}, C_1, \frac{\epsilon}{4}, \sqrt{\frac{\epsilon}{2C_2}}, \frac{(k_i - k_i^0)^2(1 - k_i^2)}{1 - k_i^2 + C_2} \right), \tag{4}$$

where $\epsilon \leq \min(1/2, (k_i - k_i^0)/8)$, and C_1 and C_2 are two explicit constants derived in the full version[11].

2. Use Subsection 5.1 to construct a quasiconformal self-homeomorphism h_i of the base z_k -marked sphere such that
 - μ_h is globally trivial (hence $h_i(z_k) = z_k$ for all k).
 - $\|\mu_h - t(\mu_i - v_i)\|_\infty < C_2 t^2$, where C_2 is the same constant as in (4).
3. Form $f_{i+1} = f_i \circ (h_i)^{-1}$. It turns out that f_{i+1} has smaller dilatation than f_i (by Lemma 11).
4. Iterate with f_{i+1} as the starting map.

The second to last step i.e., calculating the composition $f_{i+1} = f_i \circ (h_i)^{-1}$ is the main point of the construction. To our knowledge, this is the first “constructive” way to produce a map having a smaller dilatation than a given one. The heart of this step is the following crucial lemma (proof in [11]):

► **Lemma 11** (Decreasing dilatation). *Let v_f be the infinitesimally extremal Beltrami coefficient in the infinitesimal class of μ_f . Let $\mu_h(t)$ be a curve of Beltrami coefficients with the following properties:*

1. $\mu_h(t)$ is globally trivial.
2. $\mu_h(t) = t(\mu_f - v_f) + O(t^2)$.

Denote the solution to the Beltrami equation of $\mu_h(t)$ by h_t . Then $\exists \delta > 0$ such that $\forall t < \delta$, the map $f_t = f \circ (h_t)^{-1}$ has smaller dilatation than f .

Proof sketch of Theorem 8. Assume for now that the map h_i produced in each step satisfies the conditions of Lemma 11. Let $k_i = \|\mu_i\|_\infty$ be the L^∞ norm of the Beltrami coefficient of f_i (the starting map at step i), and $k_i^0 = \|v_i\|_\infty$ where v_i is infinitesimally extremal. We lower bound the decrease $d = k_i - k_{i+1}$ in the dilatation in step 3 in terms of $k_i - k_i^0$. This is bounded below further by an expression which is in terms of $k_i - k_*$ (the distance from the extremal map). This is accomplished using Teichmüller’s contraction principle, which gives a quantitative version of the following fact: If a Beltrami coefficient μ is close to the infinitesimally extremal coefficient v , then it is also close to the globally extremal coefficient μ_* . Once we have d in terms of $k_i - k_*$, a standard geometric series argument coupled with a theorem on uniform convergence of sequences of q.c.h. on the sphere completes the proof. ◀

5.1 Constructing the self homeomorphisms

Starting at the i th step with a q.c.h. f_i , we now show how to construct the self homeomorphism h_i required by Lemma 11. We simplify notation by suppressing the index i , keeping in mind that this is the i th step of the procedure. Thus μ and μ_h will denote the Beltrami coefficients of f_i and h_i , respectively. Also, v is the infinitesimally extremal Beltrami coefficient in the infinitesimal class of μ .

Let $\alpha = \mu - v$, t be as in Equation (4), and let $g^{t\alpha}$ be the normalized solution to the Beltrami equation for $t\alpha$. Denote $g^{t\alpha}(z_k)$ by z'_k . As a consequence of the mapping theorem Theorem 1 that z'_k is not very far from z_k (the “error” is $O(t^2)$).

We will first construct another homeomorphism K_∇ from $\hat{\mathbb{C}}$ to itself which satisfies $K_\nabla(z'_k) = z_k$. We then define the required self homeomorphism $h = K_\nabla \circ g^{t\alpha}$. The construction of K_∇ will be via a vector field method. A summary of this vector field method is as follows.

Let $\{D_1, \dots, D_{n-3}\}$ denote disjoint open disks centered at z_k . Choosing the radius of each disk to be $r = d/4$, where $d = \max_{1 \leq k, l \leq n-3} |z_k - z_l|$ ensures disjointness. We will fix these disks once and for all.

We first construct a self homeomorphism K_∇^k of $\hat{\mathbb{C}}$ which is the identity map outside D_k , and maps z'_k to z_k . By means of a rotation we can assume that z'_k is real and greater than z_k . Consider the vector field

$$X(z) = p(z)(z'_k - z_k) \frac{\partial}{\partial x},$$

where $p(z)$ is a C^∞ function identically zero outside D_k , and identically 1 inside the disk of radius $r/2$ around z_k , denoted as D'_k . Let γ be the one parameter family of diffeomorphisms associated with this vector field (i.e. the flow of this field). We denote the time parameter

by s and note that the diffeomorphism γ_1 sends z'_k to z_k . We denote this diffeomorphism γ at $s = 1$ by K_v^k . Now define $K_v = K_v^{n-3} \circ K_v^{n-2} \dots \circ K_v^1$, and $h = K_v \circ g^{t\alpha}$. This is the desired "correction" that ensures that the q.c.h. h is indeed a self map.

Using PDE theory of the Beltrami equation, we then prove that the Beltrami coefficient of h_i so obtained does satisfy the hypothesis of Lemma 11. This completes all the details of our continuous construction.

6 The approximation algorithm

Here we present details of our approximation algorithm. Near the marked points the mesh is made up of (triangulated) regular polygons, whose number of vertices and radii depend on δ . The mesh is a triangulation with edge lengths bounded above by an appropriate ϵ that depends on δ . We call this triangulation a canonical triangulation Δ_ϵ of size ϵ . Its precise definition can be found in the full version [11]. We describe the convex optimization part of our algorithm next.

6.1 INF-EXT

We want to discretize the operator $\mathcal{P}(\mu)$ which returns v with the least L^∞ norm satisfying $\int_R v\phi_i = \int_R \mu\phi_i$ for all ϕ_i in Equation (3). Note that the starting μ is piecewise constant at the start of every iteration.

► **Observation 12.** *The integral of ϕ_i over any triangle t_j can be computed analytically. We note that this formula involves taking the logarithm of a complex number.*

We approximate v by piecewise constant Beltrami coefficients. The constraints for infinitesimal equivalence become linear constraints of the form $Ax = b$, where $A(i, j)$ th equals $\int_{t_j} \phi_i$, x is the vector of unknown values of the piecewise constant v on a triangle, and b is the vector of $\int_{t_j} \mu_j \phi_i$, where μ_j is the value of μ on triangle t_j . If A , x and b are real, an L^∞ minimization can be formulated as a linear program. In our case, we break the vectors and matrices into their real and complex parts, and then we can formulate the program as a quadratically constrained quadratic program. Although in general they are NP-hard to solve, in the the full version [11] we show that our program involves positive semi-definite matrices, and it is known that such instances can be solved in polynomial time using interior-point methods [20].

► **Lemma 13 (INF-EXT).** *There exists an algorithm INF-EXT that, given a piecewise constant μ on Δ_ϵ returns a piecewise constant \hat{v} such that $\max_{t_j} \hat{v}(t_j) \leq \max_{t_j} \beta(t_j)$, where β is any piecewise constant (on Δ_ϵ) Beltrami coefficient that is infinitesimally equivalent to μ .*

With this, we are now in a position to prove Theorem 9, which says that this piecewise approximation \hat{v} is not very far from the true infinitesimally extremal v . The full proof is relegated to the journal version [11].

6.2 Description of EXTREMAL

Apart from the subroutine INF-EXT we require a few more subroutines to discretize our procedure.

- **TRIANG.** The input is a set of points \mathcal{S} , a size M , and a triangulation Δ_ϵ . The output of TRIANG is a triangulation $\Delta_{\epsilon'}$ of the given size M containing \mathcal{S} such that $\Delta_{\epsilon'}$ is a refinement of Δ_ϵ .

- **BELTRAMI.** The input is a triangulation Δ_ϵ of the plane, a piecewise constant Beltrami coefficient μ , and error tolerance δ . The output of **BELTRAMI** is a triangulation Δ'_ϵ that is a refinement of Δ_ϵ , and the images $\hat{f}(v_i)$ of the vertices $v_i \in \Delta'_\epsilon$ such that $|f^\mu(v_i) - \hat{f}(v_i)| < \delta$.
- **VECT-FIELD.** The input is a C^k (k sufficiently large, e.g. $k > 10$) vector field X (written as a formula in terms of elementary functions), a triangulation Δ_ϵ , and an error tolerance δ . The output is a triangulation Δ'_ϵ that is a refinement of Δ_ϵ , the images of $v_i \in \Delta_\epsilon$ up to error δ under a C^k diffeomorphism γ_x corresponding to the flow along X , and a piecewise smooth Beltrami coefficient that approximates μ_{γ_x} up to error δ .
- **PIECEWISE-COMP.** The input is a triangulation Δ_ϵ , two piece-wise constant Beltrami coefficients μ_1 and μ_2 (corresponding to q.c.h. f_1 and f_2 respectively), and error tolerances δ_1 and δ_2 . The output is a triangulation $\Delta_{\epsilon'}$ that is a refinement of Δ_ϵ , a piecewise constant Beltrami coefficient μ_{comp} that approximates the Beltrami coefficient of the composition $f_3 = f_1 \circ f_2$ within error δ_1 in the L^∞ topology, and the images $f_3(v_a)$ of the vertices v_a of $\Delta_{\epsilon'}$ up to an error of δ_2 .
- **PIECEWISE-INV.** The input is a triangulation Δ_ϵ , a piecewise constant Beltrami coefficient μ (corresponding to q.c.h. f), and error tolerances δ_1 and δ_2 . The output is a triangulation $\Delta_{\epsilon'}$ that is a refinement of Δ_ϵ , a piecewise constant Beltrami coefficient μ_{inv} that approximates the Beltrami coefficient of f^{-1} within error δ_1 in the L^∞ topology, and the images $f^{-1}(v_a)$ of the vertices of $\Delta_{\epsilon'}$ up to an error of δ_2 .

EXTREMAL The algorithm summarized below is based on Section 5.

- Use **TRIANG** to produce a triangulation of size required by **INF-EXT** to run within an error of δ^{10} .
- Loop $i = 1$ to N where N is the number of iterations in Theorem 8 to produce the result within an error of $\delta/2$.
 1. Use **INF-EXT** to produce v_i from μ_i within an error of δ^{10} . If $v_i = \mu_i$ then stop.
 2. Find t_i by Equation (4), using k_0 as $\|v_i\|_\infty$.
 3. Invoke **BELTRAMI** for the coefficient $t_i(\mu_i - v_i)$ to find the images of the marked points within an accuracy of t_i^3 .
 4. Define the vector field X as in the continuous construction using a piecewise polynomial version of the bump function (that is C^{10} for instance). Then call **VECT-FIELD** to find a piecewise constant Beltrami coefficient up to an error of t_i^3 .
 5. Use **PIECEWISE-COMP** to compose the Beltrami coefficients of step 3 and step 4 within an error $(\|\mu_i\| - \|v_i\|)^5$ for the Beltrami coefficient and δ/i^2 for the q.c.h.
 6. Use **PIECEWISE-INV** to find the Beltrami coefficient of the inverse of the q.c.h. of step 5, up to the same error as that in step 5.
 7. Call **PIECEWISE-COMP** to compose μ_i and the Beltrami coefficient of step 6 to form μ_{i+1} (up to the same error as that in step 5).

Implementing TRIANG, BELTRAMI and VECT-FIELD

1. **TRIANG.** Given a set of n points, we can obtain the Delaunay triangulation in $O(n \log n)$ time. While implementing **TRIANG** we first compute the Delaunay triangulation of all the points falling inside a triangle of the given triangulation. Then we connect the vertices on the convex hull of such a set of points to one of the three vertices of the triangle they lie in. If this complete triangulation is not yet size M , we make the mesh denser by adding points as in [21] (points are added to either the circumcenters of triangles or mid-points of edges), until we reach the desired size.

2. BELTRAMI. The solution to the Beltrami equation for μ can be expressed as a series of singular operators applied to μ . There are many efficient algorithms and implementations [6],[9] existing for BELTRAMI. Most of them can bound the L^p norm of the error, but the methods in [6] can be used to bound the L^∞ error too [5].
3. VECT-FIELD. The idea of deforming a surface by a vector field has been applied extensively in computer graphics. We refer the reader to [16] for an implementation of VECT-FIELD.

► **Remarks.** Using the composition formula for Beltrami coefficients (Equation (2)), we see that in principle one may attempt to compute a piecewise constant approximation of the Beltrami coefficient of the composition $f \circ g$ of two q.c.h. f and g , and of g^{-1} (by setting $\sigma = 0$). However, this requires the derivative of g to be well-approximated in a piecewise constant manner. Therein lies the difficulty. Basically, one needs a good way of “discretising” the definition of the Beltrami coefficient of a q.c.h. The algorithm terminates by producing μ_N . The proof of Theorem 10 is similar to that of Theorem 8 and is omitted.

7 Discussions and future work

Our algorithm for the marked sphere problem also solves as a special case what is known as the “landmark constrained” Teichmüller map problem, where the points z_i and w_i are in the interior of the polygons, and a starting map is provided that sends z_i to w_i . A reduction similar to Theorem 7 works.

Open problems abound. In addition to studying the two conjectural subroutines the extremal map problem can be further explored in many directions.

1. Most of the ideas presented here (notably Lemma 11) may be used to envision an algorithm for computing Teichmüller maps between other Riemann surfaces. The problem is challenging for multiple reasons – for instance, an explicit basis of holomorphic quadratic differentials may not be available.
2. The authors feel that building a discrete version of Teichmüller theory would be an important achievement. Given a triangulated Riemann surface, defining a discrete analog of dilatation that gives nice results (e.g. existence and uniqueness) about the extremal map would be the next step in this direction.

References

- 1 L. V. Ahlfors. *Lectures on quasiconformal mappings*, volume 38 of *University Lecture Series*. American Mathematical Society, Providence, RI, second edition, 2006. With supplemental chapters by C. J. Earle, I. Kra, M. Shishikura and J. H. Hubbard.
- 2 C. Bishop. Conformal mapping in linear time. *Discrete and Comput. Geometry*, 44(2):330–428, 2010.
- 3 Christopher Bishop. Analysis of conformal and quasiconformal maps. Results from prior NSF support, 2012. <http://www.math.sunysb.edu/~bishop/vita/nsf12.pdf>.
- 4 C. Carathéodory. Über die gegenseitige Beziehung der Ränder bei der konformen Abbildung des Inneren einer Jordanschen Kurve auf einen Kreis. *Mathematische Annalen*, 73(2):305–320, 1913.
- 5 P. Daripa and M. Goswami, 2014. Private communication.
- 6 Prabir Daripa. A fast algorithm to solve the beltrami equation with applications to quasiconformal mappings. *Journal of Computational Physics*, 106(2):355–365, 1993.
- 7 T. A. Driscoll and L. N. Trefethen. *Schwarz-Christoffel Mapping*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2002.

- 8 T. A. Driscoll and S. A. Vavasis. Numerical conformal mapping using cross-ratios and delaunay triangulation. *SIAM J. Sci. Comput.*, 19:1783–1803, 1998.
- 9 D. Gaidashev and D. Khmelev. On numerical algorithms for the solution of a beltrami equation. *SIAM Journal on Numerical Analysis*, 46(5):2238–2253, 2008.
- 10 F. P. Gardiner and N. Lakic. *Quasiconformal Teichmüller theory*. American Mathematical Society, 1999.
- 11 M. Goswami, X. Gu, V. Pingali, and G. Telang. Computing Teichmüller maps between polygons. arXiv:1401.6395 – <http://arxiv.org/abs/1401.6395>, 2014.
- 12 H. Grötzsch. Über die Verzerrung bei nichtkonformen schlichten Abbildungen mehrfach zusammenhängender Bereiche. *Leipz. Ber.*, 82:69–80, 1930.
- 13 X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S. T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, 23(7):949–958, 2004.
- 14 X. Gu and S.T. Yau. Global surface conformal parameterization. In *Symposium on Geometry Processing (SGP'03)*, volume 43, pages 127–137, 2003.
- 15 J. H. Hubbard. *Teichmüller theory and applications to geometry, topology, and dynamics*. Matrix Editions, 2006.
- 16 Ldmm – the large deformation diffeomorphic metric mapping tool. <http://cis.jhu.edu/software/ldmm-volume/tutorial.php>.
- 17 L. Lui, K. Lam, S. Yau, and X. Gu. Teichmüller Mapping (T-Map) and Its Applications to Landmark Matching Registration. *SIAM Journal on Imaging Sciences*, 7(1):391–426, 2014.
- 18 L. M. Lui, Xianfeng Gu, and Shing Tung Yau. Convergence of an iterative // algorithm for Teichmüller maps via generalized harmonic maps. arXiv:1307.2679 – <http://arxiv.org/abs/1307.2679>, 2014.
- 19 Lok Ming Lui, Tsz Wai Wong, Wei Zeng, Xianfeng Gu, Paul M. Thompson, Tony F. Chan, and Shing-Tung Yau. Optimization of surface registrations using beltrami holomorphic flow. *Journal of Scientific Computing*, 50(3):557–585, 2012.
- 20 P.M. Pardalos and M.G.C. Resende. *Handbook of applied optimization*, volume 1. Oxford University Press New York, 2002.
- 21 J. Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *J. Algorithms*, 18(3):548–585, 1995.
- 22 O. Teichmüller. Extremale quasikonforme Abbildungen und quadratische Differentiale. *Preuss. Akad. Math.-Nat.*, 1, 1940.
- 23 O. Teichmüller. Bestimmung der extremalen quasikonformen Abbildungen bei geschlossenen orientierten Riemannschen Flächen. *Preuss. Akad. Math.-Nat.*, 4, 1943.
- 24 Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samaras, and P. Huang. High Resolution Tracking of Non-Rigid Motion of Densely Sampled 3D Data Using Harmonic Maps. *International Journal of Computer Vision*, 76(3):283–300, 2008.
- 25 Y. Wang, J. Shi, X. Yin, X. Gu, T. F. Chan, S. T. Yau, A. W. Toga, and P. M. Thompson. Brain surface conformal parameterization with the ricci flow. *IEEE Transactions on Medical Imaging*, 31(2):251–264, 2012.
- 26 O. Weber, A. Myles, and D. Zorin. Computing extremal quasiconformal maps. *Comp. Graph. Forum*, 31(5):1679–1689, 2012.

On-line Coloring between Two Lines*

Stefan Felsner¹, Piotr Micek², and Torsten Ueckerdt³

- 1 Technische Universität Berlin, Berlin, Germany
felsner@math.tu-berlin.de
- 2 Theoretical Computer Science Department, Faculty of Mathematics and
Computer Science, Jagiellonian University, Poland
piotr.micek@tcs.uj.edu.pl
- 3 Department of Mathematics, Karlsruhe Institute of Technology, Germany
torsten.ueckerdt@kit.edu

Abstract

We study on-line colorings of certain graphs given as intersection graphs of objects “between two lines”, i.e., there is a pair of horizontal lines such that each object of the representation is a connected set contained in the strip between the lines and touches both. Some of the graph classes admitting such a representation are permutation graphs (segments), interval graphs (axis-aligned rectangles), trapezoid graphs (trapezoids) and cocomparability graphs (simple curves). We present an on-line algorithm coloring graphs given by convex sets between two lines that uses $O(\omega^3)$ colors on graphs with maximum clique size ω .

In contrast intersection graphs of segments attached to a single line may force any on-line coloring algorithm to use an arbitrary number of colors even when $\omega = 2$.

The *left-of* relation makes the complement of intersection graphs of objects between two lines into a poset. As an aside we discuss the relation of the class \mathcal{C} of posets obtained from convex sets between two lines with some other classes of posets: all 2-dimensional posets and all posets of height 2 are in \mathcal{C} but there is a 3-dimensional poset of height 3 that does not belong to \mathcal{C} .

We also show that the on-line coloring problem for curves between two lines is as hard as the on-line chain partition problem for arbitrary posets.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases intersection graphs, cocomparability graphs, on-line coloring

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.630

1 Introduction

In this paper we deal with on-line proper vertex coloring of graphs. In this setting a graph is created vertex by vertex where each new vertex is created with all adjacencies to previously created vertices. An *on-line coloring algorithm* colors each vertex when it is created, immediately and irrevocably, such that adjacent vertices receive distinct colors. In particular, when coloring a vertex an algorithm has no information about future vertices. This means that the color of a vertex depends only on the graph induced by vertices created before. It is convenient to imagine that vertices are created by some adaptive adversary so that the coloring process becomes a game between that adversary and an on-line algorithm.

We are interested in on-line algorithms using a number of colors that is bounded by a function of the chromatic number of the input graph. For general graphs this is too much to

* P. Micek is supported by the Polish National Science Center within a grant UMO-2011/03/D/ST6/01370.



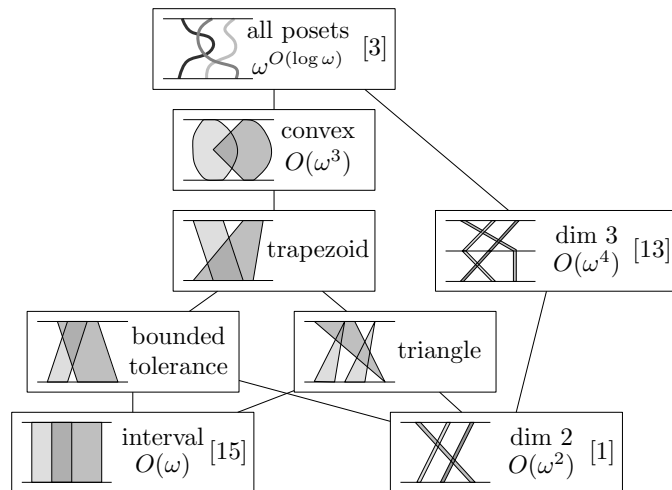
ask for. Indeed, it is a popular exercise to devise a strategy for adversary forcing any on-line algorithm to use arbitrarily many colors on a forest. However, some restricted graph classes admit competitive on-line coloring algorithms. Examples are P_5 -free graphs [13], interval graphs [15] and cocomparability graphs [12]. All of these classes are covered by the main result of Penrice, Kierstead and Trotter in [12] that says that for any tree T with radius 2, the class of graphs that do not contain an induced copy of T can be colored on-line with the number of colors depending only on T and the clique number of the input graph.

We are interested in situations where the on-line graph is presented with a geometric intersection representation. A graph G is an *intersection graph of a family \mathcal{F} of sets* if the vertices of G and the elements of \mathcal{F} are in bijection such that two vertices are adjacent in G if and only if the corresponding sets intersect. For convenience, we identify the intersection graph of the family \mathcal{F} with \mathcal{F} itself. The most important geometric intersection graphs arise from considering compact, arc-connected sets in the Euclidean plane \mathbb{R}^2 . In the corresponding on-line coloring problem such objects are created one at a time and an on-line coloring algorithm colors each set when it is created in such a way that intersecting sets receive distinct colors. For many geometric objects the on-line coloring problem is still hopeless, e.g., disks and axis-aligned squares (Erlebach and Fiala [8]). Since any intersection graph G of translates of a fixed compact convex set in the plane has a maximum degree bounded by $6\omega(G) - 7$ (Kim, Kostochka and Nakprasit [16]), any on-line algorithm that uses a new color only when it is forced to, colors G with at most $6\omega(G) - 6$ colors.

In this paper we consider the on-line coloring problem of geometric objects *spanned* between two horizontal lines, that is, arc-connected sets that are completely contained in the strip \mathbf{S} between the two lines and have non-empty intersection with each of the lines. Clearly, such a family imposes a partial order on its elements where $x < y$ if x and y are disjoint and x is contained in the left component of $\mathbf{S} \setminus y$. Hence, two sets intersect if and only if they are incomparable in the partial order, i.e., the intersection graph is a cocomparability graph. In particular, $\chi(G) = \omega(G)$ for all such graphs G . Conversely every cocomparability graph has a representation as intersection graph of y -monotone curves between two lines. The usual way to state this result is by saying that cocomparability graphs are function graphs, see [10] or [17]. If the representation is given the cocomparability graph comes with a transitive orientation of the complement. In this setting there is an on-line algorithm that uses $\omega^{O(\log \omega)}$ colors when ω is the clique number of the graph (Bosek and Krawczyk [3], see also [2]). This subexponential function in ω is way smaller than the superexponential function arising from the on-line algorithm for cocomparability graphs from [12]. The best known lower bound for on-line coloring of cocomparability graphs is of order $\Omega(\omega^2)$, see [1]. We present an on-line algorithm that uses only $O(\omega^3)$ colors on convex objects spanned between two lines.

Intersection graphs of convex sets spanned between two lines generalize several well-known graph classes.

- *Permutation graphs* are intersection graphs of segments spanned between two lines and posets admitting such cocomparability graphs are the 2-dimensional posets.
- *Interval graphs* are intersection graphs of axis-aligned rectangles spanned between two horizontal lines.
- *Bounded tolerance graphs* are intersection graphs of parallelograms with two horizontal edges spanned between two horizontal lines. (Bounded tolerance graphs were introduced in [9])
- *Triangle graphs* (a.k.a. PI-graphs) are intersection graphs of triangles with a horizontal side spanned between two horizontal lines. (Triangle graphs were introduced in [5])



■ **Figure 1** The containment order of some classes of graphs given by objects between two lines together with the performance guarantee for the best known on-line coloring algorithm (given the intersection representation as input).

- *Trapezoid graphs* are intersection graphs of trapezoids with two horizontal edges spanned between two horizontal lines. Posets admitting such cocomparability graphs are the posets of interval-dimension at most 2. (Trapezoid graphs were independently introduced in [5, 6]).

Effective on-line coloring algorithms have been known for some of these classes:

- *Permutation graphs* can be colored on-line with $\binom{\omega+1}{2}$ colors (Schmerl 1979 unpublished, see [1]). Kierstead, McNulty and Trotter [11] generalized Schmerl’s idea and gave an on-line algorithm chain partitioning d -dimensional posets, presented with d linear extensions witnessing the dimension, and using $\binom{\omega+1}{2}^{d-1}$ chains, here ω is the width of the poset.
- *Interval graphs* can be colored on-line with $3\omega - 2$ colors (Kierstead and Trotter [15]).

An easy strategy for on-line coloring is given by *First-Fit*, which is the strategy that colors each incoming vertex with the least admissible natural number. While First-Fit uses $O(\omega)$ colors on interval graphs (see [19]) it is easy to trick this strategy and force arbitrary number of colors on permutation graphs of clique-size 2 (see survey [1]). The behavior of the First-Fit algorithm on p -tolerance graphs ($0 < p < 1$), a subclass of bounded tolerance graphs, was studied in [14]. First-Fit uses there $O(\frac{\omega}{1-p})$ colors.

► **Theorem 1.** *There is an on-line algorithm coloring convex sets spanned between two lines with $O(\omega^3)$ colors when ω is the clique number of the intersection graph.*

Note that our on-line coloring algorithm is best known for bounded tolerance graphs, trapezoid and triangle graphs. The best known lower bound $\binom{\omega+1}{2}$ (see [1]) holds already for permutation graphs (segments). Proofs are deferred to later sections.

A poset is called *convex* if its cocomparability graph is an intersection graph of convex sets spanned between two lines. We give a short proof that all height 2 posets are convex. All 2-dimensional posets are convex but not all 3-dimensional.

► **Proposition 2.**

1. *Every height 2 poset is convex;*
2. *There is a 3-dimensional height 3 poset that is not convex.*

Rok and Walczak [20] have looked at intersection graphs of connected objects that are attached to a horizontal line and contained in the upper halfplane defined by this line. They show that there is a function f such that $\chi(G) \leq f(\omega(G))$ for all G admitting such a representation. However, there is no effective on-line coloring algorithm for graphs in this class, even if we restrict the objects to be segments.

► **Proposition 3.** *Any on-line algorithm can be forced to use arbitrarily many colors on a family of segments attached to a line, even if the family contains no three pairwise intersecting segments ($\omega = 2$).*

Recall that it may make a difference for an on-line coloring algorithm whether the input is an abstract cocomparability graph, or the corresponding poset, or a geometric representation. Kierstead, Penrice and Trotter [12] gave an on-line coloring algorithm for cocomparability graphs using a number of colors that is superexponential in ω . Bosek and Krawczyk [3] introduced an on-line coloring algorithm for posets using $\omega^{O(\log \omega)}$ colors where ω is the width of the poset. We show that having a poset represented by y -monotone curves between two lines does not help on-line algorithms. Indeed, such a representation can be constructed on-line if the poset is given.

► **Theorem 4.** *There is an on-line algorithm that for any poset draws y -monotone curves spanned between two lines such that $x < y$ in the poset if and only if the curves x and y are disjoint and x lies left of y . That means, for every element of the poset when it is created a curve is drawn in such a way that throughout the set of already drawn curves forms a representation of the current poset.*

Theorem 1 and Proposition 2 are proven in Section 2. Actually we define the class of quasi-convex posets and show the $O(\omega^3)$ bound for this class. Since every convex poset is quasi-convex this implies Theorem 1. The section is concluded with a proposition showing that the class of quasi-convex posets is a proper superclass of convex posets. In Section 3 we discuss general connected sets between two lines. In this context we prove Theorem 4. We conclude the paper with a proof of Proposition 3 in Section 3 and a list of four open problems related to these topics that we would very much like to see answered.

2 Quasi-Convex Sets Between Two Lines

A connected set v spanned between two parallel lines is *quasi-convex* if it contains a segment s_v that has its endpoints on the two lines. When working with a family of quasi-convex sets it is convenient to fix such a segment s_v for each v and call it the *base segment* of the set v . Clearly, every convex set spanned between two lines is also quasi-convex.

Below we show that there is an on-line algorithm coloring a family of quasi-convex sets between two parallel lines with $O(\omega^3)$ colors, when ω is the clique number of the family. This implies Theorem 1

Proof of Theorem 1. We describe an on-line coloring algorithm using at most $\binom{\omega+1}{2} \cdot 24\omega$ colors on quasi-convex sets spanned between two parallel lines with clique number at most ω . The algorithm colors incoming sets with triples (α, β, γ) of positive integers with $\alpha + \beta \leq \omega + 1$ and $\gamma \leq 24\omega$ in such a way that intersecting sets receive different triples.

Let ℓ^1, ℓ^2 be the two horizontal lines such that the quasi-convex sets of the input are spanned between ℓ^1 and ℓ^2 . With a quasi-convex set v we consider a fixed base segment s_v and the points (x -coordinates) $v^i = s_v \cap \ell^i$ for $i = 1, 2$.

A sequence (v_1, \dots, v_k) of already presented quasi-convex sets is *i-increasing* for $i = 1, 2$ if we have $v_1^i \leq v_2^i \leq \dots \leq v_k^i$. The reverse of an *i-increasing* sequence is called *i-decreasing* for $i = 1, 2$. Let α_v be the size of a maximum sequence $S_\alpha(v)$ of already presented sets that is 1-increasing and 2-decreasing and starts with v . Let β_v be the size of a maximum sequence $S_\beta(v)$ of already presented sets that is 1-decreasing and 2-increasing and starts with v .

The algorithm is going to color v with a triple $(\alpha_v, \beta_v, \gamma_v)$ where α_v and β_v are defined as above. The definition of α_v and β_v is as in Schmerl's on-line algorithm for chain partitions of 2-dimensional orders or equivalently on-line coloring of permutation graphs. Indeed, if the input consists of a set of segments, then any two segments with the same α - and β -values are disjoint.

For a fixed pair (α, β) consider the set $X = X(\alpha, \beta)$ of all quasi-convex sets u presented so far that have been colored colored with $(\alpha, \beta, *)$, where $*$ an arbitrary value of the third coordinate. Since $S_\alpha(v) \cup S_\beta(v)$ is a collection of sets with pairwise intersecting base segments we can conclude that $\alpha_v + \beta_v = |S_\alpha(v)| + |S_\beta(v)| = 1 + |S_\alpha(v) \cup S_\beta(v)| \leq 1 + \omega$.

To determine γ_v the algorithm uses First-Fit on the set $X(\alpha, \beta)$. Bosek et al. [4] have shown that First-Fit is efficient on cocomparability graphs with no induced $K_{t,t}$. The best bound is due to Dujmović, Joret and Wood [7]: First-Fit uses at most $8(2t - 3)\omega$ colors on cocomparability graphs with no induced $K_{t,t}$.

To make the result applicable we show that the intersection graph of each class $X(\alpha, \beta)$ is a cocomparability graph with no induced $K_{3,3}$. As the number of these sets is at most $\binom{\omega+1}{2}$, this will conclude the proof.

► **Claim.** *The bases of sets in $X(\alpha, \beta)$ are pairwise disjoint.*

Proof of Claim. Consider any two sets $u_1, u_2 \in X$ with the endpoints $u_j^i \in u_j \cap \ell^i$ for $i = 1, 2$ and $j = 1, 2$ of their bases. It suffices to show that we have $u_1^i < u_2^i$ for $i = 1, 2$ or $u_1^i > u_2^i$ for $i = 1, 2$.

Assume that $u_1^1 \leq u_2^1$ and $u_1^2 \geq u_2^2$ and that u_1 was presented before u_2 . Since $u_1 \in X(\alpha, \beta)$ it is part of a 1-decreasing and 2-increasing sequence $(u_1, v_2, \dots, v_\beta)$. The sequence $(u_2, u_1, v_2, \dots, v_\beta)$ is a longer 1-decreasing and 2-increasing sequence starting with u_2 . This contradicts the fact that $u_2 \in X(\alpha, \beta)$.

A similar argument applies when u_2 was presented before u_1 . In this case we compare the 1-increasing and 2-decreasing sequences $(u_2, v_2, \dots, v_\alpha)$ and $(u_1, u_2, v_2, \dots, v_\alpha)$ to arrive at a contradiction. ◀

► **Claim.** *The intersection graph of $X(\alpha, \beta)$ contains no induced $K_{3,3}$.*

Proof of Claim. Let U and V be any two disjoint triples of sets in X . We shall show that if U and V are independent, then there is a set in U which is disjoint from a set in V , i.e., that the intersection graph of these six sets is not an induced $K_{3,3}$ with bipartition classes U, V .

By the previous claim the bases of these six sets in $U \cup V$ are disjoint and hence are naturally ordered from left to right within the strip. Without loss of generality amongst the leftmost three bases at least two belong to sets in U and thus amongst the rightmost three bases at least two belong to sets in V . In particular, there are four sets $u_1, u_2 \in U, v_1, v_2 \in V$ whose left to right order of bases is u_1, u_2, v_1, v_2 .

By assumption u_1, u_2 and v_1, v_2 are non-intersecting. Since the base of each set is contained in the corresponding set (quasi-convexity) we know that u_1 lies completely to the left of the base of u_2 and v_2 lies completely to the right of the base of v_1 . Together with the order of the bases of u_2 and v_1 this makes u_1 and v_2 disjoint. ◀

◀

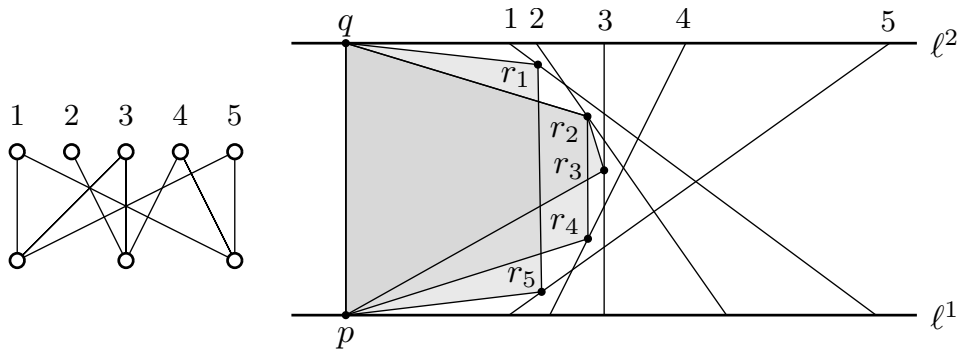


Figure 2 A poset of height 2 and its representation with convex sets spanned between two lines.

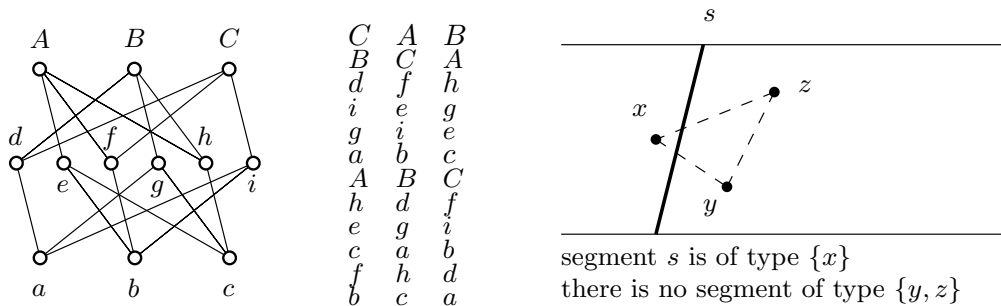


Figure 3 A height 3 and 3-dimensional poset that is not quasi-convex. Provided with its 3 linear extensions witnessing the dimension.

It is possible to decrease the number of colors used by the algorithm from $\binom{\omega+1}{2} \cdot 24\omega$ to $\binom{\omega+1}{2} \cdot 16\omega$ by showing that the pathwidth of the intersection graph of $X(\alpha, \beta)$ is at most $2\omega - 1$ and applying another result from [7]: First-Fit on cocomparability graph of pathwidth at most t uses at most $8(t + 1)$ colors.

Proof of Proposition 2. Let P be any poset of height 2, and let X and Y be the sets of minimal and maximal elements in P , respectively. We represent the elements in Y as pairwise intersecting segments so that every segment appears on the left envelope, that is, on every segment $y \in Y$ there is a point r_y such that the horizontal ray emanating from r_y to the left has no further intersection with segments from Y . Choose $p \in \ell^1$ and $q \in \ell^2$ to the left of all segments for Y and define for each $x \in X$ the convex set C_x as the convex hull of p, q and the set of all r_y for which y and x are incomparable in P . It is easy to check that in the resulting representation two sets intersect if and only if the corresponding elements in P are incomparable. See Figure 2 for an illustration.

We claim that the poset Q depicted in Figure 3 is not quasi-convex. Suppose that there is a quasi-convex realization of Q . Fix three points within the strip $x \in a \cap A$, $y \in b \cap B$, and $z \in c \cap C$. The *type* of a segment s spanned in the strip and avoiding x, y and z is the subset of $\{x, y, z\}$ consisting of the points that are to the left of s . How many different types of segments can exist for given x, y and z ? We claim that among 8 possible subsets only 7 are realizable. Indeed, consider the point $p \in \{x, y, z\}$ with the middle value with respect to the vertical axis. Then either $\{p\}$ or $\{x, y, z\} \setminus \{p\}$ is not realizable (see Figure 3). A collection of quasi-convex sets representing the elements d, e, f, g, h, i of Q must have base segments of pairwise distinct types. Moreover the types \emptyset and $\{x, y, z\}$ do not occur. This leaves 5 possible types for 6 elements, contradiction. ◀

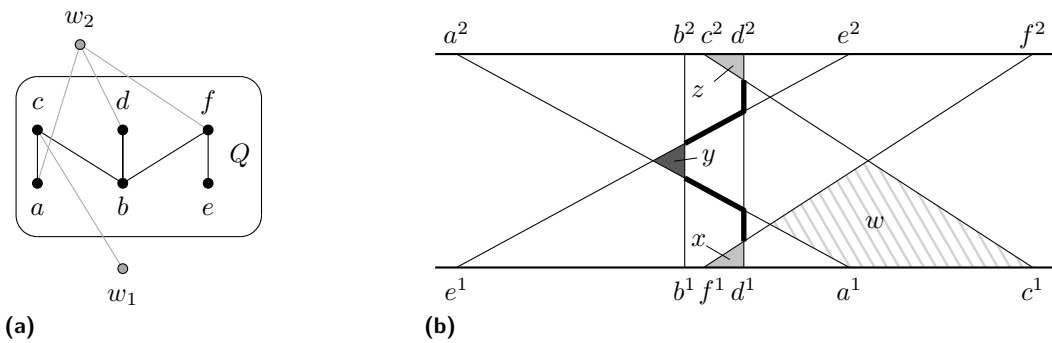


Figure 4 (a) The 6-element poset Q and the elements w_1, w_2 of P for the downset $D = \{a, b, d, e, f\}$ in Q . (b) The segment representation \mathcal{R} of Q with the cells w, x, y and z corresponding to the downsets $\{a, b, d, e, f\}, \{b, e, f\}, \{a, e\}$ and $\{a, b, c\}$ in Q , respectively.

► **Proposition 5.** *There is a quasi-convex poset that is not convex.*

Proof. Consider the poset Q on the set $E = \{a, b, c, d, e, f\}$ as shown in Figure 4a. Moreover, consider the representation \mathcal{R} of Q with segments spanned between two lines given in Figure 4b. Each cell w in \mathcal{R} naturally corresponds to the downset D_w in Q (downwards closed subset of E) formed by those segments in \mathcal{R} that lie to the left of w .

We shall construct a quasi-convex poset \bar{Q} that has Q as an induced subposet. Later we extend \bar{Q} by one point to a quasi-convex poset P and we prove that P is not convex.

Define $\bar{Q} \supset Q$ as follows. For each cell w in \mathcal{R} corresponding to a downset $D_w \subseteq E$ of Q there are two incomparable elements w_1, w_2 in \bar{Q} , where w_2 is above all elements in D_w and w_1 is below all elements in $E \setminus D_w$. There are no further comparabilities between w_1, w_2 and elements of \bar{Q} , except for those implied by transitivity. We refer again to Figure 4 for an illustration.

We extend \bar{Q} by adding an element g below d and y_2 , but incomparable to x_2 and z_2 , where y, x and z are the cells in \mathcal{R} corresponding to downsets $\{a, e\}, \{b, e, f\}, \{a, b, c\}$ in Q , respectively (Figure 4b). Let P be the poset after adding g .

To see that P is quasi-convex take the representation \mathcal{R} of Q , select a point p_w in each cell w , let s and t be two segments such that s is on the left and t is on the right of all segments in \mathcal{R} . For each cell w of \mathcal{R} define T-shaped sets for w_1 and w_2 consisting of s and t , respectively, together with a horizontal segment ending at p_w . Finally let g be the union of s and two horizontal segments, one ending at p_x and one at p_z .

Fix any quasi-convex representation $\bar{\mathcal{R}}$ of \bar{Q} . By definition each quasi-convex set in E comes with a base segment spanned between the two lines. With \mathcal{R}' we denote the configuration of the base segments corresponding to elements of Q .

► **Claim.** *The segment representations \mathcal{R} and \mathcal{R}' are equivalent in the sense that the segments together with ℓ^1 and ℓ^2 induce (up to reflection) the same plane graph where vertices are attachment points and crossings of segments and edges are pieces of segments/lines between consecutive vertices).*

Proof of Claim. Consider any cell w in \mathcal{R} and the corresponding downset $D_w \subseteq E$ of Q . By the definition of w_1, w_2 in \bar{Q} (in particular, the fact that the corresponding sets intersect in $\bar{\mathcal{R}}$), there is a cell \bar{w} in \mathcal{R}' that lies to the right of all sets in D_w and to the left of all sets in $E \setminus D_w$. Since there can be only one such cell in \mathcal{R}' , we have an injection φ from the cells of \mathcal{R} into the cells of \mathcal{R}' .

Next note that if s, t are two intersecting segments in \mathcal{R} , then there is a cell in \mathcal{R} with s to the left and t to the right, as well as another cell with s to the right and t to the left. With φ we have such cells also in \mathcal{R}' and hence the segments for s and t in \mathcal{R}' intersect as well. Disjoint segments in \mathcal{R} represent a comparability in Q , hence, the corresponding segments in \mathcal{R}' have to be disjoint as well. It follows that the number of intersections and, hence, also the number of cells, is the same in \mathcal{R} and \mathcal{R}' , proving that φ is a bijection.

To show that \mathcal{R} and \mathcal{R}' are equivalent we now consider the dual graphs. That is we take the cells between the lines as vertices and make them adjacent if and only if the corresponding downsets differ in exactly one element. These dual graphs come with a plane embedding. All the inner faces of these embeddings correspond to crossings and are therefore of degree 4. Moreover, every 4-cycle of these graphs has to be an inner face. This uniquely determines (up to reflection) the embeddings of these dual graphs and hence also of the primal graphs. For the last conclusion we have used that the union of all segments in \mathcal{R} and \mathcal{R}' is connected. ◀

► **Claim.** P is not convex.

Proof of Claim. By the previous claim every quasi-convex representation of P induces a segment representation \mathcal{R}' of Q equivalent to \mathcal{R} . We denote the segments in \mathcal{R}' for elements a, b, c, d, e, f by $a^*, b^*, c^*, d^*, e^*, f^*$, respectively, and the cells in \mathcal{R}' corresponding to x, y, z in \mathcal{R} by x^*, y^*, z^* , respectively. We claim that x^* lies strictly below y^* , which lies strictly below z^* . Indeed, we can construct a y -monotone curve as follows (Figure 4b): Start with the highest point of x^* , i.e., the crossing of f^* and d^* , follow d^* to its crossing with a^* , follow a^* to its crossing with b^* , i.e., the lowest point of the cell y^* . And symmetrically, we go from the lowest point of y^* (the crossing of b^* and e^*) along e^* to its crossing with d^* and along d^* to its crossing with c^* , i.e., the highest point of z^* .

Now, as g is below d , but incomparable to x_2 , the set for g contains a point p right of f^* and left of d^* , i.e., $p \in x$. Similarly, the set for g contains a point $q \in z$. Moreover the segment between p and q lies between the segments b^* and d^* as it starts and ends there. However, the base segment for y_2 lies to the right of d^* as y_2 is to the right of e and a . Hence, if g were a convex set, then the sets g and y_2 would intersect, contradicting that g is below y_2 in P . ◀

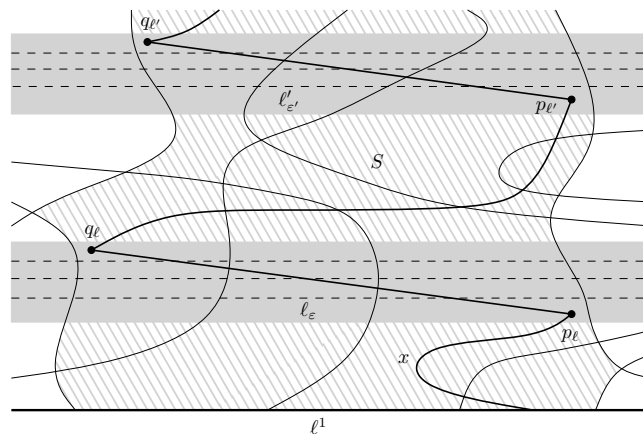
3 On-line Curve Representation

In this section we prove Theorem 4, i.e., we show that there is an on-line algorithm that produces a curve representation of any poset that is given on-line. The curves used for the representation are y -monotone.

Recall that a *linear extension* L of a poset P is a total ordering of its elements such that if $x < y$ in P this implies $x < y$ in L . Our construction maintains the invariant that at all times the curve representation \mathcal{C} of the current poset P satisfies the following property (*):

there is a set \mathcal{L} of horizontal lines such that for every linear extension L of P there is a horizontal line $\ell \in \mathcal{L}$ such that the curves in \mathcal{C} intersect ℓ from left to right in (*) distinct points in the order given by L .

For the first element of the poset use any vertical segment in the strip and property (*) is satisfied. Assume that for the current poset P we have a curve representation \mathcal{C} with y -monotone curves respecting (*).



■ **Figure 5** Constructing the curve for a new element x by using segments within ε -tubes for each $\ell \in \mathcal{L}$. Dashed horizontal lines correspond to the lines in $\mathcal{L}_{P \cup \{x\}}$.

Let x be a new element extending P . The elements of P are partitioned into the upset $U(x) = \{y : x < y\}$, the downset $D(x) = \{y : y < x\}$, and the set $I(x) = \{y : x || y\}$ of incomparable elements. Let S be the union of all points in the strip between ℓ^1 and ℓ^2 that lie strictly to the left of all curves in $U(x)$ and strictly to the right of all curves in $D(x)$. Note that S is y -monotone (its intersection with any horizontal line is connected), $S \cap \ell^i \neq \emptyset$ for $i = 1, 2$, and that S is connected since each curve in $U(x)$ lies completely to the right of each curve in $D(x)$. This implies that for any two points $p, q \in S$ with distinct y -coordinates there is a y -monotone curve connecting p and q inside of S .

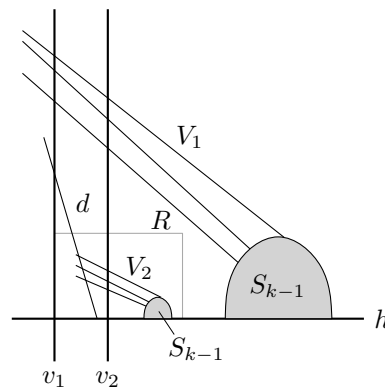
We use the set \mathcal{L} to draw the curve for x as follows:

- Choose $\varepsilon > 0$ small enough so that within the ε -tube ℓ_ε of any line $\ell \in \mathcal{L}$ no two curves get closer than ε .
- For each line $\ell \in \mathcal{L}$ choose two points $q_\ell, p_\ell \in \ell_\varepsilon \cap S$ such that q_ℓ is above ℓ and has distance at most ε to the left boundary of S while p_ℓ is below ℓ and has distance at most ε to the right boundary of S . Draw a segment from p_ℓ to q_ℓ .
- If ℓ and ℓ' are consecutive in \mathcal{L} with ℓ below ℓ' , then we connect q_ℓ and $p_{\ell'}$ by a y -monotone curve in S . We also connect the lowest p and the highest q by y -monotone curves in S to ℓ^1 and ℓ^2 respectively.
- The curve of x is the union of the segments $\overline{p_\ell q_\ell}$ and the connecting curves.

Figure 5 illustrates the construction.

We claim that the curve representation of P together with the curve of x has property (*). Let $L = (\dots, a, x, b, \dots)$ be an arbitrary linear extension of $P \cup \{x\}$ and let $L^x = (\dots, a, b, \dots)$ be the linear extension of P obtained from L by omitting x . Let $\ell^x \in \mathcal{L}$ be the horizontal line corresponding to L^x . Within the ε -tube of ℓ^x the segment $\overline{p_{\ell^x} q_{\ell^x}}$ contains a subsegment $\overline{\rho_a \rho_b}$ where ρ_a is a point ε to the right of the curve of a and ρ_b is a point ε to the left of the curve of b . The horizontal line ℓ containing the point $\frac{\rho_a + \rho_b}{2}$ is a line representing L in $P \cup \{x\}$. This proves property (*) for the extended collection of curves.

The comparabilities in the intersection of all linear extensions of $P \cup \{x\}$ are exactly the comparabilities of $P \cup \{x\}$. Therefore, property (*) implies that the curve of x is intersecting the curves of all elements of $I(x)$. Since the curve of x is in the region S it is to the right of all curves in $D(x)$ and to the left of all curves in $U(x)$. Hence, the extended family of curves represents $P \cup \{x\}$.



■ **Figure 6** Strategy S_k consists of two calls of strategy S_{k-1} and an addition of an extra segment d . Algorithm A unavoidably uses k colors on the segments intersecting v_1 or on the segments intersecting v_2 .

4 Connected Sets Attached to a Line

In this section we give the proof of Proposition 3. Actually, we prove a stronger statement by induction:

► **Claim.** *The adversary has a strategy S_k to create a family of segments attached to a horizontal line h with clique number at most 2 against any on-line coloring algorithm A such that there is a vertical line v with the properties:*

1. *any two segments pierced by v are disjoint,*
2. *every segment pierced by v is attached to h to the right of v ,*
3. *A uses at least k distinct colors on segments pierced by v .*

Proof of Claim. The strategy S_1 only requires a single segment with negative slope. Now consider $k \geq 2$. Fix any on-line algorithm A . The strategy S_k goes as follows. First the adversary uses S_{k-1} to create a family of segments \mathcal{F}_1 and a vertical line v_1 piercing a set $V_1 \subseteq \mathcal{F}_1$ of pairwise disjoint segments on which A uses at least $k-1$ colors. Define a rectangle R with bottom-side on h , the left-side in v_1 and small enough such that the vertical line supported by the right-side is piercing the same subset V_1 of \mathcal{F}_1 , moreover R is disjoint from all the segments in \mathcal{F}_1 . The adversary uses strategy S_{k-1} again, this time with the restriction that all the segments are contained in R . This creates a family \mathcal{F}_2 and a vertical line v_2 piercing a set $V_2 \subseteq \mathcal{F}_2$ of pairwise disjoint segments on which A uses at least $k-1$ colors. By construction segments from \mathcal{F}_1 and \mathcal{F}_2 are pairwise disjoint. From the definition of R it follows that line v_2 intersects all the segments in V_1 and no other segments from \mathcal{F}_1 . Strategy S_k is completed with the creation of one additional segment d such that d is attached between v_1 and v_2 , d is intersecting all the segments in V_2 and the vertical line v_1 but it intersects none of the segments in V_1 (see Figure 6).

If A uses at least k distinct colors on $V_1 \cup V_2$ then v_2 is the vertical line witnessing the invariant. Otherwise A uses exactly the same set of $k-1$ colors on V_1 and V_2 and since segment d intersects all segments in V_1 it must be colored with a different color. Thus, the vertical line v_1 intersecting $V_1 \cup \{d\}$ intersects segments of at least k distinct colors. ◀

5 Open problems

In this concluding section we collect some open problems related to the results of this paper.

In Figure 1 there are some classes of posets that contain interval orders and 2-dimensional orders and are contained in the class of convex orders. For on-line coloring of the cocomparability graphs of these classes (given with a representation) we have the algorithm from Theorem 1 that uses $O(\omega^3)$ colors.

- (1) Find an on-line algorithm that only needs $O(\omega^\tau)$ ($\tau < 3$) colors for coloring graphs in a class \mathcal{G} between 2-dimensional and convex. Interesting choices for \mathcal{G} would be trapezoid graphs, bounded tolerance graphs, triangle graphs (or simple triangle graphs; for the definition cf. [18]).

By restricting the curves or the intersection pattern of curves spanned between two lines we obtain further classes of orders which are nested between 2-dimensional orders and the class of all orders. We define *k-bend orders* by restricting the number of bends of the polygonal curves representing the elements to k . Clearly, every $k + 2$ dimensional order is a k -bend order. We define *k-simple orders* by restricting the number of intersections of pairs of curves representing elements of the order to k .

- (2) Find on-line algorithms that only need polynomially many colors for coloring cocomparability graphs of 2-simple or 1-bend orders when a representation is given.

Another direction would be the study of recognition complexity. Meanwhile the recognition complexity for all classes shown in Figure 1, except convex orders, has been determined (see [18]).

- (3) Determine the recognition complexity for convex orders.

We think that the determination of the recognition complexity of 2-simple orders and 1-bend orders are also interesting problems.

References

- 1 Bartłomiej Bosek, Stefan Felsner, Kamil Kloch, Tomasz Krawczyk, Grzegorz Matecki, and Piotr Micek. On-line chain partitions of orders: a survey. *Order*, 29(1):49–73, 2012.
- 2 Bartłomiej Bosek, Henry A. Kierstead, Tomasz Krawczyk, Grzegorz Matecki, and Matthew E Smith. An improved subexponential bound for on-line chain partitioning. arXiv preprint arXiv:1410.3247, 2014.
- 3 Bartłomiej Bosek and Tomasz Krawczyk. The sub-exponential upper bound for on-line chain partitioning. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science FOCS 2010*, pages 347–354. IEEE Computer Soc., Los Alamitos, CA, 2010.
- 4 Bartłomiej Bosek, Tomasz Krawczyk, and Edward Szczyepka. First-Fit algorithm for the on-line chain partitioning problem. *SIAM J. Discrete Math.*, 23(4):1992–1999, 2010.
- 5 Derek G. Corneil and P. A. Kamula. Extensions of permutation and interval graphs. *Congr. Numer.*, 58:267–275, 1987. Eighteenth Southeastern International Conference on Combinatorics, Graph Theory, and Computing (Boca Raton, Fla., 1987).
- 6 Ido Dagan, Martin Charles Golumbic, and Ron Yair Pinter. Trapezoid graphs and their coloring. *Discrete Appl. Math.*, 21(1):35–46, 1988.
- 7 Vida Dujmović, Gwenaël Joret, and David R. Wood. An improved bound for First-Fit on posets without two long incomparable chains. *SIAM J. Discrete Math.*, 26(3):1068–1075, 2012.
- 8 Thomas Erlebach and Jiri Fiala. On-line coloring of geometric intersection graphs. *Comput. Geom.*, 23(2):243–255, 2002.
- 9 Martin Charles Golumbic and Clyde L. Monma. A generalization of interval graphs with tolerances. In *Proceedings of the thirteenth Southeastern conference on combinatorics, graph theory and computing (Boca Raton, Fla., 1982)*, volume 35, pages 321–331, 1982.

- 10 Martin Charles Golumbic, Doron Rotem, and Jorge Urrutia. Comparability graphs and intersection graphs. *Discrete Math.*, 43(1):37–46, 1983.
- 11 Henry A. Kierstead, George F. McNulty, and William T. Trotter, Jr. A theory of recursive dimension for ordered sets. *Order*, 1(1):67–82, 1984.
- 12 Henry A. Kierstead, Stephen G. Penrice, and William T. Trotter, Jr. On-line coloring and recursive graph theory. *SIAM J. Discrete Math.*, 7(1):72–89, 1994.
- 13 Henry A. Kierstead, Stephen G. Penrice, and William T. Trotter, Jr. On-line and First-Fit coloring of graphs that do not induce P_5 . *SIAM J. Discrete Math.*, 8(4):485–498, 1995.
- 14 Henry A. Kierstead and Karin R. Saoub. First-Fit coloring of bounded tolerance graphs. *Discrete Appl. Math.*, 159(7):605–611, 2011.
- 15 Henry A. Kierstead and William T. Trotter, Jr. An extremal problem in recursive combinatorics. In *Proceedings of the Twelfth Southeastern Conference on Combinatorics, Graph Theory and Computing, Vol. II (Baton Rouge, La., 1981)*, volume 33, pages 143–153, 1981.
- 16 Seog-Jin Kim, Alexandr Kostochka, and Kittikorn Nakprasit. On the chromatic number of intersection graphs of convex sets in the plane. *Electron. J. Combin.*, 11(1):R52, 2004.
- 17 László Lovász. Perfect graphs. In *Selected topics in graph theory, 2*, pages 55–87. Academic Press, London, 1983.
- 18 George Mertzios. The recognition of simple-triangle graphs and of linear-interval orders is polynomial. In Proceedings of the 21st European Symposium on Algorithms (ESA), Sophia Antipolis, France, September 2013, pp. 719–730.
- 19 Sriram V. Pemmaraju, Rajiv Raman, and Kasturi Varadarajan. Buffer minimization using max-coloring. In *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 562–571. ACM, New York, 2004.
- 20 Alexandre Rok and Bartosz Walczak. Outerstring graphs are χ -bounded. In Siu-Wing Cheng and Olivier Devillers, editors, *30th Annual Symposium on Computational Geometry (SoCG 2014)*, pages 136–143. ACM, New York, 2014.

Building Efficient and Compact Data Structures for Simplicial Complexes

Jean-Daniel Boissonnat^{*1}, Karthik C. S.^{†2}, and Sébastien Tavenas^{‡3}

- 1 Geometrica, INRIA Sophia Antipolis – Méditerranée, France
Jean-Daniel.Boissonnat@inria.fr
- 2 Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Israel
karthik.srikanta@weizmann.ac.il
- 3 Max-Planck-Institut für Informatik, Saarbrücken, Germany
stavenas@mpi-inf.mpg.de

Abstract

The Simplex Tree (ST) is a recently introduced data structure that can represent abstract simplicial complexes of any dimension and allows efficient implementation of a large range of basic operations on simplicial complexes. In this paper, we show how to optimally compress the Simplex Tree while retaining its functionalities. In addition, we propose two new data structures called Maximal Simplex Tree (MxST) and Simplex Array List (SAL). We analyze the compressed Simplex Tree, the Maximal Simplex Tree, and the Simplex Array List under various settings.

1998 ACM Subject Classification E.1 Data structures, F.2.2 Nonnumerical Algorithms and Problems – Computations on discrete structures, Geometrical problems and computations

Keywords and phrases Simplicial complex, Compact data structures, Automaton, NP-hard

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.642

1 Introduction

Simplicial complexes are widely used in combinatorial and computational topology, and have found many applications in topological data analysis and geometric inference. The most common representation uses the Hasse diagram of the complex that has one node per simplex and an edge between any pair of incident simplices whose dimensions differ by one. A few attempts to obtain more compact representations have been reported recently.

Attali et al. [4] proposed the skeleton-blockers data structure which represents a simplicial complex by its 1-skeleton together with its set of blockers. Blockers are the simplices which are not contained in the complex but whose proper subfaces are. Flag complexes have no blockers and the skeleton-blocker representation is especially efficient for complexes that are “close” to flag complexes. An interesting property of the skeleton-blocker representation is that it enables efficient edge contraction.

Boissonnat and Maria [8] have proposed a tree representation called the Simplex Tree that can represent general simplicial complexes and scales well with dimension. The nodes

* This work was partially supported by the Advanced Grant of the European Research Council GUDHI.
† This work was partially supported by Labex UCN@Sophia scholarship, LIP fellowship and Irit Dinur’s ERC-StG grant number 239985.
‡ A part of this work was done at LIP, ENS Lyon (UMR 5668 ENS Lyon – CNRS – UCBL – INRIA, Université de Lyon).



of the tree are in bijection with the simplices (of all dimensions) of the simplicial complex. In this way, the Simplex Tree explicitly stores all the simplices of the complex but it doesn't represent explicitly all the incidences between simplices that are stored in the Hasse diagram. Storing all the simplices is useful (for example, one can then attach information to each simplex or store a filtration succinctly). Moreover, the tree structure of the Simplex Tree leads to efficient implementation of basic operations on simplicial complexes (such as retrieving incidence relations, and in particular retrieving the faces or the cofaces of a simplex).

In this paper, we propose a way to compress the Simplex Tree so as to store as few nodes and edges as possible without compromising the functionality of the data structure. The new compressed data structure is in fact a finite automaton (referred to in this paper as the Minimal Simplex Automaton) and we describe an optimal algorithm for its construction.

Previous works have looked at trie compression and have tried to establish a good trade-off between speed and size, but in most of the works, the emphasis is on one of the two. Two examples of work where the speed is of main concern are [2] where the query time is improved by reducing the number of levels in a binary trie (which corresponds to truncating the Simplex Tree at a certain height) and [1] where trie data structures are optimized for computer memory architectures. When the size of the structure is of primary concern, the work is usually focused on automata compression. For instance, in the context of natural language data processing, significant savings in memory space can be obtained if the dictionary is stored in a directed acyclic word graph (DAWG), a form of a minimal deterministic automaton, where common suffixes are shared [3]. However, theoretical analysis of compression is seldom done (if at all), in any of these works.

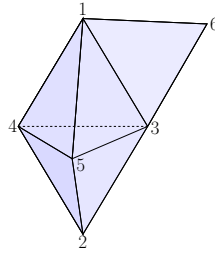
In this paper, we analyze the size of the Minimal Simplex Automaton and also demonstrate (through experiments) that compression works especially well for Simplex Tree due to the structure of simplicial complexes: namely, that all subfaces of a simplex in the complex also belong to the complex. Additionally, we consider the influence of the labeling of the vertices on compression, which can be significant. Further, we show that it is hard to find an optimal labeling for the compressed Simplex Tree and for the Minimal Simplex Automaton.

Further, we introduce two new data structures for simplicial complexes called the Maximal Simplex Tree (MxST) and the Simplex Array List (SAL). MxST is a subtree of the Simplex Tree whose leaves are in bijection with the maximal simplices (i.e. simplices with no cofaces) of the complex. We show that MxST is compact and allows efficient operations. MxST is then augmented to obtain SAL where every node uniquely represents an edge. A nice feature of SAL is its invariance over labeling of vertices. We show that SAL supports efficient basic operations and that it is compact when the dimension of the complex is fixed, a case of great interest in Manifold Learning and Topological Data Analysis. Complete proofs and more detailed discussions are presented in the full version of the paper [7].

2 Simplicial Complex: Definitions and a Lower Bound

In this paper, the class of d dimensional simplicial complexes on n vertices with m simplices, of which k are maximal, is denoted by $\mathcal{K}(n, k, d, m)$, and K denotes a simplicial complex in $\mathcal{K}(n, k, d, m)$. At times, we say $K_\theta \in \mathcal{K}_\theta(n, k, d, m)$ (where $\theta : V \rightarrow \{1, 2, \dots, |V|\}$ is a labeling of the vertex set V of K) when we want to emphasize that some of the data structures seen in this paper are influenced by the labeling of the vertices.

A maximal simplex of a simplicial complex is a simplex which is not contained in a larger simplex of the complex. A simplicial complex is pure, if all its maximal simplices are of the same dimension. Also, a free pair is defined as a pair of simplices (τ, σ) in K where τ is the



■ **Figure 1** Simplicial complex with the tetrahedra 1-3-4-5 and 2-3-4-5, and the triangle 1-3-6.

only coface of σ . In Figure 1, we have a simplicial complex on vertex set $\{1, 2, 3, 4, 5, 6\}$ which has three maximal simplices: the two tetrahedra 1-3-4-5 and 2-3-4-5, and the triangle 1-3-6. We use this complex as an example through out the paper.

We would like to note here that the case when $k = \mathcal{O}(n)$, is of particular interest. It can be observed in flag complexes, constructed from planar graphs and expanders [13], and in general, from nowhere dense graphs [15], and also from chordal graphs[14]. Generalizing, for all flag complexes constructed from graphs with degeneracy $\mathcal{O}(\log n)$ (degeneracy is the smallest integer r such that every subgraph has a vertex of degree at most r), we have that $k = n^{\mathcal{O}(1)}$ [13]. This encompasses a large class of complexes encountered in practice.

Now, we obtain a lower bound on the space needed to represent simplicial complexes by presenting a counting argument on the number of distinct simplicial complexes.

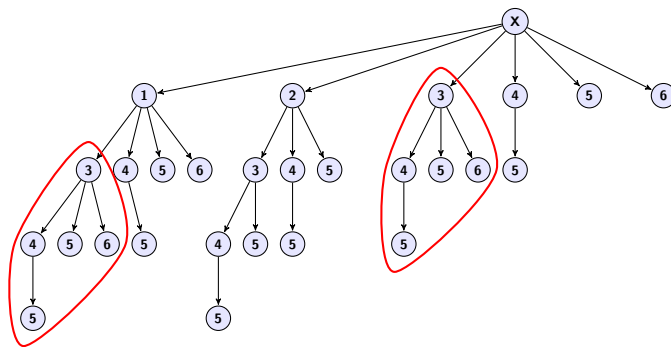
► **Theorem 1.** *Consider the set of all simplicial complexes $\mathcal{K}(n, k, d, m)$ where $d \geq 2$ and $k \geq n + 1$, and consider any data structure that can represent the simplicial complexes of this class. Such a data structure requires $\log \binom{\binom{n/2}{d+1}}{k-n}$ bits to be stored. For any constant $\varepsilon \in (0, 1)$ and for $\frac{2}{\varepsilon}n \leq k \leq n^{(1-\varepsilon)d}$ and $d \leq n^{\varepsilon/3}$, the bound becomes $\Omega(kd \log n)$.*

Proof Sketch. Define $h = k - n \geq 1$ and suppose there exists a data structure that is stored only using $s < \log \alpha \stackrel{\text{def}}{=} \log \binom{\binom{n/2}{d+1}}{h}$ bits. We will construct α simplicial complexes, all with the same set P of n vertices, the same dimension d , and with exactly k maximal simplices. Then, we will have two different complexes, say K and K' , encoded by the same word. But, by the construction of these complexes, there is a simplex which is in K and not in K' . ◀

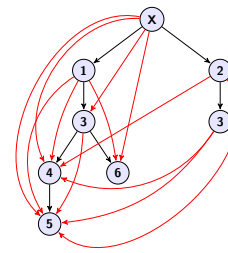
Theorem 1 applies particularly to the case of pseudomanifolds of fixed dimension where we have $k \leq n^{\frac{d}{2}}$ (i.e. $\varepsilon = \frac{1}{2}$ suffices) [6]. The case where d is small is important in Manifold Learning where it is usually assumed that the data live close to a manifold of small intrinsic dimension. The dimension of the simplicial complex should reflect this fact and ideally be equal to the dimension of the manifold.

3 Compression of the Simplex Tree

Let $K \in \mathcal{K}(n, k, d, m)$ be a simplicial complex whose vertices are labeled from 1 to n and ordered accordingly. We can thus associate to each simplex of K a word on the alphabet set $\{1, \dots, n\}$. Specifically, a j -simplex of K is uniquely represented as the word of length $j + 1$ consisting of the ordered set of the labels of its $j + 1$ vertices. Formally, let $\sigma = \{v_{\ell_0}, \dots, v_{\ell_j}\}$ be a simplex, where v_{ℓ_i} are vertices of K and $\ell_i \in \{1, \dots, n\}$ and $\ell_0 < \dots < \ell_j$. σ is represented by the word $[\sigma] = [\ell_0, \dots, \ell_j]$. The simplicial complex K can be defined as a collection of words on an alphabet of size n . To compactly represent the set of simplices of K , the corresponding words are stored in a tree and this data structure is called the Simplex



■ **Figure 2** Simplex Tree of the simplicial complex in Figure 1.



■ **Figure 3** Compressed Simplex Tree of the Simplex Tree given in Figure 2.

Tree of K and denoted by $ST(K)$ or simply ST when there is no ambiguity. It may be seen as a trie on the words representing the simplices of the complex. The depth of the root is 0 and the depth of a node is equal to the dimension of the simplex it represents plus one.

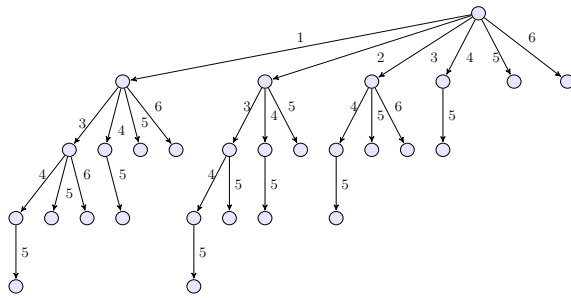
We give a constructive definition of ST . Starting from an empty tree, insert the words representing the simplices of the complex in the following manner. When inserting the word $[\sigma] = [l_0, \dots, l_j]$ start from the root, and follow the path containing successively all labels l_0, \dots, l_i , where $[l_0, \dots, l_i]$ denotes the longest prefix of $[\sigma]$ already stored in the ST . Next, append to the node representing $[l_0, \dots, l_i]$ a path consisting of the nodes storing labels l_{i+1}, \dots, l_j . In Figure 2, we give ST for the simplicial complex shown in Figure 1.

If K consists of m simplices (including the empty face) then, the associated ST contains exactly m nodes. Thus, we need $\Theta(m \log n)$ space/bits to represent ST (since each node stores a vertex which needs $\Theta(\log n)$ bits to be represented). We can compare this to the lower bound of Theorem 1. In particular, if $k = \mathcal{O}(1)$ then, ST requires at least $\Omega(2^d \log n)$ bits whereas Theorem 1 proves the necessity of only $\Omega(d \log n)$ bits. Therefore, while ST is an efficient data structure for some basic operations such as determining membership of a simplex and computing the r -skeleton of the complex, it requires storing every simplex explicitly through a node, leading to combinatorial redundancy. To overcome this, we compress ST as described below.

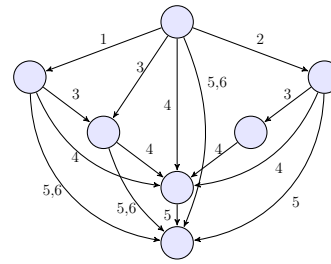
3.1 Compressed Simplex Tree

Consider the ST in Figure 2 and note that the parts marked in red appear twice. The goal of the compression is to identify these common parts and store them only once. More concretely, if the same subtree is rooted at two different nodes in ST , then the subtree is stored only once and the two root nodes now point to the unique copy of the subtree. As a consequence, the nodes are no longer in bijection with the nodes of the complex (as it was in the case of ST), but we still have the property that the paths from the root are in bijection with the simplices. We see in Figure 3, the compressed ST of the simplicial complex described in Figure 1. In the rest of the paper, we denote by \mathcal{C} , this action of compression. Also, unless otherwise stated, $|ST|$ and $|\mathcal{C}(ST)|$ refer to the number of edges in ST and $\mathcal{C}(ST)$ respectively.

Answering simplex membership queries and other queries that only require traversing the ST from root to leaves can be implemented in $\mathcal{C}(ST)$ exactly as in ST [8]. Allowing upward traversal in ST is also possible (with additional pointers from children to parents) and this has been shown to improve the efficiency of some operations, such as face or coface retrieval. In $\mathcal{C}(ST)$, parents are not unique. To account for this, we mark the parents that



■ **Figure 4** Simplex Automaton of the complex in Figure 1.



■ **Figure 5** Minimal Simplex Automaton of the complex in Figure 1.

were accessed, and use this to go back in the upward direction. This implies an additional storage of $\mathcal{O}(d \log n)$ while traversing, but a node (simplex) having many parents can assist to locate cofaces much faster.

Next, we will introduce an automaton perspective of the above compression and show how to deduce the optimal compression algorithm for the ST. We will also describe insertion and removal operations on $\mathcal{C}(\text{ST})$ through the automaton perspective.

3.2 Minimal Simplex Automaton

A Deterministic Finite state Automaton (DFA) recognizing a language is defined by a set of states and labeled transitions between these states to detect if a given word is in a predefined language or not. ST can be seen as a DFA: let us define the set of m states by $\mathcal{V} = \{\text{nodes of ST}\}$. A transition from state u to state v is labeled by a if and only if there is in ST an edge from u to v , and v contains the vertex a . We define the Simplex Automaton of K (denoted by $\text{SA}(K)$) as the automaton described above (cf. Figure 4).

SA is basically the same data structure as ST except that the labels are not put on the nodes but on the edges entering these nodes and thus, basic operations in SA can be implemented as in ST. Also, by construction of SA, it is obvious that the number of states and transitions of SA are equal to the number of nodes and edges in ST respectively.

It is known [19] that if a language L is regular (accepted by a DFA) then, L has a unique minimal automaton. DFA minimization is the task of transforming a given DFA into an equivalent DFA that has a minimum number of states. We represent the action of performing DFA minimization by \mathcal{M} . The compression of ST can be seen as DFA minimization since merging identical subtrees corresponds to merging indistinguishable states in the automaton. DFA minimization has been well studied. For instance, Hopcroft's algorithm [17] minimizes an automaton with m transitions over an alphabet of size n in $\mathcal{O}(m \log m \log n)$ steps and needs at most $\mathcal{O}(m \log n)$ space. This run-time is shown in [17] to be optimal over the set of regular languages. Additionally, Revuz showed that acyclic automaton (which SA indeed is) can be minimized in linear time [20]. For any $K \in \mathcal{K}_\theta(n, k, d, m)$, let us define the minimal simplex automaton ($\mathcal{M}(\text{SA})$) as the minimal deterministic automaton which recognizes the language L_{K_θ} . In Figure 5, we give the minimal automaton for the complex of Figure 1. Finally, it is possible to get $\mathcal{C}(\text{ST})$ from $\mathcal{M}(\text{SA})$ by duplicating states such that for each node, the labels of all its incoming edges are the same and by moving the labels from the edges to the next node. Now let us look at how to perform operations on $\mathcal{M}(\text{SA})$.

Operations on the Minimal Simplex Automaton

The set of all paths originating from the root are the same in both ST and $\mathcal{M}(\text{SA})$. All operations which involve only traversal along ST are performed with equal (if not better) efficiency in $\mathcal{M}(\text{SA})$ as, for every such operation on ST, we start by traversing from the root. As an example, consider the operation of determining if a simplex σ is in the complex. Let us adapt the algorithm described in [8] to $\mathcal{M}(\text{SA})$. Note that there is a unique path from the initial state which identifies σ in $\mathcal{M}(\text{SA})$. If $\sigma = v_{\ell_0} - \dots - v_{\ell_{d_\sigma}}$, then from the initial state we go through $d_\sigma + 1$ states by following the transitions $\ell_0, \dots, \ell_{d_\sigma}$ in that order. If at some point the requisite transition is not found then, declare that the simplex is not in the complex. Hence, performing any static operation i.e. an operation where we don't change $\mathcal{M}(\text{SA})$ in any way, can be carried out in very much the same way in both $\mathcal{M}(\text{SA})$ and ST, although it might be more efficient for $\mathcal{M}(\text{SA})$ as discussed in subsection 3.1 for $\mathcal{C}(\text{ST})$. Addition and deletion of simplices can be trickier in $\mathcal{M}(\text{SA})$ than in ST. We can always expand $\mathcal{M}(\text{SA})$ to SA, (locally) perform the operation and recompress. If the nature of the operation is itself expensive (i.e. worst-case $\Omega(m)$), then the worst-case cost does not change, which is indeed the case for operations such as removal of a face and edge contraction.

We denote by $|\text{SA}|$ and $|\mathcal{M}(\text{SA})|$, the number of states in SA and $\mathcal{M}(\text{SA})$ respectively. Analysis of $|\mathcal{M}(\text{SA})|$ will be done in section 5, after introducing a new data structure in the next section. This is done to put the impact of compression in better perspective.

4 Maximal Simplex Tree

We define the *Maximal Simplex Tree* $\text{MxST}(K)$ as an induced subgraph of $\text{ST}(K)$. All leaves in the ST corresponding to maximal simplices and the nodes encountered on the path from the root to these leaves are kept in the MxST and the remaining nodes are removed. Figure 6 shows the MxST of the simplicial complex given in Figure 1. In $\text{MxST}(K)$, the leaves are in bijection with the maximal simplices of K . Any path starting from the root provides the vertices of a simplex of K . However, in general, not all simplices in K can be associated to a path from the root in $\text{MxST}(K)$.

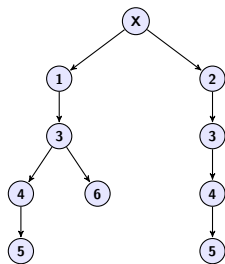


Figure 6 Simplicial Complex of Figure 1 represented using Maximal Simplex Tree.

Table 1 Cost of performing basic operations on MxST.

Operation	Cost
Identifying maximal co-faces of simplex σ / Determining membership of σ	$\mathcal{O}(kd \log n)$
Insertion of a maximal simplex σ	$\mathcal{O}(kd_\sigma \log n)$
Removal of a face	$\mathcal{O}(kd \log n)$
Elementary Collapse	$\mathcal{O}(kd \log n)$
Edge Contraction	$\Theta(kd \log n)$

We note that in MxST we add at most $d+1$ nodes per maximal simplex. Hence, $\text{MxST}(K)$ has at most $k(d+1) + 1$ nodes and at most $k(d+1)$ edges (therefore requiring $\mathcal{O}(kd \log n)$ space). We denote by $|\text{MxST}|$ the number of edges in MxST. Since MxST is a factor of ST, the size of MxST is usually much smaller than the size of ST. Further, it always meets the lower bound of Theorem 1, making it a compact data structure. We discuss below the efficiency of MxST in answering queries.

Operations on the Maximal Simplex Tree

In [8] some important basic operations (with appropriate motivation) have been discussed for ST. We will bound now the cost of these operations using MxST. Note that any node in $\text{MxST}(K)$ has $\mathcal{O}(n)$ children and we can search for a particular child of a node in time $\mathcal{O}(\log n)$ (using red–black trees). We summarize in Table 1, the asymptotic cost of some basic operations and note that it is already better than ST for some operations. Moreover, we can augment the structure of MxST without paying a lot of extra storage space, so that the above operations can be performed more efficiently. This is explained in section 6.

5 Results on Minimization of the Simplex Automaton

In this section we will see some results, both theoretical and experimental on the minimization of SA, i.e. on the extent of compression of the Simplex Tree.

5.1 Bounds on the Number of States of the Minimal Simplex Automaton

We observe below that the number of leaves in ST is large and grows linearly w.r.t. the number of nodes in ST. The proof follows by a simple induction argument on n .

► **Lemma 2.** *If $K \in \mathcal{K}(n, k, d, m)$ then, at least half the nodes of $\text{ST}(K)$ are leaves.*

Differently from ST, $\mathcal{M}(\text{SA})$ has only one leaf. The following lemma shows that $\mathcal{M}(\text{SA})$ has at most half the number of nodes of ST plus one (follows directly from Lemma 2).

► **Lemma 3.** *For any $K \in \mathcal{K}(n, k, d, m)$, $\mathcal{M}(\text{SA}(K))$ has at most $\frac{n}{2} + 1$ states.*

Similar to $\mathcal{M}(\text{SA})$, we may define $\mathcal{M}(\text{MxSA}(K))$ as the minimal DFA which recognizes only maximal simplices as words. Then the following inequality follows:

► **Lemma 4.** *For any pure complex $K \in \mathcal{K}(n, k, d, m)$, $|\mathcal{M}(\text{SA}(K))| \geq |\mathcal{M}(\text{MxSA}(K))|$.*

Proof Sketch. Notice that every maximal simplex corresponds to a path of exactly $d + 1$ transitions and vice versa. Therefore, if all transitions which do not take part in even a single such path are removed, we would obtain the minimized maximal simplex automaton. ◀

In fact, one can prove that for a large class of simplices the equality does not hold. For instance, consider $\mathcal{K}' \subset \mathcal{K}(n, k, d, m)$ such that for any $K \in \mathcal{K}'$ we have that there exists two maximal simplices which have different first letters (i.e. when the simplices are treated as words) but have the same letter at position i , for some i that is not the last position. For this subclass the equality does not hold. Observe also that Lemma 4 holds only for pure simplicial complexes because, if all complexes were allowed, then we will have complexes like in Example 5 where $|\mathcal{M}(\text{SA}(K))| < |\mathcal{M}(\text{MxSA}(K))|$.

► **Example 5.** Consider the simplicial complex on 7 vertices given by the following maximal simplices: a 4-cell 1–2–3–6–7, two triangles 2–3–5 and 4–6–7 and an edge 4–5.

5.2 Conditions for Compression

We will first see a result guaranteeing compression regardless of the labeling of the vertices:

► **Lemma 6.** *For any pure simplicial complex $K \in \mathcal{K}(n, k, d, m)$, we have that $|\mathcal{M}(\text{SA})|$ is always less than $|\text{SA}|$ when $k < d$ and $d \geq 2$.*

Proof Sketch. Since $d > k$ there should exist a free pair (τ, σ) such that τ is a maximal simplex and such that the last two vertices of τ are in σ as well. If s_τ and s_σ are the states recognizing the simplex τ and σ respectively in SA then, they are merged in $\mathcal{M}(\text{SA})$. ◀

In fact, the above result is close to tight: in Example 11 we have a pure simplicial complex with $k = d$ and $|\text{ST}| = |\mathcal{C}(\text{ST})|$. Note that we analyze compression of ST rather than minimization of SA, and we shall continue to do so through out this section because analyzing ST provides better insight into the combinatorial structures which hinder compression.

Intuitively, it seems natural that if the given simplicial complex has a large number of maximal simplices then, regardless of the labeling we should be able to compress some pairs of nodes in MxST. However, Example 7 says otherwise.

► **Example 7.** Consider the simplicial complex on $2n$ vertices of dimension $n/2$ defined by the set of maximal simplices given by:

$$\left\{ g(i) \cup \{g^r(i) + n\} \mid i \in \left\{ 1, 2, \dots, \binom{n}{n/2} \right\}, r \in \{1, 2, \dots, n/2\} \right\}$$

where g is a bijective map from $\{1, 2, \dots, \binom{n}{n/2}\}$ to the set of all simplices on n vertices of dimension $n/2 - 1$ and g^r corresponds to picking the r^{th} vertex (in lexicographic order).

Here $k = \frac{n}{2} \binom{n}{n/2} \approx 2^{n-\frac{1}{2}} \sqrt{n/\pi}$ and there is no compression in MxST. Also note that $|\mathcal{C}(\text{MxST})| < |\text{MxST}|$ does not imply $|\mathcal{C}(\text{ST})| < |\text{ST}|$ as can be seen in Example 8.

► **Example 8.** Consider the simplicial complex on seven vertices given by the maximal simplices: tetrahedron 1-2-4-6 and three triangles 2-4-5, 3-4-5 and 1-4-7.

In Example 11, we saw a simplicial complex of large dimension which cannot be compressed, but this is due to the way the vertices were labeled. Now, we state a lemma which says that there is always a labeling which ensures compression.

► **Lemma 9.** *If $K_\theta \in \mathcal{K}(n, k, d, m)$ with $d > 1$, then we can find a permutation π on $\{1, 2, \dots, n\}$ such that $|\mathcal{M}(\text{SA}(K_{\pi \circ \theta}))| < |\text{SA}(K_{\pi \circ \theta})|$.*

We would have liked to obtain better bounds for the size of $\mathcal{M}(\text{SA})$ through conditions just based on n, k, d and m , but sadly this is a hard combinatorial problem. Also, while there is always a good labeling, we show in section 7 that it is NP-Hard to find it.

5.3 Experiments

We define two parameters here, ρ_{ST} and ρ_{MxST} . The first is given by the ratio of $|\text{ST}|$ and $|\mathcal{C}(\text{ST})|$ and the second by the ratio of $|\text{MxST}|$ and $|\mathcal{C}(\text{MxST})|$.

Data Set 1: The set of points were obtained through sampling of a Klein bottle in \mathbb{R}^5 and construct the Rips Complex (see [8] for definition) with parameter α using libraries provided by the GUDHI project [16] on input of various values for α . We record in Table 2, $|\mathcal{C}(\text{ST})|$ and $|\mathcal{C}(\text{MxST})|$ for the various complexes constructed.

First, observe that for this set of data, $\frac{\log k}{\log n}$ is small (which is expected for Rips complexes) and thus $|\text{MxST}|$ would be considerably smaller than $|\text{ST}|$. Also, note that MxST hardly compresses but as α increases, ρ_{ST} increases quite fast. This indicates that compression strongly exploits the combinatorial redundancy of ST (i.e. storing each simplex explicitly through a node) in order to compress efficiently.

■ **Table 2** Analysis of experiments on Data Set 1.

No	n	α	d	k	$ \text{ST} = m - 1$	$ \text{MxST} $	$ \mathcal{C}(\text{ST}) $	ρ_{ST}	$ \mathcal{C}(\text{MxST}) $	ρ_{MxST}
1	10,000	0.15	10	24,970	604,572	96,104	218,452	2.77	90,716	1.06
2	10,000	0.16	13	25,410	1,387,022	110,976	292,974	4.73	104,810	1.06
3	10,000	0.17	15	27,086	3,543,582	131,777	400,426	8.85	123,154	1.07
4	10,000	0.18	17	27,286	10,508,485	149,310	524,730	20.03	137,962	1.08

■ **Table 3** Analysis of experiments on Data Set 2.

No	n	p	d	k	$ \text{ST} = m - 1$	$ \text{MxST} $	$ \mathcal{C}(\text{ST}) $	ρ_{ST}	$ \mathcal{C}(\text{MxST}) $	ρ_{MxST}
1	25	0.8	17	77	315,369	587	467	537.3	121	4.85
2	30	0.75	18	83	4,438,558	869	627	7,079.0	134	6.49
3	35	0.7	17	181	3,841,590	1,592	779	4,931.4	245	6.50
4	40	0.6	19	204	9,471,219	1,940	896	10,570.6	276	7.03
5	50	0.5	20	306	25,784,503	2,628	1,163	22,170.7	397	6.62

Data Set 2: All experiments conducted above are for Rips complexes with $\frac{d}{n}$ small. We now check the extent of compression for simplicial complexes with large $\frac{d}{n}$. To the aim, we look at flag complexes generated using a random graph $G_{n,p}$ on n vertices where a pair of vertices share an edge with probability p , and record in Table 3, $|\mathcal{C}(\text{ST})|$ and $|\mathcal{C}(\text{MxST})|$ for the various complexes constructed.

Here we observe staggering values for ρ_{ST} which only seems to grow as larger simplicial complexes were constructed. This is primarily because random simplicial complexes don't behave like pathological simplicial complexes which hinder compression; it is rare that there exists both large cliques and a large fraction of low dimensional maximal simplices.

6 Simplex Array List

In this section, we build a new data structure which is a hybrid of ST and MxST. The *Simplex Array List* $\text{SAL}(K)$ is a (rooted) directed acyclic graph on at most $k \binom{d(d+1)}{2} + 1$ nodes with maximum out-degree d , which is obtained by modifying MxST or constructed from the maximal simplices of K . Intuitively, SAL is representing K by storing all the edges of K explicitly as nodes in $\text{SAL}(K)$ and the edges in $\text{SAL}(K)$ are used to capture the incidence relations between simplices. More precisely, a path of length j in $\text{SAL}(K)$ corresponds to a unique j -simplex in K . We describe the construction of SAL below.

6.1 Construction

We will first see how to obtain SAL from MxST by performing three operations which we define below.

1. **Unprefixing (\mathcal{U}):** Excluding the root and the leaves, for every node v in MxST with outdegree d_v , duplicate it into d_v nodes with outdegree 1, (one copy of v for each of its children) by starting from the parents of the leaves and recursively moving up in the tree.
2. **Transitive Closure (\mathcal{T}):** For every pair of nodes (u, v) in $\mathcal{U}(\text{MxST})$ (u not being the root), if there is a path from u to v , then add an edge from u to v in $\mathcal{T}(\mathcal{U}(\text{MxST}))$ (if it doesn't already exist).
3. **Expanding Representation (\mathcal{R}):** For every node v in $\mathcal{T}(\mathcal{U}(\text{MxST}))$ with outdegree d_v , duplicate it into d_v nodes with outdegree 1, i.e. one copy of v for each of its children, by

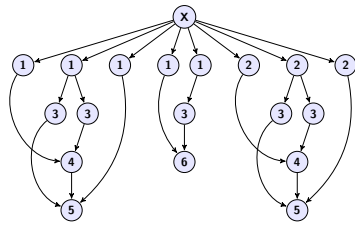


Figure 7 Simplicial Complex of Figure 1 represented using $\mathcal{R}(\mathcal{T}(\mathcal{U}(\text{MxST})))$.

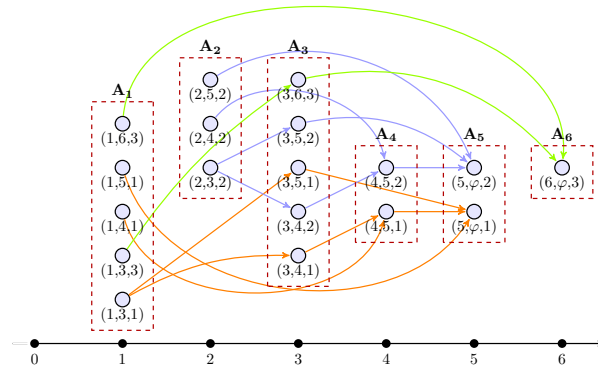


Figure 8 Simplex Array List for complex in Figure 1 embedded on the number line.

starting from the children of the root and recursively moving down to children of smallest label. Therefore, if we append i empty labels at the end of each maximal simplex, \mathcal{R} applied i times will give a graph where every node uniquely represents a i -simplex.

SAL can be seen as $\mathcal{R}(\mathcal{T}(\mathcal{U}(\text{MxST})))$ and each node uniquely represents an edge in the complex. Figure 7 shows $\mathcal{R}(\mathcal{T}(\mathcal{U}(\text{MxST})))$ of the simplicial complex given in Figure 1.

We will now see an equivalent construction of SAL from its maximal simplices and it is this construction we will use to perform operations. For a given maximal simplex $\sigma = v_{\ell_0} \cdots v_{\ell_j}$, associate a unique key between 1 and k generated using a hash function \mathcal{H} and then introduce $\frac{j(j+1)}{2} + 1$ new nodes in SAL. We build a set of $\frac{j(j+1)}{2} + 1$ labels and assign uniquely a label to each node. The set of labels is defined as the union of the following two sets: $S_1 = \{(\ell_i, \ell_{i'}, \mathcal{H}(\sigma)) \mid i \in \{0, 1, \dots, j-1\}, i' \in \{i+1, \dots, j\}\}$ and $S_2 = \{(\ell_j, \varphi, \mathcal{H}(\sigma))\}$, where φ denotes an empty label (cf. Figure 8 for an example). We introduce an edge from node with label $(\ell_p, \ell_{p'}, \mathcal{H}(\sigma))$ to node with label $(\ell_q, \ell_{q'}, \mathcal{H}(\sigma))$ if and only if $p' = q$. Additionally, we introduce an edge from every node with label $(\ell_p, \ell_j, \mathcal{H}(\sigma))$ in S_1 to the node with label $(\ell_j, \varphi, \mathcal{H}(\sigma))$ in S_2 . Thus, in SAL we represent a maximal j -simplex using a connected component containing $|S_1| + |S_2| = \frac{j(j+1)}{2} + 1$ nodes and $\frac{j(j^2+5)}{6}$ directed edges. To perform basic operations efficiently, we embed SAL on the number line such that for every $i \in \{1, 2, \dots, n\}$ on the number line we have an array A_i of nodes which has labels of the form (i, i', z) for some $z \in \{1, \dots, k\}$ and $i' \in \{i+1, \dots, n, \varphi\}$. Sort each A_i based on i' and in case of ties, sort them based on z .

If the root is removed in $\mathcal{R}(\mathcal{T}(\mathcal{U}(\text{MxST})))$, the graph we get is the same as the one described in the previous paragraph. Labels (as described above) for the nodes in $\mathcal{R}(\mathcal{T}(\mathcal{U}(\text{MxST})))$ can be easily given by just looking at the vertex represented by the node, and its children. Further, the number of nodes and number of edges in SAL are both invariant over the labeling of the vertices because SAL is constructed from $\mathcal{U}(\text{MxST})$.

6.2 Some Observations about the Simplex Array List

$\text{SAL}(K)$ has at most $k \left(\frac{d(d+1)}{2} + 1 \right)$ nodes. Also, for each maximal simplex of dimension d_σ , the outdegree of any node in the connected component corresponding to the maximal simplex, is at most d_σ . Therefore, the total number of edges in $\text{SAL}(K)$ is at most $k \left(\frac{d^2(d+1)}{2} + d \right)$.

Hence, the space required to store $SAL(K)$ is $\mathcal{O}(kd^3 \log n)$. Also, unless otherwise stated $|SAL|$ refers to number of edges in SAL .

We now see that, differently from $MxST$, the simplices of K are all associated with paths in $SAL(K)$. We say a path p is associated to a simplex σ if the sequence of numbers obtained by looking at the corresponding nodes which are embedded on the number line along p are exactly the labels of the vertices of σ in lexicographic order.

► **Lemma 10.** *Any path in $SAL(K)$ is associated to a simplex of K and any simplex of K is associated to at least one such path.*

Observe that several paths can provide the same simplex since a simplex may appear in several maximal simplices. Hence, the vertices of a given simplex cannot be accessed in a deterministic way. The previous lemma together with this observation implies that SAL is a non-deterministic finite automaton (NFA). NFA are a natural generalization of DFA. The size of a NFA is smaller than that of a DFA detecting the same language, but the operations on NFA take in general more time. We demonstrate the above fact using Example 11.

► **Example 11.** Let $K \in \mathcal{K}(2k+1, k, k, m)$ be defined on the vertices $\{1, \dots, 2k+1\}$ and the set of maximal simplices be given by $\{(\{1, \dots, k+1\} \setminus \{i\}) \cup \{k+1+i\} \mid 1 \leq i \leq k\}$.

Thus $SAL(K)$ has $\frac{k^2(k+1)}{2} + k$ nodes while $\mathcal{M}(SA(K))$ has at least 2^k states (all states reached after reading the words $s \subseteq \{1, \dots, k\}$ are pairwise distinct). Moreover, this motivates the need for considering SAL over $\mathcal{M}(SA)$, as the gap in their sizes can be exponential.

Building $SAL(K)$ can be seen as partially compressing the $ST(\sigma)$ associated to each maximal simplex σ (where σ and its subfaces are seen as a subcomplex). Compressing $ST(\sigma)$ will lead to a subtree which is exactly the same as the transitive closure of $MxST(\sigma)$. Therefore, collecting all $\mathcal{C}(ST)(\sigma)$ for all maximal simplices σ and merging the roots is the same as $\mathcal{T}(\mathcal{U}(MxST(K)))$. Now applying \mathcal{R} on $\mathcal{T}(\mathcal{U}(MxST(K)))$ can be seen as an act of uncompression. We apply \mathcal{R} once to ensure that for every node, all its children represent the same vertex and thus belong to the same A_i . If \mathcal{R} is applied multiple times then, it is equivalent to duplicating nodes (seen as an act of uncompression) to get all children of a node closer together inside A_i . Next, we discuss below how to perform operations in SAL at least as efficiently as in ST .

6.3 Operations on the Simplex Array List

Let us now analyze the cost of performing basic operations on SAL (the motivation behind these operations are well described in [8]). Denote by $\Gamma_j(\sigma, \tau)$ the number of maximal simplices that contain a j -simplex τ which is in σ . Define $\Gamma_j(\sigma) = \max_{\tau} \Gamma_j(\sigma, \tau)$ and $\Gamma_j = \max_{\sigma \in K} \Gamma_j(\sigma)$. It is easy to see that $k \geq \Gamma_0 \geq \Gamma_1 \geq \dots \geq \Gamma_d = 1$. In the case of SAL , we are interested in the value of Γ_1 which we use to estimate the worst-case cost of basic operations in SAL .

Membership of Simplex. To determine membership of $\sigma = v_{\ell_0} \dots v_{\ell_{d\sigma}}$ in K , first determine the contiguous subarray of $A_{\ell_0}, \dots, A_{\ell_{d\sigma}}$, say $B_{\ell_0}, \dots, B_{\ell_{d\sigma}}$ such that every B_{ℓ_i} contains all nodes with labels of the form (ℓ_i, ℓ_{i+1}, z) , for some z (B_{ℓ_i} 's indeed form a contiguous subarray because of the way elements in A_{ℓ_i} were sorted). We emphasize here that we determine each B_{ℓ_i} only by its starting and ending location in A_{ℓ_i} and do not explicitly read the contents of each element in B_{ℓ_i} . Thus, if P is a projection function such that $P((\ell_i, \ell_{i+1}, z)) = z$ then, we see each $P(B_{\ell_i})$ as a subset of $\{1, \dots, k\}$ because the only part of the label that distinguishes

two elements in B_{ℓ_i} is the hash value of the maximal simplex. Now we have $\sigma \in K$ if and only if $\bigcap_{0 \leq i \leq d_\sigma} P(B_{\ell_i}) \neq \emptyset$. This is because if $\sigma \in K$ then, from Lemma 10 there should exist a path corresponding to this simplex which would imply $\bigcap_i P(B_{\ell_i}) \neq \emptyset$, and if $m \in \bigcap_i P(B_{\ell_i})$, then σ is a face of m . Computing the intersection can be done in $\mathcal{O}(\gamma d_\sigma \log \zeta)$ time, where $\gamma = \min_i |B_{\ell_i}|$ and $\zeta = \max_i |A_{\ell_i}|$. Computing the subarrays can be done in $\mathcal{O}(d_\sigma \log \zeta)$ time. Thus total running time is $\mathcal{O}(d_\sigma(\gamma \log \zeta + \log \zeta)) = \mathcal{O}(d_\sigma \Gamma_1 \log(kd))$.

For example, consider the SAL of figure 8 and the task of checking membership of $\sigma = 2 - 3 - 5$ in the complex of figure 1. Then, we have $B_2 = \{(2, 3, 2)\}$, $B_3 = \{(3, 5, 1), (3, 5, 2)\}$, and $B_5 = \{(5, \varphi, 1), (5, \varphi, 2)\}$. We see each $P(B_i)$ as a subset of $\{1, 2, 3\}$ as follows: $P(B_2) = \{2\}$, $P(B_3) = \{1, 2\}$, and $P(B_5) = \{1, 2\}$. Clearly $\bigcap_i P(B_i) = \{2\}$ and σ is indeed a face of the second maximal simplex $2 - 3 - 4 - 5$.

Insertion. Suppose we want to insert a maximal simplex σ then, building a connected component takes time $\mathcal{O}(d_\sigma^3)$. Updating the arrays A_i takes time $\mathcal{O}(d_\sigma^2 \log \zeta)$. Next, we have to check if there exist maximal simplices in K which are now faces of σ , and remove them. We consider every edge σ_Δ in σ and compute Z_Δ the set of all maximal simplices which contain σ_Δ (which can be done in time $\mathcal{O}(d_\sigma^3 \Gamma_1 \log(kd))$). Then, we compute $\bigcup_{\sigma_\Delta \in \sigma} Z_\Delta$ whose size is at most $d_\sigma^2 \Gamma_1$ and check if any of these maximal simplices are faces in σ (can be done in $\mathcal{O}(d_\sigma^3 \Gamma_1)$ time). To remove all such faces of σ which were previously maximal takes at the most $\mathcal{O}(d_\sigma^4 \Gamma_1)$ time. Therefore, total time for insertion is $\mathcal{O}(d_\sigma^3 \Gamma_1 (d_\sigma + \log(kd)))$.

Removal. To remove a face σ , obtain the maximal simplices which contain it (can be done in $\mathcal{O}(d_\sigma \Gamma_1 \log(kd))$ time) and for each of them make d_σ copies of the connected component, and in the i^{th} copy delete all nodes with label (σ_i, x, y) for some x, y , and where σ_i denotes the label of the i^{th} vertex of σ . Thus, the total running time is $\mathcal{O}(d_\sigma d^3 \Gamma_1 \log(kd))$.

Elementary Collapse. Given a pair of simplices (σ, τ) , first check if it is a free pair. This is done by obtaining a list of all maximal simplices which contain σ , through a membership query (costs $\mathcal{O}(d_\sigma \Gamma_1 \log(kd))$ time) and then checking if τ is the only member in that list. If yes, remove σ (and add its facets). This takes time $\mathcal{O}(d_\sigma^4)$. Thus, total running time is $\mathcal{O}(d_\sigma(d_\sigma^3 + \Gamma_1 \log(kd)))$.

Edge Contraction. Here we cannot do better than rebuilding the entire SAL (as in MxST) and therefore the cost of the operation is $\mathcal{O}(kd^3)$. However, it's not really bad as size of SAL is already smaller than the size of ST which takes time proportional to $\mathcal{O}(md + k2^d \log n)$ to perform edge contraction.

We summarize in Table 4 the asymptotic cost of the basic operations discussed above and compare it with ST and MxST, through which the efficiency of SAL is established.

Filtration. We know from Lemma 10 that to every simplex in the complex, we can associate a set of paths in SAL. This can be used to store filtration in some cases. However, if a data structure needs to support all possible filtrations then, it can be provably shown that there

* We would like to recapitulate here the lower bound from Theorem 1 of $\Omega(kd \log n)$.
 † The space needed to represent ST is $\Theta(m \log n)$ which is written as $\mathcal{O}(k2^d \log n)$ to help in comparison.

■ **Table 4** Cost of performing basic operations on SAL in comparison with ST and MxST.

	ST	MxST	SAL
Storage*	$\mathcal{O}(k2^d \log n)^\dagger$	$\mathcal{O}(kd \log n)$	$\mathcal{O}(kd^3(\log n + \log k))$
Membership of a simplex σ	$\mathcal{O}(d_\sigma \log n)$	$\mathcal{O}(kd \log n)$	$\mathcal{O}(d_\sigma \Gamma_1 \log(kd))$
Insertion of a maximal simplex σ	$\mathcal{O}(2^{d_\sigma} d_\sigma \log n)$	$\mathcal{O}(kd_\sigma \log n)$	$\mathcal{O}(d_\sigma^3 \Gamma_1 (d_\sigma + \log(kd)))$
Removal of a face	$\mathcal{O}(m \log n)$	$\mathcal{O}(kd \log n)$	$\mathcal{O}(d_\sigma d^3 \Gamma_1 \log(kd))$
Elementary Collapse	$\mathcal{O}(2^{d_\sigma} \log n)$	$\mathcal{O}(kd \log n)$	$\mathcal{O}(d_\sigma (d_\sigma^3 + \Gamma_1 \log(kd^2)))$
Edge Contraction	$\mathcal{O}(k2^d \log n)$	$\mathcal{O}(kd \log n)$	$\mathcal{O}(kd^3)$

■ **Table 5** Values of Γ_0 , Γ_1 , Γ_2 , and Γ_3 for the simplicial complexes generated from Data Set 1.

No	n	α	d	k	m	Γ_0	Γ_1	Γ_2	Γ_3	SAL
1	10,000	0.15	10	24,970	604,573	62	53	47	37	424,440
2	10,000	0.16	13	25,410	1,387,023	71	61	55	48	623,238
3	10,000	0.17	15	27,086	3,543,583	90	67	61	51	968,766
4	10,000	0.18	17	27,286	10,508,486	115	91	68	54	1,412,310

is no better way to do so, than by storing the filtration value explicitly for each simplex in the complex. Therefore one cannot hope to find a more compact representation than ST in order to support all possible filtrations.

Performance of SAL. Plainly, if the number of maximal simplices is small (i.e. can be treated as a constant), SAL and MxST are very efficient data structures and this is indeed the case for a large class of complexes encountered in practice as discussed in section 2.

Remarkably, even if k is not small but d is small then, SAL is a compact data structure as given by the lower bound in Theorem 1. This is because $\mathcal{O}(kd^3(\log n + \log k))$ bits are sufficient to represent SAL and the lower bound is met when d is fixed (as it translates to needing $\mathcal{O}(k \log n)$ bits to represent SAL). Also, it is worth noting here that Γ_0 is usually a small fraction of k and since Γ_1 is at most Γ_0 , the above operations are performed considerably faster than in MxST where almost always the only way to perform operations is to traverse the entire tree. Indeed SAL was intended to be efficient in this regard as even if k is not small the construction of SAL replaces the dependence on k by a dependence on a more local parameter Γ_1 that reflects some “local complexity” of the simplicial complex. As a simple demonstration, we estimated $\Gamma_0, \Gamma_1, \Gamma_2$, and Γ_3 for the simplicial complexes of Data Set 1 (see section 5.3). These values are recorded in Table 5.

It is interesting to note that $|\text{SAL}|$ is larger than $|\mathcal{C}(\text{ST})|$ but much smaller than $|\text{ST}|$. This is expected, as SAL promises to perform basic operations more efficiently than ST while compromising slightly on size. Further our intuition as described previously was that Γ_0 should be much smaller than k , and this is supported by the above results. Also, we note that for larger simplicial complexes such as complexes No 3 and 4, there is a significant gap between Γ_0 and Γ_1 . Since complexity of basic operations using SAL is parametrized by Γ_1 (and not Γ_0), the above results support our claim that SAL is an efficient data structure.

Local Sensitivity of Simplex Array List. We note here that while the cost of basic operations are bounded using Γ_1 , we could use local parameters such as γ and Z_Δ (see previous paragraphs on Membership of Simplex and Insertion for definition) to get a better estimate

on the cost of these operations. γ captures local information about a simplex σ sharing an edge with other maximal simplices of the complex. More precisely, it is the minimum, over all edges of σ , of the largest number of maximal simplices that contain the edge. If σ has an edge which is contained in a few maximal simplices then, γ is very small. Z_Δ captures another local property of a simplex σ – the set of all maximal simplices that contain the edge σ_Δ . Therefore, SAL is indeed sensitive to the local structure of the complex.

7 Labeling Dependency

In this section, we discuss how the labeling of the vertices affects the size of the data structures discussed in this paper. In particular, both MxST and $\mathcal{M}(\text{SA})$ are not label invariant like ST and SAL. To see this, consider a simplicial complex which contains a maximal triangle and a maximal tetrahedron, sharing an edge. We could label the triangle and tetrahedron as 1–2–3 and 1–2–4–5, or as 1–3–4 and 2–3–4–5 respectively. Note that the two labelings give two $\mathcal{M}(\text{SA})$ (and MxST) of different sizes. We skip all the proofs in this section and instead direct the reader to the full version of the paper [7].

First, we formalize the label ordering problem on MxST (and $\mathcal{C}(\text{MxST})$, $\mathcal{C}(\text{ST})$, and $\mathcal{M}(\text{SA})$) as follows: Given an integer α and a simplicial complex $K_\theta \in \mathcal{K}_\theta(n, k, d, m)$, does there exist a permutation π of $1, 2, \dots, n$ such that $|\text{MxST}(K_{\pi \circ \theta})| \leq \alpha$ (similarly we ask $|\mathcal{C}(\text{MxST}(K_{\pi \circ \theta}))| \leq \alpha$, $|\mathcal{C}(\text{ST}(K_{\pi \circ \theta}))| \leq \alpha$, and $|\mathcal{M}(\text{SA}(K_{\pi \circ \theta}))| \leq \alpha$)? Let us refer to this problem as $\text{MxSTMINIMIZATION}(K_\theta, \alpha)$ (similarly we have $\text{CMxSTMINIMIZATION}(K_\theta, \alpha)$, $\text{CSTMINIMIZATION}(K_\theta, \alpha)$, and $\text{MSAMINIMIZATION}(K_\theta, \alpha)$). We have the following results:

► **Theorem 12** ([9]). *MxSTMINIMIZATION is NP-Complete.*

► **Theorem 13.** *CMxSTMINIMIZATION, CSTMINIMIZATION, and MSAMINIMIZATION are all NP-Complete.*

8 Discussion and Conclusion

In this paper, we introduced a compression technique for the Simplex Tree without compromising on functionality. Additionally, we have proposed two new data structures for simplicial complexes – the Maximal Simplex Tree and the Simplex Array List. We observed that the Minimal Simplex Automaton is generally smaller than the Simplex Automaton. Further, we showed that the Maximal Simplex Tree is compact and that the Simplex Array List is efficient (and compact when d is fixed). This is summarized in Table 4.

The transitive closure of MxST may have a node, with as many as kd outgoing edges to neighbors containing the same label. SAL reduces the number of outgoing edges to such neighbors with the same label from kd to d , making it much more powerful. In short, it reduces the non-determinism of their equivalent automaton representation. Also, most complexes observed in practice have k to be a low degree polynomial in n . Example 11 and Lemma 6 both deal with complexes where k is small. Further, all hardness results in section 7 are for complexes of dimension at most 2. Thus, complexes where either k or d is small are interesting to study and for these cases, SAL is very efficient.

Trie Compression, like that of $\mathcal{M}(\text{SA})$, are efficient techniques when the trie is assumed to be static. However, over the last decade, this has been extended using Dynamic Minimization – the process of maintaining an automaton minimal when insertions or deletions are performed. This has been well studied in [21], and extended to acyclic automata in [12] which would be of particular interest to us.

Another direction, is to look at approximate data structures for simplicial complexes, i.e. we store almost all the simplices (introducing an error) and gain efficiency in compression (i.e. little storage). This is a well explored topic in automata theory called hyperminimization [18] and since our language is finite, k -minimization [5] and cover automata [10] might give efficient approximate data structures by hyperminimizing SA.

Theorem 13 provides a new dimension to the hardness results obtained by Comer and Sethi in [11]. It would be worth exploring this direction further. Also, it would be interesting to find approximation algorithms for MSAMINIMIZATION. Finally proving better bounds on extent of compression remains an open problem and may be geometric constraints will eliminate pathological examples which hinder in proving good bounds on compression.

Acknowledgement. We would like to thank Eylon Yogev for helping with carrying out some experiments.

References

- 1 A. Acharya, H. Zhu, and K. Shen: Adaptive Algorithms for Cache-efficient Trie Search, *In Workshop on Algorithm Engineering and Experimentation ALENEX 99, Baltimore*, 1999.
- 2 A. Andersson and S. Nilsson: Improved Behaviour of Tries by Adaptive Branching, *In Information Processing Letters*, Vol 46, pages 295–300, 1993.
- 3 A. W. Appel and G. J. Jacobson: The world’s fastest scrabble program, *In Communications of the ACM*, Vol 31, 1988.
- 4 D. Attali, A. Lieutier, and D. Salinas: Efficient data structure for representing and simplifying simplicial complexes in high dimensions. *In International Journal of Computational Geometry and Applications*, 22(4), pages 279–303, 2012.
- 5 A. Badr, V. Geffert, and I. Shipman: Hyper-minimizing minimized deterministic finite state automata. *In RAIRO Theoretical Informatics and Applications*, pages 69–94, 2009.
- 6 L. J. Billera and A. Björner: Face numbers of polytopes on complexes. *In Handbook of Discrete and Computational Geometry*, CRC Press, pages 291–310, 1997.
- 7 J.-D. Boissonnat, Karthik C. S., and S. Tavenas: Building efficient and compact data structures for simplicial complexes. <http://arxiv.org/abs/1503.07444>.
- 8 J.-D. Boissonnat and C. Maria: The Simplex Tree: An Efficient Data Structure for General Simplicial Complexes. *In Algorithmica* 70(3), pages 406–427, 2014.
- 9 J.-D. Boissonnat and D. Mazauric: On the complexity of the representation of simplicial complexes by trees. <http://hal.inria.fr/hal-01089846>
- 10 C. Câmpeanu, N. Sântean, and S. Yu: Minimal cover-automata for finite languages. *In Theoretical Computer Science* 267(1–2), pages 3–16, 2001.
- 11 D. Comer and R. Sethi: Complexity of Trie Index Construction, *In Proceedings of Foundations of Computer Science*, pages 197–207, 1976.
- 12 J. Daciuk, S. Mihov, B. Watson, and R. Watson: Incremental construction of minimal acyclic finite-state automata. *In Comput. Linguist.*, Volume 26, pages 3–16, 2000.
- 13 D. Eppstein, M. Löffler, and D. Strash: Listing All Maximal Cliques in Sparse Graphs in Near-Optimal Time, *In ISAAC (1)*, pages 403–414, 2010.
- 14 M. Golumbic: Algorithmic Graph Theory and Perfect Graphs. *In Academic Press*, 2004.
- 15 M. Grohe, S. Kreutzer, and S. Siebertz: Characterisations of Nowhere Dense Graphs, *In FSTTCS 13*, pages 21–40, 2013.
- 16 GUDHI – Geometric Understanding in Higher Dimenions. <https://project.inria.fr/gudhi/>
- 17 J. Hopcroft: An $n \log n$ algorithm for minimizing states in a finite automaton. *In Theory of machines and computations*, pages 189–196, 1971.

- 18 A. Maletti: Notes on hyper-minimization. *In Proceedings 13th International Conference Automata and Formal Languages*, pages 34–49, 2011.
- 19 A. Nerode: Linear Automaton Transformations, *In Proceedings of the American Mathematical Society*, Volume 9, pages 541–544, 1958.
- 20 D. Revuz: Minimisation of acyclic deterministic automata in linear time, *In Theoretical Computer Science, Volume 92, Issue 1*, pages 181–189, 1992.
- 21 K. Sgarbas, N. Fakotakis, and G. Kokkinakis: Optimal insertion in deterministic DAWGs. *In Theoretical Computer Science*, pages 103–117, 2003.

Shortest Path to a Segment and Quickest Visibility Queries

Esther M. Arkin¹, Alon Efrat², Christian Knauer³,
Joseph S. B. Mitchell¹, Valentin Polishchuk⁴, Günter Rote⁵,
Lena Schlipf⁵, and Topi Talvitie⁶

- 1 Department of Applied Math and Statistics, Stony Brook University, USA
- 2 Department of Computer Science, the University of Arizona, USA
- 3 Institute of Computer Science, Universität Bayreuth, Germany
- 4 Communications and Transport Systems, ITN, Linköping University, Sweden
- 5 Institute of Computer Science, Freie Universität Berlin, Germany
- 6 Department of Computer Science, University of Helsinki, Finland

Abstract

We show how to preprocess a polygonal domain with a fixed starting point s in order to answer efficiently the following queries: Given a point q , how should one move from s in order to *see* q as soon as possible? This query resembles the well-known shortest-path-to-a-point query, except that the latter asks for the fastest way to *reach* q , instead of seeing it. Our solution methods include a data structure for a different generalization of shortest-path-to-a-point queries, which may be of independent interest: to report efficiently a shortest path from s to a query *segment* in the domain.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases path planning, visibility, query structures and complexity, persistent data structures, continuous Dijkstra

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.658

1 Introduction

Finding shortest paths is a classical problem in computational geometry, and efficient algorithms are known for computing the paths both in simple polygons and polygonal domains with holes; see [33, 34] for surveys. In the *query version* of the problem one is given a fixed source point s in the domain, and the goal is to preprocess the domain so that the length of a shortest path from s to a query point q can be reported efficiently. The problem is solved by building the *shortest path map* (SPM) from s – the decomposition of the free space into cells such that for all points q within a cell the shortest s - q path is combinatorially the same, i.e., traverses the same sequence of vertices of the domain.

The query in the shortest path problem can be stated as

Shortest path query: *Given a query point q lying in the free space, how should one move, starting from s , in order to **reach** q as soon as possible?*

Queries like this arise in surveillance and security, search and rescue, aid and delivery, and various other applications of the shortest path problem. In this paper we introduce and study a related problem that has a very similar query:

Quickest visibility query (QVQ): *Given a query point q lying in the free space, how should one move, starting from s , in order to **see** q as soon as possible?*



© Esther M. Arkin, Alon Efrat, Christian Knauer, Joseph S. B. Mitchell, Valentin Polishchuk, Günter Rote, Lena Schlipf, Topi Talvitie;
licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).
Eds.: Lars Arge and János Pach; pp. 658–673



Leibniz International Proceedings in Informatics

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

Such a query may be natural in applications in which it is important to see (or become seen by) the query point – for inspection purposes, for coming within a shooting range, for establishing communication, etc. In contrast with shortest path queries, such quickest visibility queries have not been studied before, with the single exception of [28] where the problem was considered in simple polygons (in Section 5 we give improved results for this important special case).

The other variant of the shortest path query problem, which we consider in this paper, deals with *segments* instead of points as query objects:

Shortest path to a segment query (SPSQ): *Given a query segment ab lying in the free space, how should one move, starting from s , in order to reach ab as soon as possible?*

To our knowledge such queries have not been studied before. We show that in nearly-quadratic time a nearly-quadratic-size data structure can be built to answer SPSQ in polylogarithmic time (logarithmic-time query can be achieved with nearly-cubic preprocessing time and space). We apply SPSQ as a subroutine in an algorithm for QVQ: given the query point q in an instance of QVQ, build the visibility polygon of q and use SPSQ for each “window” (edge running through the free space) of the polygon to choose the best window through which q can be seen.

1.1 Notation

Let D denote the given polygonal domain; let n, h be the number of vertices and holes of D , respectively. Assume that no two vertices of D have the same x - or y -coordinate. Two points $p, q \in D$ see each other if the segment pq fully belongs to the domain (we consider D as a closed set, so that pq may go through a vertex of D or otherwise overlap with the boundary of the domain). Let E be the size of the *visibility graph* of D – the graph on vertices of D with edges between pairs of mutually visible vertices (i.e., pairs of vertices that can be connected with a single link). We also introduce an additional definition related to “3-link visibility” between vertices of D : let Π be the number of pairs of vertices that can be connected by a right-turning 3-link path that makes 90° turns at both of its bends (refer to Fig. 3).

Let P, S, Q denote the preprocessing time, size of the built data structure and query time, respectively, for an algorithm for answering quickest visibility queries (QVQ) in D . The query point will be generally denoted by q . Let $V(q)$ denote the *visibility polygon* of q (the set of points seen by q); let K denote the complexity (the number of sides) of $V(q)$. We use P_v, S_v, Q_v to denote the preprocessing time, size of the structure and query time for an algorithm for the problem of building $V(q)$. Finally, we denote by P_s, S_s, Q_s the corresponding parameters of an algorithm for SPSQ – the problem of reporting length of the shortest path to a query segment lying in D .

Slightly abusing the terminology, we will not differentiate between the two variants of path queries: reporting the *length* of the optimal path and outputting the path itself; similarly to other path query problems, the latter can usually be done straightforwardly (by following back pointers) in additional time proportional to the combinatorial complexity of the path.

1.2 Related work

A shortest path between two points in a simple polygon ($h = 0$) can be found in linear time [7, 30]. The query version (i.e., building the SPM) can be solved within the same time [19];

using the SPM, the length of the (unique) shortest path to a query point can be reported in time $O(\log n)$.

For polygons with holes the *continuous Dijkstra* paradigm [32] leads to an $O(n \log n)$ time algorithm [25] for building the SPM, by propagating a *wave* (which we call the *p-wave*) from s through the free space at unit speed, so that the points reached by the wavefront at any time τ are exactly the points at geodesic distance τ from s (see, e.g., Fig. 4 where gray shows the area covered by the p-wave, and Fig. 6 (left), where the p-wave is blue). At any time during the propagation, the wavefront consists of a sequence of *wavelets* – circular arcs centered on vertices of D , called *generators* of the wavelets; the radius of each arc grows at unit speed. Boundaries between adjacent wavelets trace edges of the SPM (the edges are called *bisectors*, and are further classified in [13] as “walls” and “windows”¹ depending on whether there exist two homotopically distinct shortest paths to points on the bisector); this way the algorithm also builds the SPM which allows one to answer the shortest path queries in $O(\log n)$ time per query. Vertices of the SPM are vertices of D and *triple points*, at which three edges of the map meet (w.l.o.g. four edges of SPM never meet at the same point); the overall complexity of the SPM is linear [25]. Using the continuous Dijkstra method, the quickest way to see a point and the shortest path to a segment (i.e., solutions to single-shot, non-query versions of QVQ and SPSQ) can be found in $O(n \log n)$ time by simply declaring $V(q)$ and the segment as obstacles and waiting until the p-wave hits them.

Computing visibility from a point was first studied in simple polygons, for which an $O(n)$ -time solution was given already in 1987 [27]. For polygons with holes an optimal, $O(n+h \log h)$ -time algorithm was presented by Heffernan and Mitchell [23]. The query version of the problem has been well studied too: For simple polygons Guibas, Motwani and Raghavan [20] and Bose, Lubiw and Munro [3] gave algorithms with $P_v = O(n^3 \log n)$, $S_v = O(n^3)$ and $Q_v = O(\log n + K)$; Aronov, Guibas, Teichman and Zhang [2] achieve $P_v = O(n^2 \log n)$, $S_v = O(n^2)$ and $Q_v = O(\log^2 n + K)$. For polygons with holes Zarei and Ghodsi [42] achieve $P_v = O(n^3 \log n)$, $S_v = O(n^3)$, $Q_v = O(K + \min(h, K) \log n)$; Inkulu and Kapoor [26] combine and extend the approaches from [42] and [2] presenting algorithms with several tradeoffs between P_v , S_v and Q_v , in particular, with $P_v = O(n^2 \log n)$, $S_v = O(n^2)$, $Q_v = O(K \log^2 n)$ (see also [9], as well as [31] giving $P_v = O(n^2 \log n)$, $S_v = O(n^2)$, $Q_v = O(K + \log^2 n + h \log(n/h))$). A recent paper by Bunqui et al. [4] reports on practical implementation of visibility computation in an upcoming CGAL [6] package.

More generally, both visibility and shortest paths computations are textbook subjects in computational geometry – see, e.g., the respective chapters in the handbook [17] and the books [38, 15]. Visibility meets path planning in a variety of geometric computing tasks. Historically, the first approach to finding shortest paths was based on searching the visibility graph of the domain. Visibility is vital also in computing *minimum-link* paths, i.e., paths with fewest edges [37, 41, 36]. Last but not least, “visibility-driven” route planning is the subject in *watchman route* problems [5, 12, 35, 39, 11] where the goal is to find the shortest path (or a closed loop) from which every point of the domain is seen. Apart from the above-mentioned theoretical considerations, visibility and motion planning are closely coupled in practice: computer vision and robot navigation go hand-in-hand in many courses and real-world applications.

Reporting optimal paths to *non-point* query objects has not received much attention; we

¹ We admit that the term “window” is overused, since it also denotes edges of the visibility polygon $V(q)$. Still, our two different usages of the term are well separated in the text, and are always apparent from the context.

are aware of work only for simple polygons. For efficient (logarithmic-time) queries between two convex polygons within a simple polygon, preprocessing can be done in linear time for Euclidean distances [10] and cubic time (and space) for link distance [1, 10].

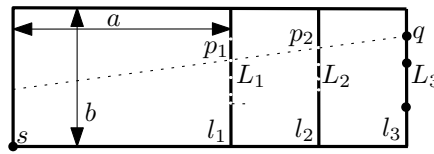
On the specific problem of quickest visibility queries addressed in this paper, Khosravi and Ghodsi [28] considered QVQs in *simple* polygons. They gave an algorithm for quickest visibility with logarithmic-time queries after quadratic-time preprocessing for building a quadratic-size structure: $P = O(n^2)$, $S = O(n^2)$, $Q = O(\log n)$. We improve the preprocessing and storage to linear, achieving $P = O(n)$, $S = O(n)$, $Q = O(\log n)$ for simple polygons (Section 5).

1.3 Overview of the results

- We start by giving a conditional lower bound connecting P and Q : Section 2 shows that 3SUM on n numbers can be solved in time $O(P + nQ)$. For instance subquadratic preprocessing time ($P = o(n^2)$) and sublinear query time ($Q = o(n)$) would lead to a subquadratic-time algorithm for 3SUM (see [18] for a recent major breakthrough on the 3SUM problem). The lower bound provides us with some justification for not obtaining sub-quadratic preprocessing time P for the QVQ. (Also more broadly, solutions to visibility and/or closely related link-distance query problems often use cubic-time preprocessing [42, 1, 10].)
- Section 3 employs the following natural approach to quickest visibility query.
 - (1) Build the visibility polygon $V(q)$ of the query point q ; $V(q)$ is a star-shaped polygon any side of which is either a piece of a boundary edge of D , or is a *window* – extension of the segment qv for some vertex v of D .
 - (2) For each window find the shortest path from s to the window, and choose the best window to go to.

The approach leads to an algorithm for QVQ with $P = P_v + P_s$, $S = S_v + S_s$, $Q = Q_v + KQ_s$ (refer to Section 1.1 for the notation). Problem (1) – building $V(q)$ – has been well studied (refer to Section 1.2 for the known bounds on P_v , S_v and Q_v). On the contrary, problem (2) – building a shortest path map for *segments* – has not been studied before. In Section 3.2 we give the first results for shortest path to a segment query (which we abbreviated SPSQ above) achieving $P_s = O(n^3 \log n)$, $S_s = O(n^3 \log n)$, $Q_s = O(\log n)$. Our solution is based on first designing a data structure for *horizontal* segments (Section 3.1) with $P_s = O(n \log n)$, $S_s = O(n \log n)$, $Q_s = O(\log n)$ – a result which may be interesting in its own right. The data structure for SPSQ for arbitrary segments is then built straightforwardly since there are $O(n^2)$ combinatorially different orientations: the data structure for arbitrarily oriented segments is thus just an $O(n^2)$ -fold replication of the structure for horizontal ones (we also give bounds in terms of sizes, E and Π , of visibility structures in D). Alternatively, in Section 3.3 we give an algorithm with $P_s = O(n^2 \log n)$, $S_s = O(n^2 \log n)$, $Q_s = O(\log^2 n)$ based on storing “snapshots” of the p-wave propagation in the continuous Dijkstra.

- In Section 4 we introduce the full *Quickest Visibility Map* (QVM) – the decomposition of D into cells such that within each cell the quickest visibility query has combinatorially the same answer: the shortest path to see any point within a cell goes through the same sequence of vertices of D . Our algorithm for building the map has $P = O(n^8 \log n)$, $S = O(n^7)$, $Q = O(\log n)$. We also observe that the QVM has $\Omega(n^4)$ complexity.
- In Section 5 we consider the case when D is a simple polygon. We give linear-size data structures that can be constructed in linear time, for answering QVQs and SPSQs



■ **Figure 1** D is long: $a \gg b$. The ray qp_2 (dotted) can reach all the way to the left, provided there exists a gap (p_1) on l_1 collinear with q and p_2 .

in logarithmic time: $P = O(n)$, $S = O(n)$, $Q = O(\log n)$, $P_s = O(n)$, $S_s = O(n)$, $Q_s = O(\log n)$.²

We invite the reader to play with our applet demonstrating QVM at <http://www.cs.helsinki.fi/group/compgeom/qvm/>.

2 A lower bound

In the *3SUM* problem the input is a set of numbers and the goal is to determine whether there are three numbers whose sum is 0. We connect P and Q (see Section 1.1 for the notation) with the *3SUM* problem.

► **Theorem 1.** *A 3SUM instance of size n can be solved in $O(P + nQ)$ time.*

Proof. We use a construction similar to the one in the proof of *3SUM*-hardness of finding minimum-link paths [36]. Start from an instance of the *GeomBase* problem: Given a set $S = L_1 \cup L_2 \cup L_3$ of n points lying on 3 vertical lines l_1, l_2, l_3 respectively, do there exist collinear points $p_1 \in L_1, p_2 \in L_2, p_3 \in L_3$? It was shown in [14] that solving *GeomBase* is as hard as solving *3SUM* with n numbers. Construct the domain D for quickest visibility queries as follows (Fig. 1): The lines l_1, l_2, l_3 are obstacles; turn each point from $L_1 \cup L_2$ into a gap punched in the obstacle. Squish vertically the whole construction, i.e., make the distances between the lines much larger than the vertical extent of S ; this way all the rays p_2p_1 with $p_2 \in L_2, p_1 \in L_1$ are confined to a narrow beam. Put the whole construction in a long box so that the beam shines onto its left side. Put s in the lower left corner of the box.

Now do quickest visibility queries to points in L_3 . If some point $q \in L_3$ is collinear with some points $p_1 \in L_1, p_2 \in L_2$, then q can be seen by traveling at most b from s ; otherwise, one needs to travel at least a to L_1 . Thus by making at most n queries we can solve the *GeomBase*. ◀

The above proof can be extended in several ways. E.g., since a can be arbitrarily large in comparison with b , even approximate answers to queries would solve the *3SUM* problem.

3 Querying shortest paths to windows

The quickest way to see the query point q from s is the quickest way to reach (the boundary of) $V(q)$, or equivalently, to reach a window of $V(q)$. Assuming the visibility polygon of q had been built by existing methods (see Section 1.2), answering QVQ boils down to determining the window closest to s . We do not have a better way of accomplishing this than to do shortest path queries to each window in succession, which leads to the problem of building

² Some results from this section were reported in EuroCG [29].

a data structure to answer efficiently shortest-path-to-a-segment query (abbreviated SPSQ above) – the subject of this section.³

3.1 Horizontal segments

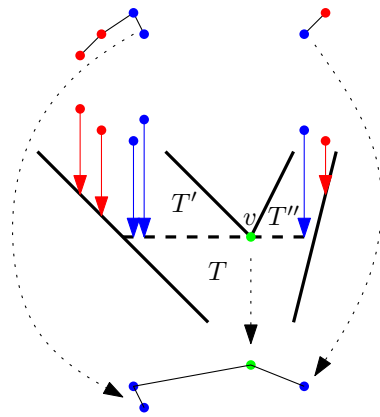
In this subsection we present a data structure for SPSQ for fixed-orientation (w.l.o.g. horizontal) segments; in the next subsection we extend the structure to handle arbitrary segments (and in Section 3.3 we present a structure for arbitrary segments, based on different techniques). The shortest path to a segment ab touches it at a , at b , or in the interior; we will focus on shortest paths to the interior, since shortest paths to a or b are answered with the SPM. Such a path follows the shortest path to some vertex v of D and then uses the perpendicular from v onto ab ; i.e., the last link of the path is vertical. We describe our data structure only for the case of paths arriving at ab from above, for which this last link is going *down*; an analogous structure is built for the paths arriving to the query from below.

The data structure is the horizontal trapezoidation of D augmented with some extra information for each trapezoid T ; specifically – the set of vertices that see the trapezoid from above (i.e., vertices from which downward rays intersect T). Of course, the information is not stored explicitly with each trapezoid (for this may require $\Omega(n)$ information in each of $\Omega(n)$ trapezoids); instead, the information is stored in persistent balanced binary trees. The vertices in the trees are sorted by x -coordinate. To enable $O(\log n)$ -time range minimum queries, each internal node stores the minimum of $d(v) + v_y$ values over all vertices v in the subtree of the node, where $d(v)$ is the geodesic distance from s to v (which can be read from the SPM) and v_y is the y -coordinate of v . Knowing the minimum of these values over the range of a segment is our ultimate goal, because the length of the shortest path that arrives to the segment at ordinate y with last link dropped from v is $d(v) + v_y - y$.

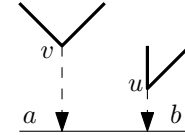
We build the trees as follows. Let \prec be the “aboveness” relation on the trapezoids (i.e., $T \prec T'$ iff T' is incident to T from above). We traverse the trapezoids using a topological order of the DAG for \prec (e.g., in the order of the y -coordinates of trapezoid top sides) and compute the trees for the trapezoids as follows (Fig. 2): If a trapezoid T does not have a successor in \prec , then T is a triangle (due to the non-degeneracy assumption on D), and the tree $\tau(T)$ for T simply stores the top vertex of T if the downward ray from the vertex goes inside T ; if the ray does not enter T (i.e., T has an obtuse angle at the base), then $\tau(T)$ is empty. If T has successors, then for each trapezoid T' that succeeds T in \prec , we take a persistent copy of the tree $\tau(T')$ and remove from it all vertices that do not see the boundary $T \cap T'$ between the trapezoids (the removal is a split operation on the copy). After the removal has been done for all successors of T , we merge the copies of the trees into the tree $\tau(T)$. Additionally, if T has a vertex of D on its top edge, then the vertex is added to $\tau(T)$.

To answer SPSQ, find the trapezoid T containing the query segment ab (recall our assumption that ab lies in the free space, and hence – in a single trapezoid) and choose the right history snapshot. Then perform the range minimum query $[a, b]$ to obtain the vertex $v \in \tau(T)$ of D with the smallest $d(v) + v_y$ (since $v \in \tau(T)$, the vertex sees ab when looking down and $a \leq v_x \leq b, v_y \geq y$); this will be the vertex from which the interior of the segment is reached in the quickest way. The shortest path via v is compared with the

³ We do not know how to take advantage of the fact that windows are quite special – maximal free-space segments anchored at vertices of D . On one hand this makes our solution more general, as it applies to arbitrary segments; on the other hand, it leaves open the possibility of designing a more efficient algorithm tailored to the special case of windows.



■ **Figure 2** Trees for the trapezoids. Red vertices are removed from persistent copies of $\tau(T')$ and $\tau(T'')$; the other vertices (blue) remain in the copies. Then the copies are merged to form $\tau(T)$. Finally, v is added to $\tau(T)$.



■ **Figure 3** u and v can be connected by a 3-link path making only right turns. ab is seen by u and v , and $d(u) + u_y = d(v) + v_y$.

shortest paths to a and b , altogether in $O(\log n)$ query time. Thus our data structure provides $P_s = O(n \log n)$, $S_s = O(n \log n)$, $Q_s = O(\log n)$ for horizontal segments.

3.2 Arbitrary segments

To support all directions of query segments, we build our structure from previous subsection for all rotations of D at which the data structure changes. The data structure changes at three types of events: (1) when two visible vertices get the same x -coordinates, (2) when two visible vertices get the same y -coordinates, and (3) when some query segment can be reached equally fast from two vertices, i.e., when the two vertices get the same $d(v) + v_y$ values (Fig. 3). The number of the first two events is bounded by the size E of the visibility graph of D , and the number of the third-type events is bounded by the number Π of pairs of vertices that can be connected by a right-turning 3-link path that turns by 90 degrees at its both bends. Thus we need to replicate our data structure only $O(E + \Pi)$ times (which may be much smaller than the naive upper bound of $O(n^2)$).

To find the rotation angles for the first two types of events, we precompute the visibility graph of D (takes $O(E + n \log n)$ time [16]). We can discover the third-type events “on-the-fly”, while actually rotating the domain. For that we make our trees “kinetic” by assigning to each internal node u of the trees the “expiration time” (rotation angle) when the vertex with lowest value of $d(v) + v_y$ in the subtree of u changes; the time for u can be computed when u is constructed, using the lowest $d(v) + v_y$ values in the subtrees of children of u . Computing the expiration time is done once per node instance of the trees.

Overall we obtain $P_s = O((E + \Pi)n \log n)$, $S_s = O((E + \Pi)n \log n)$, $Q_s = O(\log n)$.

► **Remark.** We could reuse the information between the rotations and get a persistent data structure with $P_s = O(n^2 \log^3 n)$, $S_s = O(n^2 \log^3 n)$, $Q_s = O(\log^2 n)$, but this is inferior to the performance of our data structure in the next section. Potentially one could also get a persistent data structure with $P_s = Q_s = O((E + \Pi)\text{polylog } n)$, $Q_s = O(\text{polylog } n)$; we, however, were not able to do this.

3.3 Continuous Dijkstra-based algorithm

We now give another data structure for SPSQ, based on storing “snapshots” of p-wave propagation (recall that p-wave is the wave propagated during the continuous Dijkstra algorithm for building the SPM). Recall (Section 1.2) that vertices of the SPM are vertices of D and triple points (at which three edges of the map meet). We say that time t_i is *critical* if the distance from s to a vertex of SPM is equal to t_i ; since SPM has linear complexity, there are $O(n)$ critical times. For each critical time t_i we store the *geodesic disk* D_i of radius t_i , i.e., the set of points in D whose geodesic distance to s is at most t_i ; the disk is an $O(n)$ -complexity region bounded by circular arcs (wavelets) and straight-line segments (obstacle edges). We construct data structures for two types of queries: “Given a segment ab , lying in the free space, does it intersect D_i ?” and “Given a segment ab lying outside D_i , where will the segment hit the disk if dragged perpendicularly to itself?”.

3.3.1 Determining i

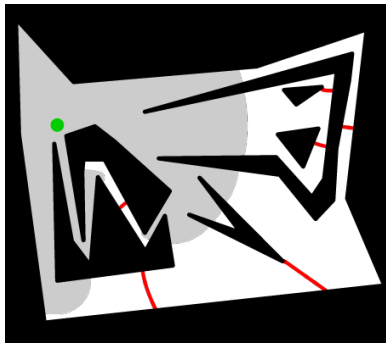
Assume that D_i has been preprocessed for point location, to test in $O(\log n)$ time whether a or b is inside D_i (in which case, obviously ab intersects D_i). To answer the intersection query when neither a nor b lies inside D_i , we look at the complement, C_i , of D_i in D ; obviously, a segment intersects the nonobstacle boundary of D_i iff it intersects the (nonobstacle) boundary of C_i . The set C_i may have several connected components (Fig. 4), at most one of which surrounds D_i . Each connected component C of C_i is preprocessed separately as follows: Let \mathcal{H} be the set of holes lying inside C . Let $\hat{C} = C \cup_{H \in \mathcal{H}} H$ be C together with the holes \mathcal{H} ; the set \hat{C} either has no holes (i.e., is simply connected) or has one hole (D_i , if C is the component that surrounds D_i). In any case \hat{C} can be preprocessed in $O(|C| \log n)$ time to answer ray shooting queries in $O(\log n)$ time [8], where $|C|$ is the complexity of C (the geodesic triangulations framework of [8] extends to regions with circular arcs on the boundary). To answer the intersection query we first determine the connected component C_a of C_i that contains a (assume that all connected components have been preprocessed for point location) and use the ray shooting data structure on \hat{C}_a to determine where the ray r from a through b exits \hat{C}_a ; ab intersects D_i iff r exits into D_i and does so before b . Note that here we crucially use the assumption that the query segment lies in the free space: we do not care if r intersects holes on the way to D_i (extending our algorithm to handle segments that may intersect holes is an open problem).

With the above data structures built for all disks D_i , we can do binary search on the critical times to determine the index i such that the query segment ab intersects D_{i+1} but does not intersect D_i , which means that ab is reached by the wavefront at some time between t_i and t_{i+1} . We spend $O(\log n)$ for ray shooting per choice of i , yielding $O(\log^2 n)$ time overall to determine i . Now the goal is to determine which wavelet of D_i hit ab first.

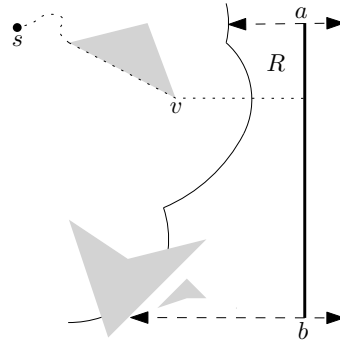
3.3.2 Determining the wavelet

Using the point location data structure on C_i we find the component C of C_i that contains ab (the segment must fully lie inside a single connected component, for otherwise it intersects D_i). Next, using the ray shooting data structure on C , we shoot rays within C , with sources at a and at b , firing orthogonal to ab , in both directions. This yields one region on each side of ab , and we consider the two regions separately; let R be the region on one of the sides (Fig. 5).

The boundary of R consists of ab , a ray shot from a to the boundary of C , a portion of the (outer) boundary of C (which may include circular-arc wavelets alternating with sequences



■ **Figure 4** s is green, D_i is gray, and C_i (the part of free space not reached by the wave) is white; it has four connected components, one of which has two holes inside it. Red curves are the walls (bisectors with more than one homotopy type of the shortest path) of the SPM.



■ **Figure 5** D_i is bounded by circular-arc wavelets (solid curves) and edges of obstacles (gray); the rays orthogonal to ab are dashed. The shortest path to ab ends with the perpendicular from v onto ab (dotted).

of straight-line segments on the boundary of obstacles), then a ray shot from b . Within R , we translate ab parallel to itself to discover the first wavelet on the boundary of R that is hit – the generator v of the wavelet is the last vertex on the shortest path to ab , with the last link of the path being the perpendicular dropped from v onto ab . This can be done by computing and storing convex hulls of pairs of consecutive wavelets on the boundary of C , pairs of pairs, pairs of pairs of pairs, etc., up to the convex hull of the whole component C . The next paragraph gives the details.⁴

Assume that the wavelets on the boundary of C are numbered in the order as they appear on the boundary. Compute convex hulls of wavelets 1 and 2, of wavelets 3 and 4, wavelets 5 and 6, etc.; then compute convex hulls of wavelets 1 through 4, wavelets 5 through 8, etc.; ...; finally, compute convex hull of all the wavelets. We thus obtain a hierarchy of convex hulls. Each convex hull of this hierarchy can be built by drawing bitangents to wavelets on the corresponding convex hulls of the preceding level, in $O(\log n)$ time per bitangent; since the complexity of each level is $O(|C|)$ and there are $O(\log n)$ levels, the whole hierarchy, for all connected components of C_i , can be stored in $O(n \log n)$ space and computed in $O(n \log^2 n)$ time. We preprocess each convex hull to answer extreme-wavelet queries – “Which wavelet is first hit by a query line moving in from infinity parallel to itself towards the convex hull?” – in $O(\log n)$ time (such preprocessing involves simply storing the bitangents to the consecutive wavelets along the convex hull in a search tree, sorted by the slope). Now, the rays shot from a and b (the ones that define the region R) hit the boundary of D_i at two wavelets, whose numbers are, say, w_1 and w_2 . The interval $[w_1, w_2]$ can be covered by $O(\log n)$ canonical intervals, for which we precomputed and stored the convex hulls; by doing the extreme-wavelet query in each of the intervals we determine the first wavelet between w_1 and w_2 hit by the sliding ab in overall $O(\log^2 n)$ time.

⁴ Note that while R may have some obstacles within it or on the boundary (e.g., in Fig. 5 the ray from b ends at an obstacle), if we sweep ab parallel to itself, it will first strike the boundary of R at a point on a circular-arc wavelet (for otherwise there would have been another critical time before the wavefront hit ab); thus, we may ignore obstacle edges on the boundary of R , and focus on storing the convex hulls only of the wavelets.

3.3.3 Putting everything together

Our data structure achieves $P_s = O(n^2 \log n)$, $S_s = O(n^2 \log n)$, $Q_s = O(\log^2 n)$: the ray shooting data structures and the convex hulls hierarchy require $O(n \log n)$ time preprocessing and storage per each of the $O(n)$ critical times, and a query involves finding the relevant D_i ($O(\log^2 n)$ time, Section 3.3.1) and then finding the first wavelet hit by the sliding ab (also $O(\log^2 n)$, Section 3.3.2).

3.4 Quickest visibility queries

Applying a data structure for SPSQ to QVQ, we obtain a solution for the latter with $P = P_v + P_s$, $S = S_v + S_s$, $Q = Q_v + KQ_s$. For instance, using [26] (which provides $P_v = O(n^2 \log n)$, $S_v = O(n^2)$, $Q_v = O(K \log^2 n)$) and the structure from Section 3.3, we obtain $P = O(n^2 \log n)$, $S = O(n^2 \log n)$, $Q = O(K \log^2 n)$. See Section 1.2 for other bounds on P_v , S_v , Q_v .

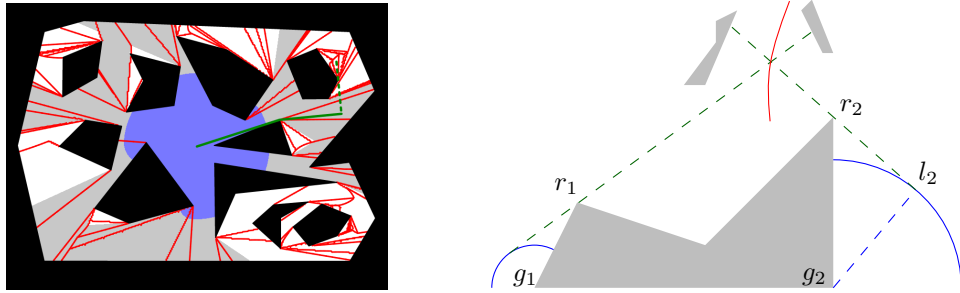
4 Quickest visibility map

Assuming the SPM has been built, the quickest way to see a query point q becomes evident as soon as the following information is specified: the window W of $V(q)$ through which to see q and the vertex g of D that is the last vertex on the shortest path to W . Let r be the vertex of D that defines W (i.e., W is part of the ray from q through r); we say that r is the *root* and g is the *generator* for q . We define the *quickest visibility map* (QVM) as the decomposition of D into cells such that all points within a cell have the same root and generator. That is, within a cell of QVM the answer to QVQ is combinatorially the same: draw the ray from q through the root r and drop the shortest segment from the generator g onto the window (this segment may be perpendicular to the window, or the segment to a window endpoint). In this section we describe an algorithm to build QVM. After the map is preprocessed for point location, QVQs can be answered in $O(\log n)$ time just by identifying the cell containing the query.

Reusing the idea of continuous Dijkstra algorithm for constructing the SPM we propagate “visibility wave” (v-wave) from s (Fig. 6, left). Similarly to the geodesic disk (the set of points that can be *reached* from s , by a certain time, starting from s and moving with unit speed), we define the *visibility disk of radius t* as the set of points that can be *seen* before time t by an observer starting from s and moving with unit speed. The ball is bounded by extensions of tangents from vertices of D to circles centered at vertices of the domain; intersections between tangents trace *bisectors* of QVM – a point q on a bisector can be seen equally fast by going to more than one side of $V(q)$ (Fig. 6, right).

To bound the complexity of QVM, we first introduce some notation. Let r, g be the root-generator pair for some cell of QVM. Let T be the line through r tangent to the wavelet centered at g at some time during the p- and v-waves propagation; let l be the point of contact of T with the wavelet. The part of the ray lr after r running through the free space (if such part exists) is called a *sweeper* – as the wavelet radius grows, T rotates around r and (parts of) the sweeper claim the cell(s) of QVM that have (r, g) as the root-generator pair. We call the segment rl the *leg* of the sweeper, and the segment gl (the radius of the wavelet) its *foot* (refer to Fig. 6, right).

Our argument below benefits from the assumption that all angles of the obstacles in D are larger than 90° ; to satisfy the assumption we can (symbolically) file the domain by replacing each acute vertex with a short edge (see the *corner arc algorithm* [21, Ch. 4] for



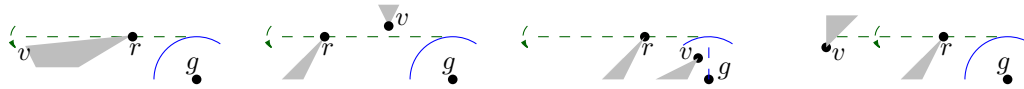
■ **Figure 6** Left: The v-wave is gray, the p-wave is blue (s is in the center of the rectangle). Red curves are bisectors in the QVM. Solid green shows the shortest path to see a query point; the path ends with a perpendicular dropped from D 's vertex (the generator) onto the ray (dashed green) from the query point through another vertex of D (the root). Right: Gray is an obstacle. As p-wave propagates, the geodesic disk grows by expanding the wavelets (blue arcs) at unit speed (wavelets are centered at generators g_1, g_2 and their radii grow at unit speed). Wavelets growth rotates tangents (dashed green) to the wavelets dropped from vertices r_1, r_2 – roots of the QVM cells. The tangents define “shadows” – the boundaries of the visibility disk; the tangents intersection traces the bisector (red) in the QVM. The QVM cell to the left of the bisector has (r_1, g_1) as the root-generator pair, while the cell on the right has (r_2, g_2) as the pair; points on the bisector have both (r_1, g_1) and (r_2, g_2) , and can be seen equally fast using paths via g_1 and via g_2 . $g_2 l_2$ is the foot of the sweeper hinged at r_2 ; $l_2 r_2$ is its leg.

similar ideas). The reason to make the assumption is that the speed of rotation of a sweeper depends on the (inverse of) the length of its leg; in particular, if the length is 0, the sweeper rotates at infinite speed, leading to a discontinuity in v-wave propagation⁵. The filing ensures that the v-wave propagation is continuous, which implies that QVM features (vertices and edges) are due only to intersections of sweepers, or (dis)appearance of sweepers, or possible sweeper extension/contraction as it passes over a vertex of D .

Consider now the subdivision S of D into maximal regions such that for any point inside a region, the set of sweepers that pass over the point is the same (i.e., if $\aleph(p)$ denotes the set of sweepers that ever pass over p , then S is the subdivision into the regions where \aleph stays the same). The vertices of QVM in the interiors of the regions are the *triple points* where three sweepers (and three bisectors) meet; since a sweeper is defined by two vertices of D (the root and the generator), there are $O(n^6)$ triple points.

What remains is to bound the number of vertices of QVM that lie on the edges of S ; to do that we define a superset \bar{S} of the edges. Specifically, disappearance of a sweeper may be due to one of the three events (Fig. 7): sweeper becoming aligned with an edge of D incident to the sweeper's root, the leg's rotation becoming blocked, or the foot's rotation becoming blocked; appearance of a sweeper is due to the reverse of the events. To account for the first-type events we add the supporting lines of edges of D to \bar{S} . The second-type events happen on supporting lines of edges of the visibility graph of D ; we add the lines to \bar{S} . Third-type events happen on lines through vertices of D perpendicular to supporting lines of the visibility graph edges; we add these perpendicular lines to \bar{S} . Finally, extension/contraction of a sweeper happens along the extension of the visibility graph edge. Overall \bar{S} consists of $O(nE)$ lines, and all $O(n^2 E^2)$ of their intersections could potentially be vertices of QVM. The only remaining vertices of QVM are intersections of bisectors with the lines in \bar{S} (all

⁵ See <http://www.cs.helsinki.fi/group/compgeom/qvm/infinitespeed.gif> for an animation



■ **Figure 7** From left to right: Sweeper aligns with rv ; leg gets blocked by v ; foot gets blocked by v ; sweeper extends at v .

the other vertices are in the interior of the cells of S); since any bisector is defined by 4 vertices of D (2 root-generator pairs for the sweepers defining the bisectors) there are $O(n^4)$ bisectors. Thus, the total number of vertices of QVM on edges of \bar{S} (and hence on the edges of S) is $O(n^2E^2 + n^4En)$.

The overall complexity of QVM (the number of vertices inside the regions of S plus on the edges of S) is thus $O(n^6 + n^2E^2 + n^5E) = O(n^7)$. The above description leads to an algorithm to compute the potential $O(n^7)$ QVM vertices by brute force; for each of them we can check in $O(n \log n)$ time whether it is indeed a vertex of QVM (see Section 1.2). We then sweep the plane to restore the QVM edges: from each vertex, extend the bisector until it hits another vertex. Putting point location data structure on top of QVM, we obtain $P = O(n^8 \log n)$, $S = O(n^7)$, $Q = O(\log n)$.

We note that any algorithm for QVM must have $P = \Omega(n^4)$, $S = \Omega(n^4)$ because it may need to store explicitly the region weakly visible from a segment, which may have $\Theta(n^4)$ complexity [40].

5 Simple polygons

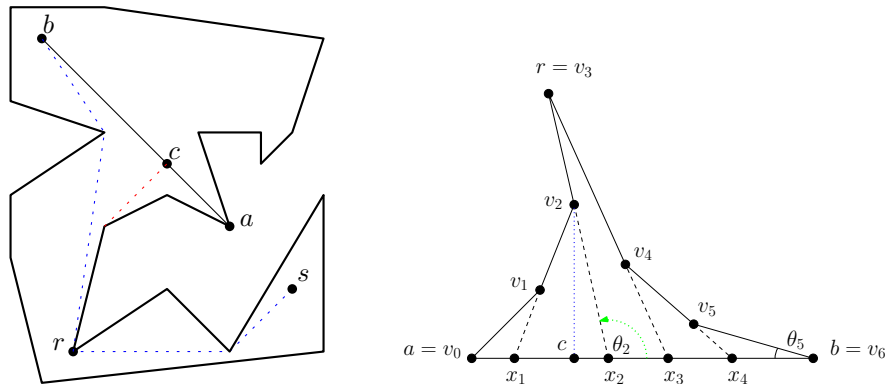
We now present an optimal ($P_s = O(n)$, $S_s = O(n)$, $Q_s = O(\log n)$) algorithm for SPSQs for the case when D is a simple polygon ($h = 0$); together with the shortest path map of D and a data structure for ray shooting queries (both can be built in $O(n)$ time to support $O(\log n)$ -time queries), it leads to an optimal algorithm ($P = O(n)$, $S = O(n)$, $Q = O(\log n)$) for QVQs as well. We start by introducing additional notation for this section.

Assume that the vertices of D are stored in an array \vec{D} sorted in clockwise order along the boundary of D . For points $x, y \in D$, let $\pi(x, y)$ denote the shortest path between x and y ; the shortest path from s to a point y is denoted simply by $\pi(y)$. Let the predecessor $\text{pred}(y)$ of y be the last vertex of D on $\pi(y)$ before y (or s if y sees s); the predecessor of any point can be read off the shortest path map (SPM) of D in $O(\log n)$ time. Let SPT be the shortest path tree from s in D ; the tree is the union of paths $\pi(v)$ for all vertices v of D . Assume that the SPT is preprocessed to support lowest common ancestor (LCA) queries in constant time [22].

Let ab be the query segment. Let r be the last common vertex of the shortest paths $\pi(a), \pi(b)$ from s to the endpoints of the segment; r can be determined from SPM and SPT in $O(\log n)$ time: either $\text{pred}(a) = \text{pred}(b) = r$, or $r = \text{LCA}(a, b)$ (Fig. 8, left). The paths $\pi(r, a)$ and $\pi(r, b)$ together with the segment ab form the *funnel* F of ab ; the vertex r is the *apex* of F .

Let $a = v_0, v_1, \dots, r = v_m, \dots, v_k, v_{k+1} = b$ be the vertices of the funnel from a to b . Note that the paths $\pi(r, a)$ and $\pi(r, b)$ are outward convex; in particular, F can be decomposed into triangles by extending the edges of F until they intersect ab (Fig. 8, right). Let x_i denote the intersection point of the extension of the edge $v_i v_{i+1}$ with ab (in particular, $x_0 = a$ and $x_k = b$). The shortest path from s to points on the segment $x_i x_{i+1}$ passes through v_{i+1} as the last vertex of D : $\forall p \in x_i x_{i+1}, \text{pred}(p) = v_{i+1}$.

Let $\theta_0, \theta_1, \dots, \theta_k$ denote the angles between the extension edges and ab : $\theta_i = \angle b x_i v_i$ for



■ **Figure 8** Left: $r = LCA(a, b)$; $\pi(c)$ is the answer to the query. Right: c is the foot of the perpendicular dropped from v_2 to ab .

$0 \leq i < k$ and $\theta_k = \pi - \angle abv_k$. The outward convexity of the paths $\pi(r, a), \pi(r, b)$ implies that the sequence $\theta_0, \theta_1, \dots, \theta_k$ is increasing. As a consequence the point $c \in ab$ closest to s can be characterized as follows [28]: c is the foot of the perpendicular from v_{i+1} to ab for i such that $\theta_i < \pi/2$ and $\theta_{i+1} \geq \pi/2$. Thus c can be found by a binary search on the angles θ_i : if $\theta_i > \pi/2$ then c lies left of x_i , whereas if $\theta_i < \pi/2$ then c lies right of x_i . We now describe how to implement the search in $O(\log n)$ time.

First, if $\theta_0 > \pi/2$ then $c = a$, and if $\theta_k < \pi/2$ then $c = b$; in both cases we are done. Next, look at the extensions of the edges emanating from the apex $r = v_m$ of the funnel. If $\theta_{m-1} \leq \pi/2 < \theta_m$, c is the foot of the perpendicular from v_m to ab and we are done.

It remains to show what to do if $\theta_{m-1} > \pi/2$ (the case $\theta_{m-1} < \pi/2$ is symmetric). In this case $\theta_i > \pi/2$ for $m \leq i \leq k$ since the angle sequence is increasing; in particular c is the foot of the perpendicular from some vertex v_i to ab , where v_i is on the left side $\pi(r, a)$ of the funnel F , i.e., $1 \leq i < m$. To determine v_i we would like to perform a binary search on the sequence v_0, \dots, v_k ; however this sequence is not directly accessible (we do not compute it during the query since it can have $\Omega(n)$ size). We therefore use the array \vec{D} , and perform a binary search on the interval $[r, a]$ in \vec{D} (if $r = s$ and s is not a vertex of D , we take the first vertex v after s on the path $\pi(a)$ and search in the interval $[v, a]$ instead).

For a vertex u in this interval we find the vertex $LCA(u, a)$, which is one of the vertices v_0, \dots, v_m on the left edge of the funnel, say v_j . By computing the angle θ_j we can decide if the binary search has to continue to the left or to the right of u . After $O(\log n)$ iterations the binary search is narrowed down to an interval between two successive vertices in \vec{D} . This implies that the point v_i from which the perpendicular to c has to be dropped is also determined. (Note that for several successive vertices u_l in $[r, a]$ we can get the same vertex v_j as a result of computing $LCA(u_l, a)$; still, since the total number of vertices in $[r, a]$ is $O(n)$, after $O(\log n)$ iterations the binary search is narrowed down to an interval between two successive vertices in \vec{D} .)

Quickest visibility queries

In a simple polygon, s is separated from q by a unique window of $V(q)$ (unless s and q see each other, which can be tested in $O(\log n)$ time by ray shooting). Since the last edge of the shortest path $\pi(q)$ is a straight-line segment, one of the window endpoints is $a = \text{pred}(q)$; the endpoint can be read off the SPM of D in $O(\log n)$ time. To find the other endpoint b of the window, shoot the ray qa until it intersects the boundary of D ; this also takes

$O(\log n)$ time using the data structure for ray shooting [24]. Once we have the window ab , our data structure described above finds the (unique) shortest path to the window in additional $O(\log n)$ time.

Acknowledgments. We thank the anonymous reviewers for their helpful comments. VP is supported by grant 2014-03476 from the Sweden's innovation agency VINNOVA. TT was supported by the University of Helsinki Research Funds.

References

- 1 Esther M. Arkin, Joseph S.B. Mitchell, and Subhash Suri. Optimal link path queries in a simple polygon. In *Proc. 3rd Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'92)*, pages 269–279, 1992.
- 2 Boris Aronov, Leonidas J. Guibas, Marek Teichmann, and Li Zhang. Visibility queries and maintenance in simple polygons. *Discrete & Computational Geometry*, 27(4):461–483, 2002.
- 3 Prosenjit Bose, Anna Lubiw, and J. Ian Munro. Efficient visibility queries in simple polygons. *Comput. Geom. Theory Appl.*, 23(3):313–335, November 2002.
- 4 Francisc Bungiu, Michael Hemmer, John Hershberger, Kan Huang, and Alexander Kröller. Efficient computation of visibility polygons. In *30th Europ. Workshop on Comput. Geom. (EuroCG'14)*, 2014.
- 5 Svante Carlsson, Håkan Jonsson, and Bengt J. Nilsson. Finding the shortest watchman route in a simple polygon. *Discrete & Computational Geometry*, 22(3):377–402, 1999.
- 6 CGAL. Computational Geometry Algorithms Library. <http://www.cgal.org>.
- 7 Bernard Chazelle. A theorem on polygon cutting with applications. In *Proc. 23rd Annu. Sympos. Found. Comput. Sci. (FOCS'82)*, pages 339–349. IEEE, 1982.
- 8 Bernard Chazelle, Herbert Edelsbrunner, Michelangelo Grigni, Leonidas Guibas, John Hershberger, Micha Sharir, and Jack Snoeyink. Ray shooting in polygons using geodesic triangulations. In Javier Leach Albert, Burkhard Monien, and Mario Rodríguez Artalejo, editors, *Automata, Languages and Programming (ICALP)*, volume 510 of *Lecture Notes in Computer Science*, pages 661–673. Springer, 1994.
- 9 Danny Z. Chen and Haitao Wang. Visibility and ray shooting queries in polygonal domains. In *Proc. 13th Int. Conf. Algorithms Data Struct. (WADS'13)*, LNCS, pages 244–255, 2013.
- 10 Yi-Jen Chiang and Roberto Tamassia. Optimal shortest path and minimum-link path queries between two convex polygons inside a simple polygonal obstacle. *Int. J. Comput. Geometry Appl.*, 7(1/2):85–121, 1997.
- 11 Moshe Dror, Alon Efrat, Anna Lubiw, and Joseph S.B. Mitchell. Touring a sequence of polygons. In *Proc. 35th Symposium on Theory of Computing (STOC'03)*, pages 473–482, 2003.
- 12 Adrian Dumitrescu and Csaba D. Tóth. Watchman tours for polygons with holes. *Comput. Geom. Theory Appl.*, 45(7):326–333, 2012.
- 13 S. Eriksson-Bique, J. Hershberger, V. Polishchuk, B. Speckmann, S. Suri, T. Talvitie, K. Verbeek, and H. Yıldız. Geometric k shortest paths. In Sanjeev Khanna, editor, *Proc. 26th Ann. ACM-SIAM Symp. Discrete Algorithms, (SODA'15)*, pages 1616–1625. SIAM, 2015.
- 14 Anka Gajentaan and Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Computational Geometry: Theory and Applications*, 5:165–185, 1995.
- 15 Subir Ghosh. *Visibility Algorithms in the Plane*. Cambridge University Press, 2007.
- 16 Subir Kumar Ghosh and David M. Mount. An output-sensitive algorithm for computing visibility graphs. *SIAM J. Comput.*, 20(5):888–910, 1991.

- 17 J.E. Goodman and J. O'Rourke, editors. *Handbook of Discrete and Computational Geometry*. Taylor & Francis, 2nd edition, 2010.
- 18 Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles. In *Proc. 55th Ann. Sympos. Found. Comput. Sci. (FOCS'14)*, pages 621–630. IEEE, 2014.
- 19 Leonidas J. Guibas, J. Hershberger, D. Leven, Micha Sharir, and R. E. Tarjan. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2:209–233, 1987.
- 20 Leonidas J. Guibas, Rajeev Motwani, and Prabhakar Raghavan. The robot localization problem. *SIAM J. Comput.*, 26(4):1120–1138, August 1997.
- 21 Olaf Andrew Hall-Holt. *Kinetic Visibility*. PhD thesis, Stanford University, 2002.
- 22 D. Harel and R. E. Tarjan. Fast algorithms for finding nearest common ancestors. *SIAM J. Comput.*, 13(2):338–355, 1984.
- 23 P. J. Heffernan and Joseph S. B. Mitchell. An optimal algorithm for computing visibility in the plane. *SIAM J. Comput.*, 24(1):184–201, 1995.
- 24 John Hershberger and Subhash Suri. A pedestrian approach to ray shooting: Shoot a ray, take a walk. *Journal of Algorithms*, 18(3):403–431, 1995.
- 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 Rajasekhar Inkulu and Sanjiv Kapoor. Visibility queries in a polygonal region. *Comput. Geom. Theory Appl.*, 42(9):852–864, 2009.
- 27 B. Joe and R. B. Simpson. Correction to Lee's visibility polygon algorithm. *BIT*, 27:458–473, 1987.
- 28 Ramtin Khosravi and Mohammad Ghodsi. The fastest way to view a query point in simple polygons. In *21st European Workshop on Computational Geometry (EuroCG'05)*, pages 187–190. Eindhoven, 2005.
- 29 Christian Knauer, Günter Rote, and Lena Schlipf. Shortest inspection-path queries in simple polygons. In *24th European Workshop on Computational Geometry (EuroCG'08)*, pages 153–156, 2008.
- 30 D. T. Lee and F. P. Preparata. Euclidean shortest paths in the presence of rectilinear barriers. *Networks*, 14:393–410, 1984.
- 31 Lin Lu, Chenglei Yang, and Jiaye Wang. Point visibility computing in polygons with holes. *Journal of Information & Computational Science*, 8(16):4165–4173, 2011.
- 32 Joseph S. B. Mitchell. Shortest paths among obstacles in the plane. *Internat. J. Comput. Geom. Appl.*, 6:309–332, 1996.
- 33 Joseph S. B. Mitchell. Geometric shortest paths and network optimization. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 633–701. Elsevier, 2000.
- 34 Joseph S. B. Mitchell. Shortest paths and networks. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 445–466. Elsevier, 2004.
- 35 Joseph S. B. Mitchell. Approximating watchman routes. In Sanjeev Khanna, editor, *Proc. 24th Annual ACM-SIAM Symp. on Discrete Algorithms, SODA'13*, pages 844–855. SIAM, 2013.
- 36 Joseph S. B. Mitchell, Valentin Polishchuk, and Mikko Sysikaski. Minimum-link paths revisited. *Comput. Geom. Theory Appl.*, 47(6):651–667, 2014.
- 37 Joseph S. B. Mitchell, Günter Rote, and Gerhard J. Woeginger. Minimum-link paths among obstacles in the plane. *Algorithmica*, 8(1):431–459, 1992.
- 38 Joseph O'Rourke. *Art Gallery Theorems and Algorithms*. Oxford University Press, 1987.

- 39 Eli Packer. Computing multiple watchman routes. In Catherine C. McGeoch, editor, *Experimental Algorithms, 7th International Workshop, WEA, Provincetown, MA, USA*, volume 5038 of *Lecture Notes in Computer Science*, pages 114–128. Springer, 2008.
- 40 S Suri and J O'Rourke. Worst-case optimal algorithms for constructing visibility polygons with holes. In *Proc. 2nd Ann. Symp. Computational Geometry*, pages 14–23. ACM, 1986.
- 41 Subhash Suri. A linear time algorithm with minimum link paths inside a simple polygon. *Computer Vision, Graphics and Image Processing*, 35(1):99–110, 1986.
- 42 Alireza Zarei and Mohammad Ghodsi. Query point visibility computation in polygons with holes. *Comput. Geom. Theory Appl.*, 39(2):78–90, 2008.

Trajectory Grouping Structure under Geodesic Distance

Irina Kostitsyna¹, Marc van Kreveld², Maarten Löffler²,
Bettina Speckmann¹, and Frank Staals²

- 1 Department of Mathematics and Computer Science, TU Eindhoven,
The Netherlands
{i.kostitsyna|b.speckmann}@tue.nl
- 2 Department of Information and Computing Sciences, Utrecht University,
The Netherlands
{m.j.vankreveld|m.loffler|f.staals}@uu.nl

Abstract

In recent years trajectory data has become one of the main types of geographic data, and hence algorithmic tools to handle large quantities of trajectories are essential. A single trajectory is typically represented as a sequence of time-stamped points in the plane. In a collection of trajectories one wants to detect maximal groups of moving entities and their behaviour (merges and splits) over time. This information can be summarized in the *trajectory grouping structure*.

Significantly extending the work of Buchin et al. [WADS 2013] into a realistic setting, we show that the trajectory grouping structure can be computed efficiently also if obstacles are present and the distance between the entities is measured by geodesic distance. We bound the number of *critical events*: times at which the distance between two subsets of moving entities is exactly ε , where ε is the threshold distance that determines whether two entities are close enough to be in one group. In case the n entities move in a simple polygon along trajectories with τ vertices each we give an $O(\tau n^2)$ upper bound, which is tight in the worst case. In case of *well-spaced* obstacles we give an $O(\tau(n^2 + m\lambda_4(n)))$ upper bound, where m is the total complexity of the obstacles, and $\lambda_s(n)$ denotes the maximum length of a Davenport-Schinzel sequence of n symbols of order s . In case of general obstacles we give an $O(\tau \min\{n^2 + m^3\lambda_4(n), n^2m^2\})$ upper bound. Furthermore, for all cases we provide efficient algorithms to compute the critical events, which in turn leads to efficient algorithms to compute the trajectory grouping structure.

1998 ACM Subject Classification F.2.2 Analysis of Algorithms and Problem Complexity

Keywords and phrases moving entities, trajectories, grouping, computational geometry

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.674

1 Introduction

Tracking moving entities like humans, vehicles and animals is becoming more and more commonplace, with applications in security (what human movement is suspicious behavior?), the social sciences (which people move together? what regions do they avoid?), biology (what are migration routes and what are the stopping places?), and traffic analysis. Technology like GPS, RFID, and video has led to large data sets with trajectories, representing the movement of entities. At a similar pace, more and more algorithmic tools to analyze such data are being developed within the areas of Geographic Information Science, data mining, and computational geometry.

In most cases, each trajectory is represented by a sequence of time-stamped points in the plane or in space. As such, trajectories can be seen as a form of time-series data with a



© Irina Kostitsyna, Marc van Kreveld, Maarten Löffler, Bettina Speckmann, and Frank Staals;
licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 674–688



Leibniz International Proceedings in Informatics

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

geometric component. Collections of trajectories can be processed for retrieving patterns like clusters, flocks, leadership, encounter, and many more [1, 7, 10, 11, 12]. Trajectory data can also be linked to the environment, available in other spatial data sets, to determine more types of patterns [3, 4].

Recent research has gone beyond identifying flocks or moving clusters separately by modelling all joint movements into the trajectory grouping structure [2]. This structure captures the joining and splitting of groups of entities by employing methods from computational topology, in particular, the Reeb graph [6]. Distances between moving entities are among the main criteria to decide if entities belong to the same group (see below for a precise definition). In this paper we significantly extend the trajectory grouping structure by incorporating obstacles and measuring distances as geodesic distances. The geodesic distance between two entities is the distance that needs to be traversed for one entity to reach the other entity. This approach gives a more natural notion of groups because it separates entities moving on opposite sides of obstacles like fences or water bodies. A threshold distance denoted by ε determines whether two entities are close enough to be in the same group. Hence we examine the number of times that a threshold distance occurs among n moving entities. Only threshold distances between the closest two entities of different groups matters, so we analyze the number of events of this type for various obstacle settings.

The combination of moving points and specific structures defined by these points has been a topic of major interest in computational geometry; for example, one of the main open problems in the area is the question “How many times can the Delaunay triangulation change its combinatorial structure in the worst case, when n points move along straight lines in the plane?” Other related research on movement in geometric algorithms concerns kinetic data structures. To our knowledge, our paper is the first to combine continuously moving points with geodesic distances in the plane. We expect that our analysis will be of interest to other distance problems on moving points than the trajectory grouping structure. For example, in a similarity measure for two trajectories that incorporates obstacles.

Terminology and notation. We are given a set \mathcal{X} of n entities, each moving along a piecewise linear trajectory with τ vertices, and a set of pairwise disjoint polygonal obstacles $\mathcal{O} = \{\mathcal{O}_1, \dots, \mathcal{O}_h\}$. Let m denote the total complexity of \mathcal{O} .

We denote the position of entity a at time t by $a(t)$. Let $\|pq\|$ denote the Euclidean distance between points p and q , and let $\xi_{ab}(t) = \|a(t)b(t)\|$ denote the (Euclidean) distance between entities a and b at time t . A path $P = p_1, \dots, p_k$ from p_1 to p_k is a polygonal line with vertices p_1, \dots, p_k , and has length $\zeta(P) = \sum_{i=1}^{k-1} \|p_i p_{i+1}\|$. A path is *obstacle-avoiding* if it is disjoint from the interior of all obstacles in \mathcal{O} . A path between p and q is a *geodesic*, denoted $g(p, q)$, if it has minimum length among all obstacle-avoiding paths. We refer to the length of $g(p, q)$ as the *geodesic distance* between p and q . We denote the geodesic distance between a and b at time t by $\varsigma_{ab}(t) = \zeta(g(a(t), b(t)))$.

To determine if a set of entities may form a group, we have to decide if they are close together. Analogous to Buchin et al. [2] we model this by a spatial parameter ε . More specifically, two entities a and b are *directly connected* at time t if they are within (geodesic) distance ε from each other, that is, $\varsigma_{ab}(t) \leq \varepsilon$. A set of entities \mathcal{X}' is ε -*connected* at time t if for any pair $a, b \in \mathcal{X}'$ there is a sequence $a = a_0, a_1, \dots, a_k = b$ such that a_i and a_{i+1} are directly connected. We refer to a time at which a and b become directly connected or disconnected as an ε -*event*. At such a time the distance between a and b is exactly ε . If an ε -event also connects or disconnects the maximal ε -connected set(s) containing a and b , it is a *critical event*. A (maximal) ε -connected set of entities \mathcal{X}' is a *group* if it is ε -connected at any time t in a time interval of length at least δ , and it has at least a certain size.

■ **Table 1** The number of critical events (i.e. the size of \mathcal{R}), and the time required to construct \mathcal{R} . Note that the input size is $\Theta(\tau n + m)$.

	Lower bound	Upper bound	Algorithm
Simple polygon	$\Omega(\tau n^2)$	$O(\tau n^2)$	$O(\tau n^2(\log^2 m + \log n) + m)$
Well-spaced obstacles	$\Omega(\tau(n^2 + nm))$	$O(\tau(n^2 + m\lambda_4(n)))$	$O(\tau n^2 m \log n)$
General obstacles	$\Omega(\tau(n^2 + nm \min\{n, m\}))$	$O(\tau \min\{n^2 + m^3\lambda_4(n), n^2 m^2\})$	$O(\tau n^2 m^2 \log n + m^2 \log m)$

Trajectory grouping structure. Since the objective of our methods is to compute the trajectory grouping structure as defined by Buchin et al. [2], we review their structure here. It captures not just the groups, but also how and when they arise, merge, split, or stop to be a group. Only maximal groups are considered, where groups can be maximal in size and in duration.

The evolution of the maximal ε -connected sets as the entities move is directly represented by a directed acyclic graph (DAG) \mathcal{R} . Edges of the graph correspond to the maximal ε -connected sets and the nodes correspond to structural changes, that is, critical events. For example, a node may represent a critical event where two maximal ε -connected sets get close enough to become one ε -connected set: the node will have in-degree 2 and out-degree 1 and represents a join. This DAG \mathcal{R} is a Reeb graph [6]. Each entity is associated with a directed path in \mathcal{R} in the natural way.

Groups are defined as above. We are interested only in maximal groups: a subset S for a time interval I is a maximal group if (i) S is in an ε -connected subset during I , (ii) I has length at least δ , (iii) S has at least the required size, (iv) no proper superset of S or proper superinterval of I exists with the same properties. Maximal groups are associated with a directed (sub)path in \mathcal{R} in a natural way.

Buchin et al. [2] show that when there are no obstacles the maximum complexity of \mathcal{R} is $\Theta(\tau n^2)$ in the worst case, and it can be computed in $O(\tau n^2 \log n)$ time. Furthermore, there are $\Theta(\tau n^3)$ maximal groups in the worst case, and they can be reported in $O(\tau n^3 \log n + N)$ time, where N is the output size (which is $O(\tau n^4)$).

Results and organization. We extend the results of Buchin et al. [2] to the case where the entities move amidst obstacles, and we thus measure the distance between two entities a and b by their geodesic distance ζ_{ab} . Instead of having $O(\tau n^2)$ events that correspond to the nodes in \mathcal{R} , we can have more events, depending on the obstacles and their complexity.

We study three settings for the obstacles. In the simplest case, all entities move inside a simple polygon with m vertices. In the most general case, obstacles can have any shape, location, and complexity, but they are disjoint and have total complexity m . As an intermediate case we assume that the distance between any two non-adjacent obstacle edges is at least ε . We say that the obstacles are *well-spaced*.

Our results on the number of critical events, and thus the size of the Reeb-graph, for the three cases, are listed in Table 1. For the simple polygon case, which we treat in Section 3, our bounds are tight. The upper bounds for the well-spaced obstacles case, and the general obstacles case include a $\lambda_4(n)$ term, where $\lambda_s(n)$ denotes the maximum length of a Davenport-Schinzel sequence of order s with n symbols. Since $\lambda_4(n)$ is only slightly superlinear, our bound for the well-spaced obstacles case is almost tight. We present these

results in Sections 4 and 5, respectively. For all cases we also bound the total number of ε -events, and we show how to compute \mathcal{R} efficiently. Omitted proofs can be found in the full version.

Once we have the Reeb graph \mathcal{R} describing connectivity events of the entities in \mathcal{X} , we can use the existing analysis by Buchin et al. [2] to bound the number of maximal groups as well as their algorithm(s) to compute these groups. So the interesting part is in analyzing the complexity of \mathcal{R} and determining how to compute it.

2 Distance Functions

Let a and b be two entities, each moving along a straight line during interval I , and let p be a fixed point in \mathbb{R}^2 . During I the Euclidean distance $\xi_{ap}(t)$ between a and p is a convex hyperbolic function in t that has the form $\sqrt{Q(t)}$, for some quadratic function Q . The Euclidean distance between a and b during I is a convex hyperbolic function of the same form. Since ξ_{ap} is convex, there are at most two times in I such that $\xi_{ap}(t) = \varepsilon$. The same applies for ξ_{ab} .

The geodesic distance $\varsigma_{ap}(t)$ between a and p is a piecewise function. At times where the geodesic $g(a(t), p)$ consists of a single line segment, the geodesic distance is simply the Euclidean distance. When the geodesic consists of more than one line segment we can decompose it into two parts: a line segment $g(a(t), u) = \overline{a(t)u}$, and a path $g(u, p)$, where u is the first obstacle vertex on $g(a(t), p)$. Similarly, if the geodesic $g(a(t), b(t))$ between a and b consists of more than one segment we can decompose it into three parts $\overline{a(t)u}$, $g(u, v)$, and $\overline{vb(t)}$ (we may have $u = v$). It follows that each piece of ς_{ap} is convex and hyperbolic. The pieces of ς_{ab} are convex as well, since they are of the form $\xi_{au}(t) + C + \xi_{vb}(t) = \sqrt{Q_1(t)} + C + \sqrt{Q_2(t)}$, for some quadratic functions Q_1 and Q_2 and a constant C . Therefore, we again have that on each piece there are at most two times where $\varsigma_{ap}(t)$ is exactly ε . The same applies for $\varsigma_{ab}(t)$.

We obtain the same results when a and b move on piecewise linear trajectories, rather than lines. The functions then simply consist of more pieces.

► **Lemma 1.** *Let $\mathcal{F} = f_1, \dots, f_n$ be a set of n piecewise (partial) functions, each function f_i consisting of τ pieces f_i^1, \dots, f_i^τ , such that any two pieces f_i^k and f_j^ℓ intersect each other at most s times. The lower envelope \mathcal{L} of \mathcal{F} has complexity $O(\tau\lambda_{s+2}(n))$.*

Analogous to Lemma 1 we can show that the upper envelope of \mathcal{F} has complexity $O(\tau\lambda_{s+2}(n))$.

3 Simple Polygon

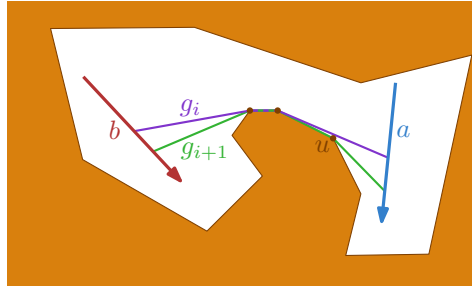
We first focus our attention on entities moving in a simply-connected polygonal domain.

3.1 Lower Bound

Buchin et al. [2] show that the number of critical events for n entities moving in \mathbb{R}^2 without obstacles can be $\Omega(\tau n^2)$. Clearly, this lower bound also holds for entities moving inside a simple polygon.

3.2 Upper Bound

Let a and b be two entities, each moving along a line during interval I , and let $\varsigma(t) = \varsigma_{ab}(t)$ be the function describing the geodesic distance between a and b during interval I .



■ **Figure 1** Geodesics g_i (purple) and g_{i+1} differ by at most one vertex; the first vertex u on g_{i+1} .

► **Lemma 2.** *The function ζ is convex.*

Proof Sketch. Let $[t_{i-1}, t_i]$ and $[t_i, t_{i+1}]$ be two consecutive time intervals, corresponding to pieces ζ_i and ζ_{i+1} of ζ . We now show that ζ is convex on $[t_{i-1}, t_{i+1}]$.

Let g_i and g_{i+1} denote the geodesic shortest paths corresponding to ζ_i and ζ_{i+1} , respectively. Geodesics g_i and g_{i+1} differ by at most one vertex u (assuming general position of the obstacle vertices), and this vertex occurs either at the beginning or the end of the geodesic. Consider the case that u is the first vertex of g_{i+1} , and u does not occur on g_i . See Figure 1. All other cases are symmetric. Let v be the second vertex of g_{i+1} (and thus the first vertex of g_i). We have $\zeta_i(t) = \|a(t)v\| + \zeta(v, b(t))$ and $\zeta_{i+1}(t) = \|a(t)u\| + \|uv\| + \zeta(v, b(t))$. It follows that the individual pieces ζ_i and ζ_{i+1} are (convex) hyperbolic functions, that $\zeta_i(t_i) = \zeta_{i+1}(t_i)$, and that for any time $t \in [t_{i-1}, t_{i+1}]$, $\zeta_{i+1}(t) \geq \zeta_i(t)$. We use these properties to show that for any three times $s, m, t \in [t_{i-1}, t_{i+1}]$, with $s \leq m \leq t$, the point $\zeta(m)$ lies below the line segment (function) $\overline{\zeta(s)\zeta(t)}$, that is $\zeta(m) \leq \overline{\zeta(s)\zeta(t)}(m)$. Since ζ_i and ζ_{i+1} are convex, the only interesting case is when s lies on ζ_i and t lies on ζ_{i+1} . We prove this by case distinction on m . It follows that ζ is convex on $[t_{i-1}, t_{i+1}]$. ◀

► **Theorem 3.** *Let \mathcal{X} be a set of n entities, each moving in a simple polygon along a piecewise linear trajectory with τ vertices. The number of ε -events is at most $O(\tau n^2)$.*

Proof. Fix a pair of entities a and b . Both a and b move along trajectories with τ vertices. So there are $2\tau - 1$ intervals during which both a and b move along a line. During each such interval ζ_{ab} is convex (Lemma 2). So there are at most two times in each interval at which $\zeta_{ab}(t) = \varepsilon$. The lemma follows. ◀

3.3 Algorithm

Next, we describe how to compute all ε -events. The high level overview of our algorithm is as follows. For each pair of entities a and b , we first find a time t_{\min} such that the geodesic distance $\zeta(t) = \zeta_{ab}(t)$ between a and b is minimal. Clearly, if $\zeta(t_{\min}) > \varepsilon$ there is no time at which a and b are at distance ε . Otherwise, we use the fact that ζ is convex (Lemma 2). This means that on $I^- = (-\infty, t_{\min}]$ it is monotonically decreasing, and on $I^+ = [t_{\min}, \infty)$ it is monotonically increasing. Hence, there are at most two times t^- and t^+ such that $\zeta(t) = \varepsilon$, and we have that $t^- \in I^-$ and $t^+ \in I^+$. We now find t^- and t^+ using parametric search [13]: t^- (t^+) is the smallest (largest) time in I^- (I^+) such that $\zeta(t) \leq \varepsilon$. To actually find t_{\min} , we basically use the same approach. At t_{\min} the derivative ζ' of ζ is zero. Since ζ is convex, its derivative is monotonically increasing. Therefore, we can find t_{\min} using a parametric search: t_{\min} is the smallest time such that $\zeta'(t) \geq 0$.

Finding the times t_{\min} , t^- , and t^+ . We use parametric search [13] to find t_{\min} , t^- , and t^+ . The global idea is as follows. For a more detailed description of parametric search and its application to our problem we refer to the full version of this paper.

To find t_{\min} we use $\mathcal{P}(t) = \zeta'(t) \geq 0$ as predicate. To find t^- and t^+ we use $\mathcal{P}(t) \leq \varepsilon$, and $\mathcal{P}(t) \geq \varepsilon$, respectively. In all these cases we need an algorithm \mathcal{A} that can test $\mathcal{P}(t)$ for a given time t . This means that we need an efficient algorithm to compute $\zeta(t)$ and a functional description of ζ . To this end, we preprocess the input polygon for shortest path queries. We triangulate the polygon in $O(m)$ time [5], and build the data structure \mathcal{D} of Guibas and Hershberger [8]. This also takes $O(m)$ time, and allows us to find the length of the shortest path between two fixed points p and q in $O(\log m)$ time. In particular, this means that for a given time t , we can compute $\zeta(t)$ and $\zeta'(t)$ in $O(\log m)$ time.

A query, and thus our algorithm \mathcal{A} , takes $O(\log m)$ time. It now immediately follows that we can compute t_{\min} , t^- , and t^+ in $O(\log^2 m)$ time each. We obtain the following result.

► **Lemma 4.** *Let \mathcal{X} be a set of n entities, each moving in a simple polygon along a piecewise linear trajectory with τ vertices. We can compute all ε -events in $O(\tau n^2 \log^2 m + m)$ time, where m is the number of vertices in the polygon.*

To compute \mathcal{R} we can now use the algorithm as described by Buchin et al. [2]. This algorithm maintains the connected components in a dynamic graph G ; at each ε -event we insert or delete an edge in G . This takes $O(\log n)$ time per ε -event, and thus $O(\tau n^2 \log n)$ time in total [2, 14]. We conclude:

► **Theorem 5.** *Let \mathcal{X} be a set of n entities, each moving in a simple polygon along a piecewise linear trajectory with τ vertices. The Reeb graph \mathcal{R} representing the movement of the entities in \mathcal{X} has size $O(\tau n^2)$ and can be computed in $O(\tau n^2(\log^2 m + \log n) + m)$ time, where m is the number of vertices in the polygon.*

4 Well-spaced Obstacles

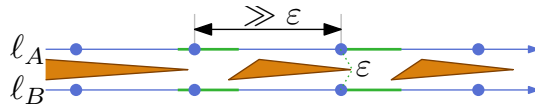
Next, we consider the situation where the entities move in a domain with multiple polygonal obstacles. We first assume that the obstacles are well-spaced, that is, the distance between any pair of non-adjacent obstacle edges is at least ε .

4.1 Lower Bound

► **Lemma 6.** *The total number of critical events for a set of n entities, each moving amidst a set of well-spaced obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices, is $\Omega(\tau(n^2 + nm))$, where m is the total complexity of \mathcal{O} .*

Proof. We describe a construction in which the entities move along lines that yields $\Omega(nm)$ critical events. We repeat this construction in $\Omega(\tau)$ steps. Since we already have a $\Omega(\tau n^2)$ lower bound for entities moving in \mathbb{R}^2 without obstacles, the lemma then follows.

The construction that we use is sketched in Fig. 2. We have two horizontal lines ℓ_A and ℓ_B that are within vertical distance ε of each other. Our obstacles essentially form a wall separating the two lines that has $\Theta(m)$ openings. Each obstacle is triangular, and thus well-spaced. Furthermore, the obstacles are at distance at least ε from each other, so \mathcal{O} is well-spaced. Our set of entities consists of two equal-sized subsets A and B . The entities move in pairs; one entity a from A and one entity b from B . Throughout the movement they maintain $a_x = b_x$, and stay far away from any other entities. It is easy to see that this yields $\Omega(nm)$ critical events as desired. ◀



■ **Figure 2** The lower bound construction for well spaced obstacles. The entities of a pair a, b are within distance ϵ from each other when both move in a green interval.

4.2 Upper Bound

In this case our obstacles are well spaced, so if two entities are at geodesic distance ϵ the geodesic consists of at most two line segments. We now start with some bounds on the total number of ϵ -events.

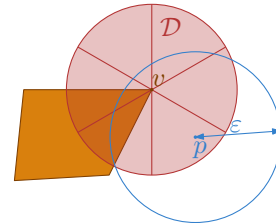
► **Observation 7.** *There are at most $O(\tau n^2)$ ϵ -events where the geodesic between the two entities involved is a single line segment.*

► **Lemma 8.** *Let \mathcal{X} be a set of n entities, each moving amidst a set of well-spaced obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices. The number of ϵ -events is at most $O(\tau n^2 m)$, where m is the total complexity of \mathcal{O} .*

Proof. By Observation 7 there are only $O(\tau n^2)$ ϵ -events in which the geodesic is a single line segment. We now bound the number of ϵ -events for which the geodesic contains an obstacle vertex v by $O(\tau n^2)$. The lemma then follows. Fix two entities a and b . Each trajectory edge intersects the ϵ -disk centered at v at most once. Hence, there are $O(\tau)$ time intervals during which both a and b move along a line, and are within distance ϵ from v . Clearly, all ϵ -events occur within one of these intervals. Since the obstacles are well spaced, the ϵ -disk contains at most three edges: the two edges connected to v and at most one edge adjacent to both these edges. It follows that the function ς_{ab} consists of at most $O(1)$ pieces during such an interval. Hence, there can be at most a constant number of ϵ -events per interval. ◀

Next, we show that the number of critical events can only be $O(\tau(n^2 + m\lambda_4(n)))$. Clearly, the number of critical events at which the geodesic is a single line segment is also at most $O(\tau n^2)$ (Observation 7). We now bound the number of critical events where two sets of entities become ϵ -connected or ϵ -disconnected, and the geodesic between them consists of two line segments, connected via an obstacle vertex v .

Let \mathcal{D} be the disk of radius ϵ centered at v , and consider a subdivision of \mathcal{D} into six equal size sectors or *wedges*. See Fig. 3. We make sure that the obstacle containing v intersects at least two wedges. Let W be such a wedge. For any pair of points p and q in W , the Euclidean distance between p and q is at most ϵ . Let $\mathcal{X}_W(t) \subseteq \mathcal{X}$ denote the set of entities that lie in W at time t .

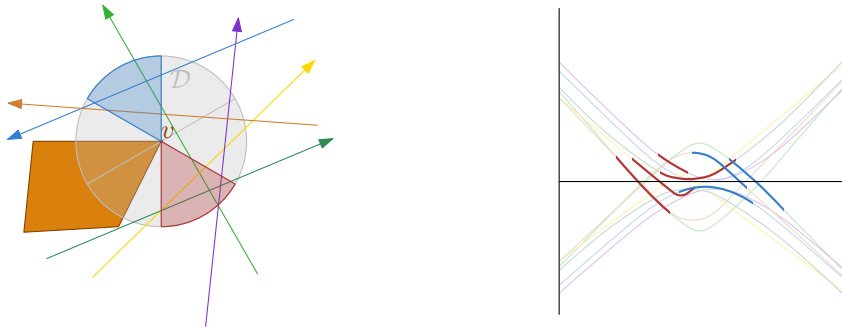


■ **Figure 3** The ϵ -disk \mathcal{D} (red) centered at v subdivided into six wedges. The distance between any pair of points p and q in the same wedge is at most ϵ .

► **Observation 9.** *At any time t , there is at most one maximal set of ϵ -connected entities G that has entities in wedge W , that is, for which $G \cap \mathcal{X}_W(t) \neq \emptyset$.*

► **Corollary 10.** *At any time t , there is at most one maximal set of ϵ -connected entities G such that $\mathcal{X}_W(t) \subseteq G$.*

When two maximal sets of ϵ -connected entities \mathcal{X}_R and \mathcal{X}_B become ϵ -connected or ϵ -disconnected at time t via vertex v , then the entities $r \in \mathcal{X}_R$ and $b \in \mathcal{X}_B$ that form their



■ **Figure 4** A set of entities on the left, and the corresponding sets of partial functions \mathcal{R} (red) and \mathcal{B} (blue). Critical events correspond to intersections between the lower envelope of \mathcal{R} and the upper envelope of \mathcal{B} .

closest pair must both lie in \mathcal{D} at time t . More specifically, since the geodesic between r and b uses vertex v , r and b must lie in different wedges. Let R and B denote the wedges that contain r and b , respectively. We now show that the total number of critical events involving entities in wedges R and B is $O(\tau\lambda_4(n))$. By Corollary 10 it then follows that each such event corresponds to exactly one pair of ε -connected sets. Since there are only 15 pairs of wedges, there are also at most $O(\tau\lambda_4(n))$ times when two maximal sets of ε -connected entities are at distance exactly ε and are connected via vertex v .

► **Lemma 11.** *The total number of critical events involving entities in wedges R and B is $O(\tau\lambda_4(n))$.*

Proof. Given an entity $a \in \mathcal{X}$ we define two partial functions ϱ_a and β_a as follows:

$$\varrho_a(t) = \begin{cases} \xi_{av}(t) - \varepsilon/2 & \text{if } a \in \mathcal{X}_R(t) \\ \perp & \text{otherwise,} \end{cases} \quad \beta_a(t) = \begin{cases} -\xi_{av}(t) + \varepsilon/2 & \text{if } a \in \mathcal{X}_B(t) \\ \perp & \text{otherwise,} \end{cases}$$

where \perp denotes undefined. Furthermore, let $\mathcal{R} = \{\varrho_r \mid r \in \mathcal{X}\}$ and $\mathcal{B} = \{\beta_b \mid b \in \mathcal{X}\}$. See Fig. 4. It now follows that for any two entities $r \in \mathcal{X}_R(t)$ and $b \in \mathcal{X}_B(t)$ the length of the path from r via v to b is ε if and only if $\varrho_r(t) = \beta_b(t)$. Thus, the number of times entities in R become ε -connected or ε -disconnected via vertex v is at most the number of intersection points between the lower envelope of \mathcal{R} and the upper envelope of \mathcal{B} . Next, we show that this number of intersection points is at most $O(\tau\lambda_4(n))$.

Each trajectory consists of $\tau - 1$ edges, each of which intersects wedge R in a single line segment. Hence, for each entity a , the function ϱ_a is defined on at most $\tau - 1$ maximal contiguous intervals $I_a^1, \dots, I_a^{\tau-1}$. Thus, by Lemma 1 the lower envelope \mathcal{L} of \mathcal{R} has complexity at most $O(\tau\lambda_4(n))$. Similarly, the upper envelope \mathcal{U} of \mathcal{B} has complexity $O(\tau\lambda_4(n))$. It follows that there are also $O(\tau\lambda_4(n))$ time intervals such that both \mathcal{L} and \mathcal{U} are represented by a simple hyperbolic function. In each such interval \mathcal{L} and \mathcal{U} intersect each other at most twice. Hence, the total number of intersection points is $O(\tau\lambda_4(n))$. ◀

It now follows that the total number of critical events at which the geodesic contains an obstacle vertex is $O(m\tau\lambda_4(n))$. We conclude:

► **Theorem 12.** *Let \mathcal{X} be a set of n entities, each moving amidst a set of well-spaced obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices. The number of critical events is at most $O(\tau(n^2 + m\lambda_4(n)))$, where m is the total complexity of \mathcal{O} .*

4.3 Algorithm

We now show how to compute the Reeb graph \mathcal{R} in case the entities move among well-spaced obstacles. At first glance, it seems that we can compute all critical events using the same approach as used in the upper bound proof. Indeed, this allows us to find all times at which critical events occur. However, to construct the Reeb graph we also need to know the sets of entities involved at each critical event, e.g. we want to know that a set \mathcal{X}' splits into subsets R and B . Unfortunately, there does not seem to be an efficient, i.e. sub-linear, way to obtain this information, nor can we easily maintain the ε -connected sets of entities without considering all ε -events. It is easy to compute all ε -events in $O(\tau n^2 m)$ time, using the approach described in Lemma 8. Once we have computed all ε -events, we can construct the Reeb graph using the same method described by Buchin et al. [2]. This takes $O(\log n)$ time per ε -event. Thus, we conclude:

► **Theorem 13.** *Let \mathcal{X} be a set of n entities, each moving amidst a set of well-spaced obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices. The Reeb graph \mathcal{R} representing the movement of the entities in \mathcal{X} has size $O(\tau(n^2 + m\lambda_4(n)))$ and can be computed in $O(\tau n^2 m \log n)$ time, where m is the total complexity of \mathcal{O} .*

5 General Obstacles

Finally, we study the most general case in which the entities move amidst multiple obstacles, and there are no restrictions on the locations, shape, or size of the obstacles.

5.1 Lower Bound

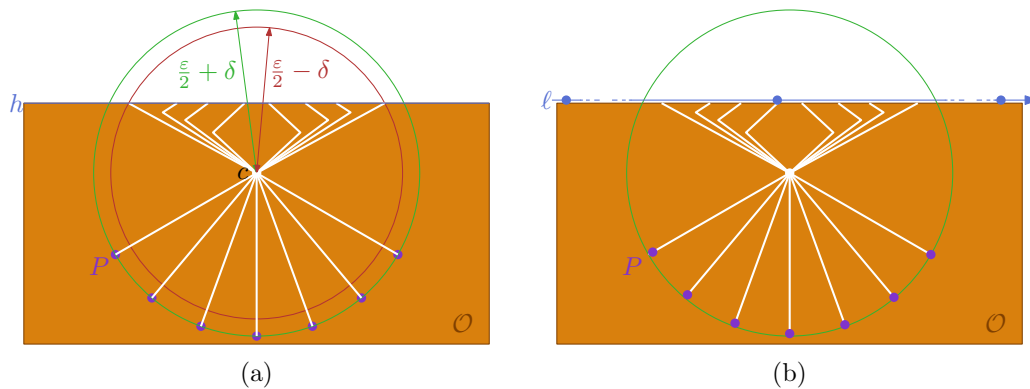
► **Lemma 14.** *The total number of critical events for a set of n entities, each moving amidst a set of obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices, is $\Omega(\tau(n^2 + nm \min\{n, m\}))$, where m is the total complexity of \mathcal{O} .*

Proof. We describe a construction in which the entities move along lines that yields $\Omega(nmk)$ critical events, with $k = \min\{n, m\}$. We again repeat this construction $\Omega(\tau)$ times.

The basic idea is to create $\Omega(k)$ stationary entities, $\Omega(n)$ moving entities, and $\Omega(m)$ “entrances” from which a moving entity can become connected with a stationary entity. Each stationary entity is surrounded by an obstacle. The distance from such a stationary entity s to an entrance leading to s , will be approximately ε . So an entity gets ε -connected with s only if it is directly in front of the entrance. We make sure that each stationary entity is reachable from all entrances. Hence, each time that one of the $\Omega(n)$ moving entities passes an entrance it will generate $\Omega(k)$ critical events. Since all $\Omega(n)$ moving entities encounter all $\Omega(m)$ entrances we get at least $\Omega(nmk)$ critical events as desired.

Let c be a point in the plane, let $\delta > 0$ be a small value, and let P be a set of $\Omega(k)$ points on the lower half of the circle with center c and radius $\varepsilon/2 + \delta$. We place a large rectangular obstacle \mathcal{O} containing c and all points in P such that the (shortest) distance from c to the top side h of \mathcal{O} is smaller than $\varepsilon/2 - \delta$. See Fig. 5(a).

We now carve $\Omega(k + m)$ passages through \mathcal{O} . The first $k' = \Omega(k)$ connect c to each point in P . The remaining $m' = \Omega(m)$ connect c to the top side h of the obstacle \mathcal{O} . The first k' passages all have length exactly $\varepsilon/2 + \delta$, and we make sure that the remaining m' passages all have length exactly $\varepsilon/2 - \delta$. We can do this with at most one bend in each passage. See Fig. 5(a). The distance from any point in P to the top side of \mathcal{O} , via any of the m' passages, is now ε , and the distance between any two points in P is strictly larger than ε .



■ **Figure 5** The lower bound construction for general obstacles. (a) Constructing the passages through obstacle \mathcal{O} . (b) The final construction.

We place a stationary entity on each point in P , and we let $\Omega(n)$ entities move from left to right on a horizontal line ℓ containing h (we can move ℓ upwards a bit later to make sure the entities do not intersect the obstacle). We make sure that at any time the distance between two of these moving entities is larger than ε , so they are never in the same ε -connected set. When an entity e arrives at an entrance, that is, an opening of one of the top passages, it is at distance ε to the points in P . Hence, we have a critical event where e connects with all entities at points in P . We can make sure that e generates an event with (the entity on) each point in P by moving each point in P by a small unique amount towards c . Fig. 5(b) shows the resulting construction. ◀

5.2 Upper Bound

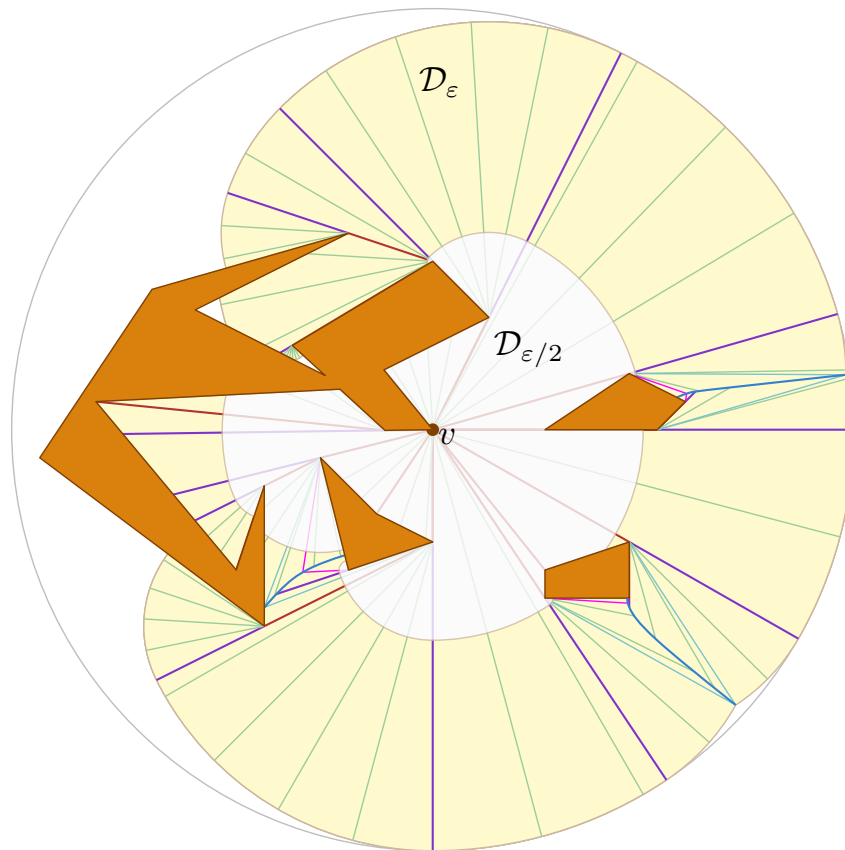
We again start by bounding the total number of ε -events.

► **Lemma 15.** *Let \mathcal{X} be a set of n entities, each moving amidst a set of obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices. The number of ε -events is at most $O(\tau n^2 m^2)$, where m is the total complexity of \mathcal{O} .*

As in the case of well-spaced obstacles, ε -events are not necessarily critical events. We now fix an obstacle vertex v , and show that there are at most $O(\tau m^2 \lambda_4(n))$ critical events involving v . To this end, we again decompose the (geodesic) ε -disk centered at v into regions such that each region corresponds to at most one maximal set of ε -connected entities. Each critical event involving v also involves two maximal ε -connected sets, and thus two regions in this decomposition. We show that we have to consider only $O(m)$ pairs of such regions, and that for each pair there can be at most $O(\tau m \lambda_4(n))$ critical events. Since we have $O(m)$ obstacle vertices this gives us a total bound of $O(\tau m^3 \lambda_4(n))$. When m is at most $O(n^2 / \lambda_4(n))$, this is an improvement over the bound in Lemma 15. It follows that the total number of critical events is thus at most $O(\tau \min\{n^2 + m^3 \lambda_4(n), n^2 m^2\})$.

Let \mathcal{D}_ε denote the geodesic ε -disk centered at v , and let $\mathcal{D}_{\varepsilon/2}$ denote the geodesic $(\varepsilon/2)$ -disk centered at v . Clearly, the geodesic distance between any two points in $\mathcal{D}_{\varepsilon/2}$ is at most ε , thus we observe:

► **Observation 16.** *At any time t there is at most one maximal ε -connected set of entities G such that $G_{\mathcal{D}_{\varepsilon/2}}(t) \neq \emptyset$, and thus $\mathcal{X}_{\mathcal{D}_{\varepsilon/2}}(t) \subseteq G$.*



■ **Figure 6** Subdivision Φ . The color of the edge indicates its type: the red edges originate from shortest paths, the purple and blue edges from the shortest path map, the cyan edges from the subdivision in “triangular sectors”, the light green edges guarantee that the maximum angle at the routing point is at most $\pi/12$, and the pink edges guarantee monotonicity.

Let $\mathcal{A} = \mathcal{D}_\varepsilon \setminus \mathcal{D}_{\varepsilon/2}$. We decompose \mathcal{A} into $O(m)$ regions such that for each region R we have that (i) the geodesic distance between two points $p, q \in R$ is at most ε , (ii) any two points $p, q \in R$ have the same (combinatorial) geodesic to v , and (iii) the boundary of R has constant complexity.

Let Φ denote this decomposition of \mathcal{A} . It follows that at any time, each region R in Φ contains entities from at most one maximal ε -connected set G . That is, $\mathcal{X}_R(t) \subseteq G$. It is now easy to see that any critical event involving v involves the maximal set of ε -connected entities $G_{\varepsilon/2}$ corresponding to $\mathcal{D}_{\varepsilon/2}$, and a maximal set of ε -connected entities G_R corresponding to a region R of Φ . Hence, there are only $O(m)$ pairs of regions that can be associated with a critical event involving v . We now show how to construct Φ , and how to bound the number of events corresponding to a single pair of regions.

Obtaining subdivision Φ . Let Φ' be the overlay of the shortest path map with root v (restricted to \mathcal{D}_ε), and all shortest paths from v to obstacle vertices in \mathcal{D}_ε .

► **Observation 17.** Φ' has complexity $O(m)$.

The edges of Φ' are either line segments or hyperbolic arcs [9]. Since Φ' is a refinement of the shortest path map, all points in a region R in Φ' have the same geodesic g to v (except

for the starting edge). Hence, each region R is star-shaped, and has a vertex c that lies inside the kernel. This vertex c is the second vertex on each geodesic g . We refer to c as the *routing point* of R .

Next, we further subdivide each region R in Φ' . We add edges \overline{cu} between the routing point c and all boundary vertices u of R . Each region is now bounded by two line segments \overline{cu} and \overline{cw} and a segment \widetilde{cw} . The segment \widetilde{cw} is either a line segment, or a hyperbolic arc. We further add edges \overline{cz} between c and points z on \widetilde{cw} such that the angle at c is at most $\theta = \pi/12$. In case \widetilde{cw} is a hyperbolic arc we make sure that the hyperbolic function describing this arc is monotonic. To this end, we add at most one additional edge \overline{cz} to the point z on \widetilde{cw} with maximum curvature. All these new edges are contained in R and do not intersect each other. It follows that the total complexity, summed over all regions in the subdivision, is still $O(m)$. Let Φ denote the resulting subdivision, restricted to \mathcal{A} . See Fig. 6.

► **Lemma 18.** *Let R be a region in Φ . For any two points $p, q \in R$ the Euclidean distance $\|pq\|$ between p and q is at most $\varepsilon\sqrt{29/4 - 4\sqrt{3}}$.*

► **Lemma 19.** *Let R be a region in Φ . For any two points $p, q \in R$ the geodesic distance $\zeta(p, q) = \zeta(g(p, q))$ between p and q is at most ε .*

► **Lemma 20.** *Subdivision Φ has complexity $O(m)$ and each region $R \in \Phi$ has the following properties:*

- (i) *the geodesic distance between two points $p, q \in R$ is at most ε ,*
- (ii) *any two points $p, q \in R$ have the same geodesic to v (excluding the starting edge), and*
- (iii) *the boundary of R has constant complexity.*

Proof. Property (i) follows directly from Lemma 19, and Property (ii) follows from the fact that Φ is a refinement of the shortest path map. Each region is bounded by three or four segments, depending if the routing point c lies in \mathcal{A} or not. If $c \in \mathcal{A}$, region R is bounded by three segments. Otherwise, R is bounded by three segments and a part of $D_{\varepsilon/2}$. However, as all shortest paths from points in R to v use point c , it follows that this part of $D_{\varepsilon/2}$ is also a single hyperbolic segment. This proves Property (iii). ◀

Bounding the number of critical events for a pair of regions. Next, we fix a region R in Φ , and show that the number of critical events involving v , R , and $D_{\varepsilon/2}$, is at most $O(\tau\lambda_4(n))$.

► **Lemma 21.** *Let R be any region of Φ , and let G_R be the maximal set of ε -connected entities corresponding to R . The (geodesic) distance between G_R and v is given by a piecewise hyperbolic function with $O(\tau\lambda_4(n))$ pieces.*

Proof. The boundary of R has constant complexity, so each entity in G_R intersects region R in $O(\tau)$ time-intervals. Furthermore, all points in R have the same combinatorial geodesic, so during any such an interval, the distance to v is given by a simple hyperbolic function. Thus, the distance function between G_R and v corresponds to the lower envelope of a set of hyperbolic functions. Lemma 1 now completes the proof. ◀

Fix a region R , let

$$\beta_a(t) = \begin{cases} -\zeta(a(t), v) + \varepsilon & \text{if } a(t) \in R \\ \perp & \text{otherwise,} \end{cases}$$

and let \mathcal{U} be the upper envelope of $\{\beta_a(t) \mid a \in \mathcal{X}\}$. It follows from Lemma 21 that \mathcal{U} has complexity $O(\tau\lambda_4(n))$.

Now consider the entities in the inner region $\mathcal{D}_{\varepsilon/2}$. The function ς_{av} expressing the geodesic distance between a and v is piecewise hyperbolic and consists of $O(m\tau)$ pieces. Let \mathcal{L} denote the lower envelope of all functions ϱ_a , $a \in \mathcal{X}$, where $\varrho_a(t) = \varsigma_{av}(t)$ if $\varsigma_{av}(t) \leq \varepsilon/2$ and \perp otherwise. It follows from Lemma 1 that \mathcal{L} has complexity $O(m\tau\lambda_4(n))$.

As with the well-spaced obstacles, all critical events in which the entities involved lie in $\mathcal{D}_{\varepsilon/2}$ and R at the time of the event correspond to intersections of \mathcal{L} and \mathcal{U} . To bound the number of intersections, and thus the number of critical events, we now (again) partition the domain of \mathcal{L} and \mathcal{U} (i.e., time) into sets D_1, \dots, D_k such that in each D_i the lower envelope \mathcal{L} and the upper envelope \mathcal{U} intersect at most twice. It is easy to partition the domain into $k = O(|\mathcal{L}| + |\mathcal{U}|) = O(\tau\lambda_4(n) + m\tau\lambda_4(n)) = O(m\tau\lambda_4(n))$ intervals with this property. Hence, we get $O(m\tau\lambda_4(n))$ critical events involving vertex v and the pair of regions $(R, \mathcal{D}_{\varepsilon/2})$. This gives a total of $O(m^3\tau\lambda_4(n))$ critical events. Together with the bound on the number of ε -events (Lemma 15) this gives us the following result:

► **Theorem 22.** *Let \mathcal{X} be a set of n entities, each moving amidst a set of obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices. The number of critical events is at most $O(\tau \min\{n^2 + m^3\lambda_4(n), n^2m^2\})$, where m is the total complexity of \mathcal{O} .*

5.3 Algorithm

We again explicitly compute all ε -events in order to construct the Reeb graph \mathcal{R} . We follow the approach from Lemma 15. That is, we compute the shortest path map Ψ with root v , and for each pair of entities a and b we trace their trajectories through Ψ . For each of the $O(\tau m)$ pairs of regions visited, we construct ς_{ab} and find the ε -events. Computing the shortest path map with root v takes $O(m \log m)$ time [9]. Tracing the trajectories and computing the distance functions takes time proportional to the number of regions visited. Hence, we spend $O(\tau m)$ time for each pair. It follows that the total time required to compute all ε -events is $O(m(m \log m + n^2\tau m)) = O(\tau n^2m^2 + m^2 \log m)$. Computing \mathcal{R} again takes $O(\log n)$ time per ε -event. We obtain the following result.

► **Theorem 23.** *Let \mathcal{X} be a set of n entities, each moving amidst a set of obstacles \mathcal{O} along a piecewise linear trajectory with τ vertices. The Reeb graph \mathcal{R} representing the movement of the entities in \mathcal{X} has size $O(\tau \min\{n^2 + m^3\lambda_4(n), n^2m^2\})$ and can be computed in $O(\tau n^2m^2 \log n + m^2 \log m)$ time, where m is the total complexity of \mathcal{O} .*

6 Concluding Remarks

We study the trajectory grouping structure for entities moving amidst obstacles. To this end, we analyze the number of times when two sets of entities are at distance ε from each other. Our results for various types of obstacles can be found in Table 1. These bounds on the number of critical events also give a bound on the size of the Reeb graph \mathcal{R} . This in turn gives bounds on the number of maximal groups: if the Reeb graph has size $O(|\mathcal{R}|)$ there are $O(|\mathcal{R}|n)$ maximal groups [2]. Furthermore, we present efficient algorithms to compute \mathcal{R} , which leads to efficient algorithms to compute the grouping structure.

One intriguing open question is whether the Reeb graph can be constructed using only the critical events, that is, in an output-sensitive manner. The difficulty with the approach as described in [2] appears to be that one would need a dynamic data structure for maintaining a subdivision of a set (the groups), that supports efficient split and merge operations. Thus,

there may be fundamental graph-theoretical obstacles to this approach. However, it is not clear that this is the only possible approach to compute \mathcal{R} .

An other direction of future work is to extend the grouping structure for entities moving in more realistic environments, for instance modeled by weighted regions. This starts with interesting modeling questions since distances are related to the speed of the entities. For example: should the distance for two entities, say sheep, to be directly connected be larger on a muddy field than it is on a concrete courtyard, or do the sheep need to be closer together in the field to be considered a group?

Although we developed the technical machinery in this paper with the goal of extending the trajectory grouping structure, we foresee wider applications for our techniques. We believe our work will serve as a starting point for more general research related to moving entities and geodesic distances. For example, we can consider trajectory similarity measures in the presence of obstacles.

Acknowledgments. M. L., F. S., I. K., and B. S. are supported by the Netherlands Organisation for Scientific Research (NWO) under grants 639.021.123, 612.001.022, 612.001.106, and 639.023.208 respectively.

References

- 1 Marc Benkert, Joachim Gudmundsson, Florian Hübner, and Thomas Wolle. Reporting flock patterns. *Computational Geometry*, 41(3):111–125, 2008.
- 2 Kevin Buchin, Maike Buchin, Marc van Kreveld, Bettina Speckmann, and Frank Staals. Trajectory grouping structure. In *Proc. 2013 WADS Algorithms and Data Structures Symposium*, LNCS, pages 219–230. Springer, 2013.
- 3 Maike Buchin, Somayeh Dodge, and Bettina Speckmann. Context-aware similarity of trajectories. In *Geographic Information Science*, volume 7478 of LNCS, pages 43–56. Springer, 2012.
- 4 Maike Buchin, Anne Driemel, and Bettina Speckmann. Computing the Fréchet distance with shortcuts is NP-hard. In *Symposium on Computational Geometry*, page 367. ACM, 2014.
- 5 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete Comput. Geom.*, 6(5):485–524, 1991.
- 6 Herbert Edelsbrunner and John L. Harer. *Computational Topology – an introduction*. American Mathematical Society, 2010.
- 7 Joachim Gudmundsson and Marc van Kreveld. Computing longest duration flocks in trajectory data. In *Proc. 14th ACM International Symposium on Advances in Geographic Information Systems*, GIS’06, pages 35–42. ACM, 2006.
- 8 Leonidas J. Guibas and John Hershberger. Optimal shortest path queries in a simple polygon. *Journal of Computer and System Sciences*, 39(2):126–152, 1989.
- 9 John Hershberger and Subhash Suri. An Optimal Algorithm for Euclidean Shortest Paths in the Plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.
- 10 Hoyoung Jeung, Man Lung Yiu, Xiaofang Zhou, Christian S. Jensen, and Heng Tao Shen. Discovery of convoys in trajectory databases. *PVLDB*, 1:1068–1080, 2008.
- 11 Panos Kalnis, Nikos Mamoulis, and Spiridon Bakiras. On discovering moving clusters in spatio-temporal data. In *Advances in Spatial and Temporal Databases*, volume 3633 of LNCS, pages 364–381. Springer, 2005.
- 12 Patrick Laube, Marc van Kreveld, and Stephan Imfeld. Finding REMO – detecting relative motion patterns in geospatial lifelines. In *Developments in Spatial Data Handling*, pages 201–215. Springer, 2005.

- 13 Nimrod Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *J. ACM*, 30(4):852–865, 1983.
- 14 Salman Parsa. A deterministic $O(m \log m)$ time algorithm for the Reeb graph. In *Proc. 28th ACM Symposium on Computational Geometry*, pages 269–276, 2012.

From Proximity to Utility: A Voronoi Partition of Pareto Optima*

Hsien-Chih Chang, Sariel Har-Peled, and Benjamin Raichel

Department of Computer Science, University of Illinois
201 N. Goodwin Avenue, Urbana, IL, 61801, USA
{hchang17,sariel,raichel2}@illinois.edu

Abstract

We present an extension of Voronoi diagrams where not only the distance to the site is taken into account when considering which site the client is going to use, but additional attributes (i.e., prices or weights) are also considered. A cell in this diagram is then the loci of all clients that consider the same set of sites to be relevant. In particular, the precise site a client might use from this candidate set depends on parameters that might change between usages, and the candidate set lists all of the relevant sites. The resulting diagram is significantly more expressive than Voronoi diagrams, but naturally has the drawback that its complexity, even in the plane, might be quite high. Nevertheless, we show that if the attributes of the sites are drawn from the same distribution (note that the locations are fixed), then the expected complexity of the candidate diagram is near linear. To this end, we derive several new technical results, which are of independent interest.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, I.1.2 Algorithms, I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Voronoi diagrams, expected complexity, backward analysis, Pareto optima, candidate diagram, Clarkson-Shor technique

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.689

1 Introduction

Informal description of the candidate diagram. Suppose you open your refrigerator one day to discover it is time to go grocery shopping.¹ Which store you go to will be determined by a number of different factors. For example, what items you are buying, and do you want the cheapest price or highest quality, and how much time you have for this chore. Naturally the distance to the store will also be a factor. On different days which store is the best to go to will differ based on that day's preferences. However, there are certain stores you will never shop at. These are stores which are worse in every way than some other store (i.e., further, more expensive, lower quality, etc.). Therefore, the stores that are relevant and therefore in the *candidate set* are those that are not strictly worse in every way than some other store. Thus, every point in the plane is mapped to a set of stores that a client at that location might use. The *candidate diagram* is the partition of the plane into regions, where each candidate set is the same for all points in the same region. Naturally, if your only consideration is distance, then this is the (classical) Voronoi diagram of the sites. However,

* Work on this paper was partially supported by NSF AF award CCF-1421231, and CCF-1217462. The paper is also available on the arXiv [11].

¹ Unless you are feeling adventurous enough that day to eat the frozen mystery food stuck to the back of the freezer, which we strongly discourage you from doing.



here deciding which shop to use is an instance of multi-objective optimization – there are multiple, potentially competing, objectives to be optimized, and the decision might change as the weighting and influence of these objectives mutate over time (in particular, you might decide to do your shopping in different stores for different products). The concept of relevant stores discussed above is often referred as the *Pareto optima*.

Pareto optima in welfare economics. Pareto efficiency, named after Vilfredo Pareto, is a core concept in economic theory and more specifically in welfare economics. Here each point in \mathbb{R}^d represents the corresponding utilities of d players for a particular allocation of finite resources. A point is said to be *Pareto optimal* if there is no other allocation which increases the utility of any individual without decreasing the utility of another. The *First Fundamental Theorem of Welfare Economics* states that any competitive equilibrium (i.e., supply equals demand) is Pareto optimal. The origins of this theorem date back to 1776 with Adam Smith’s famous (and controversial) work, “The Wealth of Nations,” but was not formally *proven* until the 20th century by Lerner, Lange, and Arrow (see [15]). Naturally such proofs rely on simplifying (i.e., potentially unrealistic) assumptions such as perfect knowledge, or absence of externalities. The *Second Fundamental Theorem of Welfare Economics* states that any Pareto optimum is achievable through lump-sum transfers (i.e. taxation and redistribution). In other words each Pareto optima is a “best solution” under some set of societal preferences, and is achievable through redistribution in one form or another (see [15] for a more in depth discussion).

Pareto optima in computer science. In computational geometry such Pareto optima points relate to the *orthogonal convex hull* [22], which in turn relates to the well known convex hull (the input points that lie on the orthogonal convex hull is a super set of those which lie on the convex hull). Pareto optima are also of importance to the database community [10, 20], in which context such points are called *maximal* or *skyline points*. Such points are of interest as they can be seen as the relevant subset of the (potentially much larger) result of a relational database query. The standard example is querying a database of hotels for the cheapest and closest hotel, where naturally hotels which are farther and more expensive than an alternative hotel are not relevant results. There is a significant amount of work on computing these points, see Kung *et al.* [21]. More recently, Godfrey *et al.* [16] compared various approaches for the computation of these points (from a databases perspective), and also introduced their own new external algorithm.²

Modeling uncertainty. Recently, there is a growing interest in modeling uncertainty in data. As real data is acquired via physical measurements, noise and errors are introduced. This can be addressed by treating the data as coming from a distribution (e.g., a point location might be interpreted as a center of a Gaussian), and computing desired classical quantities adapted for such settings. Thus, a nearest-neighbor query becomes a probabilistic question – what is the expected distance to the nearest-neighbor? What is the most likely point to be the nearest-neighbor? (See [1] and references therein for more information.)

This in turn gives rise to the question of what is the expected complexity of geometric structures defined over such data. The case where the data is a set of points, and the

² There is of course a lot of other work on Pareto optimal points, from connections to Nash equilibrium to scheduling. We resisted the temptation of including many such references which are not directly related to our paper.

locations of the points are chosen randomly was thoroughly investigated (see [23, 27, 18] and references therein). The problem, when the locations are fixed but the weights associated with the points are chosen randomly, is relatively new. Agarwal *et al.* [2] showed that for a set of disjoint segments in the plane, if they are being expanded randomly, then the expected complexity of the union is near linear. This result is somewhat surprising as in the worst case the complexity of such a union is quadratic.

Here we are interested in bounding the expected complexity of weighted generalizations of Voronoi diagrams (described below), where the weights (not the site locations) are randomly sampled. Note that the result of Agarwal *et al.* [2] can be interpreted as bounding the expected complexity of level sets of the multiplicatively weighted Voronoi diagram (of segments). On the other hand, here we want to bound the entire lower envelope (which implies the same bound on any level set). For the special case of multiplicative weighted Voronoi diagrams, a near linear expected complexity bound was provided by Har-Peled and Raichel [18]. In this work we consider a much more general class of weighted diagrams which allow multiple weights and non-linear distance functions.

1.1 Our contributions

Conceptual contribution. We define formally the *candidate diagram* in Section 2.1 – a new geometric structure that combines proximity information with utility. For every point x in the plane, the diagram associates a *candidate set* $L(x)$ of sites that are relevant to x ; that is, all the sites that are Pareto optima for x . Putting it differently, a site is not in $L(x)$ if it is further away from and worse in all parameters than some other site. Significantly, unlike the traditional Voronoi diagram, the candidate diagram allows the user to change their distance function, as long as the function respects the domination relationship. This diagram is a significant extension of the Voronoi diagram, and includes other extensions of Voronoi diagrams as special subcases, like multiplicative weighted Voronoi diagrams. Not surprisingly, the worst case complexity of this diagram can be quite high.

Technical contribution. We consider the case where each site chooses its j th attribute from some distribution \mathcal{D}_j independently for each j . We show that the candidate diagram in expectation has near linear complexity, and that, with high probability, the candidate set has poly-logarithmic size for any point in the plane. In the process we derive several results which are interesting in their own right.

- (A) **Low complexity of the minima for random points in the hypercube.** We prove that if n points are sampled from a fixed distribution (see Section 2.2 for assumptions on the distribution) over the d -dimensional hypercube then, with probability $1 - 1/n^{\Omega(1)}$, the number of Pareto optima points is $O(\log^{d-1} n)$, which is within a constant factor of the expectation. Previously, this result was only known in a weaker form that is insufficient to imply our other results. Specifically, Bai *et al.* [6] proved that after normalization the cumulative distribution function of the number of Pareto optima points is normal, up to an additive error $O(1/\text{polylog } n)$. (See [7, 8] as well.) In particular, their results (which are quite nice and mathematically involved) only imply the statement with probability $1 - 1/\text{polylog } n$. To the best of our knowledge this result is new – we emphasize, however, that for our purposes a weaker bound of $O(\log^d n)$ is sufficient, and such a weaker result follows readily from the ε -net theorem [19] (naturally, this would add a log factor to later results in the paper).
- (B) **Backward analysis with high probability.** To get this result, we prove a lemma providing high probability bounds when applying backwards analysis [24]. Such tail estimates

are known in the context of randomized incremental algorithms [13, 9], but our proof is arguably more direct and cleaner, and should be applicable to more cases. See Section 2.3 and the full version of the paper [11].

- (C) **Overlay of the k th order Voronoi cells in randomized incremental construction.** We prove that the overlay of cells during a randomized incremental construction of the k th order Voronoi diagram is of complexity $O(k^4 n \log n)$ (see Lemma 15).
- (D) **Complexity of the candidate diagram.** Combining the above results carefully yields a near-linear upper bound on the complexity of the candidate diagram (see Theorem 17).

Outline. In Section 2 we formally define our problem and introduce some tools that will be used later on. Specifically, after some required preliminaries, we formally introduce the candidate diagram in Section 2.1. The sampling model used is described in detail in Section 2.2. In Section 2.3, we discuss backward analysis with high-probability bounds.

To bound the complexity of the candidate diagram (i.e., both the size of the planar partition and the total size of the associated candidate sets), in Section 3, we use the notion of *proxy set*. Defined formally in Section 3.1, it is (informally) an enlarged candidate set. Section 3.2 bounds the size of the proxy set using backward analysis, both in expectation and with high probability. Section 3.3 shows that mucking around with the proxy set is useful, by proving that the proxy set contains the candidate set, for any point in the plane.

In Section 4, we show that the diagram induced by the proxy sets can be interpreted as the arrangement formed by the overlay of cells during the randomized incremental construction of the k th order Voronoi diagram. To this end, Section 4.1 defines the k th order Voronoi diagram, the k *environment* of a site, and states some basic properties of these entities. For our purposes, we need to bound the size of the conflict lists encountered during the randomized incremental construction, and this is done in Section 4.2 using the Clarkson-Shor moment technique. Next, in Section 4.3, we bound the expected complexity of the proxy diagram.

In Section 5, we bound the expected size of the candidate set for any point in the plane. First, we analyze the number of staircase points of random point sets in hypercubes, and we use this bound to bound the size of the candidate set.

In Section 6, we put everything together, and prove our main result, showing the desired bound on the complexity of the candidate diagram.

In the full version of the paper [11], we fill in the missing details for the results of Section 2.3, proving a high-probability bound for backward analysis.

2 Problem definition and preliminaries

Throughout, we assume the reader is familiar with standard computational geometry terms, such as arrangements [26], vertical-decomposition [9], etc. In the same vein, we assume that the variable d , the *dimension*, is a small constant and the O notation hides constants that are potentially exponential (or worse) in d .

A quantity is *bounded by $O(f)$ with high probability* with respect to n , if for any large enough constant $\gamma > 0$, there is another constant c depending on γ such that the quantity is at most $c \cdot f$ with probability at least $1 - n^{-\gamma}$. In other words, the bound holds for any sufficiently small polynomial error with the expense of a multiplicative constant factor on the size of the bound. When there's no danger of confusion, we sometimes write $O_{\text{whp}}(f)$ for short.

► **Definition 1.** Consider two points $\mathbf{p} = (p_1, \dots, p_d)$ and $\mathbf{q} = (q_1, \dots, q_d)$ in \mathbb{R}^d . The point \mathbf{p} *dominates* \mathbf{q} (denoted by $\mathbf{p} \preceq \mathbf{q}$) if $p_i \leq q_i$, for all i .

Given a point set $P \subseteq \mathbb{R}^d$, there are several terms for the subset of P that is not dominated, as discussed above, such as *Pareto optima* or *minima*. Here, we use the following term.

► **Definition 2.** For a point set $P \subseteq \mathbb{R}^d$, a point $\mathbf{p} \in P$ is a *staircase point* of P if no other point of P dominates it. The set of all such points, denoted by $\text{stair}(P)$, is the *staircase* of P .

Observe that for a finite point set P , the staircase $\text{stair}(P)$ is never empty.

2.1 Formal definition of the candidate diagram

Let $S = \{s_1, \dots, s_n\}$ be a set of n distinct *sites* in the plane. For each site s in S , there is an associated list $\alpha = \langle a_1, \dots, a_d \rangle$ of d real-valued attributes, each in the interval $[0, 1]$. When viewed as a point in the unit hypercube $[0, 1]^d$, this list of attributes is the *parametric point* of the site s_i . Specifically, a site is a point in the plane encoding a facility location, while the term *point* is used to refer to the (parametric) point encoding its attributes in \mathbb{R}^d .

Preferences. Fix a client location x in the plane. For each site, there are $d + 1$ associated variables for the client to consider. Specifically, the client distance to the site, and d additional attributes (e.g., prices of d different products) associated with the site. Conceptually, the goal of the client is to “pay” as little as possible by choosing the best site (e.g., minimize the overall cost of buying these d products together from a site, where the price of traveling the distance to the site is also taken into account).

► **Definition 3.** A client x has a *dominating preference* if for any two sites s, s' in the plane, with parametric points $\alpha, \alpha' \in \mathbb{R}^d$ respectively, the client would prefer the site s over s' if $\|x - s\| \leq \|x - s'\|$ and $\alpha \preceq \alpha'$ (that is, α dominates α').

Note that a client having a dominating preference does not identify a specific optimum site for the client, but rather a set of potential optimum sites. Specifically, given a client location x in the plane, let its distance to the i th site be $\ell_i = \|x - s_i\|$. The set of sites the client might possibly use (assuming the client uses a dominating preference) are the staircase points of the set $P(x) = \{(\alpha_1, \ell_1), \dots, (\alpha_n, \ell_n)\}$ (i.e., we are adding the distance to each site as an additional attribute of the site – this attribute depends on the location of x). The set of sites realizing the staircase of $P(x)$ (i.e., all the sites relevant to x) is the *candidate set* $L(x)$ of x :

$$L(x) = \{s_i \in S \mid (\alpha_i, \ell_i) \text{ is a staircase point of } P(x) \text{ in } \mathbb{R}^{d+1}\}. \tag{1}$$

The *candidate cell* of x is the set of all the points in the plane that have the same candidate set associated with them; that is, $\{p \in \mathbb{R}^2 \mid L(p) = L(x)\}$. The decomposition of the plane into these cells is the *candidate diagram*.

Now, the client x has the candidate set $L(x)$, and it chooses some site (or potentially several sites) from $L(x)$ that it might want to use. Note that the client might decide to use different sites for different acquisitions. As an example, consider the case when each site s_i is attached with weights $\alpha_i = (a_{i,1}, a_{i,2})$. If the client x has the preference of choosing the site with smallest value $a_{i,1} \cdot \ell_i$ among all the sites, then this preference is a dominating preference, and therefore the client will choose one of the sites from the candidate list $L(x)$. (Observe that the preference function corresponds to the weighted Voronoi diagram with respect to the first coordinate of the weights.) Similarly, if the preference function is to

choose the smallest value $a_{i,1} \cdot \ell_i^2 + a_{i,2}$ among all the sites (which again is a dominating preference), then this corresponds to a power diagram of the sites.

Complexity of the diagram. The *complexity* of a planar arrangement is the total number of edges, faces, and vertices. A candidate diagram can be interpreted as a planar arrangement, and its complexity is defined analogously. The *space complexity* of the candidate diagram is the total amount of memory needed to store the diagram explicitly, and is bounded by the complexity of the candidate diagram together with the sum of the sizes of candidate sets over all the faces in the arrangement of the diagram (which is potentially larger by a factor of n , the number of sites). Note, that the space complexity is a somewhat naïve upper bound, as using persistent data-structures might significantly reduce the space needed to store the candidate lists.

► **Lemma 4** (For proof see [11]). *Given n sites in the plane, the complexity of the candidate diagram of the sites is $O(n^4)$. The space complexity of the candidate diagram of the sites is $\Omega(n^2)$ and $O(n^5)$.*

We leave the question of closing the gap in the bounds of Lemma 4 as an open problem for further research.

2.2 Sampling model

Fortunately, the situation changes when randomization is involved. Let S be a set of n sites in the plane. For each site $s \in S$, a parametric point $\alpha = (\alpha_1, \dots, \alpha_d)$ is sampled independently from $[0, 1]^d$, with the following constraint: each coordinate α_i is sampled from a (continuous) distribution \mathcal{D}_i , independently for each coordinate. In particular, the sorted order of the n parametric points by a specific coordinate yields a uniform random permutation (for the sake of simplicity of exposition we assume that all the values sampled are distinct).

Our main result shows that, under the above assumptions, both the complexity and the space complexity of the candidate diagram is near-linear in expectation – see Theorem 17 for the exact statement.

2.3 A short detour into backward analysis

Randomized incremental construction is a powerful technique used by geometric algorithms. Here, one is given a set of elements S (e.g., segments in the plane), and one is interested in computing some structure induced by these elements (e.g., the vertical decomposition formed by the segments). To this end, one computes a random permutation $\Pi = \langle s_1, \dots, s_n \rangle$ of the elements of S , and in the i th iteration one computes the structure V_i induced by the i th prefix $\Pi_i = \langle s_1, \dots, s_i \rangle$ of Π , by inserting the i th element s_i into V_{i-1} and updating it so it becomes V_i (e.g., split all the vertical trapezoids of V_{i-1} that intersect s_i , and merge together adjacent trapezoids with the same floor and ceiling).

In *backward analysis* one is interested in computing the probability that a specific object that exists in V_i was actually created in the i th iteration (e.g., a specific vertical trapezoid in the vertical decomposition V_i). If the object of interest is defined by at most b elements of Π_i , for some constant b , then the desired quantity is the probability that s_i is one of these defining elements, which is at most b/i . In some cases, the sum of these probabilities, over the n iterations, count the number of times certain events happen during the incremental construction. However, this yields only a bound in expectation. For a high probability bound, one can not apply this argument directly, as there is a subtle dependency leakage between

the corresponding indicator variables involved between different iterations. (Without going into a detailed example, this is because the defining sets of the objects of interest can have different sizes, and these sizes depend on which elements were used in the permutation in earlier iterations.)

Let P be a set of n elements. A *property* \mathcal{P} of P is a function that maps any subset X of P to a subset $\mathcal{P}(X)$ of X .

Intuitively the elements in $\mathcal{P}(X)$ have some desired property with respect to X (for example, let X be a set of points in the plane, then $\mathcal{P}(X)$ may be those points in X who lie on the convex hull of X). The following corollary (see full version of the paper for details [11]) provides a high probability bound for backward analysis, and while the proof is an easy application of the Chernoff inequality, it nevertheless significantly simplifies some classical results on randomized incremental construction algorithms. See the full version of the paper [11] for a more detailed discussion and a proof.

► **Corollary 5.** *Let P be a set of n elements, $c > 1$ and $k \geq 1$ prespecified numbers, and let $\mathcal{P}(X)$ be a property defined over any subset $X \subseteq P$. Now, consider a uniform random permutation $\langle p_1, \dots, p_n \rangle$ of P , and let $P_i = \{p_1, \dots, p_i\}$. Furthermore, assume that, for all i , we have, with probability at least $1 - n^{-c}$, that $|\mathcal{P}(P_i)| \leq k$. Let X_i be the indicator variable of the event $p_i \in \mathcal{P}(P_i)$. Then, for any constant $\gamma \geq 2e$, we have*

$$\Pr \left[\sum_{i=1}^n X_i > \gamma \cdot (2k \ln n) \right] \leq n^{-\gamma k} + n^{-c}.$$

(If for all $X \subseteq P$ we have that $|\mathcal{P}(X)| \leq k$, then the additional error term n^{-c} is not necessary.)

3 The proxy set

Providing a reasonable bound on the complexity of the candidate diagram directly seems challenging. Therefore, we instead define for each point x in the plane a slightly different set, called the *proxy set*. First we prove that the proxy set for each point in the plane has small size (see Lemma 7 below); then we prove that, with high probability, the proxy set of x contains the candidate set of x for all points x in the plane simultaneously (see Lemma 9 below).

3.1 Definitions

As before, the input is a set of sites S . For each site $s \in S$, we randomly pick a parametric point $\alpha \in [0, 1]^d$ according to the sampling method described in Section 2.2.

Volume ordering. Given a point $p = (p_1, \dots, p_d)$ in $[0, 1]^d$, the *point volume* $pv(p)$ of point p is defined to be $p_1 p_2 \cdots p_d$; that is, the volume of the hyperrectangle with p and the origin as a pair of opposite corners. When p is specifically the associated parametric point of an input site s , we refer to the point volume of p as the *parametric volume* of s . Observe that if point p dominates another point q then p must have smaller point volume (i.e., p lies in the hyperrectangle defined by q).

The *volume ordering* of sites in S is a permutation $\langle s_1, \dots, s_n \rangle$ ordered by increasing parametric volume of the sites; that is, $pv(\alpha_1) \leq pv(\alpha_2) \leq \dots \leq pv(\alpha_n)$, where α_i is the parametric point of s_i . If α_i dominates α_j then s_i precedes s_j in the volume ordering. So if we add the sites in volume ordering, then when we add the i th site s_i we can ignore all later

sites when determining its region of influence – that is, the region of points whose candidate set s_i belongs to – as no later sites can have their parametric point dominate the one of s_i .

k nearest neighbors. For a set of sites S and a point x in the plane, let $d_k(x, S)$ denote the k th nearest neighbor distance to x in S ; that is, the k th smallest value in the multiset $\{\|x - s\| \mid s \in S\}$. The k nearest neighbors to x in S is the set $N_k(x, S) = \{s \in S \mid \|x - s\| \leq d_k(x, S)\}$.

► **Definition 6.** Let S be a set of sites in the plane, and let $V(S) = \langle s_1, \dots, s_n \rangle$ be the volume ordering of S . Let S_i denote the underlying set of the i th prefix $\langle s_1, \dots, s_i \rangle$ of $V(S)$. For a parameter k and a point x in the plane, the k th proxy set of x is the set of sites $C_k(x, S) = \bigcup_{i=1}^n N_k(x, S_i)$. In words, site s is in $C_k(x, S)$ if it is one of the k nearest neighbors to point x in some prefix of the volume ordering $V(S)$.

3.2 Bounding the size of the proxy set

The desired bound now follows by using backward analysis and Corollary 5.

► **Lemma 7.** Let S be a set of n sites in the plane, and let $k \geq 1$ be a fixed parameter. Then we have $|C_k(x, S)| = O_{whp}(k \log n)$ simultaneously for all points x in the plane.

Proof. Fix a point x in the plane. A site s gets added to the proxy set $C_k(x, S)$ if site s is one of the k nearest neighbors of x among the underlying set S_i of some prefix of the volume ordering of S . Therefore a direct application of Corollary 5 implies (by setting $\mathcal{P}(S_i)$ to be $N_k(x, S_i)$), with high probability, that $|C_k(x, S)| = O(k \log n)$.

Furthermore, this holds for all points in the plane simultaneously. Indeed, consider the arrangement determined by the $\binom{n}{2}$ bisectors formed by all the pairs of sites in S . This arrangement is a simple planar map with $O(n^4)$ vertices and $O(n^4)$ faces. Observe that within each face the proxy set cannot change since all points in this face have the same ordering of their distances to the sites in S . Therefore, picking a representative point from each of these $O(n^4)$ faces, applying the high probability bound to each one of them, and then applying the union bound implies the claim. ◀

3.3 The proxy set contains the candidate set

The following corollary is implied by a careful (but straightforward) integration argument (see full version [11]).

► **Corollary 8.** Let $F_d(\Delta)$ be the total measure of the points $p \in [0, 1]^d$, such that the point volume $\text{pv}(p)$ is at most Δ . Then for $\Delta \geq (\log n)/n$ we have $F_d(\Delta) = \Theta(\Delta \log^{d-1} n)$; in particular, $F_d(\log n/n) = \Theta((\log^d n)/n)$.

► **Lemma 9.** Let S be a set of n sites in the plane, and let $k = \Theta(\log^d n)$ be a fixed parameter. For all points x in the plane, we have that $L(x) \subseteq C_k(x, S)$, and this holds with high probability.

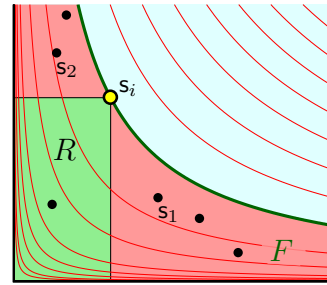
Proof. Fix a point x in the plane, and let s_i be any site *not* in $C_k(x, S)$, and let α_i be the associated parametric point. We claim that, with high probability, the site s_i is dominated by some other site which is closer to x , and hence by the definition of dominating preference (Definition 3), s_i cannot be a site used by x (and thus $s_i \notin L(x)$). Taking the union bound over all sites not in $C_k(x, S)$ then implies this claim.

By Corollary 8, the total measure of the points in $[0, 1]^d$ with point volume at most $\Delta = \log n/n$ is $\Theta((\log^d n)/n)$. As such, by Chernoff's inequality, with high probability, there

are $K = O(\log^d n)$ sites in S such that their parametric points have point volume smaller than Δ . In particular, by choosing k to be sufficiently large (i.e., $k > K$), the underlying set S_k of the k th prefix of the volume ordering of S will contain all these small point volume sites, and since $S_k \subseteq C_k(x, S)$, so will $C_k(x, S)$. Therefore, from this point on, we will assume that $s_i \notin C_k(x, S)$ and $\Delta_i = \text{pv}(\alpha_i) = \Omega(\log n/n)$.

Now any site s with smaller parametric volume than s_i is in the (unordered) prefix S_i . In particular, the k nearest neighbors $N_k(x, S_i)$ of x in S_i all have smaller parametric volume than s_i . Hence $C_k(x, S)$ contains k points all of which have smaller parametric volume than s_i , and which are closer to x . Therefore, the claim will be implied if one of these k points dominates s_i .

The probability of a site s (that is closer to x than s_i) with parametric point α to dominate s_i is the probability that $\alpha \preceq \alpha_i$ given that $\alpha \in F$, where $F = \{\alpha \in [0, 1]^d \mid \text{pv}(\alpha) \leq \Delta_i\}$. By Corollary 8, we have $\text{vol}(F) = F_d(\Delta_i) = \Theta(\Delta_i \log^{d-1} n)$. The probability that a random parametric point in $[0, 1]^d$ dominates α_i is exactly $\Delta_i = \text{pv}(\alpha_i)$, and as such the desired probability $\Pr[\alpha \preceq \alpha_i \mid \alpha \in F]$ is equal to $\Delta_i / F_d(\Delta_i)$, which is $O(1/\log^{d-1} n)$. This is depicted in the figure on the right – the probability of a random point picked uniformly from the region F under the curve $y = \Delta_i/x$, induced by s_i , to fall in the rectangle R .



As the parametric point of each one of the k points in $N_k(x, S_i)$ has equal probability to be anywhere in F , this implies the expected number of points in $N_k(x, S_i)$ which dominate s_i is $\Pr[\alpha \preceq \alpha_i \mid \alpha \in F] \cdot k = \Theta(\log n)$. Therefore by making k sufficiently large, Chernoff's inequality implies the desired result.

It follows that this holds, for all points in the plane simultaneously, by following the argument used in the proof of Lemma 7. ◀

4 Bounding the complexity of the k th order proxy diagram

The k th proxy cell of x is the set of all the points in the plane that have the same k th proxy set associated with them; that is, $\{p \in \mathbb{R}^2 \mid C_k(p, S) = C_k(x, S)\}$. The decomposition of the plane into these faces is the k th order proxy diagram. In this section, our goal is to prove that the expected total diagram complexity of the k th order proxy diagram is $O(k^4 n \log n)$. To this end, we bound the complexity by relating it to the overlay of star-polygons that rise out of the k th order Voronoi diagram.

4.1 Preliminaries

4.1.1 The k th order Voronoi diagram

Let S be a set of n sites in the plane. The k th order Voronoi diagram of S is a partition of the plane into faces such that each cell is the locus of points which have the same set of k nearest sites of S (the internal ordering of these k sites, by distance to the query point, may vary within the cell). It is well known that the worst case complexity of this diagram is $\Theta(k(n - k))$ (see [4, Section 6.5]).

Environments and overlays. For a site s in S and a constant k , the k environment of s , denoted by $\text{env}_k(s, S)$, is the set of all the points in the plane such that s is one of their k nearest neighbors in S ; that is, $\text{env}_k(s, S) = \{x \in \mathbb{R}^2 \mid s \in N_k(x, S)\}$.

See Figure 1 for an example how this environment looks like for different values of k . One can view the k environment of s as the union of the k th order Voronoi cells which have s as one of the k nearest sites. Observe that the overlay of the polygons $\text{env}_k(s_1, S), \dots, \text{env}_k(s_n, S)$ produces the k th order Voronoi diagram of S . Indeed, for any point x in the plane, if s is one of x 's k nearest sites, then by definition x is covered by $\text{env}_k(s, S)$; and conversely if x is covered by $\text{env}_k(s, S)$ then s is one of x 's k nearest neighbors. It is also known that each k environment of a site is a star-shaped polygon; this was previously observed by Aurenhammer and Schwarzkopf [5].

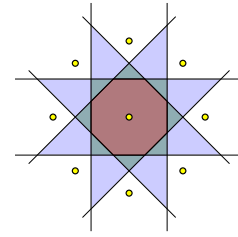


Figure 1

Going back to our original problem, let k be a fixed constant, and let $V(S) = \langle s_1, \dots, s_n \rangle$ be the volume ordering of S . As usual, we use S_i to denote the unordered i th prefix of $V(S)$. Let $\text{env}_i := \text{env}_k(s_i, S_i)$, that is, the union of all the cells in the k th order Voronoi diagram of S_i where s_i is one of the k nearest neighbors.

► **Observation 10.** *The arrangement determined by the overlay of the polygons $\text{env}_1, \dots, \text{env}_n$ is the k th order proxy diagram of S .*

4.1.2 Arrangements of planes and lines

One can interpret the k th order Voronoi diagram in terms of an arrangement of planes in \mathbb{R}^3 . Specifically, “lift” each site to the paraboloid $(x, y, -(x^2 + y^2))$. Consider the arrangement of planes H tangent to the paraboloid at the lifted locations of the sites. A point on the union of these planes is of *level k* if there are exactly k planes strictly below it. The *k -level* is the closure of the set of points of level k .³ (For any set of n hyperplanes in \mathbb{R}^d , one can define k -levels of arrangement of hyperplanes analogously.) Consider a point x in the xy -plane. The decreasing z -ordering of the planes vertically below x is the same as the ordering, by decreasing distance from x , to the corresponding sites. Hence, let $E_k(H)$ denote the set of edges in the arrangement H on the k -level, where an edge is a maximal portion of the k -level that lies on the intersection of two planes (induced by two sites). Then the projection of the edges in $E_{k-1}(H)$ onto the xy -plane results in the edges of the k th order Voronoi diagram. When there is no risk of confusion, we also use $E_k(S)$ to denote the set of edges in $E_k(H)$, where H is obtained by lifting the sites in S to the paraboloid and taking the tangential planes, as described above.

We need the notion of k -levels of arrangement of lines as well. For a set of lines L in the plane, let $E_k(L)$ denote the set of edges in the arrangement of L on the k -level.

► **Lemma 11** (For proof see [11]). *Let L be a set of n lines in general position in the plane. Fix any arbitrary insertion ordering of the lines in L , and let m be the total number of distinct vertices on the k -level of the arrangement of L seen over all iterations of this insertion process. We have $m = O(nk)$.*

³ The lifting of the sites to the paraboloid $z = -(x^2 + y^2)$ is done so that the definition of the k -level coincide with the standard definition.

4.2 Bounding the size of the below conflict-lists

4.2.1 The below conflict lists

Let H be a set of n planes in general position in \mathbb{R}^3 . (For example, in the setting of the k th order Voronoi diagram, H is the set of planes that are tangent to the paraboloid at the lifted locations of the sites.) For any subset $R \subseteq H$, let $V_k(R)$ denote the vertices on the k -level of the arrangement of R . Similarly, let $V_{\leq k}(R) = \bigcup_{i=0}^k V_i(R)$ be the set of vertices of level at most k in the arrangement of R , and let $E_{\leq k}(R)$ be the set of edges of level at most k in the arrangement of R . For a vertex v in the arrangement of R , the *below conflict list* $B(v)$ of v is the set of those planes in H that lie strictly below v ; denote b_v to be $|B(v)|$. For an edge e in the arrangement of R , the *below conflict list* $B(e)$ of e is the set of planes of H which lie below e (i.e., there is at least one point on e that lies above such a plane); denote b_e to be $|B(e)|$. Our purpose here is to bound the quantities $\mathbf{E}\left[\sum_{v \in V_{\leq k}(R)} b_v\right]$ and $\mathbf{E}\left[\sum_{e \in E_{\leq k}(R)} b_e\right]$.

4.2.2 The Clarkson-Shor technique

In the following, we use the Clarkson-Shor technique [14], stated here without proof (see [17] for details). Specifically, let S be a set of elements such that any subset $R \subseteq S$ defines a corresponding set of objects $\mathcal{T}(R)$ (e.g., S is a set of planes and any subset $R \subseteq S$ induces a set of vertices in the arrangement of planes R). Each potential object, τ , has a defining set and a stopping set. The *defining set*, $D(\tau)$, is a subset of S that must appear in R in order for the object to be present in $\mathcal{T}(R)$. We require that the defining set has at most a constant size for every object. The *stopping set*, $\kappa(\tau)$, is a subset of S such that if any of its member appear in R then τ is not present in $\mathcal{T}(R)$. We also naturally require that $\kappa(\tau) \cap D(\tau) = \emptyset$ for all object τ . Surprisingly, this already implies the following.

► **Theorem 12** (Bounded Moments [14]). *Using the above notation, let S be a set of n elements, and let R be a random sample of size r from S . Let $f(\cdot)$ be a polynomially bounded function⁴. We have that $\mathbf{E}\left[\sum_{\tau \in \mathcal{T}(R)} f(|\kappa(\tau)|)\right] = O\left(\mathbf{E}[|\mathcal{T}(R)|] f(n/r)\right)$, where the expectation is taken over random sample R .*

4.2.3 Bounding the below conflict-lists

The technical challenge. The proof of the next lemma is technically interesting as it does not follow in a straightforward fashion from the Clarkson-Shor technique. Indeed, the below conflict list is *not* the standard conflict list. Specifically, the decision whether a vertex v in the arrangement of R is of level at most k is a “global” decision of R , and as such the defining set of this vertex is neither of constant size, nor unique, as required to use the Clarkson-Shor technique. If this was the only issue, the extension by Agarwal *et al.* [3] could handle this situation. However it is even worse: a plane $h \in H \setminus R$ that is below a vertex $v \in V_{\leq k}(R)$ is not necessarily conflicting with v (i.e., in the stopping set of v) – as its addition to R will not necessarily remove v from $V_{\leq k}(R \cup \{h\})$.

The solution. Since the standard technique fails in this case, we need to perform our argument somehow indirectly. Specifically, we use a second random sample and then deploy the Clarkson-Shor technique on this smaller sample – this is reminiscent of the proof bounding

⁴ A function f is *polynomially bounded*, if (i) f is monotonically increasing, and (ii) $f(n) = n^{O(1)}$.

the size of $V_{\leq k}(H)$ by Clarkson-Shor [14], and the proof of the exponential decay lemma of Chazelle and Friedman [12].

► **Lemma 13.** *Let k be a fixed constant, and let R be a random sample (without replacement) of size r from a set of H of n planes in \mathbb{R}^3 , we have $\mathbf{E}\left[\sum_{v \in V_{\leq k}(R)} b_v\right] = O(nk^3)$.*

Proof. From the sake of simplicity of exposition, let us assume that the sampling here is done by picking every element into the random sample R with probability r/n . Doing the computations below using sampling without replacement (so we get the exact size) requires modifying the calculations so that the probabilities are stated using binomial coefficients – this makes the calculation messier, but the results remain the same. See [25] for further discussion of this minor issue.

So, fix a sample R and sample each plane in R , with probability $1/k$, to be in R' . Let us consider the probability that a vertex $v \in V_{\leq k}(R)$ ends up on the lower envelope of R' . A lower bound can be achieved by the standard argument of Clarkson-Shor. Specifically, if a vertex v is on the lower envelope then its three defining planes must be in R' and moreover as $v \in V_{\leq k}(R)$ by definition there are at most k planes below it that must not be in R' . So let X_v be an indicator variable for whether v appears on the lower envelope of R' , we then have

$$\mathbf{E}_{R'}[X_v \mid R] \geq \frac{1}{k^3}(1 - 1/k)^k \geq \frac{1}{e^2 k^3}.$$

Observe that

$$\mathbf{E}_{R'}\left[\sum_{v \in V_0(R')} b_v\right] = \mathbf{E}_R\left[\mathbf{E}_{R'}\left[\sum_{v \in V_0(R')} b_v \mid R\right]\right] = \mathbf{E}_R\left[\mathbf{E}_{R'}\left[\sum_{v \in V_{\leq k}(R)} X_v b_v \mid R\right]\right]. \quad (2)$$

Fixing the value of R , the lower bound above implies

$$\begin{aligned} \mathbf{E}_{R'}\left[\sum_{v \in V_{\leq k}(R)} X_v b_v \mid R\right] &= \sum_{v \in V_{\leq k}(R)} \mathbf{E}_{R'}[X_v b_v \mid R] = \sum_{v \in V_{\leq k}(R)} b_v \mathbf{E}_{R'}[X_v \mid R] \\ &\geq \sum_{v \in V_{\leq k}(R)} \frac{b_v}{e^2 k^3}, \end{aligned}$$

by linearity of expectations and as b_v is a constant for v . Plugging this into Eq. (2), we have

$$\mu = \mathbf{E}_{R'}\left[\sum_{v \in V_0(R')} b_v\right] \geq \mathbf{E}_R\left[\sum_{v \in V_{\leq k}(R)} \frac{b_v}{e^2 k^3}\right] = \frac{1}{e^2 k^3} \mathbf{E}_R\left[\sum_{v \in V_{\leq k}(R)} b_v\right]. \quad (3)$$

Observe that R' is a random sample of R which by itself is a random sample of H . As such, one can interpret R' as a direct random sample of H . The lower envelope of a set of planes has linear complexity, and for a vertex v on the lower envelope of R' the set $B(v)$ is the standard conflict list of v . As such, Theorem 12 implies

$$\mu = \mathbf{E}_{R'}\left[\sum_{v \in V_0(R')} b_v\right] = O\left(|R'| \cdot \frac{n}{|R'|}\right) = O(n).$$

Plugging this into Eq. 3) implies the claim. ◀

► **Corollary 14.** *Let R be a random sample (without replacement) of size r from a set H of n planes in \mathbb{R}^3 . We have that $\mathbf{E}_R\left[\sum_{e \in E_{\leq k}(R)} b_e\right] = O(nk^3)$.*

4.3 Putting it all together

The proof of the following lemma is similar in spirit to the argument of Har-Peled and Raichel [18].

► **Lemma 15.** *Let S be a set of n sites in the plane, $\langle s_1, \dots, s_n \rangle$ be a random permutation of S , and let k be a fixed number. The expected complexity of arrangement determined by the overlay of the polygons $\text{env}_1, \dots, \text{env}_n$ (and therefore, the expected complexity of the k th order proxy diagram) is $O(k^4 n \log n)$, where $\text{env}_i = \text{env}_k(s_i, S_i)$ and $S_i = \{s_1, \dots, s_i\}$ is the underlying set of the i th prefix of $\langle s_1, \dots, s_n \rangle$, for each i .*

Proof. As the arrangement of the overlay of the polygons $\text{env}_1, \dots, \text{env}_n$ is a planar map it suffices to bound the number of edges in the arrangement. For each i , let $E(\text{env}_i)$ be the edges in $E_{\leq k}(S_i)$ that appear on the boundary of env_i (for simplicity we do not distinguish between edges in $E_{\leq k}(S_i)$ in \mathbb{R}^3 and their projection in the plane). Created in the i th iteration, an edge e in $E(\text{env}_i)$ is going to be broken into several pieces in the final arrangement of the overlay. Let n_e be the number of such pieces that arise from e .

Fix an integer i . As S_i is fixed, $B(e)$ is also fixed, for all $e \in E_{\leq k}(S_i)$. Moreover, we claim that $n_e \leq c \cdot kb_e$ for some constant c . Indeed, n_e counts the number of future intersections of e with the edges of $E(\text{env}_j)$, for any $j > i$. As the edge e is on the k -level at the time of creation, and the edges in $E(\text{env}_j)$ are on the k -level when they are being created (in the future), these edges must lie below e . Namely, any future intersect on e are caused by intersections of (pairs of) planes in $B(e)$. So consider the intersection of all planes in $B(e)$ on the vertical plane containing e . On this vertical plane, $B(e)$ is a set of b_e lines, whose insertion ordering is defined by the suffix of the permutation $\langle s_{i+1}, \dots, s_n \rangle$. Now any edge of $E(\text{env}_j)$, for some $j > i$, that intersects e must appear as a vertex on the k -level at some point during the insertion of these lines. However, by Lemma 11, applied to the lines of $B(e)$ on the vertical plane of e , under any insertion ordering there are at most $O(kb_e)$ vertices that ever appear on the k -level.

For an edge $e \in E_{\leq k}(S_i)$, let X_e be the indicator variable of the event that e was created in the i th iteration, and furthermore, lies on the boundary of env_i . Observe that $\mathbf{E}[X_e \mid S_i] \leq 4/i$, as an edge appears for the first time in round i only if one of its (at most) four defining sites was the i th site inserted.

Let $Y_i = \sum_{e \in E(\text{env}_i)} n_e = \sum_{e \in E_{\leq k}(S_i)} n_e X_e$ be the total (forward) complexity contribution to the final arrangement of edges added in round i . We thus have

$$\begin{aligned} \mathbf{E}[Y_i \mid S_i] &= \mathbf{E}\left[\sum_{e \in E_{\leq k}(S_i)} n_e X_e \mid S_i\right] \leq \mathbf{E}\left[\sum_{e \in E_{\leq k}(S_i)} ckb_e X_e \mid S_i\right] \\ &= \sum_{e \in E_{\leq k}(S_i)} ckb_e \mathbf{E}[X_e \mid S_i] \leq \frac{4ck}{i} \sum_{e \in E_{\leq k}(S_i)} b_e. \end{aligned}$$

The total complexity of the overlay arrangement of the polygons $\text{env}_1, \dots, \text{env}_n$ is asymptotically bounded by $\sum_i Y_i$, and so by Corollary 14 we have

$$\begin{aligned} \mathbf{E}\left[\sum_i Y_i\right] &= \sum_i \mathbf{E}\left[\mathbf{E}[Y_i \mid S_i]\right] \leq \sum_i \mathbf{E}\left[\frac{4ck}{i} \sum_{e \in E_{\leq k}(S_i)} b_e\right] = O\left(\sum_i \frac{nk^4}{i}\right) \\ &= O(k^4 n \log n). \end{aligned}$$



5 On the expected size of the staircase

Due to space limits, the details of the following result are omitted, and can be found in the full version of the paper [11].

► **Lemma 16.** *Let S be a set of n sites in the plane, where for each site s in S , a parametric point from a distribution over $[0, 1]^d$ is sampled (as described in Section 2.2). Then, the candidate set has size $O_{whp}(\log^d n)$ simultaneously for all points in the plane.*

6 The main result

We now use the bound on the complexity of the proxy diagram, as well as our knowledge of the relationship between the candidate set and the proxy set to bound the complexity of the candidate diagram.

► **Theorem 17.** *Let S be a set of n sites in the plane, where for each site in S we sample an associated parametric point in $[0, 1]^d$, as described in Section 2.2. Then, the expected complexity of the candidate diagram is $O(n \log^{8d+5} n)$. The expected space complexity of this candidate diagram is $O(n \log^{9d+5} n)$.*

Proof. Fix k to be sufficiently large such that $k = \Theta(\log^d n)$. By Lemma 15 the expected complexity of the proxy diagram is $O(k^4 n \log n)$. Triangulating each polygonal cell in the diagram does not increase its asymptotic complexity. Lemma 7 implies that, (simultaneously) for all the points in the plane, the proxy set has size $O(k \log n)$, with high probability. Now, Lemma 9 implies that, with high probability, the proxy set contains the candidate set for any point in the plane.

The resulting triangulation has $O(k^4 n \log n)$ faces, and inside each face all the sites that might appear in the candidate set are all present in the proxy set of this face. By Lemma 4, the complexity of an m -site candidate diagram is $O(m^4)$. Therefore the complexity of the candidate diagram per face is $O((k \log n)^4)$, with high probability (clipping the candidate diagram of these sites to the containing triangle does not increase the asymptotic complexity). Multiplying the number of faces, $O(k^4 n \log n)$, by the complexity of the arrangement within each face, $O((k \log n)^4)$, yields the desired result.

The bound on the space complexity follows readily from the bound on the size of the candidate set from Lemma 16. ◀

Acknowledgments. The authors would like to thank Pankaj Agarwal, Ken Clarkson, Nirman Kumar, and Raimund Seidel for useful discussions related to this work. We are also grateful to the anonymous SoCG reviewers for their helpful comments.

References

- 1 P. K. Agarwal, B. Aronov, S. Har-Peled, J. M. Phillips, K. Yi, and W. Zhang. Nearest neighbor searching under uncertainty II. In *Proc. 32nd ACM Sympos. Principles Database Syst. (PODS)*, pages 115–126, 2013.
- 2 P. K. Agarwal, S. Har-Peled, H. Kaplan, and M. Sharir. Union of random minkowski sums and network vulnerability analysis. *Discrete Comput. Geom.*, 52(3):551–582, 2014.
- 3 P. K. Agarwal, J. Matoušek, and O. Schwarzkopf. Computing many faces in arrangements of lines and segments. *SIAM J. Comput.*, 27(2):491–505, 1998.
- 4 F. Aurenhammer, R. Klein, and D.-T. Lee. *Voronoi Diagrams and Delaunay Triangulations*. World Scientific, 2013.

- 5 F. Aurenhammer and O. Schwarzkopf. A simple on-line randomized incremental algorithm for computing higher order Voronoi diagrams. *Internat. J. Comput. Geom. Appl.*, pages 363–381, 1992.
- 6 Z.-D. Bai, L. Devroye, H.-K. Hwang, and T.-H. Tsai. Maxima in hypercubes. *Random Struct. Alg.*, 27(3):290–309, 2005.
- 7 I. Bárány and M. Reitzner. On the variance of random polytopes. *Adv. Math.*, 225(4):1986–2001, 2010.
- 8 I. Bárány and M. Reitzner. Poisson polytopes. *Annals. Prob.*, 38(4):1507–1531, 2010.
- 9 M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.
- 10 S. Börzsönyi, D. Kossmann, and K. Stocker. The skyline operator. In *Proc. 17th IEEE Int. Conf. Data Eng.*, pages 421–430, 2001.
- 11 H.-C. Chang, S. Har-Peled, and B. Raichel. From proximity to utility: A Voronoi partition of Pareto optima. *CoRR*, abs/1404.3403, 2014.
- 12 B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10(3):229–249, 1990.
- 13 K. L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3(4):185–212, 1993.
- 14 K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.
- 15 A. Feldman. Welfare economics. In S. Durlauf and L. Blume, editors, *The New Palgrave Dictionary of Economics*. Palgrave Macmillan, 2008.
- 16 P. Godfrey, R. Shipley, and J. Gryz. Algorithms and analyses for maximal vector computation. *VLDB J.*, 16(1):5–28, 2007.
- 17 S. Har-Peled. *Geometric Approximation Algorithms*, volume 173 of *Mathematical Surveys and Monographs*. Amer. Math. Soc., 2011.
- 18 S. Har-Peled and B. Raichel. On the expected complexity of randomly weighted Voronoi diagrams. In *Proc. 30th Annu. Sympos. Comput. Geom. (SoCG)*, pages 232–241, 2014.
- 19 D. Haussler and E. Welzl. ϵ -nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.
- 20 H.-K. Hwang, T.-H. Tsai, and W.-M. Chen. Threshold phenomena in k -dominant skylines of random samples. *SIAM J. Comput.*, 42(2):405–441, 2013.
- 21 H. Kung, F. Luccio, and F. Preparata. On finding the maxima of a set of vectors. *J. Assoc. Comput. Mach.*, 22(4):469–476, 1975.
- 22 T. Ottmann, E. Soisalon-Soininen, and D. Wood. On the definition and computation of rectilinear convex hulls. *Inf. Sci.*, 33(3):157–171, 1984.
- 23 R. Schneider and J. A. Wieacker. Integral geometry. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, volume B, chapter 5.1, pages 1349–1390. North-Holland, 1993.
- 24 R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 37–68. Springer-Verlag, 1993.
- 25 M. Sharir. The Clarkson-Shor technique revisited and extended. *Comb., Prob. & Comput.*, 12(2):191–201, 2003.
- 26 M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York, 1995.
- 27 W. Weil and J. A. Wieacker. Stochastic geometry. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, volume B, chapter 5.2, pages 1393–1438. North-Holland, 1993.

Faster Deterministic Volume Estimation in the Oracle Model via Thin Lattice Coverings

Daniel Dadush

Centrum Wiskunde & Informatica, The Netherlands
dadush@cwi.nl

Abstract

We give a $2^{O(n)}(1+1/\varepsilon)^n$ time and $\text{poly}(n)$ -space deterministic algorithm for computing a $(1+\varepsilon)^n$ approximation to the volume of a general convex body K , which comes close to matching the $(1+c/\varepsilon)^{n/2}$ lower bound for volume estimation in the oracle model by Bárány and Füredi (STOC 1986, Proc. Amer. Math. Soc. 1988). This improves on the previous results of Dadush and Vempala (Proc. Nat'l Acad. Sci. 2013), which gave the above result only for *symmetric bodies* and achieved a dependence of $2^{O(n)}(1+\log^{5/2}(1/\varepsilon)/\varepsilon^3)^n$.

For our methods, we reduce the problem of volume estimation in K to counting lattice points in $K \subseteq \mathbb{R}^n$ (via enumeration) for a specially constructed lattice \mathcal{L} : a so-called *thin covering of space* with respect to K (more precisely, for which $\mathcal{L} + K = \mathbb{R}^n$ and $\text{vol}_n(K)/\det(\mathcal{L}) = 2^{O(n)}$). The trade off between time and approximation ratio is achieved by scaling down the lattice.

As our main technical contribution, we give the first deterministic $2^{O(n)}$ -time and $\text{poly}(n)$ -space construction of thin covering lattices for general convex bodies. This improves on a recent construction of Alon et al. (STOC 2013) which requires exponential space and only works for symmetric bodies. For our construction, we combine the use of the M-ellipsoid from convex geometry (Milman, C.R. Math. Acad. Sci. Paris 1986) together with lattice sparsification and densification techniques (Dadush and Kun, SODA 2013; Rogers, J. London Math. Soc. 1950).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Deterministic Volume Estimation, Convex Geometry, Lattice Coverings of Space, Lattice Point Enumeration

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.704

1 Introduction

The problem of estimating the volume of a convex body is one of the most fundamental and well studied problems in high dimensional geometry. It is also one of the most striking examples of the *power of randomization*. In [11, 12], Bárány and Füredi showed that any deterministic volume algorithm for n -dimensional convex bodies having access only to a membership oracle (which returns whether a point is in the convex body or not), requires at least $(1+c/\varepsilon)^{n/2}$ membership queries to estimate volume to within a $(1+\varepsilon)^n$ factor, for $c > 0$ an absolute constant any ε small enough. In particular, an $O(1)$ -approximation requires $n^{\Omega(n)}$ queries. In a breakthrough result however, Dyer, Frieze and Kannan [9] showed that if the algorithm is allowed to err with small probability, then even a $(1+\varepsilon)$ approximation can be obtained in $\text{poly}(n, 1/\varepsilon)$ -time. Their algorithm relied on novel Monte Carlo Markov Chain techniques that spurred much further research. These works left a major open question: can the volume algorithm be made deterministic when the description of the convex body is given explicitly (e.g. a polytope given by its inequalities)?

A related (and more modest) question, which has only recently received attention, is whether one can come close to matching the lower bounds of Bárány and Füredi for



© Daniel Dadush;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 704–718



Leibniz International Proceedings in Informatics

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

deterministic volume computation in the oracle model. We note it was open to achieve such bounds deterministically even for explicitly presented polytopes. This was recently answered in the affirmative by Vempala and the author in [8], which gave a deterministic $2^{O(n)}(1 + \log^{5/2}(1/\varepsilon)/\varepsilon^3)^n$ -time and polynomial space algorithm for estimating the volume of a *symmetric* convex body K (K is symmetric if $K = -K$) to within $(1 + \varepsilon)^n$. The main tool developed there was an algorithmic version of (variants of) Milman's construction for the *M-ellipsoid* in convex geometry [18]. An *M-ellipsoid* of an n -dimensional convex body K is an ellipsoid E (an ellipsoid is a linear transformation of the Euclidean ball) satisfying that $2^{O(n)}$ translates of E suffice to cover K and vice versa. Note that the volume of an *M-ellipsoid* of K immediately provides a $2^{O(n)}$ factor approximation to the volume of K .

From the above, two natural avenues of improvement were to reduce the dependence on ε and to generalize the result to asymmetric convex bodies.

2 Main Contribution

We make improvements on both of the last two fronts. Our main result is stated below.

► **Theorem 1 (Volume Estimation).** *For a convex body $K \subseteq \mathbb{R}^n$ given by a membership oracle, and any $\varepsilon > 0$, one can compute $V \geq 0$ satisfying $\text{vol}_n(K) \leq V \leq (1 + \varepsilon)^n \text{vol}_n(K)$ in deterministic $2^{O(n)}(1 + 1/\varepsilon)^n$ -time and $\text{poly}(n)$ -space.*

Both the algorithm and that of [8] share the same high level approach, namely, reducing volume estimation to counting lattice points within a carefully chosen convex body and lattice.

We note that if we are satisfied with a c^n approximation of volume for some large enough $c > 0$, then the volume of an *M-ellipsoid* is already a good enough volume approximation for K and hence lattice point counting is not needed. This extends to asymmetric convex bodies as well, by replacing K with the symmetric body $K - K$ (an oracle for which can be efficiently computed, see [13]) and using the standard inequalities

$$2^n \text{vol}_n(K) \leq \text{vol}_n(K - K) \leq \binom{2n}{n} \text{vol}_n(K) \quad (\text{see [23]}).$$

Hence the above result is truly interesting for the case of small constant ε .

Our runtime improvement over the algorithm of [8] comes from a much more efficient reduction from volume estimation to lattice point counting. In particular, the crucial ingredient in our improved reduction is the use of so-called *thin lattice coverings of space* with respect to K (and related convex bodies). The heart of our volume algorithm, and our main technical contribution, is a deterministic construction of thin-covering lattices for general convex bodies with *good enumeration properties*, that is, where lattice point enumeration can be performed efficiently using only polynomial space. This improves on a recent thin-lattice construction of [1] which requires exponential space and only works for symmetric bodies.

Organization. The remainder of this paper is organized as follows. First, we shall explain the reduction between volume estimation and lattice point counting, which will motivate the need for thin covering lattices and other related concepts. Second, we will present the polynomial space lattice point enumeration technique we use – Schorr-Euchner enumeration – and briefly discuss its implementation and associated challenges. Third, we give the formal statements of our main thin lattice construction and related algorithms, and their relations to prior work. Finally, in the remainder, we shall detail the main ideas behind the thin covering lattice construction.

3 Preliminaries

Basic concepts. For two sets $A, B \subseteq \mathbb{R}^n$, we define their Minkowski sum $A + B = \{\mathbf{a} + \mathbf{b} : \mathbf{a} \in A, \mathbf{b} \in B\}$. For vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, we write $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_{i=1}^n x_i y_i$ to denote the standard inner product and $\|\mathbf{x}\|_2 = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle}$ for the Euclidean norm. We let $B_2^n = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\|_2 \leq 1\}$ denote the unit Euclidean ball in \mathbb{R}^n . For a set $A \subseteq \mathbb{R}^n$, we denote its interior by A° . A convex body $K \subseteq \mathbb{R}^n$ is a compact convex set with non-empty interior. A function $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is L -Lipshitz if $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n, |f(\mathbf{x}) - f(\mathbf{y})| \leq L\|\mathbf{x} - \mathbf{y}\|_2$.

Lattices. We give some basic definitions of lattice concepts.

► **Definition 2 (Lattices and Bases).** A full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ is defined as all integer combinations of some basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n) \in \mathbb{R}^{n \times n}$. In particular, $\mathcal{L} = B\mathbb{Z}^n$. The determinant of \mathcal{L} is defined as $\det(\mathcal{L}) = |\det(B)|$, which is invariant to the choice of lattice basis. We define $\mathbf{b}_1^*, \dots, \mathbf{b}_n^*$, the associated dual basis, to be the unique vectors satisfying $\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle = 1$ if $i = j$ and 0 otherwise (corresponding to the columns of B^{-T}).

► **Definition 3 (Gram Schmid Projections).** For a basis $\mathbf{b}_1, \dots, \mathbf{b}_n \in \mathbb{R}^n$, we define the i^{th} Gram-Schmidt projection $\pi_i, i \in [n+1]$, to be the orthogonal projection onto the orthogonal complement of the linear span of $\mathbf{b}_1, \dots, \mathbf{b}_{i-1}$. Note that π_1 is the identity on \mathbb{R}^n and π_{n+1} is the identically 0 map.

► **Definition 4 (Basis Parallelepiped).** For a full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ with basis B , we define $\mathcal{P}(B) = B[-1/2, 1/2)^n$ to be the half-open symmetric parallelepiped. Note that $\text{vol}_n(\mathcal{P}(B)) = \det(\mathcal{L})$.

► **Definition 5 (Sublattice Index).** For a full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ and full rank sublattice $\mathcal{L}' \subseteq \mathcal{L}$, we define the index of \mathcal{L}' in \mathcal{L} , denoted $[\mathcal{L} : \mathcal{L}']$, as $|\{\mathbf{y} + \mathcal{L}' : \mathbf{y} \in \mathcal{L}\}| < \infty$ (i.e. number of shifts of \mathcal{L}' in \mathcal{L}). Here, we have the fundamental identity $[\mathcal{L} : \mathcal{L}'] = \det(\mathcal{L}') / \det(\mathcal{L})$.

► **Definition 6 (Lattice Tiling).** A measurable set $A \subseteq \mathbb{R}^n$ tiles with respect to a full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ (and vice versa) if for every $\mathbf{x} \in \mathbb{R}^n$ there is a *unique* $\mathbf{y} \in \mathcal{L}$ such that $\mathbf{x} \in \mathbf{y} + A$. Here, A is said to be a *fundamental domain* of \mathcal{L} .

A basic fact is that every fundamental domain of \mathcal{L} has the same volume. In particular, since $\mathcal{P}(B)$ is a fundamental domain, every fundamental domain of \mathcal{L} has volume $\det(\mathcal{L})$.

Computational model. For a convex body $K \subseteq \mathbb{R}^n$, a membership oracle O_K for K takes as input $\mathbf{x} \in \mathbb{R}^n$ and returns 1 if $\mathbf{x} \in K$ and 0 otherwise. K is (\mathbf{a}_0, r, R) -centered, for $r, R > 0$ and $\mathbf{a}_0 \in \mathbb{R}^n$, if $rB_2^n \subseteq K - \mathbf{a}_0 \subseteq RB_2^n$. When we refer to K being centered, we shall mean that the *centering guarantees* (\mathbf{a}_0, r, R) exist and are implicitly passed to any algorithm operating on K and that the complexity of this algorithm may depend on these guarantees. For $\varepsilon > 0$, we define $K^\varepsilon = K + \varepsilon B_2^n$ and $K^{-\varepsilon} = \{\mathbf{x} \in K : \mathbf{x} + \varepsilon B_2^n \subseteq K\}$. A weak membership oracle O_K for K , takes an additional parameter $\varepsilon > 0$, and only guarantees that $O_K(\mathbf{x}, \varepsilon) = 1$ if $\mathbf{x} \in K^{-\varepsilon}$ and 0 if $\mathbf{x} \notin K^\varepsilon$. All our algorithms will operate on centered convex bodies equipped with (weak) membership oracles, and the complexity of our algorithms will be measured by the number of arithmetic operations and oracle calls they perform.

One of the main algorithmic tools we will use is the following classical result in convex optimization:

► **Theorem 7 (Convex Optimization [25, 13]).** *Let $K \subseteq \mathbb{R}^n$ be a centered convex body given by a weak membership oracle O_K . Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ denote an L -Lipshitz convex function.*

Then, for $\varepsilon > 0$, a vector $\mathbf{y} \in K$ satisfying

$$f(\mathbf{y}) - \varepsilon \leq \min_{\mathbf{x} \in K} f(\mathbf{x}) \leq f(\mathbf{y})$$

can be computed using a polynomial number of arithmetic operations, oracle calls and evaluations of f .

4 From Volume Estimation to Counting Lattice Points

In this section, we will show how to reduce the volume estimation problem to counting lattice points inside a well-chosen convex body. We will primarily concern ourselves with the task of minimizing the number of lattice points we need to enumerate to achieve a desired approximation factor. The important details regarding how to efficiently enumerate these lattice points is left to later sections.

To build intuition, we shall first try to estimate the volume of a convex body $K \subseteq \mathbb{R}^n$ by counting the number of points it contains in the standard integer lattice \mathbb{Z}^n . Through this attempt, we will expose some of the main ingredients needed to make volume approximation efficient.

For the integer lattice, the canonical relation between lattice point counting and volume is simply derived by associating every point in $\mathbf{y} \in \mathbb{Z}^n$ with the half open cube around it, i.e. $C = [-1/2, 1/2)^n + \mathbf{y}$. Since these shifted cubes have volume 1 and are all disjoint, the count $|K \cap \mathbb{Z}^n|$ is the same as the volume of the set $S = (K \cap \mathbb{Z}^n) + C$. Now as is, the set S may both miss parts of and “stick out” of K , so it is difficult to deduce any relationship between their volumes. To fix one of these problems, note that the cubes around each integer point form a *tiling of space*, that is every point in \mathbb{R}^n is in exactly one such cube. Hence if we enlarge S to contain all the cubes centered around \mathbb{Z}^n that *touch* K – formally, we redefine $S = ((K - C) \cap \mathbb{Z}^n) + C$ – then we are guaranteed that S covers K . In particular,

$$|(K - C) \cap \mathbb{Z}^n| = \text{vol}_n(S) \geq \text{vol}_n(K).$$

Note then that the volume of S can be computed if we can enumerate the integer points in $K - C$ (we defer for now the discussion of how to do this efficiently). So now, from the perspective of approximation, we are left with the problem that S may stick out very far from K , and hence may have very large volume compared to K . Indeed, this may easily happen (say if K is a ball of tiny radius), since we have made no assumptions on K .

Regardless, if we scale down \mathbb{Z}^n and $C = [-1/2, 1/2)^n$ by ε , then as we let $\varepsilon \rightarrow 0$, the volume of S (defined on the scaled down lattice and cube) will clearly converge to the volume of K since S will converge to K . Given this, we are lead to two basic questions. Firstly, how small do we need to make ε to get a $(1 + \varepsilon)^n$ approximation of volume? Secondly, how many lattice points do we need to enumerate to compute this approximation? Crucially, the answer to this last question will essentially determine the complexity of the algorithm.

To get a quantitative estimate, let us normalize the geometry by assuming that $\pm C \subseteq K/2$. (while requiring the condition for both C and $-C$ is essentially redundant here, it will be very important when we generalize the forthcoming analysis.) Note that this can always be achieved by an appropriate shift and scaling of K . Letting $S_\varepsilon = ((K - \varepsilon C) \cap \varepsilon \mathbb{Z}^n) + \varepsilon C$, for $\varepsilon > 0$, by the same reasoning as before we have that

$$\text{vol}_n(K) \leq \text{vol}_n(S_\varepsilon) = \text{vol}_n(\varepsilon C) |(K - \varepsilon C) \cap \mathbb{Z}^n| = \varepsilon^n |(K - \varepsilon C) \cap \mathbb{Z}^n|. \quad (1)$$

Furthermore, since $\pm C \subseteq K/2$, we have that

$$\begin{aligned} \text{vol}_n(S_\varepsilon) &= \text{vol}_n(((K - \varepsilon C) \cap \varepsilon \mathbb{Z}^n) + \varepsilon C) \leq \text{vol}_n(K + \varepsilon(C - C)) \\ &\leq \text{vol}_n(K + \varepsilon(K/2 + K/2)) = \text{vol}_n((1 + \varepsilon)K) = (1 + \varepsilon)^n \text{vol}_n(K), \end{aligned} \tag{2}$$

where the last two equalities hold by convexity of K and the homogeneity of volume respectively. Hence, from the above computing a $(1 + \varepsilon)^n$ approximation to $\text{vol}_n(K)$ reduces to enumerating the points in $(K - \varepsilon C) \cap \varepsilon \mathbb{Z}^n$. Combining (1),(2) and rearranging, we see that the number of points we must enumerate is bounded by

$$|(K - \varepsilon C) \cap \mathbb{Z}^n| \leq (1 + 1/\varepsilon)^n \text{vol}_n(K) = 2^n (1 + 1/\varepsilon)^n (\text{vol}_n(K/2)/\text{vol}_n(C)).$$

Now, if we believe that the correct measure of complexity is simply the number of lattice points we must enumerate (ignoring the actual complexity of enumeration for now), then we would achieve the complexity estimate in Theorem 1 if $\text{vol}_n(K/2)/\text{vol}_n(C) = 2^{O(n)}$. However, it is clear that not every convex body K can be scaled and shifted such that $\pm C \subseteq K/2$ and $\text{vol}_n(K/2)/\text{vol}_n(C) = 2^{O(n)}$.

On the other hand, it is easy to see that the above analysis can be substantially generalized. More precisely, instead of relying on the integer lattice, we may use an arbitrary lattice $\mathcal{L} = B\mathbb{Z}^n$, for some basis B . Instead of cubes (or parallelepipeds), we may use any measurable set $F \subseteq \mathbb{R}^n$ which tiles with respect to \mathcal{L} . From here, if there exists $\mathbf{c} \in K$, such that $\pm F \subseteq (K - \mathbf{c})/2$ (note that F need no longer be symmetric), then by the same analysis as above we have that

$$\text{vol}_n(K) \leq \varepsilon^n \cdot \text{vol}_n(F) \cdot |(K - \varepsilon F) \cap \varepsilon \mathcal{L}| \leq (1 + \varepsilon)^n \text{vol}_n(K). \tag{3}$$

When trying to use the above formula to approximate volume, one may rightly worry that the set F above maybe quite complicated, and hence of limited algorithmic use. Fortunately, it turns out that we won't actually need to know F at all – we will only need to rely on its *existence* – and, in fact, only knowledge of the point \mathbf{c} will be required. To justify this, we first remark that F is a fundamental domain, and hence $\text{vol}_n(F) = \det(\mathcal{L})$, which is easily computable given B .

Let $K[\mathbf{c}] = (K - \mathbf{c}) \cap (\mathbf{c} - K)$ denote the symmetrization of K about \mathbf{c} (note that $K[\mathbf{c}]$ is indeed symmetric). By construction, we see that

$$\pm F \subseteq \pm K[\mathbf{c}]/2 = K[\mathbf{c}]/2 \subseteq (K - \mathbf{c})/2.$$

From here, it is not hard to check that replacing $K - \varepsilon F$ by $K + \varepsilon K[\mathbf{c}]/2$ in (3) yields

$$\text{vol}_n(K) \leq \varepsilon^n \cdot \det(\mathcal{L}) \cdot |(K + \varepsilon K[\mathbf{c}]/2) \cap \varepsilon \mathcal{L}| \leq (1 + \varepsilon)^n \text{vol}_n(K). \tag{4}$$

The above formula will indeed form the basis of our algorithmic approach, where we note that a membership oracle for $K + \varepsilon K[\mathbf{c}]/2$ (under mild assumptions on \mathbf{c}) can be efficiently constructed from a membership oracle for K (see [13]). Rearranging as before, we get that the number of lattice points we need to enumerate to compute the desired approximation is bounded by

$$|(K + \varepsilon K[\mathbf{c}]/2) \cap \varepsilon \mathcal{L}| \leq 2^n (1 + 1/\varepsilon)^n \underbrace{\frac{\text{vol}_n(K)}{\text{vol}_n(K[\mathbf{c}])}}_{(a)} \underbrace{\frac{\text{vol}_n(K[\mathbf{c}]/2)}{\det(\mathcal{L})}}_{(b)} \tag{5}$$

Hence, to achieve the desired complexity bound, we will need both the expressions (a) and (b) to be bounded by $2^{O(n)}$. More precisely, we will need to compute a point $\mathbf{c} \in K$ and a lattice $\mathcal{L} \subseteq \mathbb{R}^n$ such that

1. $\text{vol}_n(K) \leq 2^{O(n)} \text{vol}_n(K[\mathbf{c}])$.
2. $\exists F \subseteq K[\mathbf{c}]$ a fundamental domain for \mathcal{L} , and $\text{vol}_n(K[\mathbf{c}]) \leq 2^{O(n)} \det(\mathcal{L})$.

We note that condition (1) becomes trivial if K is already symmetric, since we can simply choose $\mathbf{c} = \mathbf{0}$. In condition (2), note that we have, for convenience of notation, multiplied the required conditions for \mathcal{L} in (5) by 2.

We now relate some initial details of how to find \mathbf{c} and \mathcal{L} satisfying these conditions, deferring the full discussion of our methods to later sections. The plan here is to treat each condition separately. In particular, we will first choose \mathbf{c} to satisfy (1) and then pick \mathcal{L} satisfying (2).

Choosing the lattice \mathcal{L} . Once we have chosen \mathbf{c} , we wish to choose a lattice satisfying condition (2). For this purpose, we will only use the fact that $K[\mathbf{c}]$ is a symmetric convex body (which is why we can treat both conditions separately). As a first remark, we note that the existence of fundamental domain $F \subseteq K[\mathbf{c}]$ is equivalent to asking that $K[\mathbf{c}]$ *cover space* with respect to \mathcal{L} .

► **Definition 8 (Lattice Covering).** A measurable $A \subseteq \mathbb{R}^n$ is covering with respect to a full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ (and vice versa) if $\mathcal{L} + A = \mathbb{R}^n$. The covering induced by A and \mathcal{L} is said to be α -thin, $\alpha \geq 1$, if $\text{vol}_n(A) / \det(\mathcal{L}) \leq \alpha$.

Indeed, assuming that $\mathcal{L} + K[\mathbf{c}] = \mathbb{R}^n$, we can recover a suitable fundamental domain F by picking (in a measurable way) a unique representative of in $(\mathcal{L} + \mathbf{x}) \cap K[\mathbf{c}]$, for each distinct coset $\mathcal{L} + \mathbf{x}$, $\mathbf{x} \in \mathbb{R}^n$. We note that this simply corresponds to throwing away the “overrepresented” parts of $K[\mathbf{c}]$. From this discussion, we see that every covering of space must have thinness at least 1. Note that at a high level, the covering induced by $K[\mathbf{c}]$ and \mathcal{L} being α -thin means that on average points in \mathbb{R}^n are covered by at most α lattice shifts of $K[\mathbf{c}]$ (and clearly at least 1).

We can now restate our goal as that of constructing a lattice \mathcal{L} forming a $2^{O(n)}$ -thin covering with respect to $K[\mathbf{c}]$. We give a detailed accounting of how to build such lattices in section 8.1.

Choosing the center \mathbf{c} . To compute \mathbf{c} , we will require the following measure of symmetry:

► **Definition 9 (Kovner-Besicovitch Symmetry Measure).** For a convex body $K \subseteq \mathbb{R}^n$, we define its *Kovner-Besicovitch* measure of symmetry (see [15]) as

$$\text{Sym}_{kb}(K) = \max_{\mathbf{c} \in K} \text{vol}_n(K[\mathbf{c}]) / \text{vol}_n(K), \quad \text{where } K[\mathbf{c}] = (K - \mathbf{c}) \cap (\mathbf{c} - K). \tag{6}$$

Note that K is symmetric (about some center) iff $\text{Sym}_{kb}(K) = 1$. For $\mathbf{c} \in K$, we define its *KB value* to be $\text{vol}_n(K[\mathbf{c}]) / \text{vol}_n(K)$. Clearly, to satisfy condition (1), the best center we can choose is simply that of maximum KB value. For such a maximizer to be useful, we must at least convince ourselves that best center has KB value at least $2^{-O(n)}$. For this purpose, let X denote a uniform random variable over K . By a classical computation, we have that

$$\begin{aligned} \mathbb{E}_X \left[\frac{\text{vol}_n(K[X])}{\text{vol}_n(K)} \right] &= \int_K \frac{\text{vol}_n(K[\mathbf{x}])}{\text{vol}_n(K)^2} \, d\mathbf{x} = \int_K \int_K \frac{\mathbf{1}[2\mathbf{x} - \mathbf{y} \in K]}{\text{vol}_n(K)^2} \, d\mathbf{y} d\mathbf{x} \\ &= \int_K \frac{\text{vol}_n((K + \mathbf{y})/2)}{\text{vol}_n(K)^2} \, d\mathbf{y} = 2^{-n}. \end{aligned}$$

By the probabilistic method, we therefore have that $\text{Sym}_{kb}(K) \geq 2^{-n}$, which is more than good enough for us. Furthermore, it was actually shown in [19] that the centroid $\mu = \mathbb{E}[X]$ of K has KB value at least 2^{-n} . Hence, with the aid of random sampling techniques over convex bodies [9], computing a point with good KB value is rather straightforward.

Since our goal is to get a deterministic algorithm however, we cannot rely on random sampling methods. Perhaps surprisingly, our approach for computing a high KB value point will be to approximately solve the optimization problem in (6). Indeed, by the Brunn-Minkowski inequality (which states that $\text{vol}(A)^{1/n} + \text{vol}(B)^{1/n} \leq \text{vol}(A + B)^{1/n}$ for $A, B, A + B$ measurable), the function $f(\mathbf{c}) = \text{vol}_n(K[\mathbf{c}])^{1/n}$ is in fact *concave* over K . Hence, maximizing f is a concave optimization problem.

We define a point $\mathbf{c} \in K$ to be an α -approximate KB point for K , $0 < \alpha \leq 1$, if its KB value $\text{vol}_n(K[\mathbf{c}])/\text{vol}_n(K)$ is at least an α -factor of $\text{Sym}_{kb}(K)$. For our purposes, it will suffice to be able to compute a $2^{-O(n)}$ approximate KB point, which we note corresponds to computing a constant factor approximation to $\max_{\mathbf{c} \in K} f(\mathbf{c})$. We will actually be able to compute $(1 + \varepsilon)^{-n}$ -approximation KB points for any desired $\varepsilon > 0$ (see Theorem 22). Our approximation algorithm will be somewhat non-trivial, requiring many calls to our volume algorithm over symmetric bodies (noting that each $K[\mathbf{c}]$ is symmetric). We defer the full discussion to section 8.2.

5 Schnorr-Euchner Enumeration

The currently most powerful polynomial space lattice point enumeration strategy is *Schnorr-Euchner* enumeration. It is the primary enumeration method for all polynomial space solvers for the Closest Vector Problem (CVP) under the Euclidean norm (given a target \mathbf{t} and lattice \mathcal{L} , find the closest vector in \mathcal{L} to \mathbf{t}), and will form the core of our enumeration algorithm. We now explain how to adapt it to enumerate lattice points in general convex bodies (it was originally specified only for Euclidean balls, see for example [16]), and present some of its important properties.

High level algorithm. Given a basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ of \mathcal{L} and a convex body K , Schnorr-Euchner builds all feasible solutions to $\{\mathbf{z} \in \mathbb{Z}^n : \sum_{i=1}^n z_i \mathbf{b}_i \in K\}$, corresponding to $\mathcal{L} \cap K$, using a search tree over the coefficients. The nodes at level i of the tree, $i \in \{0, \dots, n\}$, correspond to integral assignments of the last i coefficients that are “feasible” for K . Precisely, a partial assignment $z_{n-i+1}, \dots, z_n \in \mathbb{Z}$ is feasible for K if $\exists r_1, \dots, r_{n-i} \in \mathbb{R}$ such that

$$\sum_{j=1}^{n-i} r_j \mathbf{b}_j + \sum_{j=n-i+1}^n z_j \mathbf{b}_j \in K. \tag{7}$$

By convention, we consider the root (level 0) to have an empty assignment, which is feasible iff $K \neq \emptyset$. From a level i node, with partial assignment $z_{n-i+1}, \dots, z_n \in \mathbb{Z}$, we recurse on all feasible extensions z_{n-i}, \dots, z_n with $z_{n-i} \in \mathbb{Z}$. By convexity of K , the set of integer assignments for z_{n-i} inducing a feasible extension form a consecutive interval, which will allow us to enumerate them efficiently.

Implementation. Since the nature of computations in the oracle model are always approximate, we will have to relax the notion of feasible partial assignment when implementing the above algorithm. In particular, we will only be able to determine where a partial assignment is either not feasible for K or feasible for K^ε , for any desired error tolerance $\varepsilon > 0$. The

exact guarantees for our enumeration algorithm, which will be sufficient for all intended applications, are stated below.

► **Lemma 10** (Enumeration Complexity). *Let $K \subseteq \mathbb{R}^n$ be a (\mathbf{a}_0, r, R) -centered convex body given a weak membership oracle, and let $\mathcal{L} \subseteq \mathbb{R}^n$ a full rank lattice with basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$. Then for $0 < \varepsilon < 1$, a set S satisfying $K \cap \mathcal{L} \subseteq S \subseteq K^\varepsilon \cap \mathcal{L}$ can be enumerated, where every point is outputted exactly once, using polynomial space and time polynomial times*

$$\sum_{i=0}^n |\pi_{n-i+1}(K^\varepsilon) \cap \pi_{n-i+1}(\mathcal{L})|,$$

where π_1, \dots, π_n are the Gram-Schmidt projections of B .

Proof. Given the high level description above, to fully describe the algorithm, it remains to describe how we compute all feasible extensions of a giving partial assignment. In the algorithm, we will guarantee that we enumerate over all partial assignments feasible for K , while enumerating at most over all partial assignments feasible for K^ε .

Extending a partial assignment. Let $\mathbf{b}_1^*, \dots, \mathbf{b}_n^*$ be the associated dual basis for B . Assume that we are at a level i recursion node, $0 \leq i \leq n$, with an associated partial assignment $z_{n-i+1}, \dots, z_n \in \mathbb{Z}$. To begin processing this node, we first check that the partial assignment is feasible. Letting $\mathbf{t} = \sum_{j=n-i+1}^n z_j \mathbf{b}_j$, and

$$d = \min_{\mathbf{x} \in K} \|\pi_{n-i+1}(\mathbf{x} - \mathbf{t})\|_2,$$

we use Theorem 7 to compute $d' \in \mathbb{R}$ satisfying $d' \leq d \leq d' + \varepsilon$. If $d' > 0$, we conclude that the partial assignment is infeasible for K and terminate the node, and if $d' \leq 0$, we conclude that it is feasible for K^ε and continue.

If $i = n$, we output the lattice point $\sum_{i=1}^n z_i \mathbf{b}_i \in K^\varepsilon \cap \mathcal{L}$ and terminate the node. If $i < n$, we now compute the possible feasible extensions with $z_{n-i} \in \mathbb{Z}$, where we shall guarantee that all integral extensions feasible for K are found and that all examined extensions are feasible for K^ε . Let $\bar{\mathbf{b}}_{n-i}^* = \pi_{n-i+1}(\mathbf{b}_{n-i}^*)$ and $\hat{\mathbf{b}}_{n-i}^* = \mathbf{b}_{n-i}^* - \bar{\mathbf{b}}_{n-i}^*$. Set $M = 4\|\hat{\mathbf{b}}_{n-i}^*\|R/\varepsilon$, and let

$$u = \max_{\mathbf{x} \in K} \langle \bar{\mathbf{b}}_{n-i}^*, \mathbf{t} \rangle + \langle \hat{\mathbf{b}}_{n-i}^*, \mathbf{x} \rangle - M\|\pi_{n-i+1}(\mathbf{x} - \mathbf{t})\|_2$$

$$l = \min_{\mathbf{x} \in K} \langle \bar{\mathbf{b}}_{n-i}^*, \mathbf{t} \rangle + \langle \hat{\mathbf{b}}_{n-i}^*, \mathbf{x} \rangle + M\|\pi_{n-i+1}(\mathbf{x} - \mathbf{t})\|_2 .$$

Using Theorem 7, we compute $u' \in \mathbb{R}$ satisfying $u \leq u' \leq u + \varepsilon\|\hat{\mathbf{b}}_{n-i}^*\|_2/2$ and $l' \in \mathbb{R}$ satisfying $l \geq l' \geq l - \varepsilon\|\hat{\mathbf{b}}_{n-i}^*\|_2/2$. We now recurse on the integral extensions $z_{n-i} \in \{z \in \mathbb{Z} : l' \leq z \leq u'\}$.

Correctness. We must guarantee that the above algorithm correctly returns a set of points between $K \cap \mathcal{L}$ and $K^\varepsilon \cap \mathcal{L}$. Due to lack of space, we defer this analysis to the full version of the paper.

Complexity analysis. To bound the runtime of the above algorithm, we remark that the work done at each node in the recursion tree is polynomial (noting that the work enumerating $\{z \in \mathbb{Z} : l' \leq z \leq u'\}$ can be charged to a node's children), hence it suffices to bound the number of nodes in the tree. Given the above analysis, for each i , $0 \leq i \leq n$, the nodes

are level i are each associated with a distinct point in $\pi_{n-i+1}(K^\varepsilon) \cap \pi_{n-i+1}(\mathcal{L})$. Hence, the complexity of the algorithm is indeed polynomial times $\sum_{i=0}^n |\pi_{n-i+1}(K^\varepsilon) \cap \pi_{n-i+1}(\mathcal{L})|$, as needed. ◀

Motivated by the above lemma, we define the following measure of enumeration complexity.

► **Definition 11** (Schnorr-Euchner Enumerable). A convex body $K \subseteq \mathbb{R}^n$ is α -Schnorr-Euchner enumerable, or α -SE, with respect to a basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$ for \mathcal{L} (or vice versa) if for every shift \mathbf{t} , $\mathbf{t} \in \mathbb{R}^n$, and level i , $i \in \{1, \dots, n\}$, we have that $|\pi_{n-i+1}(K + \mathbf{t}) \cap \pi_{n-i+1}(\mathcal{L})| \leq \alpha$, i.e. the number of distinct feasible partial assignments for $K + \mathbf{t}$ with respect to B at level i is bounded by α .

As explained previously, the total number of feasible partial assignment controls the essential complexity of Schnorr-Euchner enumeration. The usefulness of the α -SE property for K is that it will enable us to bound the complexity of Schnorr-Euchner enumeration for general convex sets via their *covering numbers* with respect to K .

► **Definition 12** (Covering Numbers). For two sets $C, D \subseteq \mathbb{R}^n$, we denote the *covering number* of C with respect to D

$$N(C, D) = \min \{|T| : T \subseteq \mathbb{R}^n, C \subseteq T + D\} .$$

C, D have covering numbers bounded by (c_1, c_2) if $N(C, D) \leq c_1$ and $N(D, C) \leq c_2$.

The following corollary, which will be crucial to making our volume algorithm efficient, is immediate:

► **Corollary 13.** Let $K \subseteq \mathbb{R}^n$ be a convex body and $\mathcal{L} \subseteq \mathbb{R}^n$ be a full rank lattice with basis B . Assume that K is α -SE with respect to B . Then for any convex body $C \subseteq \mathbb{R}^n$, C is $\alpha N(C, K)$ -SE with respect to B . In particular, if C is centered and equipped with a weak membership oracle, then for any $\varepsilon' > 0$ and $\mathbf{t} \in \mathbb{R}^n$, a set S satisfying $(C + \mathbf{t}) \cap \mathcal{L} \subseteq S \subseteq (C^{\varepsilon'} + \mathbf{t}) \cap \mathcal{L}$ can be enumerated using polynomial space in time polynomial times $\alpha \cdot N(C, K)$.

To help make the above bounds effective, we will use the fact that covering numbers for convex bodies are tightly controlled by volumes. We note that we will generally be use these estimates with respect to different scalings of the same convex body (or one of its symmetrizations).

► **Theorem 14** (Covering Bounds [24]). For convex bodies $C, D \subseteq \mathbb{R}^n$, we have that

$$\frac{\text{vol}_n(C - D)}{\text{vol}_n(D - D)} \leq N(C, D) \leq n(\log n + \log \log n + 5) \frac{\text{vol}_n(C - D)}{\text{vol}_n(D)} .$$

The next lemma two lemmas will enable us to get the main estimates we will use to bound SE-complexity.

► **Lemma 15.** Let $K \subseteq \mathbb{R}^n$ be a convex body, and let $\mathcal{L} \subseteq \mathbb{R}^n$ be a full rank lattice with basis $B = (\mathbf{b}_1, \dots, \mathbf{b}_n)$. Then K is $N(K, \mathcal{P}(B))$ -SE with respect to B .

Proof. Let $T \subseteq \mathbb{R}^n$ satisfy $K \subseteq T + \mathcal{P}(B)$ and $|T| = N(K, \mathcal{P}(B))$. Letting π_1, \dots, π_n denote the Gram-Schmidt projections of B , it is easy to check that $\pi_i(\mathcal{P}(B))$, $i \in [n]$, is the parallelepiped of the basis $\pi_i(\mathbf{b}_1), \dots, \pi_i(\mathbf{b}_n)$ for $\pi_i(\Lambda)$, and hence is a fundamental domain of $\pi_i(\Lambda)$. Given this, for each $\mathbf{x} \in T$, $|\pi_i(\mathbf{x} + \mathcal{P}(B)) \cap \pi_i(\Lambda)| = 1$. Since $\pi_i(T + \mathcal{P}(B))$ covers $\pi_i(K) \cap \pi_i(\Lambda)$, we deduce that $|\pi_i(K) \cap \pi_i(\Lambda)| \leq |T|$. Hence, K is $|T|$ -SE as needed. ◀

► **Lemma 16** (Robustness of SE-complexity). *Let $K \subseteq \mathbb{R}^n$ be a convex body, $\mathcal{L} \subseteq \mathbb{R}^n$ be a full rank lattice with basis B . If K is α -SE with respect to B , then given a basis \tilde{B} of*

1. $\mathcal{L}' \subseteq \mathcal{L}$, a full rank sublattice, a basis B' of \mathcal{L}' for which K is α -SE
 2. $\mathcal{L} \subseteq \mathcal{L}'$, a full rank superlattice, a basis B' of \mathcal{L}' for which K is $\alpha \cdot [\mathcal{L}' : \mathcal{L}]$ -SE
- can be computed in polynomial time.*

6 Lattice Packing and Covering

We now present some additional relevant lattice concepts. We refer the reader to book [14] for a comprehensive reference.

For a symmetric convex body K , we define $\|\mathbf{x}\|_K = \inf \{s \geq 0 : \mathbf{x} \in sK\}$ as the norm induced by K , which satisfies all norm properties.

► **Definition 17** (Lattice Packing). A measurable set $A \subseteq \mathbb{R}^n$ packs with respect to a full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ (and vice versa) if the translates $\mathbf{y} + A$, $\mathbf{y} \in \mathcal{L}$, are mutually disjoint.

The packing induced by A and \mathcal{L} is α -dense if $\text{vol}_n(A)/\det(\mathcal{L}) \geq \alpha$. We note that packing density is always less than 1.

► **Definition 18** (Minimum Distance). For a symmetric convex body $K \subseteq \mathbb{R}^n$ and full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$, we denote $\lambda_1(K, \mathcal{L}) = \min_{\mathbf{y} \in \mathcal{L} \setminus \{\mathbf{0}\}} \|\mathbf{y}\|_K$, the minimum distance of \mathcal{L} under $\|\cdot\|_K$ (length of shortest non-zero vector).

► **Definition 19** (Packing and Covering Radius). Let $K \subseteq \mathbb{R}^n$ be a convex body and $\mathcal{L} \subseteq \mathbb{R}^n$ be a full rank lattice.

Let $\varrho(K, \mathcal{L}) = \lambda_1(K - K, \mathcal{L})$ denote the packing radius of K with respect to \mathcal{L} . K° packs with respect to \mathcal{L} iff $\varrho(K, \mathcal{L}) \geq 1$. If K is symmetric $\varrho(K, \mathcal{L}) = \lambda_1(K, \mathcal{L})/2$.

Let $\mu(K, \mathcal{L}) = \inf \{s \geq 0 : \mathcal{L} + sK = \mathbb{R}^n\}$ denote the covering radius of K with respect to \mathcal{L} . K covers with respect to \mathcal{L} iff $\mu(K, \mathcal{L}) \leq 1$.

► **Lemma 20**. *Let $K \subseteq \mathbb{R}^n$ be a convex body and let $\mathcal{L} \subseteq \mathbb{R}^n$ be a full rank lattice. Then, if K covers with respect \mathcal{L} and $\varrho(K, \mathcal{L}) \geq 1/\beta$, $\beta > 0$, then the covering induced by K and \mathcal{L} is β^n -thin.*

Proof. By assumption $\varrho(K, \mathcal{L}) \geq 1/\beta$, and hence $(K/\beta)^\circ$ packs with respect to \mathcal{L} . In particular, $\text{vol}_n(K/\beta) \leq \det(\mathcal{L})$. Therefore, the thinness is covering induced by K and \mathcal{L} is bounded by $\text{vol}_n(K)/\det(\mathcal{L}) \leq \text{vol}_n(K)/\text{vol}_n(K/\beta) = \beta^n$, as needed. ◀

7 Thin Covering Lattices

Our main technical contribution is a deterministic construction for thin covering lattices with good Schnorr-Euchner enumeration properties. We state its guarantees below.

► **Theorem 21** (Thin Lattice). *Let $K \subseteq \mathbb{R}^n$ be (\mathbf{a}_0, r, R) -centered convex body given by a weak membership oracle. Then, there is a deterministic $2^{O(n)}$ -time and $\text{poly}(n)$ -space algorithm that constructs a basis B for a full rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$ and a point $\mathbf{c} \in K$, satisfying*

1. \mathbf{c} is a $(6/7)^n$ -approximate KB point for K and $K[\mathbf{c}]$ is $(\mathbf{c}, r/(30n), 2R)$ -centered.
2. $K[\mathbf{c}]$ covers with respect to \mathcal{L} and has packing radius $\varrho(K[\mathbf{c}], \mathcal{L}) \geq 1/3$.
3. $K[\mathbf{c}]$ is $2^{O(n)}$ -SE with respect to B .

► **Remarks.** If K is symmetric, we can specialize the above theorem by setting $\mathbf{c} = \mathbf{0}$, in which case $K[\mathbf{c}] = K$. By Lemma 20, in the above theorem, we have that \mathcal{L} forms a 3^n -thin

covering with respect to $K[\mathbf{c}]$. Next, since $K[\mathbf{c}] \subseteq K - \mathbf{c}$, \mathcal{L} also covers with respect to K . In particular, the thinness of the covering induced by K and \mathcal{L} is bounded by

$$\text{vol}_n(K)/\det(\mathcal{L}) = \text{vol}_n(K)/\text{vol}(K[\mathbf{c}]) \cdot \text{vol}_n(K[\mathbf{c}])/\det(\mathcal{L}) \leq 2^n(7/6)^n 3^n = 7^n.$$

Hence \mathcal{L} is $2^{O(n)}$ -thin covering lattice for both $K[\mathbf{c}]$ and K .

Volume estimation. We now use the above construction to prove our main volume estimation result.

Proof of Theorem 1 (Volume Estimation). We wish to compute V such that $\text{vol}_n(K) \leq V \leq (1 + \varepsilon)^n \text{vol}_n(K)$ for $0 < \varepsilon < 1$, where $K \subseteq \mathbb{R}^n$ is a (\mathbf{a}_0, r, R) -centered convex body given by a weak membership oracle.

To begin, we construct the lattice \mathcal{L} with basis B , and point $\mathbf{c} \in K$ as guaranteed by Theorem 21. From here, we construct a weak membership oracle O_C for $C = K + (\varepsilon/3)K[\mathbf{c}]$ from the weak membership oracle for K (see [13] for details). Note that $K[\mathbf{c}]$ is $(\mathbf{c}, r/(30n), 2R)$ -centered and C is $(\mathbf{a}_0, r, 2R)$ -centered. From here, letting $\varepsilon' = \varepsilon r/(180n)$, we use Corollary 13 on inputs $C, (\varepsilon/3)\mathcal{L}, (\varepsilon/3)B$ and ε' to enumerate S , satisfying

$$(K + (\varepsilon/3)K[\mathbf{c}]) \cap (\varepsilon/3)\mathcal{L} \subseteq C^{\varepsilon'} \cap (\varepsilon/3)\mathcal{L} \subseteq (K + (\varepsilon/2)K[\mathbf{c}]) \cap (\varepsilon/3)\mathcal{L}$$

in time

$$2^{O(n)} N(K + (\varepsilon/2)K[\mathbf{c}], (\varepsilon/3)K[\mathbf{c}]) = 2^{O(n)}(1 + 1/\varepsilon)^n,$$

where the last inequality follows by Theorem 14. From here, we return $V = |S| \det(\mathcal{L})(\varepsilon/3)^n$ (note that we need only count each element of S as it is outputted, which requires only polynomial space). The fact that V satisfies the required bounds follows directly from the discussions in section 4 (see Equation (4)). ◀

Comparison with prior constructions. Much work has been dedicated to proving the existence of extremely thin-lattice coverings [21, 20, 22, 4, 10] – much of instigated by C.A. Rogers – where the best construction [22] provides $n^{\log n + O(1)}$ -thin coverings for any convex body K .

All of these constructions rely on sampling from a probabilistic ensembles of lattices, occasionally with some additional post processing, and are intrinsically difficult to derandomize. More problematically however, these ensembles produce lattices that are as “hard as possible” (see for example, section 2 in [3]) to enumerate from with known polynomial space methods, severely complicating their use in our context (and in many others in fact).

Given the above discussion, the construction in Theorem 21 gives the *first existential construction* of “easy to enumerate” thin-covering lattices for general convex bodies. As an added bonus of our construction, when the convex body K is symmetric, the covering lattice we construct has packing radius at least $1/3$ and has the property that CVP under the norm $\|\cdot\|_K$ can be solved in $2^{O(n)}$ time and $\text{poly}(n)$ space (since this reduces to enumeration inside shifts of K). While building thin covering lattices for ℓ_p norms is trivial – $2n^{-1/p}\mathbb{Z}_n$ is a $2^{O(n)}$ -thin covering lattice for the ℓ_p norm – building ones with packing radius $\Omega(1)$. In fact, even for the ℓ_2 norm, there is no known explicit construction of such a lattice. While the packing radius property is not necessary in our main application, we believe it might be useful elsewhere, such as in lattice based schemes for Locality Sensitive Hashing (see [2] for an application using the 24-dimensional Leech lattice).

The only previous algorithmic construction is due Alon et al [1], whose gave a deterministic $2^{O(n)}$ -time and 2^n -space thin-lattice construction for *symmetric bodies* based on a greedy construction of Rogers [21] – which our construction is also based on – along with a 2^n space enumeration method. For their enumeration technique, they rely on the M-ellipsoid covering and Voronoi cell based enumeration algorithms of [17, 7, 5]. With these techniques, starting with a thin covering lattice \mathcal{L} for a symmetric convex body K , they can enumerate $\mathcal{L} \cap C$, for any convex body C , in time $2^{O(n)}N(C, K)$ using 2^n space. Hence, the enumeration guarantees are similar to ours, though at the cost of exponential space.

Rogers’ greedy construction. We now describe Roger’s method and our related improvements. This construction starts with essentially any lattice \mathcal{L} and symmetric convex body K such that $\varrho(K, \mathcal{L}) \geq 1$. The construction proceeds by iteratively making \mathcal{L} denser, by adding points in $\mathcal{L}/3$ to \mathcal{L} , while guaranteeing that the packing radius with respect to K stays at least 1. This will have the net effect of increasing the packing density by 3. Since the packing density cannot increase indefinitely (it can never go above 1), the densification process eventually stops, at which point one can conclude that the final lattice \mathcal{L} , after a factor 3 scaling, covers with respect to K and has packing radius at least $1/3$.

A first main problem is that even if we initialize Rogers’ construction with an “easy to enumerate” lattice \mathcal{L} , the final generated lattice maybe so far away from the initial lattice that it loses the easy enumeration property. To avoid this problem, we show that if we start the procedure with an easy to enumerate dense packing lattice for K , then the procedure converges fast enough for the final generated lattice to retain the easy enumeration property. To build the initial dense packing lattice, we begin with a lattice \mathcal{L} with basis B derived from the axes of an M-ellipsoid of K , which satisfies that B is $2^{O(n)}$ -SE with respect to K , that we then subsequently sparsify it, using techniques of [6], to make it induce a $2^{-O(n)}$ -dense packing with respect to K .

A second problem with Roger’s greedy construction is that it only directly works for symmetric bodies. In particular, if we start with an asymmetric convex body K , the final generated lattice will only be guaranteed to cover with respect to $K - K$ and not K (here, the only known relation is that $\mu(K, \mathcal{L}) \leq n\mu(K - K, \mathcal{L})$, which is far too weak). To circumvent this problem, we symmetrize K about an approximate KB point using an efficient algorithm to construct such points. Our algorithm to construct approximate KB points will in fact rely on many iterated calls of our volume algorithm and thin-lattice construction for symmetric convex bodies.

8 Techniques

We now detail the main ideas behind our thin lattice construction. We begin by describing our thin lattice construction for symmetric convex bodies, and continue with our algorithm computing approximation Kovner-Besicovitch points. We recover our full thin lattice construction (Theorem 21) by combining these two algorithms. Due to lack of space, we defer most proofs to the full version of the paper.

8.1 Thin Lattice Construction

We now describe our construction of thin covering lattices for symmetric convex bodies, corresponding to parts 2 and 3 of Theorem 21.

The construction will proceed in three stages. In the first stage, we build a base lattice Λ with a basis B derived from the axes of an M -ellipsoid E of K , for which K is $2^{O(n)}$ -SE. In

the second stage, we sparsify the base lattice Λ so that it becomes a $2^{-O(n)}$ -dense packing lattice \mathcal{N} for K using techniques from [6]. In the last stage, we densify \mathcal{N} using Rogers' procedure to derive the final $2^{O(n)}$ -thin covering lattice \mathcal{L} .

Through these stages, our goal will be to guarantee that the “distance” of the base Λ to the final lattice \mathcal{L} , quantified by the product of indexes $[\Lambda : \mathcal{N}] \cdot [\mathcal{L} : \mathcal{N}]$, is bounded by $2^{O(n)}$. Having achieved this, the robustness of SE-complexity (see Lemma 16) will allow us to construct a basis for \mathcal{L} with respect to which K is $2^{O(n)}$ -SE. We now detail the main arguments underlying each stage.

M-Lattice. For the first stage, we define the basis B of Λ , so that $\mathcal{P}(B) \subseteq E$ is a maximum volume inscribed parallelepiped, where E an M-ellipsoid for K . Here it is not hard to check that $\text{vol}_n(E)/\mathcal{P}(B) = \text{vol}_n(B_2^n)/\text{vol}_n([-1/\sqrt{n}, 1/\sqrt{n}]^n) = 2^{O(n)}$. Given this, $\mathcal{P}(B)$ inherits the covering properties of E with respect to K , in particular, $N(K, \mathcal{P}(B)), N(\mathcal{P}(B), K) = 2^{O(n)}$. In particular, $\det(\Lambda) = \text{vol}_n(\mathcal{P}(B)) = 2^{\Theta(n)}\text{vol}_n(K)$, and, by Lemma 15, K is $2^{O(n)}$ -SE with respect to B .

Packing lattice. For the second stage, to make Λ a packing lattice, it suffices to “remove” all the lattice points in $\Lambda \cap 2K \setminus \{\mathbf{0}\}$ (by symmetry of K). By the covering properties, $|\Lambda \cap 2K| \leq N(2K, K) \cdot N(K, \mathcal{P}(B)) = 2^{O(n)}$, and hence, we may expect to find a sublattice \mathcal{N} such that $[\Lambda : \mathcal{N}] = 2^{O(n)}$ and $\mathcal{N} \cap 2K = \{\mathbf{0}\}$. Indeed, a simple expectation argument shows that a “random” sublattice \mathcal{N} of index $2^{O(n)}$ avoids all the non-zero points in $\Lambda \cap 2K$ with good probability (see [6]). Furthermore, one can find the sublattice \mathcal{N} deterministically using the method of conditional expectations. Since \mathcal{N} is a sublattice, note that by Lemma 16, a basis of \mathcal{N} can be computed for which the SE-complexity of K does not increase compared to B . To see that \mathcal{N} induces a $2^{-O(n)}$ -dense packing for K , note that

$$\begin{aligned} \det(\mathcal{N}) &= [\Lambda : \mathcal{N}] \det(\Lambda) = 2^{O(n)} \det(\Lambda) \\ &= 2^{O(n)} \text{vol}_n(\mathcal{P}(B)) = 2^{O(n)} \text{vol}_n(K), \quad \text{as needed.} \end{aligned}$$

Rogers' procedure. For the last stage, we initially set $\mathcal{L} \leftarrow \mathcal{N}$ and then iteratively densify \mathcal{L} to get a $2^{O(n)}$ -thin covering lattice. By assumption, \mathcal{L} starts as a packing lattice for K , or equivalently, \mathcal{L} has minimum distance $\lambda_1(K, \mathcal{L}) \geq 2$. To make \mathcal{L} denser, we look for a point $\mathbf{x} \in \mathcal{L}/3$ at distance greater than 2 from \mathcal{L} under $\|\cdot\|_K$. If such a point \mathbf{x} is found, we set $\mathcal{L} \leftarrow \mathcal{L} + \{0, \pm\mathbf{x}\}$. By the distance assumption and symmetry of K , we maintain the invariant $\lambda_1(K, \mathcal{L}) \geq 2$, while decreasing the determinant by a factor 3.

Note that each successful iteration increases the packing density by 3. Since the packing density starts at $2^{-O(n)}$, this process must terminate in at most $O(n)$ steps. In particular, after termination, we have that $[\mathcal{L} : \mathcal{N}] = 3^{O(n)}$, and hence by Lemma 16 and our assumptions on \mathcal{N} , we can compute a B basis of \mathcal{L} for which K is $2^{O(n)}$ -SE (indeed, this can be done at every iteration). Next, at termination, we must have that every point in $\mathcal{L}/3$ is at distance less than 2 from \mathcal{L} . From here, it is not hard to show that every point in \mathbb{R}^n is at distance at most $(3/2) \cdot 2 = 3$, i.e. $\mu(K, \mathcal{L}) \leq 3$. We can therefore return $\mathcal{L}/3$ as our covering lattice, which will have packing radius at least $1/3$ as desired.

The last detail is to show that at each stage, we can find a “far away” point in $\mathcal{L}/3$ or decide that none exists in $2^{O(n)}$ -time. By the above discussion, we can assume that at the current stage, we have a basis B for \mathcal{L} for which K is $2^{O(n)}$ -SE. From here, it is easy to see that there is a point in $\mathcal{L}/3$ at distance greater than 2 iff there exists $\mathbf{x} \in B \{0, \pm 1/3\}^n$ (yielding representatives for each coset in $(\mathcal{L}/3)/\mathcal{L}$) at distance greater than 2. Since a point

$\mathbf{x} \in \mathbb{R}^n$ is at distance greater than 2 from \mathcal{L} iff $(\mathbf{x} + 2K) \cap \mathcal{L} = \emptyset$, one can test this property for any \mathbf{x} in $2^{O(n)}$ -time using Schnorr-Euchner enumeration. Repeating this test 3^n times for each point in $B\{0, \pm 1/3\}^n$ yields the result.

This completes our description thin covering lattice constructions for symmetric bodies.

8.2 Computing Approximate Kovner-Besicovitch Points

We state the guarantees for our algorithm computing approximate KB points below.

► **Theorem 22.** *Let $K \subseteq \mathbb{R}^n$ be a (\mathbf{a}_0, r, R) -centered convex body given by a weak membership oracle. Then, for $\varepsilon > 0$, one can compute a $(1 + \varepsilon)^{-n}$ approximate Kovner-Besicovitch point $\mathbf{c} \in K$, such that $K[\mathbf{c}]$ is $(\mathbf{c}, \varepsilon r/(5n), 2R)$ -centered, in deterministic $2^{O(n)}(1 + 1/\varepsilon)^{2n+1}$ time and $\text{poly}(n)$ space.*

► **Remark.** Part 1 of Theorem 21 follows by applying the Theorem 22 to K with $\varepsilon = 1/6$.

High level algorithm. First, by applying a suitable linear affine transformation to K (i.e. standard ellipsoidal rounding), we may assume that $B_2^n \subseteq K \subseteq (n + 1)n^{1/2}B_2^n$. We now define the sequence of bodies $K_i = 2^i B_2^n \cap K$, for $i \in \{0, \dots, T\}$, $T = O(\log n)$, where $K_0 = B_2^n$ and $K_T = K$. For each K_i , $i \in [T - 1]$, we will compute a 3^{-n} approximate KB point \mathbf{c}_i for K_i from a 3^{-n} -approximate KB point \mathbf{c}_{i-1} for K_{i-1} . Finally, in the last step, from K_{T-1} to K_T , we amplify this to $(1 + \varepsilon)^{-n}$ approximation. We note that we may start with $\mathbf{c}_0 = \mathbf{0}$, since this is the center of symmetry for $K_0 = B_2^n$. Furthermore, at each step, since the volume $\text{vol}_n(K_i) \leq 2^n \text{vol}_n(K_{i-1})$, the KB value of \mathbf{c}_{i-1} with respect to K_i , $i \in [T]$, is at least $2^{-n} \cdot 3^{-n} \cdot 2^{-n} = 12^{-n}$.

To compute \mathbf{c}_i starting from \mathbf{c}_{i-1} , we perform the following improvement steps: from our current solution for \mathbf{c}_i (initialized at \mathbf{c}_{i-1} during the first iteration), we begin by building a thin-covering lattice \mathcal{L} with basis B for $K_i[\mathbf{c}_i]$ (note $K_i[\mathbf{c}_i]$ is symmetric). We then construct a covering of $(1/2)(K_i + \mathbf{c}_i)$ by $(\varepsilon/2)K_i[\mathbf{c}_i]$, whose centers are computed by enumerating $S = (1/2)((K_i + \varepsilon K_i[\mathbf{c}_i] + \mathbf{c}_i) \cap \varepsilon \mathcal{L})$ via Schnorr-Euchner enumeration using B . We then replace \mathbf{c}_i by the element in S (noting that $S \subseteq K_i$) of largest approximate KB value, where for each $\mathbf{x} \in S$ we approximate $\text{vol}_n(K_i[\mathbf{x}])$ to within $(1 + \varepsilon/10)^n$ using the volume algorithm for symmetric convex bodies. The concavity of the function $\text{vol}_n(K[\mathbf{x}])^{1/n}$ will allow us to show that at each step, we improve the objective value by essentially a $(1 + c\varepsilon)^n$ factor. Hence $O(1/\varepsilon)$ iterations suffice to construct a near optimal solution.

Acknowledgments. The author would like to thank Santosh Vempala and Oded Regev for useful conversations related to this paper, as well as the anonymous referees who greatly helped improve the quality of the presentation.

References

- 1 N. Alon, A. Schraibman, T. Lee, and S. Vempala. The approximate rank of a matrix and its algorithmic applications. In *STOC*, 2013.
- 2 A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, pages 459–468, 2006.
- 3 A. Becker, N. Gama, and A. Joux. Solving shortest and closest vector problems: The decomposition approach. Cryptology Eprint. Report 2013/685, 2013.

- 4 G. J. Butler. Simultaneous packing and covering in euclidean space. *Proceedings of the London Mathematical Society*, 25(3):721–735, 1972.
- 5 D. Dadush. *Integer Programming, Lattice Algorithms, and Deterministic Volume Estimation*. PhD thesis, Georgia Institute of Technology, 2012.
- 6 D. Dadush and G. Kun. Lattice sparsification and the approximate closest vector problem. In *SODA*, 2013.
- 7 D. Dadush, C. Peikert, and S. Vempala. Enumerative lattice algorithms in any norm via M-ellipsoid coverings. In *FOCS*, 2011.
- 8 D. Dadush and S. Vempala. Near-optimal deterministic algorithms for volume computation via m-ellipsoids. *Proceedings of the National Academy of Sciences*, 2013.
- 9 M. E. Dyer, A. M. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991. Preliminary version in STOC 1989.
- 10 U. Erez, S. Litsyn, and R. Zamir. Lattices which are good for (almost) everything. *IEEE Transactions on Information Theory*, 51(10):3401–3416, 2005.
- 11 Z. Füredi and I. Bárány. Computing the volume is difficult. In *STOC*, pages 442–447, New York, NY, USA, 1986. ACM.
- 12 Z. Füredi and I. Bárány. Approximation of the sphere by polytopes having few vertices. *Proceedings of the AMS*, 102(3), 1988.
- 13 M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, 1988.
- 14 P. M. Gruber. *Convex and discrete geometry*, volume 336. Springer Science & Business Media, 2007.
- 15 B. Grünbaum. Measures of symmetry for convex sets. In *Proceedings of the 7th Symposium in Pure Mathematics of the American Mathematical Society, Symposium on Convexity*, pages 233–270, 1961.
- 16 G. Hanrot and D. Stehlé. Improved analysis of Kannan’s shortest lattice vector algorithm. In *CRYPTO*, pages 170–186, Berlin, Heidelberg, 2007. Springer-Verlag.
- 17 D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. *SIAM Journal on Computing*, 42(3):1364–1391, 2013. Preliminary version in STOC 2010.
- 18 V. D. Milman. Inégalités de Brunn-Minkowski inverse et applications at la theorie locales des espaces normes. *C. R. Math. Acad. Sci. Paris*, 302(1):25–28, 1986.
- 19 V. D. Milman and A. Pajor. Entropy and asymptotic geometry of non-symmetric convex bodies. *Advances in Mathematics*, 152(2):314–335, 2000.
- 20 C. A. Rogers. Lattice coverings of space: The Minkowski-Hlawka theorem. *Proceedings of the London Mathematical Society*, s3-8(3):447–465, 1958.
- 21 C. A. Rogers. A note on coverings and packings. *Journal of the London Mathematical Society*, s1-25(4):327–331, 1950.
- 22 C. A. Rogers. Lattice coverings of space. *Mathematika*, 6:33–39, 6 1959.
- 23 C. A. Rogers and G. C. Shephard. The difference body of a convex body. *Archiv der Mathematik*, 8:220–233, 1957.
- 24 C. A. Rogers and C. Zong. Covering convex bodies by translates of convex bodies. *Mathematika*, 44:215–218, 6 1997.
- 25 D. B. Yudin and A. S. Nemirovski. Evaluation of the information complexity of mathematical programming problems (in russian). *Ekonomika i Matematicheskie Metody*, 13(2):3–45, 1976.

Optimal Deterministic Algorithms for 2-d and 3-d Shallow Cuttings

Timothy M. Chan^{*1} and Konstantinos Tsakalidis²

1 Cheriton School of Computer Science, University of Waterloo, Canada
tmchan@uwaterloo.ca

2 Department of Computer Science and Engineering, Hong Kong University of Science and Technology, China
tsakalid@cse.ust.hk

Abstract

We present optimal deterministic algorithms for constructing shallow cuttings in an arrangement of lines in two dimensions or planes in three dimensions. Our results improve the deterministic polynomial-time algorithm of Matoušek (1992) and the optimal but randomized algorithm of Ramos (1999). This leads to efficient derandomization of previous algorithms for numerous well-studied problems in computational geometry, including halfspace range reporting in 2-d and 3-d, k nearest neighbors search in 2-d, $(\leq k)$ -levels in 3-d, order- k Voronoi diagrams in 2-d, linear programming with k violations in 2-d, dynamic convex hulls in 3-d, dynamic nearest neighbor search in 2-d, convex layers (onion peeling) in 3-d, ε -nets for halfspace ranges in 3-d, and more. As a side product we also describe an optimal deterministic algorithm for constructing standard (non-shallow) cuttings in two dimensions, which is arguably simpler than the known optimal algorithms by Matoušek (1991) and Chazelle (1993).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases shallow cuttings, derandomization, halfspace range reporting, geometric data structures

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.719

1 Introduction

Shallow cuttings were introduced by Matoušek [25] as a tool for range searching, specifically, *halfspace range reporting*. They have since found applications to numerous other central problems in computational geometry, including $(\leq k)$ -levels in arrangements of hyperplanes, order- k Voronoi diagrams, linear programming with k violations, dynamic convex hulls, and dynamic nearest neighbor search (see Section 1.4 for more information). At SoCG'99, Ramos [29] presented an optimal randomized algorithm for constructing shallow cuttings in two and three dimensions. A nagging question that has remained open is whether there is an equally efficient deterministic algorithm. The main result of this paper is a positive resolution to this question. Although the question is mainly about theoretical understanding, and derandomization isn't the most "fashionable" topic in computational geometry, we believe that in this case the fundamental nature of the problem and its wide-ranging consequences make the problem important to study.

* Part of this work was done during the author's visit to the Hong Kong University of Science and Technology.



1.1 Standard Cuttings

► **Definition 1.** Let H be a set of n hyperplanes in \mathbb{R}^d . Given a parameter $r \in [1, n]$ and a region $L \subseteq \mathbb{R}^d$, a $\frac{1}{r}$ -cutting for H covering L is a set of interior-disjoint simplices (cells) such that

- (i) the interior of every cell intersects at most $\frac{n}{r}$ hyperplanes of H , and
- (ii) the union of the cells covers L .

The *conflict list* H_Δ of a cell Δ is the set of (at most $\frac{n}{r}$) hyperplanes of H that intersect Δ . The *size* of the cutting is the number of its cells.

Cuttings are a fundamental tool in geometric divide-and-conquer. In the default “standard” setting, a cutting covers all of \mathbb{R}^d , i.e., $L = \mathbb{R}^d$.

Random sampling techniques by Clarkson [16] and Haussler and Welzl [21] imply the existence of (standard) $\frac{1}{r}$ -cuttings of size $O((r \log r)^d)$. Chazelle and Friedman [15] refined the bound to $O(r^d)$, which is optimal. (In the 2-d case, there is a simple alternative proof based on *levels* by Matoušek [23].)

Considerable effort was spent in finding efficient deterministic algorithms to construct such an optimal-size cutting. Even the 2-d case turned out to be a challenge. At SoCG’89, Matoušek [23] presented an $O(nr^2 \log r)$ -time algorithm for $d = 2$. At the same conference, Agarwal [4] (see also his PhD thesis [5]) presented an $O(nr \log n \log^{3.33} r)$ -time algorithm for $d = 2$. In a subsequent paper, Matoušek [24] improved the deterministic time bound to $O(nr)$ for $d = 2$, which is optimal if the algorithm is required to output the conflict lists of all the cells (since the worst-case total size of the conflict lists is $\Theta(r^2 \cdot \frac{n}{r}) = \Theta(nr)$). Matoušek’s later paper also described a deterministic $O(nr^{d-1})$ -time algorithm for any constant dimension d , which is again optimal if we need to output all conflict lists, but this result holds under the restriction that r is not too big, i.e., $r < n^{1-\delta}$ for some constant $\delta > 0$. Finally, Chazelle [14] obtained a deterministic $O(nr^{d-1})$ -time algorithm without any restriction on r for all constant dimensions d . All of these deterministic algorithms are complicated and/or make use of advanced derandomization techniques such as ε -approximations [21].

1.2 Shallow Cuttings

Given a point p , the *level* of p in H is the number of hyperplanes of H that are below p . We define $L_{\leq k}(H)$ to be the ($\leq k$)-*level*, i.e., the region of all points with level in H at most k . A *shallow cutting* is a variant of the standard cutting that is required to cover only points that are “shallow”, i.e., have small levels.

► **Definition 2.** Given parameters $k, r \in [1, n]$, a k -shallow $\frac{1}{r}$ -cutting is a $\frac{1}{r}$ -cutting for $L_{\leq k}(H)$.

We concentrate on the most important case of $k = \Theta(\frac{n}{r})$, which is sufficient for all of the applications encountered; in fact, shallow cuttings for any value of k can be reduced to this case—see the remarks in Section 5. Matoušek [25] proved the existence of a $\Theta(\frac{n}{r})$ -shallow $\frac{1}{r}$ -cutting of size $O(r^{\lceil d/2 \rceil})$, which is smaller than the $O(r^d)$ bound for standard $\frac{1}{r}$ -cuttings and is optimal in the worst case. In particular, for $d \in \{2, 3\}$, the size is $O(r)$.

In the same paper, Matoušek presented a deterministic algorithm that can construct such a shallow cutting in polynomial time; the running time improves to $O(n \log r)$ but only when r is small, i.e., $r < n^\delta$ for a sufficiently small constant δ . Later, Ramos [29] presented a complicated randomized algorithm for $d = 3$ (and hence $d = 2$ as well) with $O(n \log n)$ expected running time to construct not just a single shallow cutting, but a hierarchy of $O(\log n)$ such shallow cuttings for all r ’s forming a geometric sequence from 1 to n . (Such a

hierarchy is useful in certain applications.) Recently, at SODA'14, Afshani and Tsakalidis [3] managed to achieve the same bound deterministically, albeit only for an orthogonal variant of the problem where the input objects are orthants in \mathbb{R}^3 (which nonetheless has applications to *dominance range reporting*); subsequently, Afshani et al. [2] improved the time bound for a single shallow cutting to $O(n \log \log n)$ in the word RAM model. The case of orthants is indeed a special case, as orthants can be mapped to halfspaces via a certain transformation [12].

1.3 Our Contributions

We present deterministic algorithms to construct a $\Theta(\frac{n}{r})$ -shallow $\frac{1}{r}$ -cutting of size $O(r)$ for $d \in \{2, 3\}$ in $O(n \log r)$ time, which is optimal in a comparison-based model (the default model in this paper). Like Ramos' randomized algorithm [29], our algorithms can in fact construct a hierarchy of such shallow cuttings for all r 's in a geometric sequence, along with the conflict lists of all cells, in $O(n \log n)$ total time. (Note that for our 3-d algorithm, we do not insist the cutting in one layer of the hierarchy be nested inside the cutting in the next layer.)

Considering how involved known deterministic algorithms for standard cuttings are, we are happy to report that the new results are not complicated to derive. All the needed background is provided in Section 2; no advanced derandomization techniques are used. The main algorithms are describable in a few lines, as seen in Sections 3 and 4, although their analyses are not trivial.

Like in Chazelle's cutting algorithm [14], we will construct the hierarchy layer by layer, refining the shallow cutting in the previous layer to obtain the shallow cutting in the next layer. A naive implementation would cause an amplification of the constant factor in the cutting size bound, which may "blow up" after logarithmically many iterations. Chazelle used ε -approximations and *sparse ε -nets* to refine the cutting in each cell, and controlled the blow-up by charging cost to some easily summable quantity (namely, the number of vertices inside the cell). We replace ε -approximations and sparse ε -nets with the more elementary techniques by Megiddo and Dyer [28, 17]. We use a brute-force search to find the best way to refine the cutting in each cell, and control the blow-up by bounding cost in terms of the cost of an optimal cutting—this strategy is reminiscent of the analysis of approximation algorithms or PTASes (although we do not explicitly design an approximation algorithm to find the minimum-size cutting).

The strategy works beautifully in 2-d, but the constant-factor blow-up becomes tougher to deal with in 3-d, because cost of substructures along the cell boundaries becomes non-negligible. To tackle this issue, we borrow an idea from a different paper by Ramos [30], of using planar graph separators to group cells into regions, which we call "supercells", so that the total size of the boundaries of the supercells is reduced. (Ramos originally applied this idea to obtain an optimal deterministic algorithm for the 3-d diameter problem and for computing lower envelopes of certain bivariate surfaces in 3-d, but did not consider shallow cuttings in that paper. Also, the details of his algorithms appear more complicated, using ε -nets and supercells of size n^δ , whereas we use only supercells of constant size.)

In the appendix, we show that our ideas can also lead to a new presentation of a deterministic $O(nr)$ -time algorithm for constructing standard $\frac{1}{r}$ -cuttings in 2-d. This may be of independent pedagogical interest, considering the long line of previous complicated algorithms.

1.4 Applications

As mentioned, shallow cuttings are important because of their numerous applications. Below we list some of the specific implications of our new 2-d and 3-d deterministic algorithms.

1. The first optimal deterministic $O(n \log n)$ -time algorithm to preprocess a set of n points in \mathbb{R}^3 into an $O(n)$ -space data structure, so that we can answer a *halfspace range reporting* query (i.e., report all k points that lie within any given halfspace) in $O(\log n + k)$ time. This result follows from the work of Afshani and Chan [1], which was almost deterministic except for the invocation of Ramos' algorithm to construct a shallow cutting during preprocessing.
By a standard lifting transformation, the same result holds for *circular range reporting* in \mathbb{R}^2 (reporting all k points that lie inside any given circle) and *k nearest neighbors search* in \mathbb{R}^2 (reporting all k nearest neighbors to a given point, in arbitrary order, under the Euclidean metric).
2. The first optimal deterministic $O(n \log n + nk^2)$ -time algorithm to construct the $(\leq k)$ -level of an arrangement of n planes in \mathbb{R}^3 . This result follows from the work of Chan [9], which previously required randomization.
3. The currently fastest deterministic $O(n \log n + nk \cdot t(k))$ -time algorithm for constructing the k -th order Voronoi diagram of n points in \mathbb{R}^2 . Here, $t(\cdot)$ denotes the (amortized) update and query time complexity for the 2-d dynamic convex hull problem (under gift-wrapping queries). We have $t(k) = O(\log k \log \log k)$ [7], or better still, $t(k) = O(\log k)$ [8] if one has confidence in the over-100-page proof in the latter paper. This result again follows from the work of Chan [9]. Compare the result with Ramos' randomized $O(n \log n + nk2^{O(\log^* k)})$ -time algorithm [29].
4. A deterministic $O((n + k^2) \log k)$ -time algorithm for *2-d linear programming with at most k violations* (i.e., given a set of n halfspaces, find the point that lies inside all but k of the halfspaces and is extreme along a given direction). This result follows from another work of Chan [10], which was almost deterministic except for the construction of a 2-d shallow cutting in one step.
5. The first deterministic data structure for *dynamic 3-d convex hull* with polylogarithmic amortized update and query time, namely, $O(\log^3 n)$ amortized insertion time, $O(\log^6 n)$ amortized deletion time, and $O(\log^2 n)$ time for a gift-wrapping query. This result follows from another work of Chan [11], which was almost deterministic except for the construction of a hierarchy of 3-d shallow cuttings during certain update operations.
This result itself spawns countless additional consequences, for example, to *dynamic 2-d smallest enclosing circle*, *dynamic 2-d bichromatic closest pair*, *dynamic 2-d diameter*, *dynamic 2-d Euclidean minimum spanning tree*, *3-d convex layers (onion peeling)*, output-sensitive construction of 3-d k -levels, and so on.
6. A deterministic data structure for *dynamic 2-d halfspace range reporting* with $O(\log^{6+\varepsilon} n)$ amortized update time and $O(\log n + k)$ query time for any fixed $\varepsilon > 0$. In 3-d, the query time increases to $O(\log^2 n / \log \log n + k)$. This result follows from yet another work of Chan [11], which was almost deterministic except for the construction of a hierarchy of shallow cuttings during certain update operations.
7. A deterministic $O(n \log r)$ -time algorithm to construct a $\frac{1}{r}$ -net of size $O(r)$ for n points in \mathbb{R}^3 with respect to halfspace ranges. This application actually appeared in Matoušek's original paper on shallow cuttings [25]. There, he was interested in proving existence of $O(r)$ -size nets, but with our shallow cutting algorithm, the deterministic time bound follows. (Roughly speaking, in the dual, we construct a $\frac{n}{r}$ -shallow $O(\frac{1}{r})$ -cutting, construct

an ε -cutting within each cell for a sufficiently small constant ε , and output an arbitrary plane passing below each subcell.) Of course, ε -nets are well known and central to combinatorial and computational geometry. Previously, there were deterministic $nr^{O(1)}$ -time algorithms (e.g., see a recent note [20]), and an $O(n \log r)$ -time algorithm but only when r is small, i.e., $r < n^\delta$ for some constant δ [25].

By a standard lifting transformation, the same result holds for ε -nets for points in \mathbb{R}^2 with respect to circular disk ranges.

2 Preliminaries

It will be more convenient to work with the parameter $K := \frac{n}{r}$ instead of r . For brevity, a k -shallow $\frac{K}{n}$ -cutting will be referred to as a (k, K) -shallow cutting. It satisfies the properties that (i) each cell intersects at most K hyperplanes, and (ii) the cells cover $L_{\leq k}(H)$. Our goal is to compute a $(k, \Theta(k))$ -shallow cutting of size $O(\frac{n}{k})$.

For a set V of points in \mathbb{R}^d , we denote by $\text{UH}(V)$ the region underneath the *upper hull* of V . We define the *vertical decomposition* $\text{VD}(V)$ to be the set of interior-disjoint cells covering $\text{UH}(V)$, such that each cell is bounded from above by a different face of $\text{UH}(V)$, is bounded from the sides by vertical *walls*, and is unbounded from below. For example, in 2-d, the boundary of $\text{UH}(V)$ is a concave chain; a cell in $\text{VD}(V)$ is bounded by an edge of $\text{UH}(V)$ and two walls (downward vertical rays). In 3-d, the boundary of $\text{UH}(V)$ is a concave polygonal surface with triangular faces; a cell in $\text{VD}(V)$ is bounded by a triangle and three walls (trapezoids that are unbounded from below).

In the studied dimensions $d \in \{2, 3\}$, we find it simpler to work with the following equivalent form of shallow cuttings:

► **Definition 3.** Given parameters $k, K \in [1, n]$, a (k, K) -shallow cutting for H in vertex form is a set V of points such that

- (i) every point in V has level at most K , and
- (ii) $\text{UH}(V)$ covers $L_{\leq k}(H)$.

The *conflict list* of a point $v \in V$ is the set of (at most K) hyperplanes in H that are below v .

A (k, K) -shallow cutting under the original definition can be transformed into a $(k, k+K)$ -shallow cutting in vertex form simply by letting V be the set of vertices of the cells (after pruning cells that do not intersect $L_{\leq k}(H)$). In the reverse direction, a (k, K) -shallow cutting V in vertex form can be transformed to a (k, dK) -shallow cutting under the original definition simply by taking $\text{VD}(V)$, since the conflict list H_Δ of a cell Δ is contained in the union of the conflict lists of the d vertices of Δ , and thus $|H_\Delta| \leq dK$. In 2-d and 3-d, the size of $\text{VD}(V)$ is $O(|V|)$, and computing $\text{VD}(V)$ takes $O(|V| \log |V|)$ time by an optimal convex hull algorithm.

From now on, all shallow cuttings will be in vertex form by default.

Our algorithms do not require any advanced derandomization techniques at all. Only three facts are needed (the third is used only for the 3-d case):

► **Fact 4 (Constant-Size Cuttings).** Given a set of n lines in \mathbb{R}^2 or planes in \mathbb{R}^3 and any constant $\varepsilon > 0$, a (standard) ε -cutting of constant size can be computed in $O(n)$ worst-case time.

► **Fact 5 (Existence of $O(\frac{n}{k})$ -Size Shallow Cuttings).** Given a set of n lines in \mathbb{R}^2 or planes in \mathbb{R}^3 and a parameter $k \in [1, n]$, there exists a $(k, c_0 k)$ -shallow cutting (in vertex form) of maximum size $c'_0 \frac{n}{k}$, for some universal constants c_0, c'_0 .

► **Fact 6** (Planar Graph Separators). *Given a triangulated planar graph with n vertices and a parameter $t \in [1, n]$, we can group the triangles into at most $a_0 \frac{n}{t}$ connected regions where each region contains at most t triangles, and the total number of edges along the boundaries of the regions is at most $a'_0 \frac{n}{\sqrt{t}}$ for some universal constants a_0, a'_0 . Such regions can be computed in $O(n \log n)$ time.*

Fact 4 was known in the 1980s even before the term “cutting” was coined. In deriving their linear-time algorithm for 3-d linear programming, Megiddo [27] and Dyer [17] implicitly gave a linear-time construction of a $\frac{7}{8}$ -cutting of size 4 in 2-d. Megiddo [28] subsequently generalized the construction to d dimensions, yielding a $(1 - 1/2^{2^d - 1})$ -cutting of size $2^{2^d - 1}$ in linear time. (The cells may not be simplices, but we can triangulate them and the size remains bounded by a constant.) Although these constructions give ε -cuttings for one specific constant $\varepsilon > 0$, iterating a constant number of times automatically yields ε -cuttings for any given constant $\varepsilon > 0$ in linear time. The size of such a cutting may be suboptimal, but for our purposes, any constant size bound will be sufficient. More powerful techniques based on ε -approximations and ε -nets [21, 15] can yield better bounds, but a virtue of Megiddo and Dyer’s constructions is that they are completely elementary, relying on linear-time median finding as the only subroutine.

Fact 5 was proved by Matoušek [25] by using Chazelle and Friedman’s random sampling techniques [15]. (In the 2-d case, there is a simpler alternative proof using levels, similar to [23] and implicit in one of the proofs in [6].) For our purposes, we do not actually need to know how Fact 5 is proved and do not care about the construction time—we just need the existence of $O(\frac{n}{k})$ -size shallow cuttings, not for our algorithms themselves but for their analyses.

Fact 6 is a multiple-regions version [18] of the well-known planar graph separator theorem [22], as applied to the dual of the given graph. The multiple-regions version follows from the standard version by recursion. The running time $O(n \log n)$ can actually be reduced to $O(n)$ [19], although we do not need this improvement. A version by Frederickson [18] can further guarantee that each region has $O(\sqrt{r})$ boundary edges (Fact 6 guarantees the same bound but on average only); again, we do not need such an improvement.

3 A 2-d Shallow Cutting Algorithm

We begin in 2-d and prove the following theorem, from which our main result will follow as a corollary:

► **Theorem 7.** *For a set H of n lines in \mathbb{R}^2 , a parameter $k \in [1, n]$, and some suitable constants B, C, C' , given a (Bk, CBk) -shallow cutting V_{IN} (in vertex form) for H of size at most $C' \frac{n}{Bk}$ along with its conflict lists, we can compute a (k, Ck) -shallow cutting V_{OUT} (in vertex form) for H of size at most $C' \frac{n}{k}$ along with its conflict lists in $O(n + \frac{n}{k} \log \frac{n}{k})$ deterministic time.*

Proof.

Algorithm. Let ε be a constant to be set later. Our algorithm is conceptually simple:

1. For each cell $\Delta \in \text{VD}(V_{\text{IN}})$:
 - 1.1. Compute by Fact 4 an ε -cutting Γ_{Δ} for H_{Δ} of $O(1)$ size, where the cells are clipped (and re-triangulated) to lie within Δ . Let Λ_{Δ} be the set of vertices that define the cells of Γ_{Δ} .

- 1.2. Compute by brute force the *smallest subset* $V_\Delta \subseteq \Lambda_\Delta$ such that
 - (i) every vertex in V_Δ has level in H_Δ at most Ck , and
 - (ii) $\text{UH}(V_\Delta)$ covers all vertices in Λ_Δ that are in $L_{\leq 2k}(H_\Delta)$.
3. Return $V_{\text{OUT}} := \bigcup_{\Delta \in \text{VD}(V_{\text{IN}})} V_\Delta$ and all its conflict lists.

Complexity. In Line 1, computing $\text{VD}(V_{\text{IN}})$ takes $O(\frac{n}{k} \log \frac{n}{k})$ time by an optimal convex hull algorithm, since $|V_{\text{IN}}| \leq C' \frac{n}{Bk} = O(\frac{n}{k})$. Line 1.1 takes time linear in $|H_\Delta|$ by Fact 4, for a total of $\sum_{\Delta \in \text{VD}(V_{\text{IN}})} O(|H_\Delta|) = O(C' \frac{n}{Bk} \cdot 2CBk) = O(n)$ time. For Line 1.2, first we determine the level in H_Δ of every vertex in Λ_Δ by a linear scan over H_Δ , and then we probe all possible subsets of Λ_Δ . Since Γ_Δ and Λ_Δ have $O(1)$ size, there are “only” $O(1)$ subsets to test (although the constant is exponentially bigger) and each subset can be tested for the two stated conditions in $O(1)$ time. Thus, the whole step takes time linear in $|H_\Delta|$, which again totals to $O(n)$. In Line 3, computing V_{OUT} takes time linear in the output size. The conflict list of every output vertex in V_Δ can be computed by a linear scan over H_Δ , again in $O(n)$ total time.

Correctness. To show that V_{OUT} is a correct (k, Ck) -shallow cutting for H , we just check that $\text{UH}(V_{\text{OUT}})$ covers $L_{\leq k}(H)$. This follows since for any point inside a cell of Γ_Δ with level at most k , the three vertices of the cell in Γ_Δ have levels at most $k + \varepsilon|H_\Delta| \leq k + \varepsilon(2CBk) = 2k$ by setting the constant $\varepsilon := \frac{1}{2CB}$, and are thus covered by $\text{UH}(V_\Delta)$.

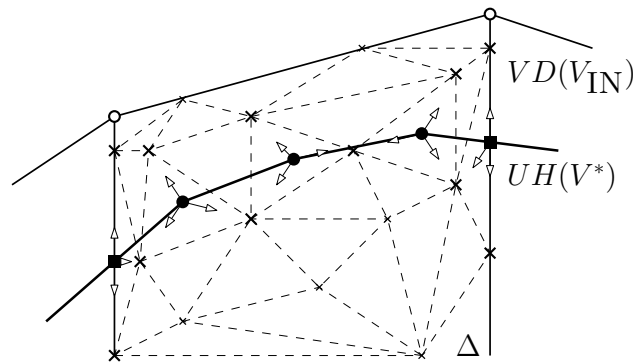
To bound the size of V_{OUT} , we compare it against a $(2k, 2c_0k)$ -shallow cutting V^* of size $c'_0 \frac{n}{2k}$ provided by Fact 5. Note that V^* is covered by $\text{VD}(V_{\text{IN}})$ by picking a constant $B \geq 2c_0$, since every vertex in V^* has level at most $2c_0k$, and $\text{VD}(V_{\text{IN}})$ covers $L_{\leq Bk}(H)$.

We render V^* comparable to V_{OUT} by modifying V^* in two steps (we emphasize that these steps are not part of the algorithm but are for the correctness proof only):

- First, we *chop* $\text{UH}(V^*)$ at the walls of the cells of $\text{VD}(V_{\text{IN}})$. A new vertex is formed at each wall; we create two copies of each such vertex (one assigned to each of the two incident cells of $\text{VD}(V_{\text{IN}})$) and add them to V^* . (See Figure 1.) For each cell $\Delta \in \text{VD}(V_{\text{IN}})$, let $V_\Delta^* := V^* \cap \Delta$. Then (i) every vertex in V_Δ^* (including the extra vertices added) has level at most $4c_0k$, and (ii) $\text{UH}(V_\Delta^*)$ is exactly $\text{UH}(V^*) \cap \Delta$ and thus covers $L_{\leq 2k}(H) \cap \Delta$. The number of extra vertices added is at most $2C' \frac{n}{Bk}$, so the size of V^* is now at most $(\frac{c'_0}{2} + 2\frac{C'}{B}) \frac{n}{k}$.
- Next, for every cell $\Delta \in \text{VD}(V_{\text{IN}})$, we *snap* the vertices in V_Δ^* to the vertices of Γ_Δ , i.e., we replace every vertex $v \in V_\Delta^*$ with the three vertices of the cell in Γ_Δ containing v . (See Figure 1.) This makes $V_\Delta^* \subseteq \Lambda_\Delta$. Then (i) every vertex in V_Δ^* now has level at most $4c_0k + \varepsilon|H_\Delta| \leq 4c_0k + \varepsilon(2CBk) = (4c_0 + 1)k$, and (ii) $\text{UH}(V_\Delta^*)$ can only increase in its coverage. The size of V^* triples to at most $(\frac{3}{2}c'_0 + 6\frac{C'}{B}) \frac{n}{k}$.

Then Line 1.2 guarantees that $|V_\Delta| \leq |V_\Delta^*|$ by setting the constant $C := 4c_0 + 1$, since the subset $V_\Delta^* \subseteq \Lambda_\Delta$ satisfies the two stated conditions and V_Δ is the smallest such subset. Therefore, totalling over all cells in $\text{VD}(V_{\text{IN}})$, we have $|V_{\text{OUT}}| \leq |V^*| \leq (\frac{3}{2}c'_0 + 6\frac{C'}{B}) \frac{n}{k} \leq C' \frac{n}{k}$ as desired, by setting the constant $C' := \frac{\frac{3}{2}c'_0}{1 - \frac{6C'}{B}}$ and picking a constant $B > 6$. ◀

► **Corollary 8.** For a set H of n lines in \mathbb{R}^2 , a parameter $k \in [1, n]$, and some suitable constants B, C, C' , we can compute a $(B^i k, CB^i k)$ -shallow cutting of size at most $C' \frac{n}{B^i k}$, along with its conflict lists, for all $i = 0, 1, \dots, \log_B \frac{n}{k}$ in $O(n \log \frac{n}{k})$ total deterministic time. In particular, we can compute a (k, Ck) -shallow cutting of size $O(\frac{n}{k})$ in the stated time.



■ **Figure 1** Modifying V^* by *chopping* (adding points marked by black squares) and *snapping* (replacing a point with three points indicated by white arrows). The cutting Γ_Δ is shown in dashed lines, and its vertices Λ_Δ are marked by crosses.

Proof. By Theorem 7, the running time $T(n, k)$ satisfies the recurrence

$$T(n, k) = T(n, Bk) + O\left(n + \frac{n}{k} \log \frac{n}{k}\right),$$

with the trivial base case $T(n, n) = O(n)$. The recurrence solves to $T(n, k) = O(n \log_B \frac{n}{k}) + O(\frac{n}{k} \log \frac{n}{k}) \sum_{i=0}^{\log_B \frac{n}{k}} \frac{1}{B^i} = O(n \log \frac{n}{k})$. ◀

4 A 3-d Shallow Cutting Algorithm

We now extend the approach from the previous section to 3-d. We need to incorporate planar separators in the algorithm and further new ideas in the analysis.

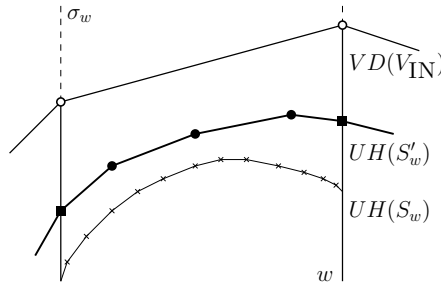
► **Theorem 9.** For a set H of n planes in \mathbb{R}^3 , a parameter $k \in [1, n]$ and some suitable constants B, C, C' , given a (Bk, Ck) -shallow cutting V_{IN} for H of size at most $C' \frac{n}{Bk}$ along with its conflict lists, we can compute a (k, Ck) -shallow cutting V_{OUT} for H of size at most $C' \frac{n}{k}$ along with its conflict lists in $O(n + \frac{n}{k} \log \frac{n}{k})$ deterministic time.

Proof.

Algorithm. Let ε and t be constants to be set later.

0. Group the faces of $\text{UH}(V_{\text{IN}})$ into regions by applying Fact 6 with parameter t . The union of the cells of $\text{VD}(V_{\text{IN}})$ defined by the triangles in a region will be called a *supercell* of $\text{VD}(V_{\text{IN}})$.
1. For each *supercell* Δ of $\text{VD}(V_{\text{IN}})$:
 - 1.1. Do as in Line 1.1 of the algorithm in Section 3.
 - 1.2. Do as in Line 1.2 of the algorithm in Section 3.
2. Do as in Line 3 of the algorithm in Section 3.

Complexity. Line 0 takes $O(\frac{n}{k} \log \frac{n}{k})$ time by Fact 6, since $|V_{\text{IN}}| \leq C' \frac{n}{Bk} = O(\frac{n}{k})$. Lines 1.1, 1.2, and 2 take $O(n + \frac{n}{k} \log \frac{n}{k})$ time by an analysis similar to Section 3, since each supercell still has $O(1)$ complexity for t constant.



■ **Figure 2** Replacing the concave chain $UH(S_w)$ with a sparse concave chain $UH(S'_w)$ at a wall w of a supercell of $VD(V_{IN})$. The set S'_w is a planar shallow cutting.

Correctness. By the same argument as in Section 3, we see that V_{OUT} is a correct (k, Ck) -shallow cutting for H , this time by setting the constant $\varepsilon := \frac{1}{3tCB}$, since $|H_\Delta| \leq 3tCBk$ for each supercell Δ of $VD(V_{IN})$.

As in Section 3, we bound the size of V_{OUT} by comparing it against a $(2k, 2c_0k)$ -shallow cutting V^* of size $c'_0 \frac{n}{2k}$ provided by Fact 5. As before, V^* is covered by $VD(V_{IN})$, this time by picking a constant $B \geq 6c_0$.

We render V^* comparable to V_{OUT} by modifying V^* in three steps, the second of which is new (again these steps are not part of the algorithm but are for the correctness proof only):

- First, we *chop* $UH(V^*)$ at the walls of the supercells of $VD(V_{IN})$. A new planar concave chain of vertices is formed at each wall; we create two copies of the chain (one assigned to each of the two incident cells of $VD(V_{IN})$) and add their vertices to V^* . For each supercell Δ of $VD(V_{IN})$, let $V_\Delta^* := V^* \cap \Delta$. Then (i) every vertex in V_Δ^* has level at most $6c_0k$, and (ii) $UH(V_\Delta^*) \cap \Delta$ is exactly $UH(V^*) \cap \Delta$ and thus covers $L_{\leq 2k}(H) \cap \Delta$. Unfortunately we do not have good enough bounds on the number of extra vertices added to V^* .
- To reduce the size of V^* , we *replace* the chain S_w of vertices at every wall w of a supercell with a *sparser* set S'_w of vertices defined as follows. (See Figure 2.) Let H_w be the set of planes in H intersecting w , and let S'_w be a planar $(6c_0k, 6c_0^2k)$ -shallow cutting provided by Fact 5 for the intersection of H_w with the vertical plane through w (a set of lines). Let σ_w be the slab delimited by the two vertical lines through the two subwalls of w . We clip $UH(S'_w)$ to σ_w , add the two new vertices to S'_w , and remove any vertices outside σ_w . Observe that S'_w is covered by w (and thus by $VD(V_{IN})$) by picking a constant $B \geq 12c_0^2$, because every vertex in S'_w (including the two extra vertices added) has level in H_w at most $12c_0^2k$, and w covers $L_{\leq Bk}(H) \cap \sigma_w = L_{\leq Bk}(H_w) \cap \sigma_w$. Then (i) every vertex in V_Δ^* now has level (in H) at most $12c_0^2k$, and (ii) $UH(V_\Delta^*)$ can only increase in its coverage, because each old set S_w is contained in $L_{\leq 6c_0k}(H) \cap \sigma_w$ and the new concave chain $UH(S'_w)$ covers $L_{\leq 6c_0k}(H) \cap \sigma_w$. For each wall w , the size of S'_w is at most $c'_0 \frac{|H_w|}{6c_0k} \leq c'_0 \frac{2CBk}{6c_0k} = \frac{c'_0CB}{3c_0}$. The number of walls of the supercells is at most $a'_0 \frac{|V_{IN}|}{\sqrt{t}} \leq \frac{a'_0C'}{B\sqrt{t}} \frac{n}{k}$. Thus, the total number of extra vertices added to V^* (two copies included) is at most $\frac{2a'_0c'_0CC'}{3c_0\sqrt{t}} \frac{n}{k}$, and the size of V^* is now at most $(\frac{c'_0}{2} + \frac{2a'_0c'_0CC'}{3c_0\sqrt{t}}) \frac{n}{k}$.
- For every supercell Δ of $VD(V_{IN})$, we *snap* the vertices in V_Δ^* to vertices of Γ_Δ i.e., we replace every vertex $v \in V_\Delta^*$ with the four vertices of the cell in Γ_Δ containing v . This makes $V_\Delta^* \subseteq \Lambda_\Delta$. Then (i) every vertex in V_Δ^* now has level at most $12c_0^2k + \varepsilon|H_\Delta| \leq 12c_0^2k + \varepsilon(3tCBk) = (12c_0^2 + 1)k$, and (ii) $UH(V_\Delta^*)$ can only increase in its coverage. The size of V^* quadruples to at most $(2c'_0 + \frac{8a'_0c'_0CC'}{3c_0\sqrt{t}}) \frac{n}{k}$.

Then Line 1.2 guarantees that $|V_\Delta| \leq |V_\Delta^*|$ by setting the constant $C := 12c_0^2 + 1$. Therefore, totalling over all cells in $\text{VD}(V_{\text{IN}})$, we have $|V_{\text{OUT}}| \leq |V^*| \leq (2c_0' + \frac{8a_0'c_0'CC'}{3c_0\sqrt{t}})\frac{n}{k} \leq C'\frac{n}{k}$ as desired, by setting the constant $C' := \frac{2c_0'}{1 - \frac{8a_0'c_0'CC'}{3c_0\sqrt{t}}}$ and picking any constant $t > (\frac{8a_0'c_0'CC'}{3c_0})^2$. ◀

As in Section 3, it follows that:

► **Corollary 10.** *For a set H of n planes in \mathbb{R}^3 , a parameter $k \in [1, n]$, and some suitable constants B, C, C' , we can compute a $(B^i k, CB^i k)$ -shallow cutting of size at most $C'\frac{n}{B^i k}$, along with its conflict lists, for all $i = 0, 1, \dots, \log_B \frac{n}{k}$ in $O(n \log \frac{n}{k})$ total deterministic time. In particular, we can compute a (k, Ck) -shallow cutting of size $O(\frac{n}{k})$ in the stated time.*

5 Final Remarks

We remark that concentrating on the $k = \Theta(\frac{n}{r})$ case is indeed without loss of generality—our algorithms can be easily applied to construct k -shallow $\frac{1}{r}$ -cuttings for any k and r . Matoušek [25] proved the existence of such cuttings of size $O(r^{\lfloor d/2 \rfloor} (\frac{kr}{n} + 1)^{\lceil d/2 \rceil})$. We can construct cuttings of this size with the following time bounds for $d \in \{2, 3\}$, which are optimal if we are required to output all conflict lists (since the worst-case total size is $\Theta(r^{\lfloor d/2 \rfloor} (\frac{kr}{n} + 1)^{\lceil d/2 \rceil} \frac{n}{r})$):

► **Corollary 11.** *For a set H of n lines in \mathbb{R}^2 and parameters $k, r \in [1, n]$, we can compute a k -shallow $\frac{1}{r}$ -shallow cutting of size $O(r(\frac{kr}{n} + 1))$, along with its conflict lists, in $O(n \log r + r(\frac{kr}{n} + 1)\frac{n}{r})$ deterministic time.*

For a set H of n planes in \mathbb{R}^3 and parameters $k, r \in [1, n]$, we can compute a k -shallow $\frac{1}{r}$ -shallow cutting of size $O(r(\frac{kr}{n} + 1)^2)$, along with its conflict lists, in $O(n \log r + r(\frac{kr}{n} + 1)^2 \frac{n}{r})$ deterministic time.

Proof. If $k \leq \frac{n}{cr}$ for a suitable constant c , then we can just apply our algorithm to compute a $\frac{n}{cr}$ -shallow $\frac{1}{r}$ -cutting of size $O(r)$ in $O(n \log r)$ deterministic time.

So assume $k > \frac{n}{cr}$. We first apply our algorithm to compute a k -shallow $\frac{ck}{n}$ -cutting of size $O(\frac{n}{k})$ in $O(n \log \frac{n}{k}) = O(n \log r)$ deterministic time. Inside each cell Δ of this cutting, the conflict list H_Δ has size at most ck and we compute a standard $\frac{n/r}{ck}$ -cutting of H_Δ of size $O((\frac{k}{n/r})^d)$ in deterministic time $O(k(\frac{k}{n/r})^{d-1})$ by known results (e.g., Chazelle's algorithm [14], or in the $d = 2$ case, our algorithm from the appendix). This yields a k -shallow $\frac{1}{r}$ -cutting of H of total size $O(\frac{n}{k} \cdot (\frac{k}{n/r})^d)$ in total time $O(n \log r + \frac{n}{k} \cdot k(\frac{k}{n/r})^{d-1})$. The size and time bounds are exactly as stated. ◀

We should mention that despite their conceptual simplicity, our algorithms are not likely to be practical in their present form, because of the huge hidden constant factors.

Our approach of incorporating brute-force search and comparing the cost of our solution to that of an optimal solution was inspired by approximation algorithms. An interesting problem is to actually find PTASes to compute the minimum-size (shallow or standard) cutting, or compute cuttings with constant factors approaching the worst-case optimum [26], with comparable running time.

The optimality of the $O(n \log r)$ time bound assumes a comparison-based model, but it remains to be seen if there are faster algorithms to compute a single shallow cutting in the word RAM model for integer input [13].

Generalization of our shallow cutting algorithms to higher dimensions is also open; odd dimensions appear particularly challenging.

References

- 1 Peyman Afshani and Timothy M. Chan. Optimal halfspace range reporting in three dimensions. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 180–186. SIAM, 2009.
- 2 Peyman Afshani, Timothy M. Chan, and Konstantinos Tsakalidis. Deterministic rectangle enclosure and offline dominance reporting on the RAM. In *Proceedings of the Forty-First International Colloquium on Automata, Languages, and Programming, Part I*, ICALP '14, pages 77–88, 2014.
- 3 Peyman Afshani and Konstantinos Tsakalidis. Optimal deterministic shallow cuttings for 3d dominance ranges. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pages 1389–1398. SIAM, 2014.
- 4 Pankaj K. Agarwal. Partitioning arrangements of lines I: An efficient deterministic algorithm. *Discrete & Computational Geometry*, 5(1):449–483, 1990.
- 5 Pankaj K. Agarwal. *Intersection and Decomposition Algorithms for Planar Arrangements*. Cambridge University Press, New York, NY, USA, 1991.
- 6 Pankaj K. Agarwal, Boris Aronov, Timothy M. Chan, and Micha Sharir. On levels in arrangements of lines, segments, planes, and triangles. *Discrete & Computational Geometry*, 19(3):315–331, 1998.
- 7 Gerth Stølting Brodal and Riko Jacob. Dynamic planar convex hull with optimal query time. In *Proceedings of the Seventh Scandinavian Workshop on Algorithm Theory*, SWAT '00, pages 57–70, 2000.
- 8 Gerth Stølting Brodal and Riko Jacob. Dynamic planar convex hull. In *Proceedings of the Forty-Third Symposium on Foundations of Computer Science*, FOCS '02, pages 617–626. IEEE, 2002. Current draft of full paper at <https://pwgrp1.inf.ethz.ch/Current/DPCH/Journal/topdown.pdf>.
- 9 Timothy M. Chan. Random sampling, halfspace range reporting, and construction of ($\leq k$)-levels in three dimensions. *SIAM Journal on Computing*, 30(2):561–575, 2000.
- 10 Timothy M. Chan. Low-dimensional linear programming with violations. *SIAM Journal on Computing*, 34(4):879–893, April 2005.
- 11 Timothy M. Chan. Three problems about dynamic convex hulls. *International Journal of Computational Geometry & Applications*, 22(04):341–364, 2012.
- 12 Timothy M. Chan, Kasper Green Larsen, and Mihai Pătraşcu. Orthogonal range searching on the RAM, revisited. In *Proceedings of the Twenty-Seventh Symposium on Computational Geometry*, SOCG '11, pages 1–10. ACM, 2011.
- 13 Timothy M. Chan and Mihai Pătraşcu. Transdichotomous results in computational geometry, I: point location in sublogarithmic time. *SIAM J. Comput.*, 39(2):703–729, 2009.
- 14 Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete & Computational Geometry*, 9(1):145–158, 1993.
- 15 Bernard Chazelle and Joel Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10(3):229–249, 1990.
- 16 Kenneth L. Clarkson. New applications of random sampling in computational geometry. *Discrete & Computational Geometry*, 2:195–222, 1987.
- 17 Martin E. Dyer. Linear time algorithms for two- and three-variable linear programs. *SIAM Journal on Computing*, 13(1):31–45, 1984.
- 18 Greg N. Frederickson. Fast algorithms for shortest paths in planar graphs, with applications. *SIAM Journal on Computing*, 16(6):1004–1022, 1987.
- 19 Michael T. Goodrich. Planar separators and parallel polygon triangulation. *Journal of Computer and System Sciences*, 51(3):374–389, 1995.
- 20 Sarel Har-Peled, Haim Kaplan, Micha Sharir, and Shakhar Smorodinsky. Epsilon-nets for halfspaces revisited. *CoRR*, abs/1410.3154, 2014.

- 21 David Haussler and Emo Welzl. ϵ -nets and simplex range queries. *Discrete & Computational Geometry*, 2(1):127–151, 1987.
- 22 Richard J. Lipton and Robert E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.
- 23 Jiří Matoušek. Construction of ϵ -nets. *Discrete & Computational Geometry*, 5(1):427–448, 1990.
- 24 Jiří Matoušek. Cutting hyperplane arrangements. *Discrete & Computational Geometry*, 6(1):385–406, 1991.
- 25 Jiří Matoušek. Reporting points in halfspaces. *Computational Geometry*, 2(3):169–186, 1992.
- 26 Jiří Matoušek. On constants for cuttings in the plane. *Discrete & Computational Geometry*, 20(4):427–448, 1998.
- 27 Nimrod Megiddo. Linear-time algorithms for linear programming in R^3 and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.
- 28 Nimrod Megiddo. Linear programming in linear time when the dimension is fixed. *Journal of the ACM*, 31(1):114–127, 1984.
- 29 Edgar A. Ramos. On range reporting, ray shooting and k -level construction. In *Proceedings of the Fifteenth Annual Symposium on Computational Geometry*, SoCG '99, pages 390–399. ACM, 1999.
- 30 Edgar A. Ramos. Deterministic algorithms for 3-d diameter and some 2-d lower envelopes. In *Proceedings of the Sixteenth Annual Symposium on Computational Geometry*, SoCG '00, pages 290–299. ACM, 2000.

A Appendix: A 2-d Standard Cutting Algorithm

In this appendix, we describe how our ideas can be used to rederive known results by Matoušek [24] and Chazelle [14] for standard cuttings in 2-d.

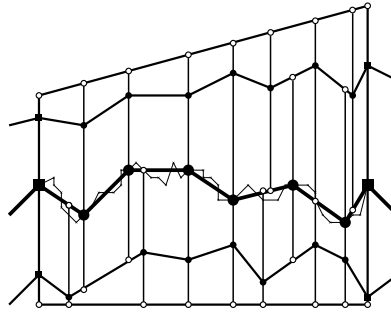
As before, it will be more convenient to work with the parameter $K := \frac{n}{r}$ instead of r . The target $O(nr)$ time bound becomes $O(\frac{n^2}{K})$. Our cuttings will be the vertical decompositions of noncrossing line segments. Given a set S of noncrossing line segments inside a cell Δ , we define the *vertical decomposition* $VD(S)$ to be the subdivision into trapezoids, obtained by drawing a vertical upward/downward ray at each vertex till the ray hits another segment. We define $VD_{\Delta}(S)$ to be $VD(S)$ clipped inside a given cell Δ .

► **Theorem 12.** *For a set H of n lines in \mathbb{R}^2 , a parameter $K \in [1, n]$ and suitable constants B, C , given a $\frac{BK}{n}$ -cutting T_{IN} for H of size at most $C(\frac{n}{BK})^2$ along with its conflict lists, we can compute a $\frac{K}{n}$ -cutting T_{OUT} for H of size at most $C(\frac{n}{K})^2$ along with its conflict lists in $O(\frac{n^2}{K})$ deterministic time.*

Proof.

Algorithm. Let ϵ be a constant to be set later.

1. For each cell $\Delta \in T_{IN}$:
 - 1.1. Compute by Fact 4 an ϵ -cutting Γ_{Δ} for H_{Δ} of $O(1)$ size, where the cells are clipped (and re-triangulated) to lie within Δ . Further refine the cells of Γ_{Δ} by drawing a vertical line at every vertex of Γ_{Δ} . Let Λ_{Δ} be the set of vertices that define the cells of (the refined) Γ_{Δ} .



■ **Figure 3** *Simplifying a level.*

- 1.2. Compute by brute force the *smallest set* of noncrossing line segments S_Δ , whose endpoints are from Λ_Δ , such that each trapezoid in $\text{VD}_\Delta(S_\Delta)$ intersects at most K lines of H_Δ .
2. Return $T_{\text{OUT}} := \bigcup_{\Delta \in T_{\text{IN}}} \text{VD}_\Delta(S_\Delta)$ and all its conflict lists.

Complexity. Line 1.1 takes time linear in $|H_\Delta|$ by Fact 4, for a total of $\sum_{\Delta \in T_{\text{IN}}} O(|H_\Delta|) = O(C(\frac{n}{BK})^2 \cdot BK) = O(\frac{n^2}{K})$ time. For Line 1.2, we probe all possible sets S_Δ of line segments with endpoints from Λ_Δ . Since Γ_Δ and Λ_Δ have $O(1)$ size, there are “only” $O(1)$ sets to test and each set can be tested in $O(|H_\Delta|)$ time. Thus, the whole step takes time linear in $|H_\Delta|$, which again totals to $O(\frac{n^2}{K})$.

Correctness. Clearly T_{OUT} is a $\frac{K}{n}$ -cutting for H . To bound the size of T_{OUT} , we compare it against some optimal $\frac{K}{n}$ -cutting for H of size $O((\frac{n}{K})^2)$, specifically, the cutting produced by Matoušek’s construction [23] using levels. (We would have preferred a cleaner proof that compares T_{OUT} against an arbitrary optimal-size cutting, like in our earlier proofs, but were unable to make the details work.) We adapt his construction to incorporate our earlier ideas of *chopping* and *snapping*.

- We first pick a random index $j_0 \in [1, 0.5K]$. For each $j \equiv j_0 \pmod{0.5K}$, consider the j -level (the set of points on the arrangement with level j), which is an x -monotone chain. Since the arrangement has $O(n^2)$ vertices in total, the expected total number of vertices in these chains is $O(\frac{n^2}{K})$.
 - We *chop* these chains into subchains at the boundaries of the cells of T_{IN} . Since the total number of vertices along cell boundaries is $O(C(\frac{n}{BK})^2 \cdot BK) = O(\frac{C}{B})\frac{n^2}{K}$, the expected total number of subchains created is at most $O(\frac{C}{B})\frac{n^2}{K^2}$.
 - We *simplify* each subchain by selecting every $0.1k$ -th vertex from the subchain and forming a shorter x -monotone chain through these vertices, while keeping the start and end vertex. (See Figure 3.) Note that the levels of points on a simplified subchain can deviate from the original level by at most $\pm 0.1K$. Let S^* be the set of the edges of the simplified subchains. Since the size of a simplified subchain is at most one plus $\frac{1}{0.1K}$ -th the original size of the subchain, the expected size of S^* is at most $O(1 + \frac{C}{B})(\frac{n}{K})^2$. We pick a j_0 so that the size of S^* is at most its expectation.
- For each boundary edge of the cells of T_{IN} , we subdivide it by selecting every $0.1K$ -th vertex of the arrangement lying on the edge. We add two copies of the resulting edges to S^* (one assigned to each of the two incident cells). Since the number of extra edges added is $O(C(\frac{n}{BK})^2 \cdot \frac{BK}{0.1K}) = O(\frac{C}{B})(\frac{n}{K})^2$, the size of S^* remains at most $O(1 + \frac{C}{B})(\frac{n}{K})^2$.

Let $S_\Delta^* := S^* \cap \Delta$. We claim that each trapezoid in $\text{VD}_\Delta(S_\Delta^*)$ intersects at most $0.9K$ lines. This follows because the left side of the trapezoid intersects at most $0.5K + 0.1K + 0.1K$ lines, and the top or bottom side intersects at most $0.1K$ lines.

- For every cell $\Delta \in T_{\text{IN}}$, we *snap* the endpoints of the segments in S_Δ^* to the vertices of Γ_Δ , i.e., we replace each such endpoint v with the rightmost vertex of the cell in Γ_Δ containing v . For each endpoint v that lie on a boundary edge of Δ , we snap it to a vertex of Γ_Δ on that edge.

Note that the x -order of the vertices in S_Δ^* is preserved after snapping, because we have refined Γ_Δ with extra vertical lines. Thus, the simplified subchains inside Δ of a common chain remain x -monotone and noncrossing. Furthermore, two simplified subchains of two different chains remain noncrossing for a sufficiently small ε , since the two chains have levels at least $0.5K$ apart, simplification changes levels by at most $0.1K$, and snapping changes levels by at most $O(\varepsilon BK)$. Thus, S_Δ^* remains noncrossing.

By modifying the previous argument, we see that each trapezoid in $\text{VD}_\Delta(S_\Delta^*)$ intersects at most $0.9K + O(\varepsilon BK)$ lines; the number can be made at most K for a sufficiently small constant ε .

Then Line 1.2 guarantees that $|S_\Delta| \leq |S_\Delta^*|$. Therefore, $|T_{\text{OUT}}| \leq O(|S^*|) \leq O(1 + \frac{C}{B})(\frac{n}{K})^2$, which can be made at most $C(\frac{n}{K})^2$ as desired, by choosing a sufficiently large constant B . (Note that in the entire correctness proof, constants hidden in the O notation are universal constants.) ◀

► **Corollary 13.** *For a set H of n lines in \mathbb{R}^2 , a parameter $K \in [1, n]$, and some suitable constants B, C , we can compute a $\frac{B^i K}{n}$ -cutting of size at most $C(\frac{n}{B^i K})^2$ for all $i = 0, 1, \dots, \log_B \frac{n}{K}$, along with its conflict lists, in $O(\frac{n^2}{K})$ total deterministic time. In particular, we can compute a $\frac{1}{r}$ -cutting of size $O(r^2)$ in $O(nr)$ deterministic time.*

Proof. The recurrence $T(n, K) = T(n, BK) + O((\frac{n}{K})^2)$, with the trivial base case $T(n, n) = O(n)$, solves to $T(n, K) = O(\sum_{i=0}^{\log_B \frac{n}{K}} (\frac{n}{B^i K})^2) = O((\frac{n}{K})^2)$. ◀

Our above algorithm can be viewed as a reinterpretation of Chazelle's algorithm [14], where ε -approximations and sparse ε -nets are replaced by a brute-force component that is more self-contained to describe. Our analysis only works in 2-d, however; Chazelle's approach is still more powerful.

A Simpler Linear-Time Algorithm for Intersecting Two Convex Polyhedra in Three Dimensions

Timothy M. Chan*

Cheriton School of Computer Science, University of Waterloo, Canada
tmchan@uwaterloo.ca

Abstract

Chazelle [FOCS'89] gave a linear-time algorithm to compute the intersection of two convex polyhedra in three dimensions. We present a simpler algorithm to do the same.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases convex polyhedra, intersection, Dobkin–Kirkpatrick hierarchy

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.733

1 Introduction

This note concerns the following problem: given two convex polyhedra of size $O(n)$ in 3-d, compute their intersection. Equivalently, the dual problem is to compute the convex hull of the two convex polyhedra, i.e., *merge* two convex hulls.

This is one of the most basic computational problems about convex polyhedra. Algorithms for the problem have been used as subroutines to solve many other problems in computational geometry (see [2] for just one example).

In the 70s, Preparata and Hong [13] observed that two linearly separated convex hulls in 3-d can be merged in linear time. (Earlier Shamos and Hoey [14] observed the same for the special case of two linearly separated Delaunay triangulations in 2-d, and later Kirkpatrick [9] showed how to merge two arbitrary Delaunay triangulations in 2-d in linear time.) The general problem was eventually settled a decade later when Chazelle [4] announced a linear-time algorithm for intersecting/merging two arbitrary convex polyhedra in 3-d.

Chazelle's algorithm, like many of his other works, is a tour de force. It started with a standard construction of the *Dobkin–Kirkpatrick (DK) hierarchies* [6, 7] of the input polyhedra in both primal and dual space, but incorporated pages of intricate ideas and details. To give a flavor of the overall plan, we only mention that the running time satisfies a recurrence of the form $T(n) = 4T(\delta n) + O(n)$, which indeed solves to $T(n) = O(n)$ for a sufficiently small constant $\delta > 0$.

A thesis by Martin [11] described a simplification of Chazelle's algorithm that avoided switching back and forth with duality, but needed to navigate simultaneously in the DK hierarchies of the insides and outsides of the polyhedra. The details were still lengthy, and the recurrence was “improved” to $T(n) = 2T(\delta n) + O(n)$.

Chazelle's work dated back to a time when the unifying techniques of randomized geometric divide-and-conquer [5, 12] were just starting to flourish. This prompts the question of whether more modern concepts like sampling, conflict lists, etc. might give a simpler explanation for why the problem can be solved in linear time. After all, at a gut level, this

* Part of this work was done during the author's visit to the Hong Kong University of Science and Technology.



textbook problem shouldn't be that hard to solve (although one could say the same for the problem of triangulating a simple polygon [3, 1]!).

In this note, we show that there is indeed a simpler linear-time algorithm for intersecting two convex polyhedra. Our solution ends up not requiring random sampling, but falls back to the DK hierarchy. We only need to navigate in the hierarchies of the outsides, and we don't need to switch between primal and dual space. Furthermore, we get a more usual recurrence $T(n) = T(\delta n) + O(n)$ – in other words, a more conventional way of using DK hierarchies turns out to work after all! There are concrete advantages to having the better recurrence when considering other computational models; for example, the algorithm is more efficiently parallelizable. However, we believe the simplicity of the solution is what is the most valuable aspect of the work.

2 Preliminaries

We begin by computing a point o in the intersection of the two convex polyhedra; this can be done in linear time by 3-d linear programming [8] (known randomized algorithms are particularly simple), or in polylogarithmic time using DK hierarchies [6, 7]. By translation, we may make o the origin. It suffices to compute the part of the intersection in $\{z > 0\}$. By a projective transformation $(x, y, z) \mapsto (x/z, y/z, -1/z)$, we can move o to $(0, 0, -\infty)$ and thus assume that both input polyhedra are unbounded from below, i.e., they are (the regions underneath) lower envelopes of planes. We assume that the planes are in general position, by standard perturbation techniques.

Given a set H of planes, let $\mathcal{P}(H)$ denote the region underneath its lower envelope. We say that H is *nonredundant* if all planes of H participate in the boundary of $\mathcal{P}(H)$. Given $P = \mathcal{P}(H)$, let $\mathcal{T}(P)$ denote a triangulation of P . More precisely, we triangulate each face of P , and for each triangle $v_1v_2v_3$ we take the region underneath $v_1v_2v_3$ (a prism unbounded from below) as a cell of $\mathcal{T}(P)$. For any region Δ , the *conflict list* $H_{|\Delta}$ is the subset of all planes of H intersecting Δ .

A standard approach to computing the lower envelope of H is to pick a random sample H' of H , construct the lower envelope of the conflict list $H_{|\Delta}$ inside Δ for each cell $\Delta \in \mathcal{T}(\mathcal{P}(H'))$, and then glue the results together. Although we will not use randomization, we will adapt similar ideas.

Given $\mathcal{P}(H_1)$ and $\mathcal{P}(H_2)$ for two nonredundant sets H_1 and H_2 of planes, our problem is to compute $P = \mathcal{P}(H_1) \cap \mathcal{P}(H_2)$ (i.e., $P = \mathcal{P}(H_1 \cup H_2)$). In order to allow for a recursive algorithm, we need to strengthen the output requirement and require further information to be reported for each vertex v of P . Our key idea is this. Since v is in the intersection, we know that v is on or below $\mathcal{P}(H_j)$ for each $j \in \{1, 2\}$. Thus, there exist three vertices w_1, w_2, w_3 of $\mathcal{P}(H_j)$ that “witness” this fact, i.e., that have v below¹ $\Delta w_1w_2w_3$. We will require the algorithm to output one such triple for each v and j . It is important that we do not insist $w_1w_2w_3$ be a face of (a triangulated) $\mathcal{P}(H_j)$. Otherwise, one can show that finding such witnesses may require $\Omega(n \log n)$ comparisons in the worst case! Witnesses will make the generation of conflict lists easy; on the other hand, extra work will be required to find witnesses.

To summarize, we will solve the following stronger problem:

Problem: Given $\mathcal{P}(H_1)$ and $\mathcal{P}(H_2)$ for two nonredundant sets H_1 and H_2 of n planes, compute $P = \mathcal{P}(H_1) \cap \mathcal{P}(H_2)$, and for each vertex v of P and each $j \in \{1, 2\}$, report some vertices w_1, w_2, w_3 of $\mathcal{P}(H_j)$, called the *j -witnesses* of v , such that v is below $\Delta w_1w_2w_3$.

¹ Throughout the paper, “below” means “below or is incident to” unless preceded by “strictly”.

3 The Algorithm

We are now ready to give the algorithm outline to solve the problem:

```

Intersect( $\mathcal{P}(H_1), \mathcal{P}(H_2)$ ):
0. if  $H_1$  and  $H_2$  have size below a constant then return answer directly
1. for  $j \in \{1, 2\}$ :
2.   choose an independent set of faces of  $\mathcal{P}(H_j)$ 
3.   let  $I_j$  be the planes defining these faces, and let  $H'_j = H_j \setminus I_j$ 
4.   obtain  $\mathcal{P}(H'_j)$  from  $\mathcal{P}(H_j)$ 
5.  $P' = \mathbf{Intersect}(\mathcal{P}(H'_1), \mathcal{P}(H'_2))$ 
// now compute the intersection  $P$  of  $\mathcal{P}(H_1)$  and  $\mathcal{P}(H_2)$ 
6. for each  $\Delta \in \mathcal{T}(P')$ :
7.   for  $j \in \{1, 2\}$ :
8.     find the conflict list  $H_{j|\Delta}$  by searching in the candidate list
            $C_{j,\Delta} := \{ h \in H_j : h \text{ lies below a } j\text{-witness of a vertex of } \Delta \}$ 
9.     compute the intersection of  $\mathcal{P}(H_{1|\Delta})$  and  $\mathcal{P}(H_{2|\Delta})$  inside  $\Delta$ 
10.  glue all the polyhedra from line 9 to get  $P$ 
// now compute new witnesses for  $P$ 
11. for each  $\Delta \in \mathcal{T}(P)$ :
12.   for  $j \in \{1, 2\}$ :
13.     for each vertex  $v$  of  $P$  inside  $\Delta$ :
14.       find  $j$ -witnesses of  $v$  by searching in the candidate witness list
            $W_{j,\Delta} := \{ \text{vertices } w \text{ of } \mathcal{P}(H_j) : w \text{ is a } j\text{-witness of a vertex of } \Delta \text{ or} \\ w \text{ is on a plane in } I_j \cap C_{j,\Delta} \}$ 
15. return  $P$  with all its witnesses

```

We explain the algorithm in more detail. In line 2, independence means that the chosen faces do not share any edges. By applying a standard linear-time greedy algorithm to a planar graph in the dual, we can always choose an independent set of at least αn faces each with at most c vertices, for some constants α and c ; for example, see Kirkpatrick's well known paper [10], which has $\alpha = 1/24$ and $c = 11$.

Line 4 takes linear time, since the difference of two polyhedra $\mathcal{P}(H_j)$ and $\mathcal{P}(H'_j)$ consists of disjoint constant-size *pockets*, as we are removing an independent set of constant-size faces; each pocket can be constructed from the lower envelope of $O(1)$ planes. (The hierarchy of polyhedra produced from the recursion is commonly referred to as the *Dobkin–Kirkpatrick hierarchy* [6, 7].)

Line 5 contains the main recursive call, where the number of planes in either input set drops to at most $(1 - \alpha)n$.

In line 8, we use witnesses for P' to help generate conflict lists. Any plane h in the conflict list $H_{j|\Delta}$ must lie below one u of the three vertices of Δ . Since u (a vertex of P') lies below $\Delta w'_1 w'_2 w'_3$ for its j -witnesses w'_1, w'_2, w'_3 , it follows that h lies below some w'_i and must indeed be in the candidate list $C_{j,\Delta}$.

There are at most nine j -witnesses for the three vertices of Δ . Each j -witness w'_i (a vertex of $\mathcal{P}(H'_j)$) has at most $O(1)$ planes of H_j below it: namely, its three defining planes, and at most one plane from I_j strictly below it (which we can easily identify after initializing some pointers). Thus, the candidate list $C_{j,\Delta}$ has constant size, and so each conflict list $H_{j|\Delta}$ can be generated in constant time.

Line 9 takes constant time even by a brute-force algorithm. Line 10 then takes linear total time.

We show, with a slightly subtle proof, that in line 14, we can indeed always find j -witnesses from the candidate witness list $W_{j,\Delta}$:

► **Lemma 1.** *For a vertex v of P inside Δ , there exist $w_1, w_2, w_3 \in W_{j,\Delta}$ such that v lies below $\Delta w_1 w_2 w_3$.*

Proof. Let W' be the at most nine j -witnesses of the three vertices of Δ . Then v lies below the upper hull of W' and is thus below the upper hull of some three points $w'_1, w'_2, w'_3 \in W'$ (which are vertices of $\mathcal{P}(H'_j)$). Let Γ be the region underneath $\Delta w'_1 w'_2 w'_3$.

Since v is in $\mathcal{P}(H_j) \cap \Gamma$, there exist three vertices u_1, u_2, u_3 of $\mathcal{P}(H_j) \cap \Gamma$ such that v is below $\Delta u_1 u_2 u_3$. (Note that each u_i may not necessarily be a vertex of $\mathcal{P}(H_j)$ as it could lie on the boundary of Γ .) For each $i \in \{1, 2, 3\}$, we claim that u_i is below the upper hull of $W_{j,\Delta}$:

Case 1: u_i is a vertex of Γ , i.e., $u_i \in \{w'_1, w'_2, w'_3\}$. Then u_i is a vertex of $\mathcal{P}(H'_j)$. Since u_i is in $\mathcal{P}(H_j)$, it follows that u_i is a vertex of $\mathcal{P}(H_j)$. Thus, u_i is in $W_{j,\Delta}$ by definition of $W_{j,\Delta}$, and the claim is trivially true.

Case 2: u_i is not a vertex of Γ . Then u_i must be defined by at least one plane $h \in H_j$ that intersects the interior of Γ . This plane h is strictly below at least one of w'_1, w'_2, w'_3 and so must be a member of I_j and also a member of $C_{j,\Delta}$. Now, u_i lies in the face of $\mathcal{P}(H_j)$ defined by h ; all the vertices of this face are in $W_{j,\Delta}$ by definition of $W_{j,\Delta}$, and the claim is again true.

Since v is below $\Delta u_1 u_2 u_3$, it follows that v is below the upper hull of $W_{j,\Delta}$ and is thus below $\Delta w_1 w_2 w_3$ for some three vertices $w_1, w_2, w_3 \in W_{j,\Delta}$. ◀

The candidate witness list $W_{j,\Delta}$ has constant size, since there are at most nine j -witnesses for the three vertices of Δ , each plane in I_j contains at most c vertices, and there are $O(1)$ planes in $C_{j,\Delta}$. So, line 13 can be done in constant time by brute force. The entire loop in lines 11–14 then takes linear total time.

The overall running time of the algorithm satisfies the recurrence $T(n) = T((1 - \alpha)n) + O(n)$, which solves to $T(n) = O(n)$.

4 Remarks

An alternative, slightly slower algorithm. There is a more “standard” algorithm based on sampling, without using witnesses, that gives almost linear $n2^{O(\log^* n)}$ expected time. For the readers who are familiar with randomization techniques [5, 12] and enjoy comparisons of different approaches, we briefly sketch the alternative:

First consider a multiset version of H_j where the multiplicity $w_j(h)$ (the *weight*) of each plane $h \in H_j$ is the size of the face of $\mathcal{P}(H_j)$ defined by h . The multiset still has $O(n)$ size. We draw a random sample H'_j of the multiset of size $r = O(n/\log n)$. We construct $\mathcal{P}(H'_1)$, $\mathcal{P}(H'_2)$, and their intersection P' by an $O(r \log r)$ -time algorithm, which takes $O(n)$ time.

For each vertex v of $\mathcal{P}(H'_j)$, we can construct its conflict list $H_{j|v}$ (the list of all planes of H_j below v) as follows: first find an initial plane of H_j below v by a point location query in the xy -projection of $\mathcal{P}(H_j)$; then find all planes of H_j below v by a graph search over the faces of $\mathcal{P}(H_j)$. This works because the planes below v correspond to the faces visible to v , which are connected in the boundary of $\mathcal{P}(H_j)$ (assuming that H_j is nonredundant). We can preprocess in linear time for point location in $O(\log n)$ time [10], so the $O(r)$ point location

queries cost $O(n)$ total time. The graph search takes time proportional to the weight of $H_{j|v}$. The total time over all v is $O(r \cdot n/r) = O(n)$ in expectation, by Clarkson and Shor's analysis [5].

Next, for each vertex v of P' , we can compute its conflict list $H_{j|v}$ as follows: first find a cell $\Delta \in \mathcal{T}(\mathcal{P}(H'_j))$ containing v by a point location query in the xy -projection of $\mathcal{T}(\mathcal{P}(H'_j))$; then search in the conflict lists of the three vertices of Δ (which are vertices of $\mathcal{P}(H'_j)$) found in the preceding paragraph. The $O(r)$ point location queries again cost $O(n)$ total time. So, this step again takes $O(n)$ expected total time.

For each cell $\Delta \in \mathcal{T}(P')$, we can now generate the conflict list $H_{j|\Delta}$ from the conflict lists of the three vertices of Δ (which are vertices of P') found in the preceding paragraph. We then recursively compute the intersection of $\mathcal{P}(H_{1|\Delta})$ and $\mathcal{P}(H_{2|\Delta})$ inside Δ , and glue the polyhedra together.

By Clarkson and Shor's analysis [5], the total expected running time satisfies the recurrence $T(n) = \sum_i T(n_i) + O(n)$ where $\max_i n_i = O((n/r) \log n) = O(\log^2 n)$ with high probability and $\sum_i n_i$ has expected value $O(r \cdot n/r) = O(n)$. With $O(\log^* n)$ iterations, this yields an expected time bound of $n2^{O(\log^* n)}$.

An open problem. An interesting question is whether we can similarly merge lower envelopes of *pseudo-planes* in 3-d in linear time, under an appropriate definition of pseudo-planes where three pseudo-planes may intersect in at most one point. This would have applications to merging two additively weighted Voronoi diagrams in 2-d, for instance. Our concept of j -witnesses unfortunately doesn't seem immediately generalizable, although the alternative $n2^{O(\log^* n)}$ -time randomized algorithm can be adapted at least for the case of 2-d additively weighted Voronoi diagrams.

Acknowledgement. The author thanks Stefan Langerman for discussion on these problems.

References

- 1 Nancy M. Amato, Michael T. Goodrich, and Edgar A. Ramos. A randomized algorithm for triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 26(2):245–265, 2001.
- 2 Timothy M. Chan. Deterministic algorithms for 2-d convex programming and 3-d online linear programming. *Journal of Algorithms*, 27(1):147–166, 1998.
- 3 Bernard Chazelle. Triangulating a simple polygon in linear time. *Discrete and Computational Geometry*, 6:485–524, 1991.
- 4 Bernard Chazelle. An optimal algorithm for intersecting three-dimensional convex polyhedra. *SIAM Journal on Computing*, 21(4):671–696, 1992.
- 5 Kenneth L. Clarkson and Peter W. Shor. Application of random sampling in computational geometry, II. *Discrete and Computational Geometry*, 4:387–421, 1989.
- 6 David P. Dobkin and David G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *Journal of Algorithms*, 6(3):381–392, 1985.
- 7 David P. Dobkin and David G. Kirkpatrick. Determining the separation of preprocessed polyhedra—A unified approach. In *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, pages 400–413, 1990.
- 8 Martin E. Dyer, Nimrod Megiddo, and Emo Welzl. Linear programming. In Jacob E. Goodman and Joseph O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, chapter 45. CRC Press, second edition, 2004.
- 9 David G. Kirkpatrick. Efficient computation of continuous skeletons. In *Proceedings of the 20th Annual Symposium on Foundations of Computer Science*, pages 18–27, 1979.

- 10 David G. Kirkpatrick. Optimal search in planar subdivisions. *SIAM Journal on Computing*, 12(1):28–35, 1983.
- 11 Andrew K. Martin. A simple primal algorithm for intersecting 3-polyhedra in linear time. Master's thesis, Department of Computer Science, University of British Columbia, 1991. <https://circle.ubc.ca/handle/2429/30114> or <http://www.cs.ubc.ca/cgi-bin/tr/1991/TR-91-16>.
- 12 Ketan Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1993.
- 13 Franco P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, 1977.
- 14 Michael Ian Shamos and Dan Hoey. Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, pages 151–162, 1975.

Approximability of the Discrete Fréchet Distance

Karl Bringmann^{*1} and Wolfgang Mulzer^{†2}

1 Institute of Theoretical Computer Science, ETH Zurich, Switzerland
karlb@inf.ethz.ch

2 Institut für Informatik, Freie Universität Berlin, Germany
mulzer@inf.fu-berlin.de

Abstract

The Fréchet distance is a popular and widespread distance measure for point sequences and for curves. About two years ago, Agarwal et al [SIAM J. Comput. 2014] presented a new (mildly) subquadratic algorithm for the discrete version of the problem. This spawned a flurry of activity that has led to several new algorithms and lower bounds.

In this paper, we study the approximability of the discrete Fréchet distance. Building on a recent result by Bringmann [FOCS 2014], we present a new conditional lower bound that strongly subquadratic algorithms for the discrete Fréchet distance are unlikely to exist, even in the *one-dimensional* case and even if the solution may be approximated up to a factor of 1.399.

This raises the question of how well we can approximate the Fréchet distance (of two given d -dimensional point sequences of length n) in strongly subquadratic time. Previously, no general results were known. We present the first such algorithm by analysing the approximation ratio of a simple, linear-time greedy algorithm to be $2^{\Theta(n)}$. Moreover, we design an α -approximation algorithm that runs in time $O(n \log n + n^2/\alpha)$, for any $\alpha \in [1, n]$. Hence, an n^ε -approximation of the Fréchet distance can be computed in strongly subquadratic time, for any $\varepsilon > 0$.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems – Geometrical problems and computations

Keywords and phrases Fréchet distance, approximation, lower bounds, Strong Exponential Time Hypothesis

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.739

1 Introduction

Let P and Q be two polygonal curves with n vertices each. The *Fréchet distance* provides a meaningful way to define a distance between P and Q that overcomes some of the shortcomings of the classic Hausdorff distance [6]. Since its introduction to the computational geometry community by Alt and Godau [6], the concept of Fréchet distance has proven extremely useful and has found numerous applications (see [4, 6, 7, 8, 9, 10] and the references therein).

The Fréchet distance has two classic variants: *continuous* and *discrete* [6, 12]. In this paper, we focus on the discrete variant. In this case, the Fréchet distance between two sequences P, Q of n points in d dimensions is defined as follows: imagine two frogs traversing the sequences P and Q , respectively. In each time step, a frog can jump to the next vertex along its sequence, or it can stay where it is. The discrete Fréchet distance is the minimal length of a leash required to connect the two frogs while they traverse the two sequences from start to finish.

* Supported by an ETH Zurich Postdoctoral Fellowship.

† Supported in part by DFG Grants MU 3501/1 and MU 3501/2.



The original algorithm for the continuous Fréchet distance by Alt and Godau has running time $O(n^2 \log n)$ [6]; while the algorithm for the discrete Fréchet distance by Eiter and Mannila needs time $O(n^2)$ [12]. These algorithms have remained the state of the art until very recently: in 2013, Agarwal et al [4] presented a slightly subquadratic algorithm for the discrete Fréchet distance. Building on their work, Buchin et al [9] managed to find a slightly improved algorithm for the continuous Fréchet distance a year later. At the time, Buchin et al thought that their result provides evidence that computing the Fréchet distance may not be 3SUM-hard [13], as had previously been conjectured by Alt [5]. Even though a recent result by Grønlund and Pettie [15], showing that 3SUM has subquadratic decision trees, casts new doubt on the connection between 3SUM and the Fréchet distance, the conclusions of Buchin et al motivated Bringmann [7] to look for other explanations for the apparent difficulty of the Fréchet distance.

He found a possible explanation in the *Strong Exponential Time Hypothesis* (SETH) [16, 17], which roughly speaking asserts that satisfiability cannot be decided in time¹ $O^*((2 - \varepsilon)^n)$ for any $\varepsilon > 0$ (see Section 2 for details). Since exhaustive search takes time $O^*(2^n)$ and since the fastest known algorithms are only slightly faster than that, SETH is a reasonable assumption that formalizes an algorithmic barrier. It has been shown that SETH can be used to prove conditional lower bounds even for polynomial time problems [1, 2, 18, 20]. In this line of research, Bringmann [7] showed, among other things, that there are no strongly subquadratic algorithms for the Fréchet distance unless SETH fails. Here, *strongly subquadratic* means any running time of the form $O(n^{2-\varepsilon})$, for constant $\varepsilon > 0$. Bringmann's lower bound works for two-dimensional curves and both classic variants of the Fréchet distance. Thus, it is unlikely that the algorithms by Agarwal et al and Buchin et al can be improved significantly, unless a major algorithmic breakthrough occurs.

1.1 Our Contributions

In this extended abstract we focus on the discrete Fréchet distance. In Section 6, we will discuss how far our results carry over to the continuous version. Our main results are as follows.

Conditional Lower Bound. We strengthen the result of Bringmann [7] by showing that even in the one-dimensional case computing the Fréchet distance remains hard. More precisely, we show that any 1.399-approximation algorithm in strongly subquadratic time for the one-dimensional discrete Fréchet distance violates the Strong Exponential Time Hypothesis. Previously, Bringmann [7] had shown that no strongly subquadratic algorithm approximates the two-dimensional Fréchet distance by a factor of 1.001, unless SETH fails.

One can embed any one-dimensional sequence into the two-dimensional plane by fixing some $\varepsilon > 0$ and by setting the y -coordinate of the i -th point of the sequence to $i \cdot \varepsilon$. For sufficiently small ε , this embedding roughly preserves the Fréchet distance. Thus, unless SETH fails, there is also no strongly subquadratic 1.399-approximation for the discrete Fréchet distance on (1) two-dimensional curves without self-intersections, and (2) two-dimensional x -monotone curves (also called time-series). These interesting special cases had been open.

Approximation: Greedy Algorithm. A simple greedy algorithm for the discrete Fréchet distance goes as follows: in every step, make the move that minimizes the current distance,

¹ The notation $O^*(\cdot)$ hides polynomial factors in the number of variables n and the number of clauses m .

where a “move” is a step in either one sequence or in both of them. This algorithm has a straightforward linear time implementation. We analyze the approximation ratio of the greedy algorithm, and we show that, given two sequences of n points in d dimensions, the maximal distance attained by the greedy algorithm is a $2^{\Theta(n)}$ -approximation for their discrete Fréchet distance. We emphasize that this approximation ratio is *bounded*, depending only on n , but not the coordinates of the vertices. This is surprising, since so far no bounded approximation algorithm that runs in strongly subquadratic time was known at all. Moreover, although an approximation ratio of $2^{\Theta(n)}$ is huge, the greedy algorithm is the best *linear time* approximation algorithm that we could come up with.

Approximation: Improved Algorithm. For the case that slightly more than linear time is acceptable, we provide a much better approximation algorithm: given two sequences P and Q of n points in d dimensions, we show how to find an α -approximation of the discrete Fréchet distance between P and Q in time $O(n \log n + n^2/\alpha)$, for any $1 \leq \alpha \leq n$. In particular, this yields an $n/\log n$ -approximation in time $O(n \log n)$, and an n^ε -approximation in strongly subquadratic time for any $\varepsilon > 0$. We leave it open whether these approximation ratios can be improved.

2 Preliminaries and Definitions

We call an algorithm an α -approximation for the Fréchet distance if, given curves P, Q , it returns a number that is at least the Fréchet distance of P, Q and at most α times the Fréchet distance of P, Q .

2.1 Discrete Fréchet Distance

Since we focus on the discrete Fréchet distance throughout, we will sometimes omit the term “discrete”. Let $P = \langle p_1, \dots, p_n \rangle$ and $Q = \langle q_1, \dots, q_n \rangle$ be two sequences of n points in d dimensions. A *traversal* β of P and Q is a sequence of pairs in $(p, q) \in P \times Q$ such that (i) the traversal β begins with the pair (p_1, q_1) and ends with the pair (p_n, q_n) ; and (ii) the pair $(p_i, q_j) \in \beta$ can be followed only by one of (p_{i+1}, q_j) , (p_i, q_{j+1}) , or (p_{i+1}, q_{j+1}) . We call β *simultaneous*, if it only makes steps of the third kind, i.e., if in each step β advances in both P and Q . We define the *distance* of the traversal β as $\delta(\beta) := \max_{(p,q) \in \beta} d(p, q)$, where $d(\cdot, \cdot)$ denotes the Euclidean distance. The *discrete Fréchet distance* of P and Q is now defined as $\delta_{\text{dF}}(P, Q) := \min_{\beta} \delta(\beta)$, where β ranges over all traversals of P and Q .

We review a simple $O(n^2 \log n)$ time algorithm to compute $\delta_{\text{dF}}(P, Q)$ that is the starting point of our second approximation algorithm. First, we describe a *decision procedure* that, given a value γ , decides whether $\delta_{\text{dF}}(P, Q) \leq \gamma$. For this, we define the *free-space matrix* F . This is a Boolean $n \times n$ matrix such that for $i, j = 1, \dots, n$, we set $F_{ij} = 1$ if $d(p_i, q_j) \leq \gamma$, and $F_{ij} = 0$, otherwise. Then $\delta_{\text{dF}}(P, Q) \leq \gamma$ if and only if F allows a *monotone traversal from* $(1, 1)$ to (n, n) , i.e., if we can go from entry F_{11} to F_{nn} while only going down, to the right, or diagonally, and while only using 1-entries. This is captured by the *reach matrix* R , which is again an $n \times n$ Boolean matrix. We set $R_{11} = F_{11}$, and for $i, j = 1, \dots, n$, $(i, j) \neq (1, 1)$, we set $R_{ij} = 1$ if $F_{ij} = 1$ and either one of $R_{(i-1)j}$, $R_{i(j-1)}$, or $R_{(i-1)(j-1)}$ equals 1 (we define any entry of the form $R_{(-1)j}$ or $R_{i(-1)}$ to be 0). Otherwise, we set $R_{ij} = 0$. From these definitions, it is straightforward to compute F and R in total time $O(n^2)$. Furthermore, by construction we have $\delta_{\text{dF}}(P, Q) \leq \gamma$ if and only if $R_{nn} = 1$.

With this decision procedure at hand, we can use binary search to compute $\delta_{\text{dF}}(P, Q)$ in total time $O(n^2 \log n)$ by observing that the optimum must be achieved for one of the n^2

distances $d(p_i, q_j)$, for $i, j = 1, \dots, n$. Through a more direct use of dynamic programming, the running time can be reduced to $O(n^2)$ [12].

2.2 Hardness Assumptions

Strong Exponential Time Hypothesis (SETH). As is well-known, the k -SAT problem is as follows: given a CNF-formula Φ over boolean variables x_1, \dots, x_n with clause width k , decide whether there is an assignment of x_1, \dots, x_n that satisfies Φ . Of course, k -SAT is NP-hard, and it is conjectured that no subexponential algorithm for the problem exists [14]. The Strong Exponential Time Hypothesis (SETH) goes a step further and basically states that the exhaustive search running time of $O^*(2^n)$ cannot be improved to $O^*(1.99^n)$ [16, 17].

► **Conjecture 2.1 (SETH).** For no $\varepsilon > 0$, k -SAT has an $O(2^{(1-\varepsilon)n})$ algorithm for all $k \geq 3$.

The fastest known algorithms for k -SAT take time $O(2^{(1-c/k)n})$ for some constant $c > 0$ [19]. Thus, SETH is reasonable and, due to lack of progress in the last decades, can be considered unlikely to fail. It is by now a standard assumption for conditional lower bounds.

Orthogonal Vectors (OV) is the following problem: Given $u_1, \dots, u_N, v_1, \dots, v_N \in \{0, 1\}^D$, decide whether there are $i, j \in [N]$ with $(u_i)_k \cdot (v_j)_k = 0$ for all $1 \leq k \leq D$. Here we denote by u_i the i -th vector and by $(u_i)_k$ its k -th entry. We write $u_i \perp v_j$ if $(u_i)_k \cdot (v_j)_k = 0$ for all $1 \leq k \leq D$. This problem has a trivial $O(DN^2)$ algorithm. The fastest known algorithm runs in time $N^{2-1/O(\log(D/\log N))}$ [3], which is only slightly subquadratic for $D \gg \log N$. It is known that OV has no strongly subquadratic time algorithms unless SETH fails [21]; we present a proof for completeness.

► **Lemma 2.2.** For no $\varepsilon > 0$, OV has an $D^{O(1)} \cdot N^{2-\varepsilon}$ algorithm, unless SETH fails.

Proof. Given a k -SAT formula Φ on variables x_1, \dots, x_n and clauses C_1, \dots, C_m , we build an equivalent OV-instance with $N = 2^{n/2}$ and $D = m$. Denote all possible assignments of true and false to the variables $x_1, \dots, x_{n/2}$ (the first half of the variables) by ϕ_1, \dots, ϕ_N , $N = 2^{n/2}$. For every such assignment ϕ_i we construct a vector u_i where $(u_i)_k$ is 0 if ϕ_i causes C_k to evaluate to true, and 1 otherwise. Similarly, we enumerate the assignments ψ_1, \dots, ψ_N of the variables $x_{n/2+1}, \dots, x_n$ (the second half of the variables), and for every ψ_j we construct a vector v_j where $(v_j)_k$ is 0 if ψ_j causes C_k to evaluate to true, and 1 otherwise. Then, $(u_i)_k \cdot (v_j)_k = 0$ if and only if one of ϕ_i and ψ_j satisfies clause C_k . Thus, we have $(u_i)_k \cdot (v_j)_k = 0$ for all $1 \leq k \leq D$ if and only if (ϕ_i, ψ_j) forms a satisfying assignment of the formula Φ . Hence, we constructed an equivalent OV-instance of the required size. The constructed OV instance can be computed in time $O(DN)$.

It follows that any algorithm for OV with running time $D^{O(1)} \cdot N^{2-\varepsilon}$ gives an algorithm for k -SAT with running time $m^{O(1)} 2^{(1-\varepsilon/2)n}$. Since $m \leq n^k \leq 2^{o(n)}$, this contradicts SETH. ◀

A problem P is *OV-hard* if there is a reduction that transforms an instance I of OV with parameters N, D , to an equivalent instance I' of P of size $n \leq D^{O(1)}N$, in time $D^{O(1)}N^{2-\varepsilon}$ for some $\varepsilon > 0$. A strongly subquadratic algorithm (i.e., $O(n^{2-\varepsilon'})$ for some $\varepsilon' > 0$) for an OV-hard problem P would yield an algorithm for OV with running time $D^{O(1)}N^{2-\min\{\varepsilon, \varepsilon'\}}$. Thus, by Lemma 2.2 an OV-hard problem does not have strongly subquadratic time algorithms unless SETH fails. Most known SETH-based lower bounds for polynomial time problems are actually OV-hardness results; our lower bound in the next section is no exception. Note that OV-hardness is potentially stronger than a SETH-based lower bound, since it is conceivable that SETH fails, but OV still has no strongly subquadratic algorithms.

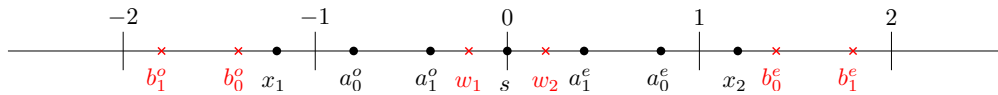


Figure 1 The point set \mathcal{P} constructed in the conditional lower bound.

3 Hardness of Approximation in One Dimension

We prove OV-hardness of the discrete Fréchet distance on one-dimensional curves. By Lemma 2.2, this also yields a SETH-based lower bound.

Let $u_1, \dots, u_N, v_1, \dots, v_N \in \{0, 1\}^D$ be an instance of the Orthogonal Vectors problem. Without loss of generality, we assume that D is even (if not, we duplicate a dimension). We show how to construct two sequences P and Q of $O(DN)$ points in \mathbb{R} in time $O(DN)$ such that there are $i, j \in \{1, \dots, N\}$ with $u_i \perp v_j$ if and only if $\delta_{\text{dF}}(P, Q) \leq 1$. Our sequences P and Q consist of elements from the following set \mathcal{P} of 13 points; see Figure 1.

- $a_0^o = -0.8, a_1^o = -0.4, a_1^e = 0.4, a_0^e = 0.8.$
- $b_1^o = -1.8, b_0^o = -1.4, b_0^e = 1.4, b_1^e = 1.8.$
- $s = 0, x_1 = -1.2, x_2 = 1.2$
- $w_1 = -0.2, w_2 = 0.2.$

We first construct *vector gadgets*. For each $u_i, i \in \{1, \dots, N\}$, we define a sequence A_i of D points from \mathcal{P} as follows: For $1 \leq k \leq D$ let $p \in \{o, e\}$ be the parity of k (odd or even). Then the k th point of A_i is $a_{(u_i)_k}^p$. Similarly, for each v_j , we define a sequence B_j of D points from \mathcal{P} . For B_j , we use the points b_*^p instead of a_*^p . The next claim shows how the vector gadgets encode orthogonality.

► **Claim 3.1.** Fix $i, j \in \{1, \dots, N\}$ and let β be a traversal of (A_i, B_j) . (i) If β is not a simultaneous traversal, then $\delta(\beta) \geq 1.8$; (ii) if β is a simultaneous traversal and $u_i \perp v_j$, then $\delta(\beta) \leq 1$; and (iii) if β is a simultaneous traversal and $u_i \not\perp v_j$, then $\delta(\beta) = 1.4$.

Proof. First, suppose that β is not a simultaneous traversal. Consider the first time when β makes a move on one sequence but not the other. Then, the current points on A_i and B_j lie on different sides of s , which forces $\delta(\beta) \geq \min\{d(a_1^o, b_0^e), d(a_1^e, b_0^o)\} = 1.8$.

Next, suppose that $u_i \perp v_j$. Then, the simultaneous traversal β of A_i and B_j has $\delta(\beta) \leq 1$. Indeed, for each dimension $1 \leq k \leq D$ at least one of $(u_i)_k$ and $(v_j)_k$ is 0. Thus, if we consider the k th point of A_i and the k th point of B_j , both of them lie on the same side of s , and at least one of them is in $\{a_0^o, a_0^e, b_0^o, b_0^e\}$. It follows that the distance between the k th points in β is at most 1, for all k .

Finally, suppose that $(u_i)_k = (v_j)_k = 1$ for some k . Let β be the simultaneous traversal of A_i and B_j , and consider the time when β reaches the k th points of A_i and B_j . These are either $\{a_1^o, b_1^o\}$ or $\{a_1^e, b_1^e\}$, so $\delta(\beta) = \min\{d(a_1^o, b_1^o), d(a_1^e, b_1^e)\} = 1.4$. ◀

Let W be the sequence of $D(N - 1)$ points that alternates between a_0^o and a_0^e , starting with a_0^o . We set

$$P = W \cdot x_1 \cdot s \cdot A_1 \cdot s \cdot A_2 \cdot \dots \cdot s \cdot A_N \cdot s \cdot x_2 \cdot W$$

and

$$Q = w_1 \cdot B_1 \cdot w_2 \cdot w_1 \cdot B_2 \cdot w_2 \cdot \dots \cdot w_1 \cdot B_N \cdot w_2,$$

where \cdot denotes the concatenation of sequences. The idea is to implement an *or-gadget*. If there is a pair of orthogonal vectors, then P and Q should be able to reach the corresponding

vector gadgets and traverse them simultaneously. If there is no such pair, it should not be possible to “cheat”. The purpose of the sequences W and the points w_1 and w_2 is to provide a buffer so that one sequence can wait while the other sequence catches up. The purpose of the points x_1 , x_2 , and s is to synchronize the traversal so that no cheating can occur. The next two claims make this precise. First, we show completeness.

► **Claim 3.2.** *If there are $i, j \in \{1, \dots, N\}$ with $u_i \perp v_j$, then $\delta_{\text{dF}}(P, Q) \leq 1$.*

Proof. Let u_i, v_j be orthogonal. We traverse P and Q as follows:

1. P goes through $D(N - j)$ points of W ; Q stays at w_1 .
2. For $k = 1, \dots, j - 1$: we perform a simultaneous traversal of B_k and the next portion of W starting with a_0^o and the first point on B_k . When the traversal reaches a_0^e and the last point of B_k , P stays at a_0^e while Q goes to w_2 and w_1 . If $k < j - 1$, the traversal continues with a_0^o on P and the first point of B_{k+1} on Q . If $k = j - 1$, we go to Step 3.
3. P proceeds to x_1 and walks until the point s before A_i , Q stays at w_1 before B_j .
4. P and Q go simultaneously through A_i and B_j , until the pair (s, w_2) after A_i and B_j .
5. P continues to x_2 while Q stays at w_2 .
6. For $k = j + 1, \dots, N$: P goes to the next a_0^o on W while Q goes to w_1 . We then perform a simultaneous traversal of B_k and the next portion of W . When the traversal reaches a_0^e and the last point of B_k , P stays at a_0^e while Q continues to w_2 . If $k < N$, the traversal continues with the next iteration, otherwise we go to Step 7.
7. P finishes the traversal of W while Q stays at w_2 .

We use the notation $\max\text{-}d(S, T) := \max_{s \in S, t \in T} d(s, t)$, and $\max\text{-}d(s, T) := \max\text{-}d(\{s\}, T)$, $\max\text{-}d(S, t) := \max\text{-}d(S, \{t\})$. The traversal maintains a maximum distance of 1: for Step 1, this is implied by $\max\text{-}d(\{a_0^o, a_0^e\}, w_1) = 1$. For Step 2, it follows from D being even and from

$$\max\text{-}d(a_0^o, \{b_1^o, b_0^o\}) = \max\text{-}d(a_0^e, \{b_1^e, b_0^e, w_1, w_2\}) = 1.$$

For Step 3, it is because $\max\text{-}d(\{x_1, a_0^o, a_1^o, s, a_1^e, a_0^e\}, w_1) = 1$. For Step 4, we use Claim 3.1 and $d(s, w_2) = 0.2$. In Step 5, it follows from $\max\text{-}d(\{a_0^o, a_1^o, s, a_1^e, a_0^e, x_2\}, w_2) = 1$. In Step 6, we again use that D is even and that

$$\max\text{-}d(a_0^o, \{b_1^o, b_0^o, w_1\}) = \max\text{-}d(a_0^e, \{b_1^e, b_0^e, w_2\}) = 1.$$

Step 7 uses $\max\text{-}d(\{a_0^o, a_0^e\}, w_2) = 1$. ◀

The second claim establishes the soundness of the construction.

► **Claim 3.3.** *If there are no $i, j \in \{1, \dots, N\}$ with $u_i \perp v_j$, then $\delta_{\text{dF}}(P, Q) \geq 1.4$.*

Proof. Let β be a traversal of (P, Q) . Consider the time when β reaches x_1 on P . If Q is not at either w_1 or at a point from $B^o = \{b_0^o, b_1^o\}$, then $\delta(\beta) \geq 1.4$ and we are done. Next, suppose that the current position is in $\{x_1\} \times B^o$. In the next step, β must advance P to s or Q to $\{b_0^e, b_1^e\}$ (or both).² In each case, we get $\delta(\beta) \geq 1.4$. From now on, suppose we reach x_1 in position (x_1, w_1) . After that, P must advance to s , because advancing Q to B^o would take us to a position in $\{x_1\} \times B^o$, implying $\delta(\beta) \geq 1.4$ as we saw above.

Now consider the next step when Q leaves w_1 . Then Q must go to a point from B^o . At this time, P must be at a point from $A^o = \{a_0^o, a_1^o\}$ or we would get $\delta(\beta) \geq 1.4$ (note that P has already passed the point x_1). This point on P belongs to a vector gadget A_i or to the

² Recall that we assumed D to be even.

final gadget W (again because P is already past x_1). In the latter case, we have $\delta(\beta) \geq 1.4$, because in order to reach the final W , P must have gone through x_2 and $d(x_2, w_1) = 1.4$. Thus, P is at a point in A^o in a vector gadget A_i , and Q is at the starting point (from B^o) of a vector gadget B_j .

Now β must alternate simultaneously in P and Q among both sides of s , or again $\delta(\beta) \geq 1.4$, see Claim 3.1. Furthermore, if P does not start in the first point of A_i , then eventually P has to go to s while Q has to go to a point in B^o or stay in $\{b_0^o, b_1^e\}$, giving $\delta(\beta) \geq 1.4$. Thus, we may assume that β simultaneously reached the starting points of A_i and B_j and traverses A_i and B_j simultaneously. By assumption, the vectors u_i, v_j are not orthogonal, so Claim 3.1 gives $\delta(\beta) \geq 1.4$. ◀

► **Theorem 3.4.** *Fix $\alpha \in [1, 1.4)$. Computing an α -approximation of the discrete Fréchet distance in one dimension is OV-hard. In particular, the discrete Fréchet distance in one dimension has no strongly subquadratic α -approximation unless SETH fails.*

Proof. We use Claims 3.2 and 3.3 and the fact that P and Q can be computed in time $O(DN)$ from $u_1, \dots, u_N, v_1, \dots, v_N$: any $O(n^{2-\epsilon})$ time α -approximation for the discrete Fréchet distance would yield an OV algorithm in time $D^{O(1)}N^{2-\epsilon}$, which by Lemma 2.2 contradicts SETH. ◀

► **Remark.** The proofs of Claims 3.2 and 3.3 yield a system of linear inequalities that constrain the points in \mathcal{P} . Using this system, one can see that the inapproximability factor 1.4 in Theorem 3.4 is best possible for our current proof.

4 Approximation Quality of the Greedy Algorithm

In this section we study the following greedy algorithm. Let $P = \langle p_1, \dots, p_n \rangle$ and $Q = \langle q_1, \dots, q_n \rangle$ be two sequences of n points in \mathbb{R}^d . We construct a traversal $\beta_{\text{greedy}} = \beta_{\text{greedy}}(P, Q)$. We begin at (p_1, q_1) . If the current position is (p_i, q_j) , there are at most three possible successor configurations: (p_{i+1}, q_j) , (p_i, q_{j+1}) , and (p_{i+1}, q_{j+1}) (or fewer, if we have already reached the last point from P or Q). Among these, we pick the pair $(p_{i'}, q_{j'})$ that minimizes the distance $d(p_{i'}, q_{j'})$. We stop when we reach (p_n, q_n) . We denote the largest distance taken by the greedy traversal by $\delta_{\text{greedy}}(P, Q) := \delta(\beta_{\text{greedy}}(P, Q))$.

► **Theorem 4.1.** *Let P and Q be two sequences of n points in \mathbb{R}^d . Then, $\delta_{\text{dF}}(P, Q) \leq \delta_{\text{greedy}}(P, Q) \leq 2^{O(n)}\delta_{\text{dF}}(P, Q)$. Both inequalities are tight, i.e., there are polygonal curves P, Q with $\delta_{\text{greedy}}(P, Q) = \delta_{\text{dF}}(P, Q) > 0$ and $\delta_{\text{greedy}}(P, Q) = 2^{\Omega(n)}\delta_{\text{dF}}(P, Q) > 0$, respectively.*

The inequality $\delta_{\text{dF}}(P, Q) \leq \delta_{\text{greedy}}(P, Q)$ follows directly from the definition, since the traversal $\beta_{\text{greedy}}(P, Q)$ is a candidate for an optimal traversal. Furthermore, one can check that if P and Q are increasing one-dimensional sequences, then the greedy traversal is optimal (this is similar to the merge step in mergesort). Thus, there are examples where $\delta_{\text{greedy}}(P, Q) = \delta_{\text{dF}}(P, Q)$. It remains to show the upper bound $\delta_{\text{greedy}}(P, Q) \leq 2^{O(n)}\delta_{\text{dF}}(P, Q)$ and to provide an example where this inequality is tight.

4.1 Upper Bound

We call a pair $p_i p_{i+1}$ of consecutive points on P an *edge* of P , for $i = 1, \dots, n - 1$, and similarly for Q . Let m be the total number of edges of P and Q , and let $\ell_1 \leq \ell_2 \leq \dots \leq \ell_m$

be the sorted sequence of the edge lengths. We pick $k^* \in \{0, \dots, m\}$ minimum such that

$$4\delta_{\text{dF}}(P, Q) + 2 \sum_{i=1}^{k^*} \ell_i < \ell_{k^*+1},$$

where we set $\ell_{m+1} = \infty$. We define δ^* as the left hand side, $\delta^* := 4\delta_{\text{dF}}(P, Q) + 2 \sum_{i=1}^{k^*} \ell_i$.

► **Lemma 4.2.** *We have (i) $\delta^* \geq 4\delta_{\text{dF}}(P, Q)$; (ii) $\sum_{i=1}^{k^*} \ell_i \leq \delta^*/2 - 2\delta_{\text{dF}}(P, Q)$; (iii) there is no edge with length in $(\delta^*/2 - 2\delta_{\text{dF}}(P, Q), \delta^*)$; and (iv) $\delta^* \leq 3^{k^*} 4\delta_{\text{dF}}(P, Q)$.*

Proof. Properties (i) and (ii) follow by definition. Property (iii) holds since for $i = 1, \dots, k^*$, we have $\ell_i \leq \delta^*/2 - 2\delta_{\text{dF}}(P, Q)$, by (ii), and for $i = k^* + 1, \dots, m$, we have $\ell_i \geq \delta^*$, by definition. It remains to prove (iv): for $k = 0, \dots, k^*$, we set $\delta_k = 4\delta_{\text{dF}}(P, Q) + 2 \sum_{i=1}^k \ell_i$, and we prove by induction that $\delta_k \leq 3^k 4\delta_{\text{dF}}(P, Q)$. For $k = 0$, this is immediate. Now suppose we know that $\delta_{k-1} \leq 3^{k-1} 4\delta_{\text{dF}}(P, Q)$, for some $k \in \{1, \dots, k^*\}$. Then, $k \leq k^*$ implies $\ell_k \leq \delta_{k-1}$, so $\delta_k = \delta_{k-1} + 2\ell_k \leq 3\delta_{k-1} \leq 3^k 4\delta_{\text{dF}}(P, Q)$, as desired. Now (iv) follows from $\delta^* = \delta_{k^*}$. ◀

We call an edge *long* if it has length at least δ^* , and *short* otherwise. In other words, the short edges have lengths $\ell_1, \dots, \ell_{k^*}$, and the long edges have lengths $\ell_{k^*+1}, \dots, \ell_m$. Let β be an optimal traversal of P and Q , i.e., $\delta(\beta) = \delta_{\text{dF}}(P, Q)$.

► **Lemma 4.3.** *The sequences P and Q have the same number of long edges. Furthermore, if $p_{i_1}p_{i_1+1}, \dots, p_{i_k}p_{i_k+1}$ and $q_{j_1}q_{j_1+1}, \dots, q_{j_k}q_{j_k+1}$ are the long edges of P and of Q , for $1 \leq i_1 < \dots < i_k < n$ and $1 \leq j_1 < \dots < j_k < n$, then both β and β_{greedy} contain the steps $(p_{i_1}, q_{j_1}) \rightarrow (p_{i_1+1}, q_{j_1+1}), \dots, (p_{i_k}, q_{j_k}) \rightarrow (p_{i_k+1}, q_{j_k+1})$.*

Proof. First, we show that for every long edge $p_i p_{i+1}$ of P , the optimal traversal β contains the step $(p_i, q_j) \rightarrow (p_{i+1}, q_{j+1})$, where q_j, q_{j+1} is a long edge of Q . Consider the step of β from p_i to p_{i+1} . This step has to be of the form $(p_i, q_j) \rightarrow (p_{i+1}, q_{j+1})$ for some $q_j \in Q$: since $\max\{d(p_i, q_j), d(p_{i+1}, q_j)\} \geq d(p_i, p_{i+1})/2 \geq \delta^*/2 \geq 2\delta_{\text{dF}}(P, Q)$, by Lemma 4.2(i), staying in q_j would result in $\delta(\beta) \geq 2\delta_{\text{dF}}(P, Q)$. Now, since $\max\{d(p_i, q_j), d(p_{i+1}, q_{j+1})\} \leq \delta(\beta) = \delta_{\text{dF}}(P, Q)$, the triangle inequality gives $d(q_j, q_{j+1}) \geq d(p_i, p_{i+1}) - 2\delta_{\text{dF}}(P, Q) \geq \delta^* - 2\delta_{\text{dF}}(P, Q)$. Lemma 4.2(iii) now implies $d(q_j, q_{j+1}) \geq \delta^*$, so the edge $q_j q_{j+1}$ is long.

Thus, β traverses every long edge of P simultaneously with a long edge of Q . A symmetric argument shows that β traverses every long edge of Q simultaneously with a long edge of P . Since β is monotone, it follows that P and Q have the same number of long edges, and that β traverses them simultaneously in their order of appearance along P and Q .

It remains to show that the greedy traversal β_{greedy} traverses the long edges of P and Q simultaneously. Set $i_0 = j_0 = 0$. We will prove for $a \in \{0, \dots, k-1\}$ that if β_{greedy} contains the position (p_{i_a+1}, q_{j_a+1}) , then it also contains the step $(p_{i_{a+1}}, q_{j_{a+1}}) \rightarrow (p_{i_{a+1}+1}, q_{j_{a+1}+1})$ and hence the position $(p_{i_{a+1}+1}, q_{j_{a+1}+1})$. The claim on β_{greedy} then follows by induction on a , since β_{greedy} contains the position (p_1, q_1) by definition. Thus, fix $a \in \{0, \dots, k-1\}$ and suppose that β_{greedy} contains (p_{i_a+1}, q_{j_a+1}) . We need to show that β_{greedy} also contains the step $(p_{i_{a+1}}, q_{j_{a+1}}) \rightarrow (p_{i_{a+1}+1}, q_{j_{a+1}+1})$. For better readability, we write i for i_a , j for j_a , i' for i_{a+1} , and j' for j_{a+1} . Consider the first position of β_{greedy} when β_{greedy} reaches either $p_{i'}$ or $q_{j'}$. Without loss of generality, this position is of the form $(p_{i'}, q_l)$, for some $l \in \{j+1, \dots, j'\}$. Then, $d(p_{i'}, q_l) \leq \delta^*/2 - \delta_{\text{dF}}(P, Q)$, since we saw that $d(p_{i'}, q_{j'}) \leq \delta(\beta) = \delta_{\text{dF}}(P, Q)$ and since the remaining edges between q_l and $q_{j'}$ are short and thus have total length at most $\delta^*/2 - 2\delta_{\text{dF}}(P, Q)$, by Lemma 4.2(ii). The triangle inequality now gives $d(p_{i'+1}, q_l) \geq d(p_{i'}, p_{i'+1}) - d(p_{i'}, q_l) \geq \delta^*/2 + \delta_{\text{dF}}(P, Q)$. If

$l < j'$, the same argument applied to q_{l+1} shows that $d(p_{i'}, q_{l+1}) \leq \delta^*/2 - \delta_{\text{dF}}(P, Q)$ and thus $d(p_{i'+1}, q_{l+1}) \geq \delta^*/2 + \delta_{\text{dF}}(P, Q)$. Thus, β_{greedy} moves to $(p_{i'}, q_{l+1})$. If $l = j'$, then β_{greedy} takes the step $(p_{i'}, q_{j'}) \rightarrow (p_{i'+1}, q_{j'+1})$, as $d(p_{i'+1}, q_{j'+1}) \leq \delta(\beta) = \delta_{\text{dF}}(P, Q)$, but $d(p_{i'}, q_{j'+1}), d(p_{i'+1}, q_{j'}) \geq \delta^* - \delta_{\text{dF}}(P, Q) \geq 3\delta_{\text{dF}}(P, Q)$, by Lemma 4.2(i). ◀

Finally, we can show the desired upper bound on the quality of the greedy algorithm.

► **Lemma 4.4.** *We have $\delta_{\text{greedy}}(P, Q) \leq \delta^*/2$.*

Proof. By Lemma 4.3, P and Q have the same number of long edges. Let $p_{i_1}p_{i_1+1}, \dots, p_{i_k}p_{i_k+1}$ and $q_{j_1}q_{j_1+1}, \dots, q_{j_k}q_{j_k+1}$ be the long edges of P and of Q , where $1 \leq i_1 < \dots < i_k < n$ and $1 \leq j_1 < \dots < j_k < n$. By Lemma 4.3, β_{greedy} contains the positions (p_{i_a}, q_{j_a}) and (p_{i_a+1}, q_{j_a+1}) for $a = 1, \dots, k$, and $d(p_{i_a}, q_{j_a}), d(p_{i_a+1}, q_{j_a+1}) \leq \delta_{\text{dF}}(P, Q)$ for $a = 1, \dots, k$. Thus, setting $i_0 = j_0 = 0$ and $i_{k+1} = j_{k+1} = n$, we can focus on the subtraversals $\beta_a = (p_{i_a+1}, q_{j_a+1}), \dots, (p_{i_{a+1}}, q_{j_{a+1}})$ of β_{greedy} , for $a = 0, \dots, k$. Now, since all edges traversed in β_a are short, and since $d(p_{i_a+1}, q_{j_a+1}) \leq \delta_{\text{dF}}(P, Q)$, we have $\delta(\beta_a) \leq \delta_{\text{dF}}(P, Q) + \delta^*/2 - 2\delta_{\text{dF}}(P, Q) \leq \delta^*/2$ by Lemma 4.2(iii) and the triangle inequality. Thus, $\delta(\beta_{\text{greedy}}) \leq \max\{\delta_{\text{dF}}(P, Q), \delta(\beta_1), \dots, \delta(\beta_k)\} \leq \delta^*/2$, as desired. ◀

Lemmas 4.2(iv) and 4.4 prove the desired inequality $\delta_{\text{greedy}}(P, Q) \leq 2^{O(n)}\delta_{\text{dF}}(P, Q)$, since $k^* \leq m = 2n - 2$.

4.2 Tight Example for the Upper Bound

Fix $1 < \alpha < 2$. Consider the sequence $P = \langle p_1, \dots, p_n \rangle$ with $p_i := (-\alpha)^i$ and the sequence $Q = \langle q_1, \dots, q_{n-2} \rangle$ with $q_i := (-\alpha)^{i+2}$. We show the following:

1. The greedy traversal $\beta_{\text{greedy}}(P, Q)$ makes $n - 2$ simultaneous steps in P and Q followed by 2 single steps in P . This results in a maximal distance of $\delta_{\text{greedy}}(P, Q) = \alpha^n + \alpha^{n-1}$.
2. The traversal which makes 2 single steps in P followed by $n - 2$ simultaneous steps in both P and Q has distance $\alpha^3 + \alpha^2$.

Together, this shows that $\delta_{\text{greedy}}(P, Q)/\delta_{\text{dF}}(P, Q) = \Omega(\alpha^n) = 2^{\Omega(n)}$, proving that the inequality $\delta_{\text{greedy}}(P, Q) \leq 2^{O(n)}\delta_{\text{dF}}(P, Q)$ is tight.

To see (1), assume that we are at position (p_i, q_i) . Moving to (p_i, q_{i+1}) would result in a distance of $d(p_i, q_{i+1}) = \alpha^{i+3} + \alpha^i$. Similarly, the other possible moves to (p_{i+1}, q_i) and to (p_{i+1}, q_{i+1}) would result in distances $\alpha^{i+2} + \alpha^{i+1}$, and $\alpha^{i+3} - \alpha^{i+1}$, respectively. It can be checked that for all $\alpha > 1$ we have $\alpha^{i+3} + \alpha^i > \alpha^{i+2} + \alpha^{i+1}$. Moreover, for all $\alpha < 2$ we have $\alpha^{i+2} + \alpha^{i+1} > \alpha^{i+3} - \alpha^{i+1}$. Thus, the greedy algorithm makes the move to (p_{i+1}, q_{i+1}) . Using induction, this shows that the greedy traversal starts with $n - 2$ simultaneous moves in P and Q . In the end, the greedy algorithm has to take two single moves in P . Thus, the greedy traversal contains the pair (p_{n-1}, q_{n-2}) , which is in distance $d(p_{n-1}, q_{n-2}) = \alpha^n + \alpha^{n-1} = 2^{\Omega(n)}$.

To see (2), note that the traversal which makes 2 single steps in P followed by $n - 2$ simultaneous moves in P and Q starts with (p_1, q_1) and (p_2, q_1) followed by (p_i, q_{i-2}) for $i = 2, \dots, n$. Note that $d(p_1, q_1) = \alpha^3 - \alpha$, $d(p_2, q_1) = \alpha^3 + \alpha^2$, and $p_i = q_{i-2}$, so that the remaining distances are 0. Thus, we have $\delta_{\text{dF}}(P, Q) \leq \alpha^3 + \alpha^2 = O(1)$.

5 Improved Approximation Algorithm

Let $P = \langle p_1, \dots, p_n \rangle$ and $Q = \langle q_1, \dots, q_n \rangle$ be two sequences of n points in \mathbb{R}^d , where d is constant. Let $1 \leq \alpha \leq n$. We show how to find a value δ^* with $\delta_{\text{dF}}(P, Q) \leq \delta^* \leq \alpha\delta_{\text{dF}}(P, Q)$

in time $O(n \log n + n^2/\alpha)$. For simplicity, we will assume that all points on P and Q are pairwise distinct. This can be achieved by an infinitesimal perturbation of the point set.

5.1 Decision Algorithm

We begin by describing an approximate decision procedure. For this, we prove the following theorem.

► **Theorem 5.1.** *Let P and Q be two sequences of n points in \mathbb{R}^d , and let $1 \leq \alpha \leq n$. Suppose that the points of P and Q have been sorted along each coordinate axis. There exists a decision algorithm with running time $O(n^2/\alpha)$ and the following properties: if $\delta_{\text{dF}}(P, Q) \leq 1$, the algorithm returns YES; if $\delta_{\text{dF}}(P, Q) \geq \alpha$, the algorithm returns NO; if $\delta_{\text{dF}}(P, Q) \in (1, \alpha)$, the algorithm may return either YES or NO. The running time depends exponentially on d .*

Consider the regular d -dimensional grid with diameter 1 (all cells are axis-parallel cubes with side length $1/\sqrt{d}$). The distance between two grid cells C and D , $d(C, D)$, is defined as the smallest distance between a point in C and a point in D . The distance between a point x and a grid cell C , $d(x, C)$, is the distance between x and the closest point in C . For a point $x \in \mathbb{R}^d$, we write B_x for the closed unit ball with center x and C_x for the grid cell that contains x (since we are interested in approximation algorithms, we may assume that all points of $P \cup Q$ lie strictly inside the cells). We compute for each point $r \in P \cup Q$ the grid cell C_r that contains it. We also record for each nonempty grid cell C the number of points from Q contained in C . This can be done in total linear time as follows: we scan the points from $P \cup Q$ in x_1 -order, and we group the points according to the grid intervals that contain them. Then we split the lists that represent the x_2, \dots, x_d -order correspondingly, and we recurse on each group to determine the grouping for the remaining coordinate axes. Each iteration takes linear time, and there are d iterations, resulting in a total time of $O(n)$. In the following, we will also need to know for each non-empty cell the neighborhood of all cells that have a certain constant distance from it. These neighborhoods can be found in linear time by modifying the above procedure as follows: before performing the grouping, we make $O(1)$ copies of each point $r \in P \cup Q$ that we translate suitably to hit all neighboring cells for r . By using appropriate cross-pointers, we can then identify the neighbors of each non-empty cell in total linear time. Afterwards, we perform a clean-up step, so that only the original points remain.

A grid cell C is *full* if $|C \cap Q| \geq 5n/\alpha$. Let \mathcal{F} be the set of full grid cells. Clearly, $|\mathcal{F}| \leq \alpha/5$. We say that two full cells $C, D \in \mathcal{F}$ are *adjacent* if $d(C, D) \leq 4$. This defines a graph H on \mathcal{F} of constant degree. Using the neighborhood finding procedure from above, we can determine H and its connected components L_1, \dots, L_k in time $O(n + \alpha)$. For $C \in \mathcal{F}$, the *label* L_C of C is the connected component of H containing C .

For each $q \in Q$, we search for a full cell $C \in \mathcal{F}$ with $d(q, C) \leq 2$. If such a cell exists, we label q with $L_q = L_C$; otherwise, we set $L_q = \perp$. Similarly, for each $p \in P$, we search a full cell $C \in \mathcal{F}$ with $d(p, C) \leq 1$. In case of success, we set $L_p = L_C$; otherwise, we set $L_p = \perp$. Using the neighborhood finding procedure from above, this takes linear time. Let $P' = \{p \in P \mid L_p \neq \perp\}$ and $Q' = \{q \in Q \mid L_q \neq \perp\}$. The labeling has the following properties.

► **Lemma 5.2.** *We have*

1. *for every $r \in P \cup Q$, the label L_r is uniquely determined;*
2. *for every $x, y \in P' \cup Q'$ with $L_x = L_y$, we have $d(x, y) \leq \alpha$;*
3. *if $p \in P'$ and $q \in B_p \cap Q$, then $L_p = L_q$; and*
4. *if $p \in P \setminus P'$, there are $O(n/\alpha)$ points $q \in Q$ with $d(p, C_q) \leq 1$. Hence, $|B_p \cap Q| = O(n/\alpha)$.*

Proof. Let $r \in P \cup Q$ and suppose there are $C, D \in \mathcal{F}$ with $d(r, C) \leq 2$ and $d(r, D) \leq 2$. Then $d(C, D) \leq d(C, r) + d(r, D) \leq 4$, so C and D are adjacent in H . It follows that $L_C = L_D$ and that L_r is determined uniquely.

Fix $x, y \in P' \cup Q'$ with $L_x = L_y$. By construction, there are $C, D \in \mathcal{F}$ with $d(x, C) \leq 2$, $d(y, D) \leq 2$ and $L_C = L_D$. This means that C and D are in the same component of H . Therefore, C and D are connected by a sequence of adjacent cells in \mathcal{F} . We have $|\mathcal{F}| \leq \alpha/5$, any two adjacent cells have distance at most 4, and each cell has diameter 1. Thus, the triangle inequality gives $d(x, y) \leq 2 + 4(|\mathcal{F}| - 1) + |\mathcal{F}| + 2 \leq \alpha$.

Let $p \in P'$ and $q \in B_p \cap Q$. Take $C \in \mathcal{F}$ with $d(p, C) \leq 1$. By the triangle inequality, $d(q, C) \leq d(q, p) + d(p, C) \leq 2$, so $L_q = L_p = L_C$.

Take $p \in P$ and suppose there is a grid cell C with $|C \cap Q| > 5n/\alpha$ and $d(p, C) \leq 1$. Then $C \in \mathcal{F}$, so $L_p \neq \perp$, which means that $p \in P'$. The contrapositive gives (4). ◀

Lemma 5.2 enables us to design an efficient approximation algorithm. For this, we define the *approximate free-space matrix* F . This is an $n \times n$ matrix with entries from $\{0, 1\}$. For $i, j \in \{1, \dots, n\}$, we set $F_{ij} = 1$ if either (i) $p_i \in P'$ and $L_{p_i} = L_{q_j}$; or (ii) $p_i \in P \setminus P'$ and $d(p_i, q_j) \leq 1$. Otherwise, we set $F_{ij} = 0$. The matrix F is approximate in the following sense:

▶ **Lemma 5.3.** *If $\delta_{\text{dF}}(P, Q) \leq 1$, then F allows a monotone traversal from $(1, 1)$ to (n, n) . Conversely, if F has a monotone traversal from $(1, 1)$ to (n, n) , then $\delta_{\text{dF}}(P, Q) \leq \alpha$.*

Proof. Suppose that $\delta_{\text{dF}}(P, Q) \leq 1$. Then there is a monotone traversal β of (P, Q) with $\delta(\beta) \leq 1$. By Lemma 5.2(3), β is also a traversal of F .

Now let β be a monotone traversal of F . By Lemma 5.2(2), we have $\delta(\beta) \leq \alpha$, as desired. ◀

Additionally, we define the *approximate reach matrix* R , which is an $n \times n$ matrix with entries from $\{0, 1\}$. We set $R_{ij} = 1$ if F allows a monotone traversal from $(1, 1)$ to (i, j) , and $R_{ij} = 0$, otherwise. By Lemma 5.3, R_{nn} is an α -approximate indicator for $\delta_{\text{dF}} \leq 1$. We describe how to compute the rows of R successively in total time $O(n^2/\alpha)$.

First, we perform the following preprocessing steps: we break Q into *intervals*, where an interval is a maximal consecutive subsequence of points $q \in Q$ with the same label $L_q \neq \perp$. For each point in an interval, we store pointers to the first and the last point of the interval. This takes linear time. Furthermore, for each $p_i \in P \setminus P'$, we compute a sparse representation T_i of the corresponding row of F , i.e., a sorted list of all the column indices j for which $F_{ij} = 1$. Using hashing and bucketing, this can be done in total time $O(n^2/\alpha)$, by Lemma 5.2(4).

Now we successively compute a sparse representation for each row i of R , i.e., a sorted list I_i of disjoint intervals $[a, b] \in I_i$ such that for $j = 1, \dots, n$, we have $R_{ij} = 1$ if and only if there is an interval $[a, b] \in I_i$ with $j \in [a, b]$. We initialize I_1 as follows: if $F_{11} = 0$, we set $I_1 = \emptyset$ and abort. Otherwise, if $p_1 \in P'$, then I_1 is initialized with the interval of q_1 (since $F_{11} = 1$, we have $L_{p_1} = L_{q_1}$ by Lemma 5.2(3)). If $p_1 \in P \setminus P'$, we determine the maximum b such that $F_{1j} = 1$ for all $j = 1, \dots, b$, and we initialize I_1 with the *singleton* intervals $[j, j]$ for $j = 1, \dots, b$. This can be done in time $O(n/\alpha)$, irrespective of whether p_i lies in P' or not.

Now suppose we already have the interval list I_i for some row i , and we want to compute the interval list I_{i+1} for the next row. We consider two cases.

Case 1: $p_{i+1} \in P'$. If $L_{p_{i+1}} = L_{p_i}$, we simply set $I_{i+1} = I_i$. Otherwise, we go through the intervals $[a, b] \in I_i$ in order. For each interval $[a, b]$, we check whether the label of q_b or the label of q_{b+1} equals the label of p_{i+1} . If so, we add the maximal interval $[b', c]$ to I_{i+1} with

$b' = b$ or $b' = b + 1$ and $L_{p_{i+1}} = L_{q_j}$ for all $j = b', \dots, c$. With the information from the preprocessing phase, this takes $O(1)$ time per interval. The resulting set of intervals may not be disjoint (if $p_i \in P \setminus P'$), but any two overlapping intervals have the same endpoint. Also, intervals with the same endpoint appear consecutively in I_{i+1} . We next perform a clean-up pass through I_{i+1} : we partition the intervals into consecutive groups with the same endpoint, and in each group, we only keep the largest interval. All this takes time $O(|I_i| + |I_{i+1}|)$.

Case 2: $p_{i+1} \in P \setminus P'$. In this case, we have a sparse representation T_{i+1} of the corresponding row in F at our disposal. We simultaneously traverse I_i and T_{i+1} to compute I_{i+1} as follows: for each $j \in \{1, \dots, n\}$ with $F_{(i+1)j} = 1$, if I_i has an interval containing $j - 1$ or j or if $[j - 1, j - 1] \in I_{i+1}$, we add the singleton $[j, j]$ to I_{i+1} . This takes total time $O(|I_i| + |I_{i+1}| + n/\alpha)$.

The next lemma shows that the interval representation remains sparse throughout the execution of the algorithm, and that the intervals I_i indeed represent the approximate reach matrix R .

► **Lemma 5.4.** *We have $|I_i| = O(n/\alpha)$ for $i = 1, \dots, n$. Furthermore, the intervals in I_i correspond exactly to the 1-entries in the approximate reach matrix R .*

Proof. First, we prove that $|I_i| = O(n/\alpha)$ for $i = 1, \dots, n$. This is done by induction on i . We begin with $i = 1$. If $p_1 \in P'$, then $|I_1| = 1$. If $p_1 \in P \setminus P'$, then Lemma 5.2(4) shows that the first row of F contains at most $O(n/\alpha)$ 1-entries, so $|I_1| = O(n/\alpha)$. Next, suppose that we know by induction that $|I_i| = O(n/\alpha)$. We must argue that $|I_{i+1}| = O(n/\alpha)$. If $p_{i+1} \in P \setminus P'$, then the $(i + 1)$ -th row of F contains $O(n/\alpha)$ 1-entries by Lemma 5.2(4), and $|I_{i+1}| = O(n/\alpha)$ follows directly by construction. If $p_{i+1} \in P'$ and $L_{p_{i+1}} = L_{p_i}$, then $I_{i+1} = I_i$, and the claim follows by induction. Finally, if $p_{i+1} \in P'$ and $L_{p_{i+1}} \neq L_{p_i}$, then by construction, every interval in I_i gives rise to at most one new interval in I_{i+1} . Thus, by induction, $|I_{i+1}| \leq |I_i| = O(n/\alpha)$.

Second, we prove that I_i represents the i -th row of R , for $i = 1, \dots, n$. Again, the proof is by induction. For $i = 1$, the claim holds by construction, because the first row of R consists of the initial segment of 1s in F . Next, suppose we know that I_i represents the i -th row of R . We must argue that I_{i+1} represents the $(i + 1)$ th row of R . If $p_{i+1} \in P \setminus P'$, this follows directly by construction, because the algorithm explicitly checks the conditions for each possible 1-entry of R ($R_{(i+1)j}$ can only be 1 if $F_{(i+1)j} = 1$). If $p_{i+1} \in P'$ and $L_{p_{i+1}} = L_{p_i}$, then the $(i + 1)$ -th row of F is identical to the i -th row of F , and the same holds for R : there can be no new monotone paths, and all old monotone paths can be extended by one step along Q . Finally, consider the case $p_{i+1} \in P'$ and $L_{p_{i+1}} \neq L_{p_i}$. If $p_i \in P \setminus P'$, then every interval in I_i is a singleton $[b, b]$, from which a monotone path could potentially reach $(i + 1, b)$ and $(i + 1, b + 1)$, and from there walk to the right. We explicitly check both of these possibilities. If $p_i \in P'$, then for every interval $[a, b] \in I_i$ and for all $j \in [a, b]$ we have $L_{q_j} = L_{p_i} \neq L_{p_{i+1}}$. Thus, the only possible move is to $(i + 1, b + 1)$, and from there walk to the right, which is what we check. ◀

The first part of Lemma 5.4 implies that the total running time is $O(n^2/\alpha)$, since each row is processed in time $O(n/\alpha)$. By Lemma 5.3 and the second part of Lemma 5.4, if I_n has an interval containing n then $\delta_{\text{dF}}(P, Q) \leq \alpha$, and if $\delta_{\text{dF}}(P, Q) \leq 1$ then n appears in I_n . Since the intervals in I_n are sorted, this condition can be checked in $O(1)$ time. Theorem 5.1 follows.

5.2 Optimization Procedure

We now leverage Theorem 5.1 to an optimization procedure.

► **Theorem 5.5.** *Let P and Q be two sequences of n points in \mathbb{R}^d , and let $1 \leq \alpha \leq n$. There is an algorithm with running time $O(n^2 \log n / \alpha)$ that computes a number δ^* with $\delta_{\text{dF}}(P, Q) \leq \delta^* \leq \alpha \delta_{\text{dF}}(P, Q)$. The running time depends exponentially on d .*

Proof. If $\alpha \leq 5$, we compute $\delta_{\text{dF}}(P, Q)$ directly in $O(n^2)$ time. Otherwise, we set $\alpha' = \alpha/5$. We sort the points of $P \cup Q$ according to the coordinate axes, and we compute a $(1/3)$ -well-separated pair decomposition $\mathcal{P} = \{(S_1, T_1), \dots, (S_k, T_k)\}$ for $P \cup Q$ in time $O(n \log n)$ [11]. Recall the properties of a well-separated pair decomposition: (i) for all pairs $(S, T) \in \mathcal{P}$, we have $S, T \subseteq P \cup Q$, $S \cap T = \emptyset$, and $\max\{\text{diam}(S), \text{diam}(T)\} \leq d(S, T)/3$ (here, $\text{diam}(S)$ denotes the maximum distance between any two points in S); (ii) the number of pairs is $k = O(n)$; and (iii) for every distinct $q, r \in P \cup Q$, there is exactly one pair $(S, T) \in \mathcal{P}$ with $q \in S$ and $r \in T$, or vice versa.

For each pair $(S_i, T_i) \in \mathcal{P}$, we pick arbitrary $s \in S_i$ and $t \in T_i$, and set $\delta_i = 3d(s, t)$. After sorting, we can assume that $\delta_1 \leq \dots \leq \delta_k$. We call δ_i a *YES-entry* if the algorithm from Theorem 5.1 on input α' and the point sets P and Q scaled by a factor of δ_i returns YES; otherwise, we call δ_i a *NO-entry*. First, we test whether δ_1 is a YES-entry. If so, we return $\delta^* = \alpha' \delta_1$. If δ_1 is a NO-entry, we perform a binary search on $\delta_1, \dots, \delta_k$: we set $l = 1$ and $r = k$. Below, we will prove that δ_k must be a YES-entry. We set $m = \lceil (l+r)/2 \rceil$. If δ_m is a NO-entry, we set $l = m$, otherwise, we set $r = m$. We repeat this until $r = l + 1$. In the end, we return $\delta^* = \alpha' \delta_r$. The total running time is $O(n \log n + n^2 \log n / \alpha)$. Our procedure works exactly like binary search, but we presented it in detail in order to emphasize that $\delta_1, \dots, \delta_k$ is not necessarily monotone: NO-entries and YES-entries may alternate.

We now argue correctness. The algorithm finds a YES-entry δ_r such that either $r = 1$ or δ_{r-1} is a NO-entry. By Theorem 5.1, any δ_i is a NO-entry if $\delta_i \leq \delta_{\text{dF}}(P, Q) / \alpha'$. Thus, we certainly have $\delta^* = \alpha' \delta_r > \delta_{\text{dF}}(P, Q)$. Now take a traversal β with $\delta(\beta) = \delta_{\text{dF}}(P, Q)$, and let $(p, q) \in P \times Q$ be a position in β that has $d(p, q) = \delta(\beta)$. There is a pair $(S_{r^*}, T_{r^*}) \in \mathcal{P}$ with $p \in S_{r^*}$ and $q \in T_{r^*}$, or vice versa. Let $s \in S_{r^*}$ and $t \in T_{r^*}$ be the points we used to define δ_{r^*} . Then

$$d(s, t) \geq d(p, q) - \text{diam}(S_{r^*}) - \text{diam}(T_{r^*}) \geq d(p, q) - 2d(S_{r^*}, T_{r^*})/3 \geq d(p, q)/3,$$

and

$$d(s, t) \leq d(p, q) + \text{diam}(S_{r^*}) + \text{diam}(T_{r^*}) \leq d(p, q) + 2d(S_{r^*}, T_{r^*})/3 \leq 5d(p, q)/3,$$

so $\delta_{r^*} = 3d(s, t) \in [\delta(\beta), 5\delta(\beta)]$. Since by Theorem 5.1 any δ_i is a YES-entry if $\delta_i \geq \delta_{\text{dF}}(P, Q)$, all δ_i with $i \geq r^*$ are YES-entries (in particular, δ_k is a YES-entry). Thus, $\delta^* \leq \alpha' \delta_{r^*} \leq 5\alpha' \delta_{\text{dF}}(P, Q) \leq \alpha \delta_{\text{dF}}(P, Q)$. ◀

The running time of Theorem 5.5 can be improved as follows.

► **Theorem 5.6.** *Let P and Q be two sequences of n points in \mathbb{R}^d , and let $1 \leq \alpha \leq n$. There is an algorithm with running time $O(n \log n + n^2 / \alpha)$ that computes a number δ^* with $\delta_{\text{dF}}(P, Q) \leq \delta^* \leq \alpha \delta_{\text{dF}}(P, Q)$. The running time depends exponentially on d .*

Proof. If $\alpha \leq 4$, we can compute $\delta_{\text{dF}}(P, Q)$ exactly. Otherwise, we use Theorem 5.5 to compute a number δ' with $\delta_{\text{dF}}(P, Q) \leq \delta' \leq n \cdot \delta_{\text{dF}}(P, Q)$, or, equivalently, $\delta_{\text{dF}}(P, Q) \in [\delta'/n, \delta']$. This takes time $O(n \log n)$. Set $i^* = \lceil \log(n/\alpha) \rceil + 1$ and for $i = 1, \dots, i^*$ let $\alpha_i = n/2^{i+1}$. Also, set $a_1 = \delta'/n$ and $b_1 = \delta'$.

We iteratively obtain better estimates for $\delta_{\text{dF}}(P, Q)$ by repeating the following for $i = 1, \dots, i^* - 1$. As an invariant, at the beginning of iteration i , we have $\delta_{\text{dF}}(P, Q) \in [a_i, b_i]$ with $b_i/a_i = 4\alpha_i$. We use the algorithm from Theorem 5.1 with inputs α_i and P and Q scaled by a factor $2\alpha_i$ (since $\alpha_i \geq \alpha_{i^*-1} = n/2^{\lceil \log(n/\alpha) \rceil + 1} \geq \alpha/4$, the algorithm can be applied). If the answer is YES, it follows that $\delta_{\text{dF}}(P, Q) \leq \alpha_i 2\alpha_i = b_i/2$, so we set $a_{i+1} = a_i$ and $b_{i+1} = b_i/2$. If the answer is NO, then $\delta_{\text{dF}}(P, Q) \geq 2\alpha_i$, so we set $a_{i+1} = 2\alpha_i$ and $b_{i+1} = b_i$. This needs time $O(n^2/\alpha_i)$ and maintains the invariant.

In the end, we return a_{i^*} . The invariant guarantees $\delta_{\text{dF}}(P, Q) \in [a_{i^*}, b_{i^*}]$ and $b_{i^*}/a_{i^*} = 4\alpha_{i^*} \leq \alpha$, as desired. The total running time is proportional to

$$n \log n + \sum_{i=1}^{i^*-1} n^2/\alpha_i = n \log n + \sum_{i=1}^{i^*-1} n2^{i+1} \leq n \log n + n2^{i^*+1} = O(n \log n + n^2/\alpha). \quad \blacktriangleleft$$

6 Conclusions

We have obtained several new results on the approximability of the discrete Fréchet distance. As our main results,

1. we showed a conditional lower bound for the *one-dimensional* case that there is no 1.399-approximation in strongly subquadratic time unless the Strong Exponential Time Hypothesis fails. This sheds further light on what makes the Fréchet distance a difficult problem.
2. we determined the approximation ratio of the *greedy* algorithm as $2^{\Theta(n)}$ in any dimension $d \geq 1$. This gives the first general linear time approximation algorithm for the problem; and
3. we designed an α -*approximation* algorithm running in time $O(n \log n + n^2/\alpha)$ for any $1 \leq \alpha \leq n$ in any constant dimension $d \geq 1$. This significantly improves the greedy algorithm, at the expense of a (slightly) worse running time.

Our lower bounds exclude only (too good) constant factor approximations with strongly subquadratic running time, while our best strongly subquadratic approximation algorithm has an approximation ratio of n^ε . It remains a challenging open problem to close this gap.

References

- 1 Amir Abboud and Virginia Vassilevska Williams. Popular conjectures imply strong lower bounds for dynamic problems. In *Proc. 55th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS)*, pages 434–443, 2014.
- 2 Amir Abboud, Virginia Vassilevska Williams, and Oren Weimann. Consequences of faster alignment of sequences. In *Proc. 41st Internat. Colloq. Automata Lang. Program. (ICALP)*, volume 8572 of *LNCS*, pages 39–51, 2014.
- 3 Amir Abboud, Ryan Williams, and Huacheng Yu. More applications of the polynomial method to algorithm design. In *Proc. 26th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 218–230, 2015.
- 4 Pankaj K. Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the discrete Fréchet distance in subquadratic time. *SIAM J. Comput.*, 43(2):429–449, 2014.
- 5 Helmut Alt. Personal communication. 2012.
- 6 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *Internat. J. Comput. Geom. Appl.*, 5(1–2):78–99, 1995.
- 7 Karl Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Proc. 55th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS)*, pages 661–670, 2014.

- 8 Karl Bringmann and Marvin Künnemann. Improved approximation for Fréchet distance on c -packed curves matching conditional lower bounds. [arXiv:1408.1340](https://arxiv.org/abs/1408.1340), 2014.
- 9 Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four soviets walk the dog – with an application to Alt’s conjecture. In *Proc. 25th Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 1399–1413, 2014.
- 10 Kevin Buchin, Maike Buchin, Rolf van Leusden, Wouter Meulemans, and Wolfgang Mulzer. Computing the Fréchet distance with a retractable leash. In *Proc. 21st Annu. European Sympos. Algorithms (ESA)*, pages 241–252, 2013.
- 11 Paul B. Callahan and S. Rao Kosaraju. A decomposition of multidimensional point sets with applications to k -nearest-neighbors and n -body potential fields. *J. ACM*, 42(1):67–90, 1995.
- 12 Thomas Eiter and Heikki Mannila. Computing Discrete Fréchet Distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory, 1994.
- 13 Anka Gajentaan and Mark H. Overmars. On a class of $O(n^2)$ problems in computational geometry. *Comput. Geom. Theory Appl.*, 5(3):165–185, 1995.
- 14 Michael R. Garey and David S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness*. W. H. Freeman, 1979.
- 15 Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles. In *Proc. 55th Annu. IEEE Sympos. Found. Comput. Sci. (FOCS)*, pages 621–630, 2014.
- 16 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k -SAT. *J. Comput. System Sci.*, 62(2):367–375, 2001.
- 17 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity. *J. Comput. System Sci.*, 63(4):512–530, 2001.
- 18 Mihai Pătraşcu and Ryan Williams. On the possibility of faster SAT algorithms. In *Proc. 21st Annu. ACM-SIAM Sympos. Discrete Algorithms (SODA)*, pages 1065–1075, 2010.
- 19 Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k -sat. *J. ACM*, 52(3):337–364, 2005.
- 20 Liam Roditty and Virginia Vassilevska Williams. Fast approximation algorithms for the diameter and radius of sparse graphs. In *Proc. 45th Annu. ACM Sympos. Theory Comput. (STOC)*, pages 515–524, 2013.
- 21 Ryan Williams. A new algorithm for optimal 2-constraint satisfaction and its implications. *Theoret. Comput. Sci.*, 348(2):357–365, 2005.

The Hardness of Approximation of Euclidean k -Means

Pranjal Awasthi¹, Moses Charikar², Ravishankar Krishnaswamy³,
and Ali Kemal Sinop⁴

- 1 Computer Science Department, Princeton University, USA
pawasthi@cs.cmu.edu
- 2 Computer Science Department, Princeton University, USA
moses@cs.princeton.edu
- 3 Microsoft Research, India
ravishan@cs.cmu.edu
- 4 Simons Institute for the Theory of Computing, USA
University of California, Berkeley
asinop@cs.cmu.edu

Abstract

The Euclidean k -means problem is a classical problem that has been extensively studied in the theoretical computer science, machine learning and the computational geometry communities. In this problem, we are given a set of n points in Euclidean space \mathbb{R}^d , and the goal is to choose k center points in \mathbb{R}^d so that the sum of squared distances of each point to its nearest center is minimized. The best approximation algorithms for this problem include a polynomial time constant factor approximation for general k and a $(1 + \epsilon)$ -approximation which runs in time $\text{poly}(n) \exp(k/\epsilon)$. At the other extreme, the only known computational complexity result for this problem is NP-hardness [1]. The main difficulty in obtaining hardness results stems from the Euclidean nature of the problem, and the fact that any point in \mathbb{R}^d can be a potential center. This gap in understanding left open the intriguing possibility that the problem might admit a PTAS for all k, d .

In this paper we provide the first hardness of approximation for the Euclidean k -means problem. Concretely, we show that there exists a constant $\epsilon > 0$ such that it is NP-hard to approximate the k -means objective to within a factor of $(1 + \epsilon)$. We show this via an efficient reduction from the vertex cover problem on *triangle-free graphs*: given a triangle-free graph, the goal is to choose the fewest number of vertices which are incident on all the edges. Additionally, we give a proof that the current best hardness results for vertex cover can be carried over to triangle-free graphs. To show this we transform G , a known hard vertex cover instance, by taking a graph product with a suitably chosen graph H , and showing that the size of the (normalized) maximum independent set is almost exactly preserved in the product graph using a spectral analysis, which might be of independent interest.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Euclidean k -means, Hardness of Approximation, Vertex Cover

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.754

1 Introduction

Clustering is the task of partitioning a set of items such as web pages, protein sequences etc. into groups of related items. This is a fundamental task in machine learning, information retrieval, computational geometry, computer vision, data visualization and many other



© Pranjal Awasthi, Moses Charikar, Ravishankar Krishnaswamy, and Ali Kemal Sinop;
licensed under Creative Commons License CC-BY
31st International Symposium on Computational Geometry (SoCG'15).
Editors: Lars Arge and János Pach; pp. 754–767



Leibniz International Proceedings in Informatics
LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

domains. In many applications, clustering is often used as a first step toward other fine grained tasks such as classification. Needless to say, the problem of clustering has received significant attention over the years and there is a large body of work on both the applied and the theoretical aspects of the problem [6, 4, 10, 13, 19, 21, 26, 33, 8, 28, 34]. A common way to approach the task of clustering is to map the set of items into a metric space where distances correspond to how different two items are from each other. Using this distance information, one then tries to optimize an objective function to get the desired clustering. Among the most commonly used objective function used in the clustering literature is the k -means objective function. In the k -means problem, the input is a set S of n data points in Euclidean space \mathbb{R}^d , and the goal is to choose k center points $C^* = \{c_1, c_2, \dots, c_k\}$ from \mathbb{R}^d so as to minimize $\Phi = \sum_{x \in S} \min_i \|x - c_i\|^2$, where $c_i(x) \in C^*$ is the center closest to x . Aside from being a natural clustering objective, an important motivation for studying this objective function stems from the fact that a very popular and widely used heuristic (appropriately called the k -means heuristic [28]) attempts to minimize this k -means objective function.

While the k -means heuristic is very much tied to the k -means objective function, there are many examples where it converges to a solution which is far away from the optimal k -means solution. This raises the important question of whether there exist provable algorithms for the k -means problem in general Euclidean space, which is the focus problem of our paper. Unfortunately though, the approximability of the problem is not very well understood. From the algorithmic side, there has been much focus on getting $(1 + \epsilon)$ -approximations that run as efficiently as possible. Indeed, for fixed k , Euclidean k -means admits a PTAS [26, 16]. These algorithms have exponential dependence in k , but only linear dependence in the number of points and the dimensionality of the space. As mentioned above, there is also empirical and theoretical evidence for the effectiveness of very simple heuristics for this problem [33, 28, 25]. For arbitrary k and d , the best known approximation algorithm for k -means achieves a factor of $9 + \epsilon$ [21]. In contrast to the above body of work on getting algorithms for k -means, lower bounds for k -means have remained elusive. In fact, until recently, even NP-hardness was not known for the k -means objective [11, 1]. This is perhaps due to the fact that as opposed to many discrete optimization problems, the k -means problem allows one to choose any point in the Euclidean space as a center. The above observations lead to the following intriguing possibility:

“Is there a PTAS for Euclidean k -means for arbitrary k and dimension d ?”

In this paper we answer this question in the negative and provide the first hardness of approximation for the Euclidean k -means problem.

► **Theorem 1.1.** *There exists a constant $\epsilon > 0$ such that it is NP-hard to approximate the Euclidean k -means to a factor better than $(1 + \epsilon)$.*

The starting point for our reduction is the Vertex-Cover problem on triangle-free graphs: here, given a triangle-free graph, the goal is to choose the fewest number of vertices which are incident on all the edges in the graph. This naturally leads us to our other main result in this paper, that of showing hardness of approximation of vertex cover on triangle-free graphs. Kortsarz et al [24] show that if the vertex cover problem is hard to approximate to a factor of $\alpha \geq 3/2$, then it is hard to approximate vertex cover on triangle-free graphs to the same factor of α . While such a hardness (in fact, a factor of $2 - \epsilon$ [22]) is known assuming the stronger unique games conjecture, the best known NP-hardness results do not satisfy $\alpha \geq 3/2$. We settle this question by showing NP-hardness results for approximating vertex cover on triangle-free graphs, which match the best known hardness on general graphs.

► **Theorem 1.2.** *It is NP-hard to approximate Vertex Cover on triangle-free graphs to within any factor smaller than 1.36.*

2 Main Technical Contribution

In Section 4, we show a reduction from Vertex-Cover on triangle-free graphs to Euclidean k -means where the vertex cover instances have small cover size if and only if the corresponding k -means instances have a low cost. A crucial ingredient is to relate the cost of the clusters to the structural properties of the original graph, which lets us transition from the Euclidean problem to a completely combinatorial problem. Then in Section 5, we prove that the known hardness of approximation results for Vertex-Cover carry over to triangle-free graphs. This improves over existing hardness results for vertex cover on triangle-free graphs [24]. Furthermore, we believe that our proof techniques are of independent interest. Specifically, our reduction transforms known hard instances G of vertex cover, by taking a graph product with an appropriately chosen graph H . We then show that the size of the vertex cover in the new graph (in proportion to the size of the graph) can be related to spectral properties of H . In fact, by choosing H to have a bounded spectral radius, we show that the vertex covers in G and the product graph are roughly preserved, while also ensuring that the product graph is triangle-free. Combining this with our reduction to k -means completes the proof.

3 Related Work

Arthur and Vassilvitskii [5] proposed k -means++, a random sampling based approximation algorithm for Euclidean k -means which achieves a factor of $O(\log k)$. This was improved by Kanungo et al. [21] who proposed a local search based algorithm which achieves a factor of $(9 + \epsilon)$. This is currently the best known approximation algorithm for k -means. For fixed k and d , Matousek [31] gave a PTAS for k -means which runs in time $O(n\epsilon^{-2k^2d} \log^k n)$. Here n is the number of points and d is the dimensionality of the space. This was improved by Badoiu et al. [7] who gave a PTAS for fixed k and any d with run time $O(2^{(k/\epsilon)^{O(1)}} \text{poly}(d)n \log^k n)$. Kumar et al. [26] gave an improved PTAS with exponential dependence in k and only linear dependence in n and d . Feldman et al. [16] combined this with efficient coresets constructions to give a PTAS for fixed k with improved dependence on k . The work of Dasgupta [11] and Aloise et al. [1] showed that Euclidean k -means is NP-hard even for $k = 2$. Mahajan et al. [30] also show that the k -means problem is NP-hard for points in the plane.

There are also many other clustering objectives related to k -means which are commonly studied. The most relevant to our discussion are the k -median and the k -center objectives. In the first problem, the objective is to pick k centers to minimize the sum of distances of each point to the nearest center (note that the distances are not squared). The problem deviates from k -means in two crucial aspects, both owing to the different contexts in which the two problems are studied: (i) the k -median problem is typically studied in the setting where the centers are one of the data points (or come from a set of possible centers specified in the input), and (ii) the problem is also very widely studied on general metrics, without the Euclidean restriction. The k -median problem has been a testbed of developing new techniques in approximation algorithms, and has constantly seen improvements even until very recently [20, 19, 27]. Currently, the best known approximation for k -median is a factor of $2.611 + \epsilon$ due to Bykra et al. [9]. On the other hand, it is also known that the k -median objective (on general metrics) is NP-hard to approximate to a factor better than $(1 + 1/e)$ [19]. When restricted to Euclidean metrics, Kolliopoulos et al. [23] show a PTAS for k -median

on constant dimensional spaces. On the negative side for k -median on Euclidean metrics, it is known that the discrete problem (where centers come from a specified input) cannot have a PTAS under standard complexity assumptions [17]. As mentioned earlier, all these results are for the version when the possible candidate centers is specified in the input. For the problem where any point can be a center, Arora et al. [4] show a PTAS when the points are on a 2-dimensional plane.

In the k -center problem the objective is to pick k center points such that the maximum distance of any data point to the closest center point is minimized. In general metrics, this problem admits a 2-factor approximation which is also optimal assuming $P \neq NP$ [18]. For Euclidean metric when the center could be any point in the space, the upper bound is still 2 and the best hardness of approximation is a factor 1.82 [15].

4 Our Hardness Reduction: From Vertex Cover to Euclidean k -means

In this section, we show a reduction from the Vertex-Cover problem (on triangle-free graphs) to the k -means problem. Formally, the vertex cover problem can be stated as follows: Given an undirected graph $G = (V, E)$, choose a subset S of vertices (with minimum $|S|$) such that S is incident on every edge of the graph. More specifically, our reduction establishes the following theorem.

► **Theorem 4.1.** *There is an efficient reduction from instances of Vertex Cover (on triangle-free graphs with m edges) to those of Euclidean k -means that satisfies the following properties:*

- (i) *if the Vertex Cover instance has value k , then the k -means instance has cost $\leq m - k$.*
- (ii) *if the Vertex Cover instance has value at least $k(1 + \epsilon)$, then the optimal k -means cost is $\geq m - (1 - \Omega(\epsilon))k$. Here, ϵ is some fixed constant > 0 .*

In Section 5, we show that there exist triangle-free graph instances of vertex cover on $m = \Theta(n)$ edges, and $k = \Omega(n)$ such that it is NP-hard to distinguish if the instance has a vertex cover of size at most k , or all vertex covers have size at least $(1 + \epsilon)k$, for some constant $\epsilon > 0$.

Now, let $k = m/\Delta$ where $\Delta = \Omega(1)$ from the hard vertex cover instances. Then, from Theorem 4.1, we get that if the vertex cover has value k , then the k -means cost is at most $m(1 - \frac{1}{\Delta})$, and if the vertex cover is at least $k(1 + \epsilon)$, then the optimal k -means cost is at least $m(1 - \frac{1 - \Omega(\epsilon)}{\Delta})$. Therefore, the vertex cover hardness says that it is also NP-hard to distinguish if the resulting k -means instance has cost at most $m(1 - \frac{1}{\Delta})$ or cost more than $m(1 - \frac{1 - \Omega(\epsilon)}{\Delta})$. Since Δ is a constant, this implies that it is NP-hard to approximate the k -means problem within some factor $(1 + \Omega(\epsilon))$, thereby establishing our main result Theorem 1.1. In what follows, we prove Theorem 4.1.

4.1 Proof of Theorem 4.1

Let $G = (V, E)$ denote the graph in the Vertex Cover instance \mathcal{I} , with parameter k denoting the number of vertices we can select. We associate the vertices with natural numbers $[n]$. Therefore, we refer to vertices by natural numbers i , and edges by pairs of natural numbers (i, j) .

Construction of k-means Instance \mathcal{I}_{km}

For each vertex $i \in [n]$, we have a unit vector $x_i = (0, 0, \dots, 1, \dots, 0)$ which has a 1 in the i^{th} coordinate and 0 elsewhere. Now, for each edge $e \equiv (i, j)$, we have a vector $x_e \stackrel{\text{def}}{=} x_i + x_j$. Our data points on which we solve the k -means problem is precisely $\{x_e : e \in E\}$. This completes the definition of \mathcal{I}_{km} .

► **Remark.** As stated, the dimensionality of the points we have constructed is n , and we get a hardness factor of $(1 + \epsilon)$. However, by using the dimensionality reduction ideas of Johnson and Lindenstrauss (see, e.g. [12]), without loss of generality, we can assume that the points lie in $O(\log n/\epsilon^2)$ dimensions and our hardness results still hold true. This is because, after the transformation, all pairwise distances (and in particular, the k -means objective function) are preserved up to a factor of $(1 + \epsilon/10)$ of the original values, and so our hardness factor is also (almost) preserved, i.e., we would get hardness of approximation of $(1 + \Omega(\epsilon))$.

However, for simplicity, we stick with the n dimensional vectors as it makes the presentation much cleaner.

4.2 Completeness

Suppose \mathcal{I} is such that there exists a vertex cover $S^* = \{v_1, v_2, \dots, v_k\}$ of k vertices which can cover all the edges. We will now show that we can recover a good clustering of low k -means cost. To this end, let E_{v_ℓ} denote the set of edges which are covered by v_ℓ for $1 \leq \ell \leq k$. If an edge is covered by two vertices, we assume that only one of them covers it. As a result, note that the E_{v_ℓ} 's are pairwise disjoint (and their union is E), and each E_{v_ℓ} is of the form $\{(v_\ell, w_{\ell,1}), (v_\ell, w_{\ell,2}), \dots, (v_\ell, w_{\ell,p_\ell})\}$.

Now, to get our clustering, we do the following: for each $v \in S^*$, form a cluster out of the data points $\mathcal{F}_v := \{x_e : e \in E_v\}$. We now analyze the average connection cost of this solution. To this end, we begin with some easy observations about the k -means clustering. Indeed, since any cluster is of a set of data points (corresponding to a subset of edges in the graph G), we shall abuse notation and associate any cluster \mathcal{F} also with the corresponding subgraph on V , i.e., $\mathcal{F} \subseteq E$. Moreover, we use $d_{\mathcal{F}}(i)$ to denote the degree of node i in \mathcal{F} and $m_{\mathcal{F}}$ to denote the number of edges in \mathcal{F} , $m_{\mathcal{F}} = |\mathcal{F}|$. Finally, we refer by $d_G(i)$ the degree of vertex i in G .

► **Claim 4.2.** For any clustering $\{\mathcal{F}\}$: (a) $\sum_{\mathcal{F}} d_{\mathcal{F}}(i) = d_G(i)$; (b) $\sum_i \sum_{\mathcal{F}} d_{\mathcal{F}}(i) = 2m = 2|E|$.

Proof. Immediate, because every edge $e \in E$ belongs to exactly one cluster in $\{\mathcal{F}\}$. ◀

Our next claim relates the connection cost of any cluster \mathcal{F} to the structure of the associated subgraph, which forms the crucial part of the analysis.

► **Claim 4.3.** The total connection cost of any cluster \mathcal{F} is $\sum_i d_{\mathcal{F}}(i)(1 - \frac{1}{m_{\mathcal{F}}}d_{\mathcal{F}}(i))$.

Proof. Firstly, note that $\sum_i d_{\mathcal{F}}(i) = 2m_{\mathcal{F}}$. Now consider the center $\mu_{\mathcal{F}}$ of cluster \mathcal{F} . By definition, we have that at coordinate $i \in V$:

$$\mu_{\mathcal{F}}(i) = \frac{1}{m_{\mathcal{F}}} \sum_{S \in \mathcal{F}: i \in S} 1 = \frac{d_{\mathcal{F}}(i)}{m_{\mathcal{F}}}.$$

So $\|\mu_{\mathcal{F}}\|^2 = \frac{1}{m_{\mathcal{F}}^2} \sum_i d_{\mathcal{F}}(i)^2$. Hence the total cost of this clustering, $c_{\mathcal{F}}$, is:

$$\sum_{e \in \mathcal{F}} (\|x_e - \mu_{\mathcal{F}}\|^2) = \sum_{e \in \mathcal{F}} (\|x_e\|^2 - \|\mu_{\mathcal{F}}\|^2) = 2m_{\mathcal{F}} - \frac{1}{m_{\mathcal{F}}} \sum_{i \in V} d_{\mathcal{F}}(i)^2 = \sum_i d_{\mathcal{F}}(i) - \frac{1}{m_{\mathcal{F}}} d_{\mathcal{F}}(i)^2.$$

Here we used $m_{\mathcal{F}}\mu_{\mathcal{F}} = \sum_{e \in \mathcal{F}} x_e$ in the first equality and $\|x_e\|^2 = 2$ in the second one. ◀

► **Claim 4.4.** *There exists a clustering of our k -means instance \mathcal{I}_{km} with cost at most $m - k$, where m is the number of edges in the graph $G = (V, E)$ associated with the vertex cover instance \mathcal{I} , and k is the size of the optimal vertex cover.*

Proof. Consider a cluster \mathcal{F}_v , which consists of data points associated with edges covered by a single vertex v . Then, by Claim 4.3, the connection cost of this cluster is precisely $m_{\mathcal{F}_v} - 1$, since the sub-graph associated with a cluster is simply a star rooted at v . Here, $m_{\mathcal{F}_v}$ is the number of edges which v covers in the vertex cover (if an edge is covered by different vertices in the cover, it is included in only one vertex). Then, summing over all clusters, we get the claim. ◀

4.3 Soundness

In this section, we show that if there is a clustering of low k -means cost, then there is a very good vertex cover for the corresponding graph. We begin with some useful notation.

► **Notation 4.5.** *Given a set $E' \subseteq \binom{V}{2}$ of $m_{E'} = |E'|$ edges with corresponding node degrees (d_1, \dots, d_n) , we define $\text{Cost}(E')$ as the following:*

$$\text{Cost}(E') \stackrel{\text{def}}{=} \sum_{u \in V} d_u \left(1 - \frac{d_u}{m_{E'}}\right).$$

Note that, by Claim 4.3, the connection cost of a clustering $\Gamma = \{\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_k\}$ of the n points is equal to $\sum_i \text{Cost}(\mathcal{F}_i)$. Recall that we abuse notation slightly and view each cluster \mathcal{F}_i of the data points also as a subset of E . Moreover, because Γ clusters all points, the subgraphs $\mathcal{F}_1, \mathcal{F}_2, \dots, \mathcal{F}_k$ form a partition of E . Using this analogy, we study the properties of each subgraph and show that if the k -means cost of Γ is small, then most of these subgraphs in fact are stars. This will in turn help us recover a small vertex cover for G . We begin with a simple property of $\text{Cost}(E')$.

► **Proposition 4.6.** *For any set of $m_{E'}$ edges E' , $m_{E'} - 1 \leq \text{Cost}(E') \leq 2m_{E'} - 1$.*

Proof. We have $\text{Cost}(E') = \sum_{u \in V} d_u \left(1 - \frac{d_u}{m_{E'}}\right) = 2m_{E'} - \frac{\sum_{u \in V} d_u^2}{m_{E'}}$. The proof follows from noting that $\frac{\sum_{u \in V} d_u^2}{m_{E'}} \geq \frac{\sum_{u \in V} d_u}{m_{E'}} = 2$ and $\frac{\sum_{u \in V} d_u^2}{m_{E'}} \leq m_{E'} + 1$. The last inequality is due to the fact that $\sum_{u \in V} d_u^2$ is maximized by the degree sequence $(m_{E'}, 1, 1, \dots, 1)$. ◀

► **Theorem 4.7.** *If the k -means instance \mathcal{I}_{km} has a clustering $\Gamma = \{\mathcal{F}_1, \dots, \mathcal{F}_k\}$ with $\sum_{\mathcal{F} \in \Gamma} \text{Cost}(\mathcal{F}) \leq m - (1 - \delta)k$, then there exists a $(1 + O(\delta))k$ -vertex cover of G in the instance \mathcal{I} .*

Note that this, along with Claim 4.4 will imply the proof of Theorem 4.1.

Proof. For each $i \in [k]$, let $m_i \stackrel{\text{def}}{=} |\mathcal{F}_i|$ and $\nu_i \stackrel{\text{def}}{=} \sum_u d_u(\mathcal{F}_i)^2$. Note that $\text{Cost}(\mathcal{F}_i) = 2m_i - \frac{\nu_i}{m_i}$. By Proposition 4.6, each $i \in [k]$ satisfies $m_i - 1 \leq \text{Cost}(\mathcal{F}_i) \leq 2m_i - 1$. Hence if we define δ_i as $\delta_i \stackrel{\text{def}}{=} \text{Cost}(\mathcal{F}_i) - (m_i - 1)$, then $0 \leq \delta_i \leq m_i$. Moreover $\frac{\nu_i}{m_i} = m_i + 1 - \delta_i$. Thus:

$$m - (1 - \delta)k \geq \sum_i \text{Cost}(\mathcal{F}_i) = \sum_i (\delta_i + m_i - 1) = \sum_i \delta_i + m - k \implies \delta k \geq \sum_i \delta_i.$$

This means, except $\leq 2\delta k$ clusters, the remaining clusters all have $\delta_i \leq \frac{1}{2}$. Moreover, Theorem 4.8 implies all these $(1 - 2\delta)k$ clusters are either stars or triangles and have $\delta_i = 0$.

Since the graph is triangle free, they are all stars, and hence the corresponding center vertices cover all the edges in the respective clusters. It now remains to cover the edges in the remaining $2\delta k$ clusters which have larger δ_i values. Indeed, even for these clusters, we can appeal to Theorem 4.8, and choose *two* vertices per cluster to cover all but δ_i edges in each cluster. So the size of our candidate vertex cover is at most $k(1 + 2\delta)$, and we have covered all but $\sum_i \delta_i$ edges. But now, we notice that $\sum_i \delta_i \leq \delta k$, and so we can simply include one vertex per uncovered edge and would obtain a vertex cover of size at most $k(1 + 3\delta)$, thus completing the proof. \blacktriangleleft

► **Lemma 4.8.** *Given a graph $G_{\mathcal{F}} = (V, \mathcal{F})$ with $m = |\mathcal{F}|$ edges and degrees (d_1, \dots, d_n) ; let δ be such that $\frac{1}{m} \sum_u d_u^2 = m + 1 - \delta$. Then there always exists an edge $\{u, v\} \in \mathcal{F}$ with $d_u + d_v \geq m + 1 + \delta$. Furthermore, if $\delta < \frac{1}{2}$, then $\delta = 0$ and $G_{\mathcal{F}}$ is either a star or triangle.*

Proof. Since $\sum_u d_u^2 = \sum_{u \sim v} (d_u + d_v)$, we can think of $\frac{1}{m} \sum_u d_u^2$ as the the expectation of $d_u + d_v$ over a random edge chosen uniformly, $\{u, v\} \in E$:

$$\frac{1}{m} \sum_u d_u^2 = \mathbb{E}_{u \sim v} [d_u + d_v].$$

From this, we can immediately conclude the existence of an edge $\{u, v\}$ with $d_u + d_v \geq m + 1 - \delta$. Now to complete the second part of the Lemma statement, suppose $d_u \geq d_v$. The number of edges incident to $\{u, v\}$ is:

$$d_u + d_v - 1 \geq m - \delta \xrightarrow{\delta < 1} d_u + d_v - 1 = m.$$

So all edges are incident to u or v , and $d_w \leq 2$ if $w \notin \{u, v\}$. If $d_v \leq 1$, then we are done. In the other case, we have $d_v \geq 2 \geq d_w$ for all $w \notin \{u, v\}$. Let $\alpha \stackrel{\text{def}}{=} d_u$ and $\beta \stackrel{\text{def}}{=} d_v$. The degree sequence (d_1, \dots, d_n) is strongly majorized by the following sequence, d' :

$$d' \stackrel{\text{def}}{=} \left(\alpha, \beta, \underbrace{2, \dots, 2}_{\beta - 1 \text{ many}}, \underbrace{1, \dots, 1}_{\alpha - \beta \text{ many}} \right).$$

Since $\sum_u d_u^2$ is Schur-convex, its value increases under majorization:

$$\begin{aligned} (\alpha + \beta - 1)(\alpha + \beta - \delta) &= m(m + 1 - \delta) = \sum_u d_u^2 \leq \sum_u d_u'^2 \\ &= \alpha^2 + \beta^2 + 4(\beta - 1) + (\alpha - \beta). \\ \implies 0 &\leq (\alpha + \beta - 1)\delta + 2\alpha + 4\beta - 4 - 2\alpha\beta \\ &= (\alpha + \beta - 1)\delta + 2\alpha(1 - \beta) + 4(\beta - 1). \end{aligned}$$

So we obtain $2\alpha(\beta - 1) \leq (\alpha + \beta - 1)\delta + 4(\beta - 1)$. Since $\beta \geq 2$, we divide both sides by $\beta - 1$:

$$2\alpha \leq \frac{\alpha}{\beta - 1}\delta + 4 + \delta \leq \delta\alpha + 4 + \delta.$$

In particular, $(2 - \delta)\alpha \leq 4 + \delta \implies \alpha \leq \frac{4 + \delta}{2 - \delta} < 3$ as $\delta < 1/2$. Hence $\alpha \leq 2$. Consequently, $d_u = d_v = 2$ and $m = d_u + d_v - 1 = 3$. There are two possible cases: The graph is either a 3-cycle or 4-path. In the latter case, the corresponding δ is:

$$\delta = m + 1 - \frac{1}{m} \sum_u d_u^2 = 4 - \frac{1}{3}(2^2 + 2^2 + 1 + 1) = 4 - \frac{10}{3} = \frac{2}{3} > \frac{1}{2};$$

which is a contradiction and the graph is a triangle. \blacktriangleleft

Putting the pieces together, we get the proof of Theorem 4.1.

► **Remark (Unique Games Hardness).** Khot and Regev [22] show that approximating Vertex-Cover to factor $(2 - \epsilon)$ is hard assuming the Unique Games conjecture. Furthermore, Kortsarz et al. [24] show that any approximation algorithm with ratio $\alpha \geq 1.5$ for Vertex-Cover on 3-cycle-free graphs implies an α approximation algorithm for Vertex-Cover (on general graphs). This result combined with the reduction in this section immediately implies APX hardness for k -means under the unique games conjecture. In the next section we generalize the result of Kortsarz et al. [24] by giving an approximation preserving reduction from Vertex-Cover on general graphs to Vertex-Cover on triangle-free graphs. This would enable us to get APX hardness for the k -means problem.

5 Hardness of Vertex Cover on Triangle-Free Graphs

In this section, we show that the Vertex Cover problem is as hard on triangle-free graphs as it is on general graphs. To this end, for any graph $G = (V, E)$, we define $\text{IS}(G)$ as the size of maximum independent set in G . For convenience, we define $\text{rel-IS}(G)$ as the ratio of $\text{IS}(G)$ to the number of nodes in G :

$$\text{rel-IS}(G) \stackrel{\text{def}}{=} \frac{\text{IS}(G)}{|V|}.$$

Similarly, let $\text{VC}(G)$ be the size of minimum vertex cover in G and $\text{rel-VC}(G)$ be the ratio $\frac{\text{VC}(G)}{|V|}$. The following is well known, which says independent sets and vertex covers are duals of each other.

► **Proposition 5.1.** *Given $G = (V, E)$, $I \subseteq V$ is an independent set if and only if $C = V \setminus I$ is a vertex cover. In particular, $\text{IS}(G) + \text{VC}(G) = |V|$.*

We will prove the following theorem.

► **Theorem 5.2.** *For any constant $\epsilon > 0$, there is a $(1 + \epsilon)$ -approximation-preserving reduction for independent set from any graph $G = (V, E)$ with maximum degree Δ to triangle-free graphs with $\text{poly}(\Delta, \epsilon^{-1})|V|$ nodes and degree $\text{poly}(\Delta, \epsilon^{-1})$ in deterministic polynomial time.*

Combining Theorem 5.2 with the best known unconditional hardness result for Vertex Cover, due to Dinur and Safra [14], we obtain the following corollary.

► **Corollary 5.3.** *Given any unweighted triangle-free graph G with bounded degrees, it is NP-hard to approximate Vertex Cover within any factor smaller than 1.36.*

Given two simple graphs $G = (V_1, E_1)$ and $H = (V_2, E_2)$, we define the Kronecker product of G and H , $G \otimes H$, as the graph with nodes $V(G \otimes H) = V_1 \times V_2$ and edges:

$$E(G \otimes H) = \left\{ \{(u, i), (v, j)\} \mid \{u, v\} \in E(G), \{i, j\} \in E(H) \right\}.$$

Observe that, if A_G and A_H denote the adjacency matrix of G and H , then $A_{G \otimes H} = A_G \otimes A_H$.

Given any symmetric matrix M , we will use $\sigma_i(M)$ to denote the i^{th} largest eigenvalue of M . For any graph G on n -nodes, we define the spectral radius of G , $\rho(G)$, as the following:

$$\rho(G) \stackrel{\text{def}}{=} \max_{p \perp \mathbf{e}} \frac{|p^T A_G p|}{\|p\|^2} = \max(\sigma_2(A_G), |\sigma_n(A_G)|).$$

Here \mathbf{e} is the all 1's vector of length n .

► **Proposition 5.4.** *If H is triangle-free, then so is $G \otimes H$.*

Proof. Suppose $G \otimes H$ has a 3-cycle of the form $((a, i), (b, j), (c, k), (a, i))$. Then (i, j, k, i) is a closed walk in H . H is triangle-free, therefore $i = j$ wlog; a contradiction as H has no loops. ◀

The following Lemma says that as long as H has good spectral properties, the relative size of maximum independent sets in G will be preserved by $G \otimes H$.

► **Lemma 5.5.** *Suppose H is a d -regular graph with spectral radius $\leq \rho$. For any graph G with maximum degree Δ , $\text{rel-IS}(G \otimes H) \geq \text{rel-IS}(G) \geq \left(1 - \frac{\rho\Delta}{2d}\right) \text{rel-IS}(G \otimes H)$.*

Proof. Suppose $V(G) = [n]$ and $V(H) = [N]$. Let $A \stackrel{\text{def}}{=} A_G$ be the adjacency matrix of G and B be the normalized adjacency matrix of H , $B \stackrel{\text{def}}{=} \frac{1}{d}A_H$.

For the lower bound, consider an independent set I in G . It is easy to check that $I \times [N]$ is an independent set in $G \otimes H$, thus $\text{IS}(G \otimes H) \geq N \cdot \text{IS}(G)$ so $\text{rel-IS}(G \otimes H) \geq \text{rel-IS}(G)$.

For the upper bound, consider the indicator vector $f \in \{0, 1\}^{[n] \times [N]}$ of an independent set in $G \otimes H$. The corresponding set contains no edges from $G \otimes H$, so $f^T(A \otimes B)f = 0$. Define $p : V \rightarrow [0, 1]$ as $p_u \stackrel{\text{def}}{=} \frac{1}{N} \sum_{j \in [N]} f_{u,j}$. For each $u \in [n]$, pick u with probability p_u . Let $I_0 \subseteq [n]$ be the set of picked nodes. Next, start with $I \leftarrow I_0$. As long as there is an edge of G contained in I , arbitrarily remove one of its endpoints from I . At the end of this process, the remaining set I is an independent set in G , and its size is at least the size of I_0 minus the number of edges contained in I_0 . Hence $|I| \geq |I_0| - |E_G(I_0, I_0)|$. Observe that

$$\mathbb{E}[|I_0|] = \sum_u p_u = \frac{1}{N} \|f\|^2 \quad \text{since } f \text{ is a } \{0, 1\} \text{ vector.}$$

The probability of any pair $i \neq j$ being contained in I_0 is given by $\text{Prob}[\{i, j\} \subseteq I_0] = p_i p_j$. Therefore, the expected number of edges contained in I_0 is:

$$\begin{aligned} \mathbb{E}[|E_G(I_0, I_0)|] &= \sum_{u < v} A_{uv} p_u p_v = \frac{1}{2} p^T A p \stackrel{(\text{Theorem 5.6})}{=} \frac{1}{2N} f^T (A \otimes \widetilde{J}_N) f \\ &\stackrel{(\text{Theorem 5.7})}{\leq} \frac{1}{2N} \left| f^T A \otimes (\widetilde{J}_N - B) f \right| \stackrel{(\text{Theorem 5.8})}{\leq} \frac{\rho\Delta}{2Nd} \|f\|^2. \end{aligned}$$

Putting it all together:

$$\mathbb{E}[|I|] \geq \mathbb{E}[|I_0|] - \mathbb{E}[|E_G(I_0, I_0)|] = \frac{1}{N} \|f\|^2 - \frac{1}{2} p^T A p \geq \frac{\|f\|^2}{N} \left(1 - \frac{\rho\Delta}{2d}\right).$$

Therefore, $\text{IS}(G) \geq \frac{1 - \frac{\rho\Delta}{2d}}{N} \text{IS}(G \otimes H) \implies \text{rel-IS}(G) \geq \left(1 - \frac{\rho\Delta}{2d}\right) \text{rel-IS}(G \otimes H)$. ◀

In the remaining part, we prove the supporting claims.

► **Claim 5.6.** $p^T A p = \frac{1}{N} f^T (A \otimes \widetilde{J}_N) f$ where \widetilde{J}_N is the N -by- N matrix of all $1/N$'s.

Proof. Let $\mathbf{e}^{u,v} \in \mathbb{R}^{V \times V}$ be the matrix whose entry at u^{th} row and v^{th} column is 1, and all others 0. Notice $A = \sum_{u,v} A_{u,v} \mathbf{e}^{u,v}$. Let J_N be the N -by- N matrix of all 1's. For any pair $(u, v) \in V^2$,

$$\begin{aligned} p_u p_v &= \frac{1}{N^2} \sum_{i,j} f_{u,i} f_{v,j} = \frac{1}{N^2} f^T (\mathbf{e}^{u,v} \otimes J_N) f = \frac{1}{N} f^T (\mathbf{e}^{u,v} \otimes \widetilde{J}_N) f. \\ p^T A p &= \frac{1}{N} \sum_{u,v} A_{u,v} f^T (\mathbf{e}^{u,v} \otimes \widetilde{J}_N) f = \frac{1}{N} f^T \left[\left(\sum_{u,v} A_{u,v} \mathbf{e}^{u,v} \right) \otimes \widetilde{J}_N \right] f \\ &= \frac{1}{N} f^T (A \otimes \widetilde{J}_N) f. \end{aligned}$$

The second-to-last identity follows from the bi-linearity of Kronecker product. ◀

► **Claim 5.7.** $f^T(A \otimes \widetilde{J}_N)f \leq |f^T A \otimes (B - \widetilde{J}_N)f|$.

Proof. We have $f^T(A \otimes \widetilde{J}_N)f = f^T[A \otimes (\widetilde{J}_N - B)]f + f^T(A \otimes B)f$. As noted above, f being an independent set implies $f^T(A \otimes B)f = 0$:

$$f^T(A \otimes \widetilde{J}_N)f = f^T[A \otimes (\widetilde{J}_N - B)]f \leq |f^T A \otimes (\widetilde{J}_N - B)f|. \quad \blacktriangleleft$$

► **Claim 5.8.** $|f^T A \otimes (B - \widetilde{J}_N)f| \leq \frac{\Delta \rho}{d} \|f\|^2$.

Proof. Define $C \stackrel{\text{def}}{=} B - \widetilde{J}_N$. For any symmetric matrix M , let $\rho(M)$ be its spectral radius, $\rho(M) \stackrel{\text{def}}{=} \max_p \frac{|p^T M p|}{\|p\|^2}$. Observe that $\rho(M) = \max(|\sigma_i(M)|)$. We have:

$$|f^T A \otimes (B - \widetilde{J}_N)f| = |f^T A \otimes C f| \leq \rho(A \otimes B) \|f\|_2^2.$$

We know that the spectrum of the Kronecker product of two symmetric matrices correspond to the pairwise product of the spectrum of corresponding matrices, i.e., all eigenvalues of $A \otimes C$ are of the form $\sigma_i(A) \cdot \sigma_j(C)$ for each i and j . Therefore,

$$\rho(A \otimes C) = \max(|\sigma_i(A)\sigma_j(C)|) = \max(|\sigma_i(A)|) \max(|\sigma_j(C)|) = \rho(A) \cdot \rho(C).$$

Observe that $\rho(A) \leq \Delta$, since A is the adjacency matrix of a graph with degree $\leq \Delta$. Now we will upper bound $\rho(C)$. Since H is a regular graph and B is its normalized adjacency matrix, the largest eigenvector of B is all 1's and the corresponding eigenvalue is 1. Therefore C has the same eigenspace with B . Moreover $C\mathbf{e} = 0$, thus:

$$\begin{aligned} \rho(C) &= \max(|\sigma_i(C)| : 1 \leq i \leq n) = \max(|\sigma_i(B)| : 2 \leq i \leq n) \\ &= \max(\sigma_2(B), |\sigma_n(B)|) = \frac{1}{d} \rho(G). \end{aligned} \quad \blacktriangleleft$$

We now prove the main theorem needed for our reduction.

► **Theorem 5.9.** *Given a graph $G = (V, E)$ with maximum degree Δ , for any $\varepsilon > 0$, we can construct in polynomial time, a triangle-free graph $\widehat{G} = (\widehat{V}, \widehat{E})$ with:*

$$\text{rel-IS}(G) \leq \text{rel-IS}(\widehat{G}) \leq (1 + \varepsilon) \text{rel-IS}(G).$$

Moreover \widehat{G} has (a) $\text{poly}(\Delta, \varepsilon^{-1})|V|$ nodes, (b) degree $O(\Delta^3 \varepsilon^{-2})$.

Proof. For any d and N , it is known how to construct [29, 32] in deterministic polynomial time, a $O(d)$ -regular Ramanujan graph H with girth $\Omega(\log_d N)$ and spectral radius at most $\rho \leq O(\sqrt{d})$. Thus for some choice of $d = O(\Delta^2 \varepsilon^{-2})$ and $N = d^{O(1)} = \text{poly}(\Delta, \varepsilon^{-1})$, we can find a d -regular graph H with girth at least $\Omega(1)$ and spectral radius $\rho \leq d\varepsilon/\Delta$. For such H , let $\widehat{G} \leftarrow G \otimes H$. We have $\left(1 - \frac{\rho\Delta}{2d}\right)^{-1} \leq (1 - \varepsilon/2)^{-1} \leq 1 + \varepsilon$. Proposition 5.4 implies $G \otimes H$ is triangle free. By Theorem 5.5:

$$\text{rel-IS}(G) \leq \text{rel-IS}(G \otimes H) \leq \left(1 - \frac{\rho\Delta}{2d}\right)^{-1} \text{rel-IS}(G) \leq (1 + \varepsilon) \text{rel-IS}(G).$$

Now we prove the remaining properties:

- (a) $|V(G \otimes H)| = |V(G)| \cdot |V(H)| \leq |V| \cdot \text{poly}(\Delta, \varepsilon^{-1})$.
- (b) $d_{\max}(G \otimes H) \leq d_{\max}(G) \times d_{\max}(H) \leq O(\Delta d) = O(\Delta^3 \varepsilon^{-2})$. ◀

► **Note.** Noga Alon has provided an alternate construction where one can obtain a triangle free graph \hat{G} such that $\text{rel-IS}(\hat{G}) = \text{rel-IS}(G)$. This however, does not lead to improved constant in our analysis. For the sake of completeness, we include the alternate theorem in the Appendix (see Theorem A.1).

Before we end the section with the proof of Theorem 5.3, we need the following hardness result from [14], which follows from Corollary 2.3 and Appendix 8 (weighted to unweighted reduction) of [14]. As noted in [14], the construction produces bounded degree graphs.

► **Theorem 5.10** (Dinur, Safra [14]). *For any constant $\varepsilon > 0$, given any unweighted graph G with bounded degrees, it is NP-hard to distinguish between:*

■ (Yes) $\text{rel-IS}(G) > c - \varepsilon$,

■ (No) $\text{rel-IS}(G) < s + \varepsilon$;

where c and s are constants such that $\frac{1-s}{1-c} \approx 1.36$.

Proof of Theorem 5.3. Given a bounded degree graph G , consider the graph \hat{G} given by Theorem 5.9 for some small constant $\varepsilon_0 < \varepsilon$. Since G is bounded degree and ε_0 is constant, \hat{G} is also bounded degree. Furthermore, \hat{G} satisfies $\text{rel-IS}(G) \leq \text{rel-IS}(\hat{G}) \leq (1 + \varepsilon_0) \text{rel-IS}(G)$. Completeness follows immediately: $\text{rel-IS}(\hat{G}) > c - \varepsilon$. For the soundness, suppose $\text{rel-IS}(\hat{G}) > s + \varepsilon$. Then $\text{rel-IS}(G) \geq \frac{s + \varepsilon}{1 + \varepsilon_0} \geq s + \varepsilon$ for suitable ε_0 . The hardness of Vertex Cover follows from Proposition 5.1. ◀

6 Conclusions

In this paper we provide the first hardness of approximation for the fundamental Euclidean k -means problem. Although our work clears a major hurdle of going beyond NP-hardness for this problem, there is still a big gap in our understanding with the best upper bound being a factor $(9 + \epsilon)$. We believe that our result and techniques will pave way for further work in closing this gap. Our reduction from vertex cover produces high dimensional instances ($d = \Omega(n)$) of k -means. However, by using the Johnson-Lindenstrauss transform [12], we can project the instance onto $O(\log n/\epsilon^2)$ dimensions and still preserve pairwise distances by a factor $(1 + \epsilon)$ and the k -means cost by a factor of $(1 + \epsilon)^2$. We leave it as an open question to investigate inapproximability results for k -means in constant dimensions. It would also be interesting to study whether our techniques give hardness of approximation results for the Euclidean k -median problem. Finally, our hardness reduction in Section 5 provides a novel analysis by using the spectral properties of the underlying graph to argue about independent sets in graph products – this connection could have applications beyond the present paper.

Acknowledgments. We would like to thank Noga Alon and Oded Regev for valuable feedback on the results in Section 5, in particular for suggesting alternate proofs of Proposition 5.4 and Theorem 5.5. We would also like to thank Noga for pointing out that the graph product construction in Section 5 does not eliminate even cycles. Finally we thank the anonymous reviewers for their comments.

References

- 1 Daniel Alose, Amit Deshpande, Pierre Hansen, and Preyas Papat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.
- 2 Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, 1992.

- 3 Noga Alon and Joel Spencer. *The Probabilistic Method*. John Wiley, 1992.
- 4 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 the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 106–113, 1998.
- 5 David Arthur and Sergei Vassilvitskii. k -means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 1027–1035, 2007.
- 6 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.
- 7 Mihai Bădoiu, Sariel 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*, pages 250–257, 2002.
- 8 Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2009, New York, NY, USA, January 4–6, 2009*, pages 1068–1077, 2009.
- 9 Jaroslaw Byrka, Thomas Pensyl, Bartosz Rybicki, Aravind Srinivasan, and Khoa Trinh. An improved approximation for k -median, and positive correlation in budgeted optimization. *CoRR*, abs/1406.2951, 2014.
- 10 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.
- 11 Sanjoy Dasgupta. The hardness of k -means clustering. Technical report, University of California, San Diego, 2008.
- 12 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Struct. Algorithms*, 22(1):60–65, 2003.
- 13 Wenceslas Fernandez de la Vega, Marek Karpinski, Claire Kenyon, and Yuval Rabani. Approximation schemes for clustering problems. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9–11, 2003, San Diego, CA, USA*, pages 50–58, 2003.
- 14 Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005.
- 15 Tomás Feder and Daniel H. Greene. Optimal algorithms for approximate clustering. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2–4, 1988, Chicago, Illinois, USA*, pages 434–444, 1988.
- 16 Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A PTAS for k -means clustering based on weak coresets. In *Proceedings of the 23rd ACM Symposium on Computational Geometry, Gyeongju, South Korea, June 6–8, 2007*, pages 11–18, 2007.
- 17 Venkatesan Guruswami and Piotr Indyk. Embeddings and non-approximability of geometric problems. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12–14, 2003, Baltimore, Maryland, USA.*, pages 537–538, 2003.
- 18 Dorit S. Hochbaum and David B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *J. ACM*, 33(3):533–550, 1986.
- 19 Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19–21, 2002, Montréal, Québec, Canada*, pages 731–740, 2002.

- 20 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.
- 21 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.
- 22 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2 - \epsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.
- 23 Stavros G. Kolliopoulos and Satish Rao. A nearly linear-time approximation scheme for the Euclidean k -median problem. *SIAM J. Comput.*, 37(3):757–782, 2007.
- 24 Guy Kortsarz, Michael Langberg, and Zeev Nutov. Approximating maximum subgraphs without short cycles. *SIAM J. Discrete Math.*, 24(1):255–269, 2010.
- 25 Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k -means algorithm. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23–26, 2010, Las Vegas, Nevada, USA*, pages 299–308, 2010.
- 26 Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A simple linear time $(1 + \epsilon)$ -approximation algorithm for k -means clustering in any dimensions. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17–19 October 2004, Rome, Italy, Proceedings*, pages 454–462, 2004.
- 27 Shi Li and Ola Svensson. Approximating k -median via pseudo-approximation. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 901–910, 2013.
- 28 Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.
- 29 Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.
- 30 Meena Mahajan, Prajakta Nimbhorkar, and Kasturi R. Varadarajan. The planar k -means problem is NP-hard. *Theor. Comput. Sci.*, 442:13–21, 2012.
- 31 Jiri Matoušek. On approximate geometric k -clustering. *Discrete and Computational Geometry*, 24(1), 2000.
- 32 Moshe Morgenstern. Existence and explicit constructions of $q + 1$ regular ramanujan graphs for every prime power q . *J. Comb. Theory, Ser. B*, 62(1):44–62, 1994.
- 33 Rafail Ostrovsky, Yuval Rabani, Leonard J. Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k -means problem. *J. ACM*, 59(6):28, 2012.
- 34 Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus F. M. Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, 14(1):1–37, 2008.

A An Alternative Proof of Theorem 5.2

The following were suggested by Noga Alon as an alternative to Theorem 5.2.

► **Theorem A.1.** *Let $G = (V, E)$ be an arbitrary graph with maximum degree Δ . It is possible to construct in polynomial time a triangle free graph \hat{G} such that $\text{rel-IS}(\hat{G}) = \text{rel-IS}(G)$.*

Before proving the theorem, we need the following standard facts about (n, d, λ) graphs.

► **Lemma A.2.** *Let $H = (U, F)$ be an (n, d, λ) graph, assume $\lambda < d/4$ and let B be a set of vertices of H . Let $N(B)$ denote the set of all neighbors of B in H . Then:*

1. *If $|B| > \frac{\lambda}{d}n$, then $|N(B)| > n - \frac{\lambda}{d}n$.*
2. *If $|B| \leq \frac{\lambda}{d}n$, then $|N(B)| \geq \frac{\lambda}{2d}n$.*

Proof. (1) is proved in Corollary 1 in [2]. We will prove (2). When $\frac{2\lambda^2}{d^2}n \leq |B| \leq \frac{\lambda}{d}n$, it follows from the same corollary again, which implies that in this range $|N(B)| \geq \frac{n}{2}$. For $|B| \leq \frac{2\lambda^2}{d^2}n$, the result follows from the expander mixing lemma (see [3], corollary 9.2.5), as there are $d|B|$ edges between B and $N(B)$. ◀

Proof of Theorem A.1. Let $H = (U, F)$ be a (n, d, λ) -expander with $\lambda \leq 2\sqrt{d-1}$ ¹. Let $\hat{G} = G \otimes H$. Further, let $\frac{d}{2\lambda} \geq \Delta$. It is well known that such graphs exist. It is easy to see that $\text{rel-IS}(G \otimes H) \geq \text{rel-IS}(G)$, since any independent set S in G leads to an independent set $S \otimes U$ in $G \otimes H$.

For the other direction, let $S \subset V \times U$ be an independent set in $G \otimes H$. Define T as:

$$T \stackrel{\text{def}}{=} \{v \in V : |\{u \in U : (v, u) \in S\}| \geq \frac{\lambda}{d}n\}.$$

By Lemma A.2 (1), T is an independent set in G . Let T' be a maximal (with respect to containment) independent set in G that contains T . By maximality, every vertex in $V \setminus T'$ has at least one neighbor in T' . Thus T' is a dominating set in G and there is a collection of stars $\{S_v : v \in T'\}$, covering all the vertices of G . As T' is an independent set, $|T'| \leq \text{rel-IS}(G)|V|$. To complete the proof it suffices to show that for each of the stars S_v in our collection, whose set of vertices in G is V_v , we have:

$$|\{(v', u) : (v', u) \in S, v' \in V_v\}| \leq |U| = n \tag{1}$$

The number of leaves of the star S_v is at most Δ . For each such leaf v' , the set of vertices of H given by $B_{v'} \stackrel{\text{def}}{=} \{u \in U : (v', u) \in S\}$ is of cardinality smaller than $\frac{\lambda}{d}n$. Moreover, all its neighbors in H cannot belong to the set $B_v = \{u \in U : (v, u) \in S\}$ where v is the center of the star S_v . By Lemma A.2 (2), the number of these neighbors is at least $\frac{d}{2\lambda} \geq \Delta$ times the cardinality of $B_{v'}$. This implies that the total size of all sets $B_{v'}$ where the sum ranges over all leaves v' of S_v is at most the number of vertices in $U - B_v$, implying (1) and completing the proof. ◀

¹ This means that all non-trivial eigenvalues of H are bounded by λ .

A Fire Fighter’s Problem

Rolf Klein¹, Elmar Langetepe¹, and Christos Levcopoulos²

1 Institute of Computer Science I, University of Bonn, Germany

2 Department of Computer Science, Lund University, Sweden

Abstract

Suppose that a circular fire spreads in the plane at unit speed. A fire fighter can build a barrier at speed $v > 1$. How large must v be to ensure that the fire can be contained, and how should the fire fighter proceed? We provide two results. First, we analyze the natural strategy where the fighter keeps building a barrier along the frontier of the expanding fire. We prove that this approach contains the fire if $v > v_c = 2.6144\dots$ holds. Second, we show that any “spiralling” strategy must have speed $v > 1.618$, the golden ratio, in order to succeed.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, Geometrical problems and computations, G. Mathematics of Computing, G.1.6 Optimization

Keywords and phrases Motion Planning, Dynamic Environments, Spiralling strategies, Lower and upper bounds

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.768

1 Introduction

Fighting wildfires and epidemics has become a serious issue in the last decades. Professional fire fighters need models and simulation tools on which strategic decisions can be based; for example see [5]. Thus, a good understanding of the theoretical foundations seems necessary.

Substantial work has been done on the fire fighting problem in graphs; see, e.g., the survey article [3]. Here, initially one vertex is on fire. Then an immobile firefighter can be placed at one of the other vertices. Next, the fire spreads to each adjacent vertex that is not defended by a fighter, and so on. The game continues until the fire cannot spread anymore. The objective, to save a maximum number of vertices from the fire, is NP-hard to achieve, even for trees.

A more geometric setting has recently been studied in [6]. Suppose that inside a simple polygon P a candidate set of disjoint diagonal barriers has been defined. If a fire starts at some point inside P one wants to build a subset of these barriers in order to save a maximum area from the fire. But each point on a barrier must be built before the fire arrives there. This maximization problem is also NP-hard, even if the candidate barriers are the diagonals of a convex polygon, but there exists an 11.65 approximation algorithm.

In this paper we study a purely geometric version of the fire fighter problem. Suppose there is a circular fire of initial radius A in the plane, centered at the origin. The fire spreads at unit speed. Initially, the plane is empty, except for a single fire fighter who is placed on the boundary of the fire. The fighter can move at speed v , and build a barrier along his path. The fire cannot cross this barrier, and the fighter cannot move into the fire. Will the fighter be able to contain the fire, and how should she proceed to achieve this?

Clearly, the answer depends on speed v . For $v = 1$ the fighter can barely save herself by moving along a straight line away from the fire.



© Rolf Klein, Elmar Langetepe, and Christos Levcopoulos;
licensed under Creative Commons License CC-BY

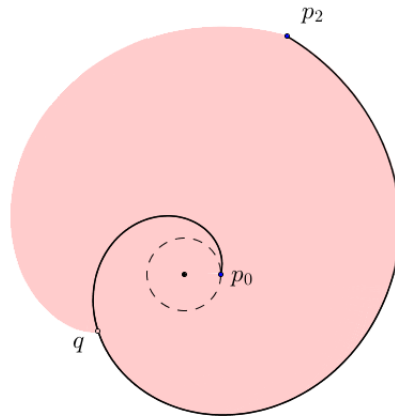
31st International Symposium on Computational Geometry (SoCG’15).

Editors: Lars Arge and János Pach; pp. 768–780



Leibniz International Proceedings in Informatics

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



■ **Figure 1** The race between the fire and the fighter for speed $v = 3.738$. The firebreak was constructed from p_0 to p_2 whereas the fire expands along the outer side of the barrier up to point q . Can the fire fighter finally catch the fire?

At speed $v > 2\pi + 1$, the fire fighter can move a distance x away from the fire and build a complete circular barrier before the fire can reach it. This requires $(x + 2\pi(x + A))/v \leq x$ or $(2\pi + 1) + 2\pi A/x \leq v$.

What happens in between 1 and $2\pi + 1$? *In this paper we show that a speed $v > 2.6144$ is sufficient to contain a fire, and that a speed $v > 1.618$ is necessary, at least for a reasonably large class of strategies.*

The first bound is established in the following way. We consider a conscientious fire fighter who tries to contain the fire by building a barrier along its ever expanding frontier, at her maximum speed v . Let us denote this strategy by FF (short for Follow Fire). A spiralling barrier curve results. While the fighter keeps building the barrier, the fire is coming after her along the outside of the barrier, as shown in Figure 1. Intuitively, the fighter can only win this race, and contain the fire, if the last coil of the barrier hits the previous one.

In the hand-drawn example shown in Figure 2 this happens in the second round if $v = 4.1932$; but for smaller values of v , more rounds may be necessary.

We have the following result.

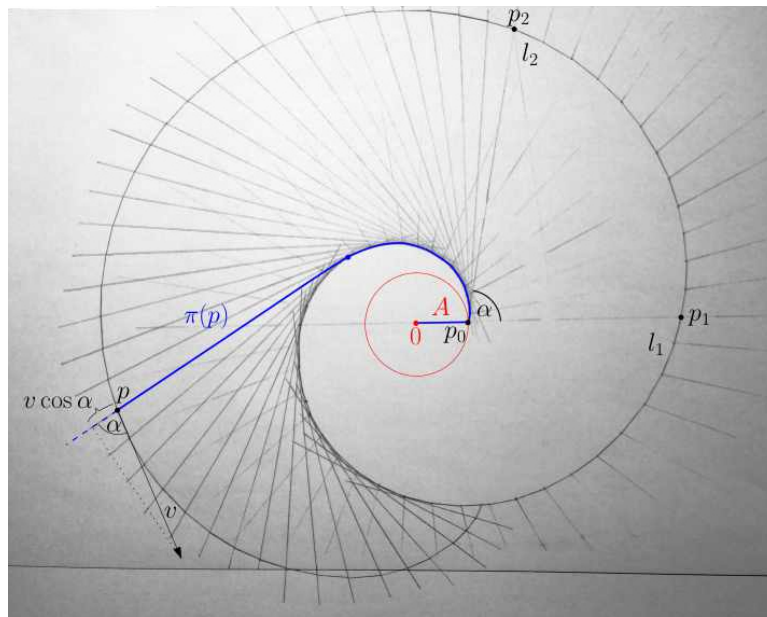
► **Theorem 1.**

- (i) *Strategy FF contains the fire if $v > v_c \approx 2.6144$ holds.*
- (ii) *As v decreases to v_c , the number of rounds to containment tends to infinity.*

Although strategy FF is rather simple, the proof of Theorem 1 is not. First, we establish a recursive system of linear differential equations associated with each round. They can be solved easily by standard methods, but the resulting recursions are complicated. Therefore, we apply techniques from analytic combinatorics. We look at the generating function $F(Z)$ that arises from these recursions, and find a presentation of $F(Z)$ as a ratio of analytic functions. The denominator equals

$$e^{wZ} - sZ = 0, \tag{1}$$

where $w = \frac{2\pi + \alpha}{\sin \alpha}$ and $s = e^{(2\pi + \alpha)\cot \alpha}$ are functions of a real variable α which equals $\cos^{-1}(1/v)$ in our setting. Our targets are the coefficients of $F(Z)$; they are linked to the zeroes of equation 1.



■ **Figure 2** At speed $v = 4.1932$ the fire will be fully contained by the fire fighter’s barrier in the second round.

Let $\alpha_c \approx 1.1783$ be the smallest positive solution of $s = ew$, corresponding to $v_c \approx 2.6144$. For this value of α , equation 1 has a real zero $Z = 1/w$, as direct substitution shows. For $\alpha > \alpha_c$, corresponding to $v > v_c$, this real zero splits into a complex zero $z_0 = \rho(\cos \phi + \sin \phi i)$ and its conjugate, where $\phi \in (0, \pi)$, and no real zeroes of equation 1 remain.

At this point, part (i) of Theorem 1 follows from a Theorem of Pringsheim’s in complex function theory; see Section 6. To find out how many rounds it takes to contain the fire, we apply Cauchy’s residue theorem and find that their number is $\approx \pi/\phi$. Since ϕ , the angle of the complex root z_0 , tends to zero as z_0 becomes real for $\alpha \rightarrow \alpha_c$, part (ii) of Theorem 1 also follows. How j , the number of rounds, depends on v is shown in Figure 3. For speeds $v \geq 3$ strategy FF needs at most 4 rounds to contain the fire.

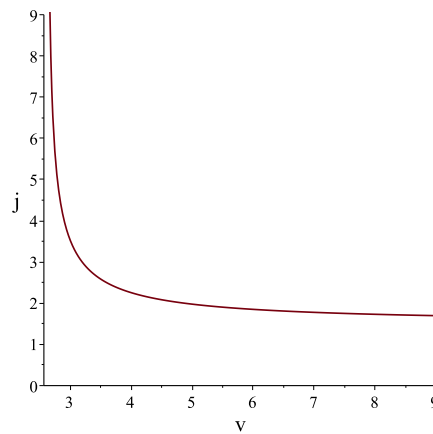
In addition to the above upper bound we prove the following lower bound. To this end we restrict ourselves to the class of “spiralling” strategies that visit the four coordinate half-axes in cyclic order, and at increasing distances from the origin. Note that strategy FF is spiralling even though the fighter’s distance to the origin may be decreasing: the barrier’s intersection points with any ray from 0 are in increasing order since the curve does not self-intersect. Here we have the following.

► **Theorem 2.** *In order to enclose the fire, a spiralling strategy must be of speed*

$$v > \frac{1 + \sqrt{5}}{2} \approx 1.618,$$

the golden ratio.

The proof of Theorem 2 is given in Section 7. An (almost) complete proof of Theorem 1 (i) is given in the main text; only for some details we refer to the technical report of this paper; see [7]. Proving part (ii) of Theorem 1 requires considerably more work; we sketch only the essential ideas in the main text. A complete proof of (i) and (ii), which can be read independently of the main text, is given in the Appendix of the technical report [7].



■ **Figure 3** The approximate number of rounds needed by strategy FF, as a function of speed v .

2 The barrier curve generated by strategy FF

We would like to show how the barrier curve shown in Figure 2 has been developed. A more detailed view of the starting situation of Figure 2 from p_0 to p_2 is depicted in Figure 4.

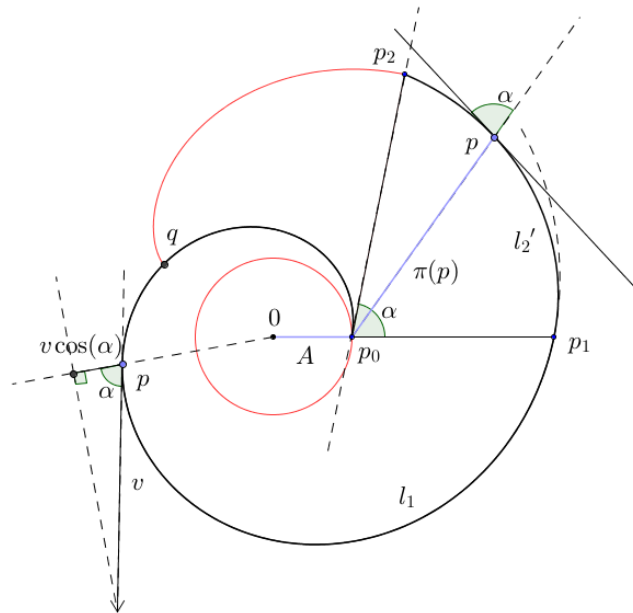
Consider some point p in the first round between p_0 and p_1 as shown in Figure 4. If α denotes the angle between the fighter’s velocity vector at p and the ray from 0 through p , the fighter advances at speed $v \cos \alpha$ away from 0. This implies $v \cos \alpha = 1$ because the fire expands at unit speed and the fighter stays on its frontier, by definition of strategy FF. Consequently, the barrier curve between p_0 and p_1 is part of a logarithmic spiral centered at 0, whose tangents forms the angle $\alpha = \cos^{-1}(1/v)$ with the extensions of the rays from 0 through p .

In polar coordinates a logarithmic spiral (with excentricity α) is defined by $(\varphi, A \cdot e^{\varphi \cot \alpha})$ and the barrier curve from p_0 to p_1 is represented by the interval $\varphi \in [0, 2\pi]$. The curve length of the logarithmic spiral of excentricity α around origin O between two points C and D appearing on the spiral in this order is given by $\frac{1}{\cos \alpha} (|DO| - |CO|)$, where $|CO|$ and $|DO|$ denote the distances from D and C to the origin 0, respectively. Thus, for example the curve length from p_0 to p_1 is given by $l_1 = \frac{A}{\cos(\alpha)} \cdot (e^{2\pi \cot(\alpha)} - 1)$.

From point p_1 on, the geodesic shortest paths $\pi(p)$ from 0 to p , along which the fire spreads, start with segment $0p_0$, followed by segment p_0p , until the fighter reaches the point p_2 on the barrier’s tangent to p_0 ; see Figure 4. Thus, by the previous argument, between p_1 and p_2 the barrier curve constructed by FF is part of a logarithmic spiral of excentricity α now centered at p_0 . This spiral starts at p_1 with distance $A' = A(e^{2\pi \cot(\alpha)} - 1)$ from its origin p_0 , and the curve length from p_1 to p_2 is given by $l'_2 = \frac{A'}{\cos(\alpha)} (e^{\alpha \cot(\alpha)} - 1) = \frac{A}{\cos(\alpha)} (e^{2\pi \cot(\alpha)} - 1)(e^{\alpha \cot(\alpha)} - 1)$. This means that the overall curve length from p_0 to p_2 is given by $l_1 + l'_2 = l_2 = \frac{A}{\cos(\alpha)} (e^{2\pi \cot(\alpha)} - 1)e^{\alpha \cot(\alpha)}$.

How does the curve constructed by FF develop from p_2 on? We turn over to Figure 2. From p_2 on, the geodesic shortest path $\pi(p)$ from 0 to fighter’s current position p starts wrapping around the existing spiral part of the curve, beginning at p_0 . The last edge of $\pi(p)$ ending at p will be called the *free string* in the sequel. The fire will be contained if and only if the free string ever attains length 0.

Thus, after the first round the curve is drawn by endpoint p of the free string. But unlike an involute, the string is not normal to the outer layer. Rather, its extension beyond p forms the angle α with the barrier’s tangent at p . This causes the string to grow in length by $\cos \alpha$



■ **Figure 4** The first part of the barrier curve constructed by FF consists of two different logarithmic spirals of eccentricity α where $\alpha = \cos^{-1}(1/v)$ holds. Namely, a logarithmic spiral around the origin 0 from p_0 to p_1 and a logarithmic spiral around p_0 from p_1 to p_2 . At p_2 the fire fighter’s curve starts wrapping around the constructed barrier as show in Figure 2.

for each unit drawn. At the same time, part of the string gets wrapped around the inner layer. It is this interplay between growing and shrinking that we will investigate below. Note that the curve starting at p_2 is no longer a logarithmic spiral.

As the fighter is building the barrier at speed $1/\cos \alpha$, the fire is coming after her at unit speed along the outside of the barrier, as indicated in Figure 1. Thus, each barrier point p is caught by fire twice, once from the inside, when the fighter passes through p , and a second time from the outside, if the fire is not stopped before.

3 Linkages

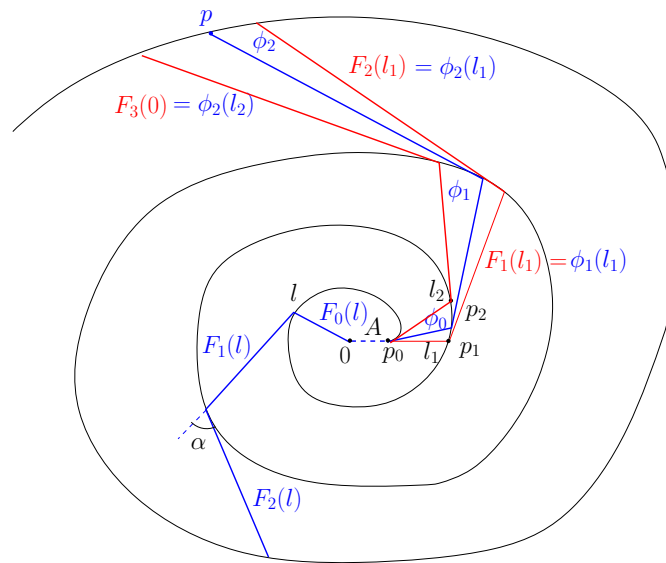
That the innermost part of the curve consists of two different spiral segments, around 0 and around p_0 , carries over to subsequent layers. The structure of the curve can be described as follows. Let

$$l_1 = \frac{A}{\cos(\alpha)} \cdot (e^{2\pi \cot(\alpha)} - 1)$$

$$l_2 = \frac{A}{\cos(\alpha)} \cdot (e^{2\pi \cot(\alpha)} - 1)e^{\alpha \cot(\alpha)}$$

denote the curve lengths from p_0 to p_1 and p_2 , respectively, as derived before in Section 2. For $l \in [0, l_1]$ let $F_0(l)$ denote the segment connecting 0 to the point of curve length l ; see the sketch given in Figure 5.

At the endpoint of $F_0(l)$ we construct the tangent and extend it until it hits the next layer of the curve, creating a segment $F_1(l)$, and so on. This construction gives rise to a “linkage” connecting adjacent layers of the curve. Each edge of the linkage is turned counterclockwise by α with respect to its predecessor. The outermost edge of a linkage is the free string



■ **Figure 5** A sketch of the general situation. Two types of linkages defining subsegments of the curve.

mentioned above. As parameter l increases from 0 to l_1 , edge $F_0(l)$, and the whole linkage, rotate counterclockwise. While $F_0(0)$ equals the line segment from the center to p_0 , edge $F_0(l_1)$ equals segment $0p_1$.

Analogously, let $l \in [l_1, l_2]$, and let $\phi_0(l)$ denote the segment from p_0 to the point at curve length l from p_1 . This segment can be extended into a linkage in the same way. We observe that

$$F_{j+1}(l_1) = \phi_{j+1}(l_1) \tag{2}$$

$$F_{j+1}(0) = \phi_j(l_2) \tag{3}$$

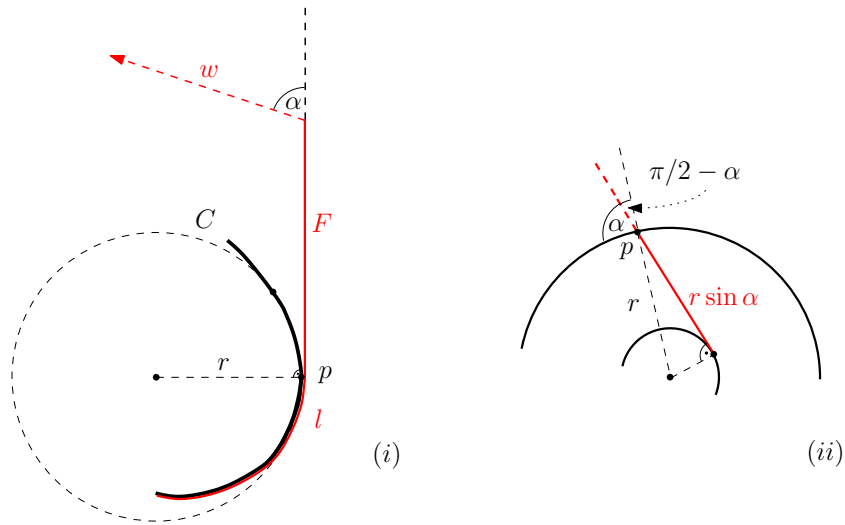
hold. But initially, we have $F_0(l) = A + \cos(\alpha)l$ and $\phi_0(l) = \cos(\alpha)l$, so that $F_0(l_1) \neq \phi_0(l_1)$. Clearly, each point on the curve can be reached by a linkage, as tangents can be constructed backwards. We refer to the two types of linkages by F -type and ϕ -type.

4 Analysis

A detailed proof of the following general facts is given in the Appendix of the technical report [7] in Lemma 7 and 8. We present the intuitive ideas here.

As the endpoint of a taut string of length F , tangent to a smooth curve C at some point p , is moved in direction α , as shown in Figure 6 (i), the length l of the wrapped string grows at rate $r \sin \alpha / F$, where r denotes the curve's radius of curvature at p . (Intuitively, the more perpendicular motion w acts on the string and the larger the osculating circle, the more of the string gets wrapped; but the larger F , the smaller is the effect of the perpendicular motion.)

The center of the osculating circle at p is known to be the limit of the intersections of the normals of all points near p with the normal at p . If, instead of the normals, we consider the lines turned by the angle $\pi/2 - \alpha$, their limit intersection point has distance $r \sin \alpha$ from p ; an example is shown in Figure 6 (ii) for the case where curve C itself is a circle.



■ **Figure 6** In (i), the wrapped string grows at a rate of $r \sin \alpha/F$. In (ii), the turned normals meet at a point $r \sin \alpha$ away from p .

For the barrier curve, the limit intersection point of the turned normals near p is just the tangent point from p to the previous layer of the curve. If we denote by L_i the length of the barrier curve from p_0 to the outer endpoint of the i th edge of an F -linkage, the above observations imply the following for L_{j-1}, F_j and F_{j-1} as functions of L_j .

$$\frac{L'_{j-1}}{L'_j} = \frac{L'_{j-1}}{1} = \frac{r \sin \alpha}{F_j} = \frac{F_{j-1}}{F_j}.$$

Now we change the former variable L_j to $L_j(l)$ for $l \in [0, l_1]$ introduced in Section 3. Observing that the derivatives of the inner functions cancel out we obtain

► **Lemma 3.**

$$\frac{L'_{j-1}(l)}{L'_j(l)} = \frac{F_{j-1}(l)}{F_j(l)}.$$

By multiplication, Lemma 3 generalizes to non-consecutive edges. Thus,

$$\frac{F_j(l)}{F_0(l)} = \frac{L'_j(l)}{l'} = L'_j(l) \tag{4}$$

holds.

On the other hand, a point p on the j th layer of the barrier curve has geodesic distance $L_{j-1}(l) + F_j(l)$ from the initial fire of radius A , and the fire arrives at p (from the inside) simultaneously with the fighter, who has then completed a barrier of length $L_j(l)$ at speed $1/\cos \alpha$. This yields, $F_j(l) + L_{j-1}(l) = \cos \alpha L_j(l)$ and after taking derivatives,

$$F'_j(l) + L'_{j-1}(l) = \cos \alpha L'_j(l). \tag{5}$$

From 5 and 4 we obtain a linear differential equation for $F_j(l)$,

$$F'_j(l) - \frac{\cos(\alpha)}{F_0(l)} F_j(l) = -\frac{F_{j-1}(l)}{F_0(l)}.$$

The textbook solution for $y'(x) + f(x)y(x) = g(x)$ is

$$y(x) = \exp(-a(x)) \left(\int g(t) \exp(a(t)) dt + \kappa \right),$$

where $a = \int f$ and κ denotes a constant that can be chosen arbitrarily. In our case,

$$a(l) = \int -\frac{\cos(\alpha)}{A + \cos(\alpha)l} = -\ln(F_0(l))$$

because of $F_0(l) = A + \cos(\alpha)l$, and we obtain

$$F_j(l) = F_0(l) \left(\kappa_j - \int \frac{F_{j-1}(t)}{F_0^2(t)} dt \right). \tag{6}$$

Next, we consider a linkage of ϕ -type, for parameters $l \in [l_1, l_2]$, and obtain analogously

$$\phi_j(l) = \phi_0(l) \left(\lambda_j - \int \frac{\phi_{j-1}(t)}{\phi_0^2(t)} dt \right). \tag{7}$$

Now we determine the constants κ_j, λ_j such that the solutions 6 and 7 describe a contiguous curve. To this end, we must satisfy conditions 2 and 3.

We define $\kappa_0 := 1$ and

$$\kappa_{j+1} := \frac{\phi_j(l_2)}{F_0(0)} + \int \frac{F_j(t)}{F_0^2(t)} dt|_{l=0}$$

so that 6 becomes

$$F_{j+1}(l) = F_0(l) \left(\frac{\phi_j(l_2)}{F_0(0)} - \int_0^l \frac{F_j(t)}{F_0^2(t)} dt \right),$$

which, for $l = 0$, yields $F_{j+1}(0) = \phi_j(l_2)$ (condition 3).

Similarly, we set $\lambda_0 := 1$ and

$$\lambda_{j+1} := \frac{F_{j+1}(l_1)}{\phi_0(l_1)} + \int \frac{\phi_j(t)}{\phi_0^2(t)} dt|_{l=l_1}$$

so that 7 becomes

$$\phi_{j+1}(l) = \phi_0(l) \left(\frac{F_{j+1}(l_1)}{\phi_0(l_1)} - \int_{l_1}^l \frac{\phi_j(t)}{\phi_0^2(t)} dt \right),$$

and for $l = l_1$ we get $F_{j+1}(l_1) = \phi_{j+1}(l_1)$ (condition 2).

For simplicity, let us write

$$G_j(l) := \frac{F_j(l)}{F_0(l)} \quad \text{and} \quad \chi_j(l) := \frac{\phi_j(l)}{\phi_0(l)}, \tag{8}$$

which leads to

$$G_{j+1}(l) = \frac{\phi_0(l_2)}{F_0(0)} \chi_j(l_2) - \int_0^l \frac{G_j(t)}{F_0(t)} dt \tag{9}$$

$$\chi_{j+1}(l) = \frac{F_0(l_1)}{\phi_0(l_1)} G_{j+1}(l_1) - \int_{l_1}^l \frac{\chi_j(t)}{\phi_0(t)} dt. \tag{10}$$

In order to find out if the fire fighter is successful we only need to check the values of $F_j(l)$ at the end of each round, as the following lemma shows.

► **Lemma 4.** *The curve encloses the fire if and only if there exists an index j such that $F_j(l_1) \leq 0$ holds.*

Proof. The free string shrinks to zero if and only if there exist an index j and argument l such that $F_j(l) \leq 0$ or $\phi_j(l) \leq 0$. Clearly, G_j and F_j have identical signs, as well as χ_j and ϕ_j do. Suppose that $G_j > 0$ and $G_{j+1}(l) = 0$, for some j and some $l \in [0, l_1]$. By 9, function G_{j+1} is decreasing, therefore $G_{j+1}(l_1) \leq 0$. Now assume that $G_i > 0$ holds for all i , and that we have $\chi_{j-1} > 0$ and $\chi_j(l) = 0$ for some j and some $l \in [l_1, l_2]$. By 10 this implies $\chi_j(l_2) \leq 0$, and from 9 we conclude $G_{j+1} \leq 0$, in particular $G_{j+1}(l_1) \leq 0$. ◀

5 Recursions

The integrals in 9 and 10 disappear by iterated substitution. This process is not entirely trivial, and the calculations can be found in Section C in the Appendix of the technical report [7]. After plugging in values, one obtains cross-wise recursions

$$F_j(l_1) = \frac{F_0(l_1)}{F_0(0)} \sum_{\nu=0}^j \frac{(-1)^\nu}{\nu!} \left(\frac{2\pi}{\sin \alpha}\right)^\nu \phi_{j-1-\nu}(l_2) \tag{11}$$

$$\phi_j(l_2) = \frac{\phi_0(l_2)}{\phi_0(l_1)} \sum_{\nu=0}^j \frac{(-1)^\nu}{\nu!} \left(\frac{\alpha}{\sin \alpha}\right)^\nu \hat{F}_{j-\nu}(l_1) \tag{12}$$

where $\phi_{-1}(l_2) := F_0(0)$, $\hat{F}_0(l_1) := \phi_0(l_1)$, and $\hat{F}_{i+1}(l_1) := F_{i+1}(l_1)$.

In order to solve the cross-wise recursions 11 and 12 for the numbers $F_j(l_1)$ we define the formal power series

$$F(X) := \sum_{j=0}^{\infty} F_j X^j \quad \text{and} \quad \phi(X) := \sum_{j=0}^{\infty} \phi_j X^j$$

where $F_j := F_j(l_1)$ and $\phi_j := \phi_j(l_2)$, for short. From 11 we obtain

$$F(X) = \frac{F_0}{F_0(0)} e^{-\frac{2\pi}{\sin \alpha} X} (X \phi(X) + F_0(0)), \tag{13}$$

and from 12,

$$\phi(X) = \frac{\phi_0}{\phi_0(l_1)} e^{-\frac{\alpha}{\sin \alpha} X} (X F(X) - F_0 + \phi_0(l_1)); \tag{14}$$

both equalities can be easily verified by computing the products and comparing coefficients. Now we substitute 14 into 13, solve for $F(X)$, divide both sides by F_0 and expand by $e^{\frac{2\pi+\alpha}{\sin \alpha} X}$ to obtain

$$\frac{F(X)}{F_0} = \frac{e^{vX} - rX}{e^{wX} - sX}, \tag{15}$$

where v, r, w, s are the following functions of α :

$$\begin{aligned} v &= \frac{\alpha}{\sin \alpha} & \text{and} & \quad r = e^{\alpha \cot \alpha} \\ w &= \frac{2\pi + \alpha}{\sin \alpha} & \text{and} & \quad s = e^{(2\pi+\alpha) \cot \alpha}. \end{aligned} \tag{16}$$

Note that here the parameter v does no longer represent the speed parameter, the speed is given by $\frac{1}{\cos \alpha}$.

It is possible to expand the inverse of the denominator in 15 into a power series. This leads to interesting expressions for the F_j ; but how to derive their signs seems not obvious.

6 Singularities and Residues

Now we consider the right hand side of (15) as a function

$$f(z) := \frac{e^{vz} - rz}{e^{wz} - sz}, \tag{17}$$

of a complex variable, z . Both numerator and denominator of f are analytic on the complex plane. Thus, singularities of f can only arise from zeroes of the denominator $e^{wz} - sZ$. This equation has received some attention in the area of delay differential equations [2]. As in the Introduction, let $\alpha_c \approx 1.1783$ be the unique solution of $s = ew$ in $(0, \pi/2]$, corresponding to speed $v_c = 1/\cos \alpha_c \approx 2.6144$.

► **Lemma 5.** *For $\alpha = \alpha_c$, equation $e^{wZ} - sZ$ has a real root $1/w \approx 0.1238$. For $\alpha > \alpha_c$ (corresponding to speed $v > v_c$), this root splits into a complex conjugate pair z_0 and \bar{z}_0 , whose absolute values are < 0.31 . All other zeroes of numerator and denominator in 15 are strictly complex, and of absolute values ≥ 1 . Function $f(z)$ in 17 has only poles as singularities.*

For a proof of Lemma 5 see Lemmata 10 to 13 in the Appendix of the technical report [7].

From now on we assume that $\alpha > \alpha_c$ holds. Now we would like to make use of a general Theorem concerning the sign of coefficients of power series within their convergence radius, in order to prove the first part of Theorem 1.

► **Theorem 6** (Pringsheim’s Theorem (see for example [4, p. 240])). *Let $h(z) = \sum_{n=0}^{\infty} a_n z^n$ be a power series with finite convergence radius R . If $h(z)$ has non-negative coefficients, a_j , then point $z = R$ is a singularity of $h(z)$.*

Proof of Theorem 1 (i). Let $\alpha > \alpha_c$. Because of the singularities z_0 and \bar{z}_0 , the power series expansion of $f(z)$ in 17 has a finite radius, R , of convergence. If all coefficients F_i were ≥ 0 then, by Pringsheim’s Theorem function $f(z)$ would have a singularity at R . But, by Lemma 5, there can be only complex singularities. Thus, there must be coefficients $F_j < 0$, proving that the fire fighter succeeds. ◀

Now we sketch the proof of Theorem 1(ii). A complete version can be found in the Appendix Sections E and F of the technical report [7]. This will also lead to another, and constructive, proof of part (i) of Theorem 1.

We are using a technique described in [4, p. 258 ff.]. Let Γ denote the circle of radius 0.9 around the origin. By Cauchy’s Residue Theorem,

$$\frac{1}{2\pi i} \int_{\Gamma} \frac{f(u)}{u^{j+1}} du = \sum_{z \text{ inside } \Gamma} \text{res}(z)$$

holds, where the sum is over all residues of the poles of $\frac{f(z)}{z^{j+1}}$ encircled by Γ . By Lemma 5, these poles are z_0 , \bar{z}_0 , and 0, which has residue F_j/F_0 . Computing the residues of z_0 , \bar{z}_0 yields

$$\frac{F_j}{F_0} = \sin(j\phi + p) \frac{|z_0|^{-j}}{|z_0 - x_0|} \Theta(1) + \frac{1}{2\pi i} \int_{\Gamma} \frac{f(u)}{u^{j+1}} du, \tag{18}$$

where $z_0 = \rho(\cos \phi + \sin \phi i)$, with $0 < \phi < \pi$, and $x_0 = (1/w, 0)$ is the limit of z_0 as α_c tends to α . The rightmost term’s absolute value is upper bounded by the maximum of $|f(z)|$ on Γ , times 0.9^{-j} ; its influence turns out to be negligible.

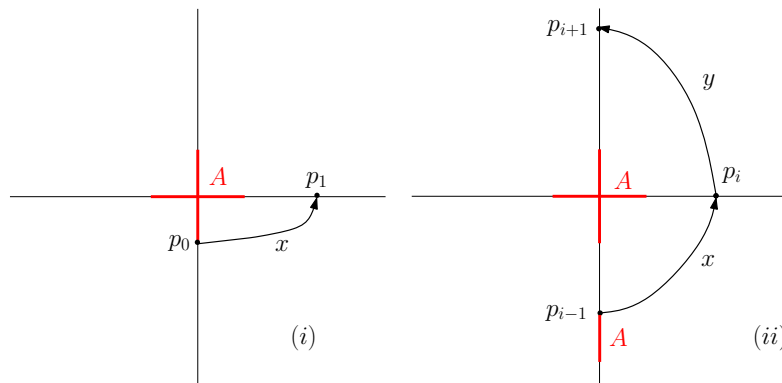


Figure 7 Proof of Lemma 7.

The oscillation $\sin(t\phi + p)$ has wavelength $2\pi/\phi$. For j near its negative minimum, the value of 18 becomes negative. This proves that the fire fighter will succeed in containing the fire in round j , for some $j \leq c \cdot 2\pi/\phi$ (in fact, one can choose $c = 1$). As α decreases towards α_c , both ϕ and phase p tend to zero, but

$$\lim_{\alpha \rightarrow \alpha_c} \frac{p}{\phi} \approx 1.315$$

holds. This value denotes how much the graph of $\sin(t\phi + p)$ is shifted to the left, as compared to $\sin t$. We see that j must increase through almost the whole positive halfwave of $\sin(t\phi + p)$ before negative values can occur. Since wavelength $2\pi/\phi$ goes to infinity, so does the number of rounds the fire fighter needs. This completes the proof of Theorem 1. All details are given in the Appendix of the technical report [7].

7 Lower bound

Let us recall that a barrier building strategy S is *spiralling* if it starts on the boundary of a fire of radius A , and visits the four coordinate half-axes in counterclockwise order and at increasing distances from the origin.

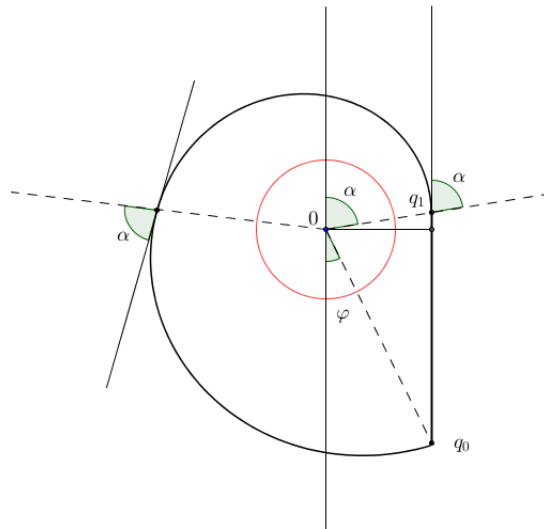
Now let S be a spiralling strategy of maximum speed $v \leq (1 + \sqrt{5})/2 \approx 1.618$, the golden ratio. We can assume that S proceeds at constant speed v . Let p_0, p_1, p_2, \dots denote the points on the coordinate axes visited, in this order, by S . The following lemma shows that S cannot succeed because there is still fire burning outside the barrier on the axis previously visited.

► **Lemma 7.** *Let A be the initial fire radius. When S visits point p_{i+1} , the interval $[p_i, p_i + \text{sign}(p_i)A]$ on the axis visited before is on fire.*

Proof. The proof is by induction on i . Suppose strategy S builds a barrier of length x between p_0 and p_1 , as shown in Figure 7 (i). During this time the fire advances x/v along the positive X -axis, so that $A + x/v \leq p_1 \leq x$ must hold, or

$$\frac{x}{v} \geq \frac{1}{v-1}A > A;$$

the last inequality follows from $v < 2$. Thus, the fire has enough time to move a distance of A from p_0 downwards along the negative Y -axis.



■ **Figure 8** A completion time optimal single closed loop solution for $v \approx 6.25$ starts with a line segment outside the fire and ends with a logarithmic spiral along the boundary of the fire. A single loop solution exists only for $v \geq 3.7788\dots$

Now let us assume that strategy S builds a barrier of length y between p_i and p_{i+1} , as shown in Figure 7 (ii). By induction, the interval of length A below p_{i-1} is on fire. Also, when the firefighter moves on from p_i , there must be a burning interval of length at least $A + x/v$ on the positive Y -axis which is not bounded by a barrier from above. This is clear if p_{i+1} is the first point visited on the positive Y -axis, and it follows by induction, otherwise. Thus, we must have $A + x/v + y/v \leq p_{i+1} \leq y$, hence

$$\frac{y}{v} \geq \frac{1}{v-1}A + \frac{1}{v(v-1)}x > A + x,$$

since the assumption on v implies $v^2 \leq v + 1$. This shows that the fire can crawl along the barrier from p_{i-1} to p_i , and a distance A to the right, as the firefighter moves to p_{i+1} , completing the proof of Theorem 2. ◀

8 Conclusions

A number of interesting questions arise. Are there strategies that can contain the fire at a speed $v < v_c$? How about starting points away from the fire? Given a speed $v \geq v_c$, there can be many barrier curves that contain a fire. Which one should the firefighter choose, to minimize the time to completion, or the area burned? Is it possible to generalize to fires of more realistic shapes, as they result under the influence of wind as for example suggested in [5]? These problems define a new and nice area in the field of path planning in dynamic environments, where obstacle shapes depend on the agent's actions.

For practical purposes, one would wish for a strategy that contains the fire in a single closed round. Also, starting points away from the fire could be allowed. If the firefighter is free to pick her starting point she can contain the fire in a single closed round if, and only if, her speed is at least $v \geq 3.7788\dots$ In this case the shortest possible (i.e., completion time optimal) solution consists of a line segment q_0q_1 followed by a segment of a logarithmic spiral

of excentricity α , where $v = \frac{1}{\cos(\alpha)}$. See Figure 8 for an example of the time optimal single closed loop for $\alpha = 1.41$ and $v \approx 6.25$.

A single closed loop solution only exists for

$$\alpha > \arctan \left(\frac{\frac{3}{2}\pi}{W\left(\frac{3}{2}\pi\right)} \right) \approx 74.66^\circ$$

in which W denotes Lambert's W function [1] defined by the functional equation $W(x)e^{W(x)} = x$. This gives $\alpha \geq 1.3029\dots$ or $v \geq 3.7788\dots$

Acknowledgements. We would like to thank the anonymous referees for their valuable comments and suggestions.

References

- 1 R. M. Corless and G. H. Gonnet and D. E. G. Hare and D. J. Jeffrey. Lambert's W function in Maple. The Maple Technical Newsletter, Issue 9, pp. 12–22, 1993.
- 2 C. E. Falbo. Analytic and Numerical Solutions to the Delay Differential Equation $y'(t) = \alpha y(t - \delta)$. Joint Meeting of the Northern and Southern California Sections of the MAA, San Luis Obispo, CA, 1995. Revised version at <http://www.mathfile.net>
- 3 S. Finbow and G. MacGillivray. The Firefighter Problem: A survey of results, directions and questions. Australasian J. Comb, 43, pp. 57-78, 2009.
- 4 P. Flajolet and R. Sedgewick. Analytic Combinatorics. Cambridge, 2009.
- 5 Food and Agriculture Organization of the United Nations (FAO). International Handbook on Forest Fire Protection. <http://www.fao.org/forestry/27221-06293a5348df37bc8b14e24472df64810.pdf>
- 6 R. Klein, Ch. Levcopoulos, and A. Lingas. Approximation algorithms for the geometric firefighter and budget fence problems. in A. Pardo and A. Viola (eds.) LATIN 2014, Montevideo, LNCS 8392, pp. 261–272.
- 7 R. Klein, E. Langetepe, and Ch. Levcopoulos. A Fire Fighter's Problem. Technical Report, <http://arxiv.org/abs/1412.6065>, 2014

Approximate Geometric MST Range Queries

Sunil Arya^{*1}, David M. Mount^{†2}, and Eunhui Park²

- 1 Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong, China
arya@cse.ust.hk
- 2 Department of Computer Science
University of Maryland
College Park, Maryland 20742, USA
{mount,ehpark}@cs.umd.edu

Abstract

Range searching is a widely-used method in computational geometry for efficiently accessing local regions of a large data set. Typically, range searching involves either counting or reporting the points lying within a given query region, but it is often desirable to compute statistics that better describe the structure of the point set lying within the region, not just the count.

In this paper we consider the geometric minimum spanning tree (MST) problem in the context of range searching where approximation is allowed. We are given a set P of n points in \mathbb{R}^d . The objective is to preprocess P so that given an admissible query region Q , it is possible to efficiently approximate the weight of the minimum spanning tree of $P \cap Q$. There are two natural sources of approximation error, first by treating Q as a fuzzy object and second by approximating the MST weight itself. To model this, we assume that we are given two positive real approximation parameters ε_q and ε_w . Following the typical practice in approximate range searching, the range is expressed as two shapes Q^- and Q^+ , where $Q^- \subseteq Q \subseteq Q^+$, and their boundaries are separated by a distance of at least $\varepsilon_q \cdot \text{diam}(Q)$. Points within Q^- must be included and points external to Q^+ cannot be included. A weight W is a valid answer to the query if there exist point sets P' and P'' such that $P \cap Q^- \subseteq P' \subseteq P'' \subseteq P \cap Q^+$ and $wt(\text{MST}(P')) \leq W \leq (1 + \varepsilon_w) \cdot wt(\text{MST}(P''))$.

In this paper, we present an efficient data structure for answering such queries. Our approach uses simple data structures based on quadtrees, and it can be applied whenever Q^- and Q^+ are compact sets of constant combinatorial complexity. It uses space $O(n)$, and it answers queries in time $O(\log n + 1/(\varepsilon_q \varepsilon_w)^{d+O(1)})$. The $O(1)$ term is a small constant independent of dimension, and the hidden constant factor in the overall running time depends on d , but not on ε_q or ε_w . Preprocessing requires knowledge of ε_w , but not ε_q .

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Geometric data structures, Minimum spanning trees, Range searching, Approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.781

1 Introduction

Range searching is a fundamental tool in computational geometry. Given a set P of n points in \mathbb{R}^d , the objective is to preprocess the points into a data structure so that, given any

* Research supported by the Research Grants Council of Hong Kong, China under project number 16200014.

† Research supported by NSF grant CCF-1117259 and ONR grant N00014-08-1-1015.



© Sunil Arya, David M. Mount, and Eunhui Park;
licensed under Creative Commons License CC-BY

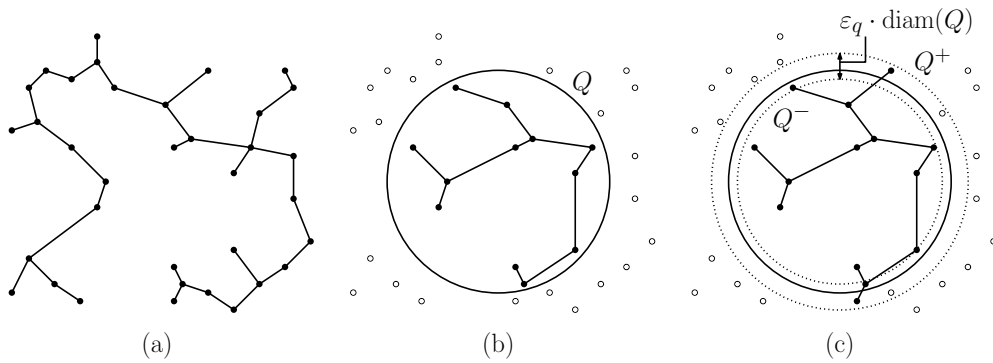
31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 781–795



Leibniz International Proceedings in Informatics

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



■ **Figure 1** (a) Euclidean MST, (b) MST query, and (c) approximate MST query.

range Q from some class of admissible ranges (e.g., axis-aligned rectangles, balls, halfspaces, simplices), it is possible to efficiently count or report the points of P that lie within Q . Range searching is a powerful method for exploring local regions of a large geometric data set, and it finds many applications in science and engineering.

In many of these applications it is desirable to obtain more detailed information than simple counts. In this paper we explore the question of whether it possible to compute more interesting properties of the subset of points lying within a range, properties that depend on the geometric structure of the points. There are numerous statistics that describe the structure of a point set. Often, such properties are based on graph structures that are implicitly defined by the points set. Perhaps the most fundamental example of such a graph is the Euclidean minimum spanning tree (see Fig. 1(a)). Given a point set P in a Euclidean space, let $\text{MST}(P)$ denote P 's minimum weight spanning tree, and let $wt(\text{MST}(P))$ denote its total edge weight. Given a query range Q , an *MST query* returns $wt(\text{MST}(P \cap Q))$ (see Fig. 1(b)). The MST weight (and more generally the distribution of its edge weights) can provide useful information about the density properties of a point set.

Because of the high computational complexities of exact range searching and computing exact geometric spanning trees in multi-dimensional spaces, it is natural to consider the problem in an approximate context. We assume that we are given two positive real parameters ϵ_q and ϵ_w , which represent the allowable errors in approximating the query shape and the MST weight, respectively. A range is modeled as a “fuzzy” region of space, so that points near the range’s boundary may be included or excluded at the algorithm’s discretion. To make this more formal, an ϵ_q -approximate range Q is presented as a pair of compact bodies Q^- and Q^+ (called the *inner range* and *outer range*, respectively), where $Q^- \subseteq Q \subseteq Q^+$ and the boundaries of Q^- and Q^+ are separated by a distance of at least $\epsilon_q \cdot \text{diam}(Q)$. In standard approximate range searching, the objective is to compute the size (or generally weight) of any set P' , such that $P \cap Q^- \subseteq P' \subseteq P \cap Q^+$. Thus, a natural formulation¹ would be to return any weight W such that

$$wt(\text{MST}(P')) \leq W \leq (1 + \epsilon_w) \cdot wt(\text{MST}(P')), \text{ where } P \cap Q^- \subseteq P' \subseteq P \cap Q^+.$$

Because we amortize the cost of our result against the weight of the MST in a slightly larger

¹ Note that the “obvious” formulation of returning a weight W such that $wt(\text{MST}(P \cap Q^-)) \leq W \leq (1 + \epsilon_w) \cdot wt(\text{MST}(P \cap Q^+))$ is not well defined because (in dimensions three and higher) there exist point sets such that, even for spherical ranges, $wt(\text{MST}(P \cap Q^-)) > wt(\text{MST}(P \cap Q^+))$. The phenomenon is related to the effect of decreasing the MST weight through the addition of Steiner points.

region, we introduce two sets in our formulation. In particular, we return a weight W such that

$$wt(\text{MST}(P')) \leq W \leq (1 + \varepsilon_w) \cdot wt(\text{MST}(P'')), \text{ where } P \cap Q^- \subseteq P' \subseteq P'' \subseteq P \cap Q^+.$$

We refer to this as an $(\varepsilon_q, \varepsilon_w)$ -approximate MST query.

Our main result is given in the following theorem. For our purposes, a range $Q \subseteq \mathbb{R}^d$ is *admissible* if it is compact and has the property that in $O(1)$ time it is possible to determine for any hypercube b : (1) whether b is contained within Q^+ and (2) whether b is disjoint from Q^- . Thus, the inner and out ranges need not be convex, but should be of constant combinatorial complexity. To simplify the complexity bounds (which are stated in full detail at the end of Section 3.3), we use the notation O^* to ignore factors of the form $1/\varepsilon^{O(1)}$, where the $O(1)$ term does not depend on d (and is roughly 2 in our case).

► **Theorem 1.** *Given a set P of n points in \mathbb{R}^d and a weight-approximation parameter $\varepsilon_w > 0$, P can be preprocessed into a data structure of space $O(n)$ such that given any admissible ε_q -approximate query Q , it is possible to answer $(\varepsilon_q, \varepsilon_w)$ -approximate MST queries in time $O^*(\log n + 1/(\varepsilon_q \varepsilon_w)^d)$.*

Preprocessing time will be discussed in the full version of the paper, where we show that (ignoring logarithmic factors) the data structure can be built in time $\tilde{O}(n/\varepsilon^{d/2})$. While preprocessing assumes knowledge of ε_w , it is interesting to note that the space bounds do not depend on ε_w . In [5] it is shown that answering ε_q -approximate range counting queries even for hypercube ranges by searching a partition tree requires $\Omega(\log n + 1/\varepsilon_q^{d-1})$ time. Thus, ignoring the ε_w term, the query time is not far from optimal assuming an approach based on partition trees (as is the approach presented here).

The notion of extracting more complex information than simple counts (or more generally evaluating sums over a commutative semigroup) in range searching has been studied before. One broad class of results involve extensions of *aggregate range searching* [19, 1]. Papadias *et al.* [16] and Shan *et al.* [18] both present data structures that answer various types of nearest neighbor queries over ranges. Nekrich and Smid [15] present a generic data structure that returns an ε -coreset for orthogonal query ranges in \mathbb{R}^d . Brass *et al.* [9] present data structures for answering orthogonal range queries in \mathbb{R}^2 involving extent measures of the points lying within a query range, including width, area and perimeter of the convex hull, and the smallest enclosing disk. MST queries are particularly challenging because, due to the requirement that the MST must be connected, it is not possible to merely aggregate information in order to answer the query.

Extracting structural information has also been explored in a temporal setting in the work of Bannister *et al.* [8, 7]. In [8] a collection of pairwise relational events are given with time stamps, and it is shown how to extract graph properties efficiently for the events lying within a given query time interval. In [7], this is extended to geometric structures for points with time stamps. Because we are interested in constructing information about the MST in sublinear time, our methods bear similarity to sublinear time algorithms for computing geometric spanning trees, as exemplified in the work of Czumaj, Sohler, and others [12, 13] and Frahling *et al.* [14]. We note, however, that in contrast to these algorithms that are randomized and return only an approximation to the weight (not the edges), our query algorithm is deterministic and implicitly provides a certificate in the form of a connected graph (possibly containing cycles) that spans the point set P' and satisfies the stated weight requirements. Given this certificate, it is possible to enumerate or randomly sample from the edges of this graph.

Our approach borrows some standard techniques for computing approximate geometric spanning trees, such as quadtrees, well-separated pair decompositions (WSPDs), bottom-up construction, and randomized shifting (see, e.g., [11, 4, 2]). Due to the special nature of our problem, we have developed a number of new twists on these ideas. For example, in order to avoid problems with bad quadtree alignments, we develop a local variant of the well-known technique of randomly shifting the coordinate system [3]. We also develop a more efficient method for computing the closest pair of points in the pairs of a WSPD, which exploits the fact that (in our context) the approximation error can be amortized against the weight of the MST within the dumbbell heads of the WSPD.

2 Preliminaries

In this section we provide basic definitions of a number of concepts that will be used throughout the paper.

2.1 Minimum Spanning Trees

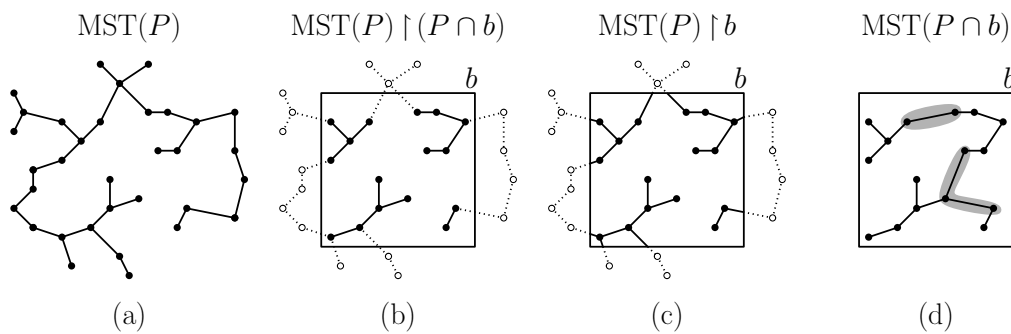
Consider a finite point set $P \in \mathbb{R}^d$. Given two points $p, q \in \mathbb{R}^d$, we denote their Euclidean distance by $\|pq\|$. Formally, the *minimum spanning tree* of P , denoted $\text{MST}(P)$, is any minimum spanning tree of the complete graph on P whose edge weights are the interpoint distances. (Our results can be extended easily to any Minkowski distance, with a slight adjustment in the constant factors.) The edges of $\text{MST}(P)$ are line segments, and we will often treat the relevant portions of the MST as a finite set of line segments. Define the *weight* of any such set S of segments, denoted $wt(S)$, to be the sum of the segment lengths.

Throughout, we will need to refer to various restrictions of the edges/weight of the MST to a region of space. We use the term *global MST* to refer to $\text{MST}(P)$. Given subsets $P', P'' \subseteq P$, define the *induced MST* on (P', P'') , denoted $\text{MST}(P) \upharpoonright (P', P'')$, to be the subset of global MST edges that have one endpoint in P' and one in P'' . Let $\text{MST}(P) \upharpoonright P'$ denote $\text{MST}(P) \upharpoonright (P', P')$.

Given a closed region of space b (which for us will be a hypercube or the difference of two nested hypercubes), there are two natural ways of restricting $\text{MST}(P)$ to b , depending on whether we include edges entirely or partially. Define $\text{MST}(P) \upharpoonright (P \cap b)$ to be the subset of the edges of $\text{MST}(P)$ both of whose endpoints lie within b (see Fig. 2(b)), and define $\text{MST}(P) \upharpoonright b$ to be intersection of $\text{MST}(P)$ (as a set of segments) with b (see Fig. 2(c)). Observe that $\text{MST}(P) \upharpoonright (P \cap b)$ is a subgraph of $\text{MST}(P \cap b)$. When P is understood from context, define the *local connectors* of b , denoted $\Delta(b)$, to be the segments of $\text{MST}(P \cap b)$ that are not in $\text{MST}(P) \upharpoonright (P \cap b)$ (highlighted in Fig. 2(d)).

Our algorithm will classify edges of the MST as being “short” or “long,” and process each group differently. Given any $\gamma > 0$, define the γ -*restricted MST*, denoted $\text{MST}_\gamma(P)$, to be the subgraph of $\text{MST}(P)$ consisting of edges of weight at most γ , and define $\text{MST}_{>\gamma}(P)$ similarly but for edges of weight greater than γ .

We will organize the edges of the MST using a quadtree decomposition. In general, a uniform grid of hypercubes overlaid on P naturally induces a graph whose vertices are the grid cells and two cells (b, b') are connected by an edge if $\text{MST}(P) \upharpoonright (P \cap b, P \cap b')$ is nonempty. (Note that this graph may contain cycles and self-loop edges.) It is well known that the MST of any finite point set P in \mathbb{R}^d has constant degree (depending on the dimension), and it is easy to show that this is true for this induced graph as well. We omit the proof.



■ **Figure 2** Geometric minimum spanning tree definitions.

► **Lemma 2.** *Given a finite point set P in \mathbb{R}^d and a uniform grid of hypercubes, there exists a constant c (depending only on the dimension d) such that the MST induced on the grid is of degree at most c .*

2.2 BBD-trees and Blocks

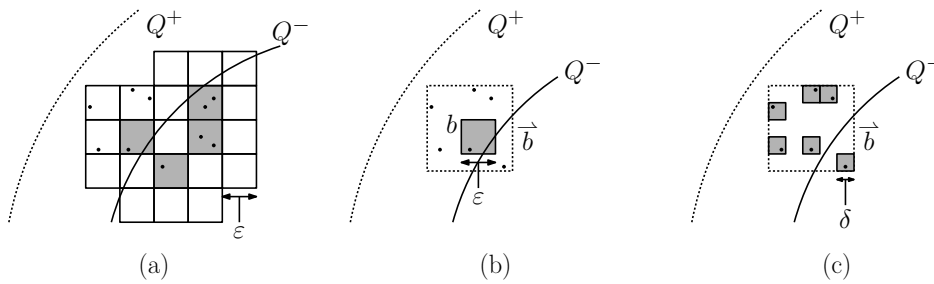
Our solution will be based on a balanced variant of a quadtree, called a BBD-tree. We refer the reader to [6] for details, but informally, a BBD-tree is based on a quadtree-like subdivision of space, which introduces a decomposition operator, called *shrinking*, that allows the data structure to zoom into regions of dense concentration. The relevant properties of the BBD-tree are given in the following lemma, which was proved by Arya *et al.* [6].

► **Lemma 3 (BBD-tree Construction and Packing Lemma).** *Given an n -element point set P in \mathbb{R}^d , in $O(n \log n)$ time it is possible to construct a BBD-tree of size $O(n)$ and height $O(\log n)$. Furthermore, the number of cells of this tree with pairwise disjoint interiors, each of side length at least s , that intersect a ball of radius r is at most $O((1 + r/s)^d)$.*

For the purposes of processing queries, it will be convenient to conceptualize the subset of points contributing to the query as union of the points lying within a subset of sufficiently small disjoint quadtree boxes all of equal side length. To make this more formal, we introduce the notions of mini-blocks and micro-blocks.

For a sufficiently small constant c (specified later), define $\varepsilon = c \cdot \varepsilon_q \cdot \text{diam}(Q)$. We will assume that c is chosen so that ε is power of two and $c \leq 1/2d$. Define a *mini-block* to be a nonempty quadtree box of side length ε . Let $B_\varepsilon(Q)$ denote the set of mini-blocks that overlap Q^- (the shaded squares of Fig. 3(a)). (This set depends on P and ε_q as well, but since P and ε_q will be fixed throughout, we omit reference to them.) Also, define $B_\varepsilon^+(Q)$ to be the set of quadtree boxes of side length ε such that at least one of its 3^d neighboring blocks is in $B_\varepsilon(Q)$ (all the squares of Fig. 3(a)). A box of side length ε has diameter at most $d\varepsilon \leq (\varepsilon_q/2) \cdot \text{diam}(Q)$, and therefore, all the boxes of $B_\varepsilon(Q)$ and $B_\varepsilon^+(Q)$ lie within Q^+ .

An important part of our construction will involve expanding and shifting mini-blocks. Each mini-block b of $B_\varepsilon(Q)$ will be associated with a hypercube that contains b and whose side length is twice as large as b 's (see Fig. 3(b)). We call this the *shifted block* and denote it by \vec{b} . Observe that each shifted block lies within the union of the 3^d neighboring blocks of b , and therefore each shifted block lies within $B_\varepsilon^+(Q)$. For a sufficiently small positive constant c' (specified later), define $\delta = c' \varepsilon_w \varepsilon$. Again, we will assume that c' is chosen so that δ is a power of two. Define a *micro-block* (associated with Q) to be a nonempty quadtree box of side length δ . Our preprocessing algorithm will construct \vec{b} so that it is aligned with the



■ **Figure 3** (a) Mini-blocks (all blocks are in $B_\epsilon^+(Q)$ and shaded blocks are in $B_\epsilon(Q)$), (b) a mini-block b and its shifted block \tilde{b} , and (c) the micro-blocks associated with b .

quadtree grid of side length δ . Define $B_\delta(b)$ to be the set of micro-blocks lying within \tilde{b} (see Fig. 3(c)), and define $B_\delta(Q)$ to be the union of these micro-blocks over all $b \in B_\epsilon(Q)$.

Assuming the existence of these quantities for now, define $P(Q)$ to be the subset of P that is covered by all the shifted miniblocks, and similarly define $P^+(Q)$ to be the subset of P lying within the blocks of $B_\epsilon^+(Q)$. The following results are straightforward consequences of our definitions. (Due to space limitations, proofs have been omitted from this version.)

► **Lemma 4.** *There exist constants c and c' (for the above definitions) such that, given a point set P in \mathbb{R}^d and an ϵ_q -approximate range Q :*

- (i) $B_\epsilon(Q)$ and $B_\epsilon^+(Q)$ are both of size $O(1/\epsilon_q^d)$.
- (ii) $B_\delta(Q)$ is of size $O(1/(\epsilon_q \epsilon_w)^d)$.
- (iii) $P \cap Q^- \subseteq P(Q) \subseteq P^+(Q) \subseteq P \cap Q^+$.

This lemma suggests a means by which to construct a solution to an (ϵ_q, ϵ_w) -approximate MST query. Namely, find the weight of the edges of $\text{MST}(P) \upharpoonright P(Q)$, and then include additional edges of low weight to join the connected components of this forest. Our approach will be of this general form, and the additional edges will be classified as being of one of two types, short edges and long edges. At a first reading it is reasonable to think of the sets $P(Q)$ and $P^+(Q)$ as playing the roles of P' and P'' in the definition of an approximate MST query. (But a twist will enter at the end.)

Our next lemma shows that these block sets can be computed efficiently. It is a straightforward adaptation of standard algorithms on BBD-trees.

► **Lemma 5.** *Given a BBD-tree storing P and an ϵ_q -approximate range Q , it is possible to compute $B_\epsilon(Q)$ and $B_\epsilon^+(Q)$ in $O(\log n + 1/\epsilon_q^d)$ time and $B_\delta(Q)$ in $O(\log n + 1/(\epsilon_q \epsilon_w)^d)$ time.*

We would like to identify the mini- and micro-blocks with subsets of nodes of the BBD-tree. This is complicated by the fact that a given block need not exist as the cell of any node within the tree because the decomposition ended at a leaf node before reaching this level. In order to focus on the key issues, it will greatly simplify matters to ignore the BBD-tree structure for now and assume that we have instantaneous access to the data stored in any quadtree box. In the full version we will discuss the technical details underlying this assumption.

3 Computing the MST Weight

In this section, we will present our data structure and discuss query processing. Let us begin with a high-level overview of our approach. First, recall that $B_\epsilon(Q)$ denotes the set of mini-blocks of side length roughly $\epsilon_q \cdot \text{diam}(Q)$ that overlap the inner query range. These

mini-blocks all lie within the outer query range, and so if we could compute (approximately) the MST of the point set lying within them we would be done. We know that the global MST induced on this set of points is a subset of the final MST. Thus, a natural strategy would be to store the weights of edges of the global MST locally in the nodes of the quadtree, and then at query time combine the MST edge weights for the nodes representing $B_\varepsilon(Q)$ and explicitly compute the additional connecting edges needed to join the connected components of this forest into a single tree.

The difficulty in carrying out this strategy is that there may be many ($\Omega(n)$) connected components of the global MST, and like the tangled branches of a vine, these components can be quite long and intricate and may be separated by arbitrarily small distances. To overcome this problem, within each mini-block we would like to compute (as a part of the preprocessing) a set of edges that will connect the components within this block. Because this will be done independently for each block, without consideration of global connectivity, the problem is determining how to do this without significantly increasing the total edge weight within the query region.

To overcome this problem, we will modify a common strategy used in the computation of geometric MSTs. First, let us focus on “short edges.” Recall that δ is roughly $\varepsilon_w \varepsilon$, and the δ -restricted MST is the subgraph of the $MST(P)$ consisting of edges of length at most δ . Rather than connecting all the components, we will focus instead on connecting just the components of the δ -restricted MST lying within each mini-block b in order to form the δ -restricted MST of $P \cap b$. (For technical reasons, we will do this for a slightly larger value, $\hat{\delta} = 2d\delta$, but we will ignore this small variation for now.) Unfortunately, such a local strategy may introduce unnecessarily long edges if the quadtree structure is badly aligned with respect to the point set. In traditional MST approximation algorithms this difficulty is handled by introducing a modified distance function that penalizes very short edges (of length at most δ) that cross the mini-block boundary. This relies on the fact that if a random shift is applied to the coordinate system, then in expectation this added penalty increases the global MST weight by only a small amount. This approach cannot be applied in our setting however, because we need to show that the weight increase is bounded within *every* possible query region.

Rather than shifting the coordinate system, we instead expand each mini-block b by a factor of two and take an appropriately translated copy of this *shifted block*, denoted \vec{b} , that contains b . (For technical reasons, this will be applied to a slight enlargement of the shifted box, called \vec{b}^+ .) Because this is computed at preprocessing time, query processing is deterministic. The key property possessed by \vec{b} is that the total weight needed to connect the δ -restricted global MST within \vec{b} is within a factor of roughly ε_w of the total weight of the global MST induced in the neighborhood of b , more formally, within the region covered by the 3^d blocks that surround b . We call these additional edges *local connectors*. Recall that $P(Q)$ denotes the union of the points of P lying within these shifted blocks.

Given the weight of the δ -restricted global MST induced on the shifted blocks and the weight of the local connectors, we can now resume our original strategy. We decompose the shifted blocks into micro-blocks of side length δ , accumulate the weights of the δ -restricted global MST and local connectors on these blocks. The number of such blocks is $O(1/(\varepsilon_q \varepsilon_w)^d)$, and this accumulation can be performed within this time bound by a traversal of the BBD-tree. These edges induce a graph on the δ -blocks, called the *global connection graph*. We compute the connected components of this graph. This provides us with an approximation to the δ -restricted MST of $P(Q)$, with the caveat that the approximation error is expressed with respect to the larger point set that lies in $B_\varepsilon^+(Q)$, the neighboring blocks of $B_\varepsilon(Q)$. We refer to all of this as the *short-edge processing*.

To finish the job, we need to add the “long edges” (of length greater than δ) in order to connect the components of the global connection graph. To do this, we employ a strategy based on the well-separated pair decomposition of the micro-blocks. Callahan and Kosaraju [11] observed that, even with a constant factor separation, the MST could be well approximated by computing an approximation to the closest pair within each well-separated pair, and then computing the MST of these pairs. We will apply the same idea with two modifications. Because we are only interested in well-separated pairs at distance greater than $\Theta(\delta)$, the number of pairs is proportional to the number of micro-blocks. Second, we ignore any pairs that join two points whose micro-blocks are within the same component of the global-connection graph.

The problem with applying the Callahan and Kosaraju approach directly is that in order to compute an ε_w -approximation to the closest pair, we would need to decompose each micro-block further into $O(1/\varepsilon_w^{\Omega(d)})$ subblocks, which would increase the running time considerably. In order to avoid this additional blow-up, we employ a novel idea. The pairs that are difficult to process are those having many subblocks within the dumbbell head of the well-separated pairs. In such cases, however, the weight of the MST within the dumbbell head is relatively large. Rather than charging the approximation error to the length of the pair returned, we instead charge the error to the weight of the MST within the dumbbell heads. We show that by doing this, the running time is $O((1/\varepsilon_w^2) \log^2(1/\varepsilon_q \varepsilon_w))$, which avoids ε dependencies that grow exponentially in the dimension.

The final answer to the query is the sum of the weights from the short-edge and long-edge processing. As mentioned above, our algorithm is deterministic and implicitly provides a certificate to the answer in the form of a connected graph on $P(Q)$.

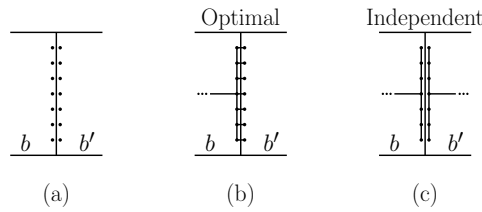
3.1 Short-Edge Processing

Let us discuss now the short-edge processing in greater detail. Recall ε , δ , $\widehat{\delta}$, $B_\varepsilon(Q)$, $B_\delta(Q)$, $P(Q)$, and $P^+(Q)$ introduced earlier. Also recall that each mini-block b is associated with a shifted block \vec{b} (to be specified below), which contains b and is contained within b 's neighbors. The objective of this phase is to compute a locally connected augmentation of the $\widehat{\delta}$ -restricted global MST within each of the shifted mini-blocks. This will involve three things: (1) the weight of the edges of the $\widehat{\delta}$ -restricted global MST induced on each shifted block, (2) the weight of a set of local connectors that join components of this graph to form the $\widehat{\delta}$ -restricted MST within each shifted block, and (3) a global-connection graph on the micro-blocks of $B_\delta(Q)$ that connects these components throughout the query range. In this section, we will show that these structures satisfy two properties:

Low weight: The total weight of the local connectors over all the mini-blocks of $B_\varepsilon(Q)$ is at most $(\varepsilon_w/2) \cdot wt(\text{MST}_{\widehat{\delta}}(P^+(Q)))$. (This will be established in Lemma 8 below.)

Local connectivity: Given two points $p, p' \in P(Q)$ such that $\|pp'\| \leq \widehat{\delta}$, the micro-blocks of $B_\delta(Q)$ that contain these points are in the same connected component of the global-connection graph.

The challenge in achieving these two properties arises from the possible poor placement of partitioning cuts in the quadtree. For example, suppose we have a pair b and b' of neighboring mini-blocks, and we have a large number of point pairs where one element of each pair lies in b and the other in b' , and further the segment joining each pair is extremely short (see Fig. 4(a)). If we build the MSTs independently within each mini-block, the local weight will be nearly twice the optimum (see Figs. 4(b) and (c)). Since this instance is the result of an



■ **Figure 4** (a) Two points sets lying close to a quadtree splitting edge, (b) the optimal MST, and (c) two MSTs computed independently within each box.

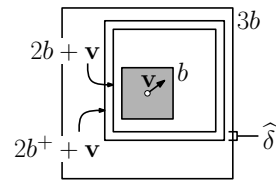
unlucky choice of quadtree cuts, this is usually remedied by applying a random translation to the coordinate system before building the quadtree. While it can be shown that this fixes the problem (in expectation) for the global MST, it does not necessarily fix the problems at the local level, which is what we need for range searching.

As mentioned above, our solution will involve expanding each mini-block by a factor of two, and applying a shift to this expanded block. Before presenting our shifting algorithm, we present a useful lemma. To motivate this lemma, for any $\gamma > 0$ consider the γ -restricted MST of a point set P and a sufficiently large hypercube b . As observed earlier, the γ -restricted global MST induced on $P \cap b$ (formally, $\text{MST}_\gamma(P) \upharpoonright (P \cap b)$) is a subgraph of the γ -restricted MST on $P \cap b$ (formally, $\text{MST}_\gamma(P \cap b)$). Define $\Delta_\gamma(b)$ to be the edges in the set-theoretic difference of these two graphs. We will show that $wt(\Delta_\gamma(b))$ is proportional to the weight of the global spanning tree within distance γ of b 's boundary. Intuitively, this holds because the components of $\text{MST}_\gamma(P) \upharpoonright (P \cap b)$ that are connected in $\text{MST}_\gamma(P \cap b)$ must be connected by paths consisting of edges of the MST of length at most γ that lie outside of b .

Before stating the lemma we introduce some terminology. Given a hypercube b of side length at least 2γ , define the γ -shell of b , denoted $\text{shell}_\gamma(b)$, to be the set-theoretic difference of two hypercubes b^+ and b^- , where $b^- \subset b \subset b^+$, and the boundaries of these hypercubes are separated from b 's boundary by a distance of γ .

► **Lemma 6.** Consider a point set P in \mathbb{R}^d , $\gamma > 0$, and a hypercube b of side length at least 2γ . Then, $wt(\Delta_\gamma(b)) \leq 3 \cdot wt(\text{MST}_\gamma(P) \upharpoonright \text{shell}_\gamma(b))$.

Resuming the discussion of the short-edge processing, consider a mini-block b . Recall that its side length is ε . For the sake of our construction, let us assume that the origin is centered at b 's center. Let $2b$ and $3b$ denote centrally scaled copies of b by factors of 2 and 3, respectively (see Fig. 5). Because we are interested in edges of length up to $\hat{\delta}$ that might have one endpoint within $2b$ and one endpoint outside, let $2b^+$ denote the hypercube that results by translating each of the bounding hyperplanes of $2b$ outwards by distance $\hat{\delta}$. Given a vector \mathbf{v} let $2b^+ + \mathbf{v}$ denote a translation of $2b^+$ by \mathbf{v} .

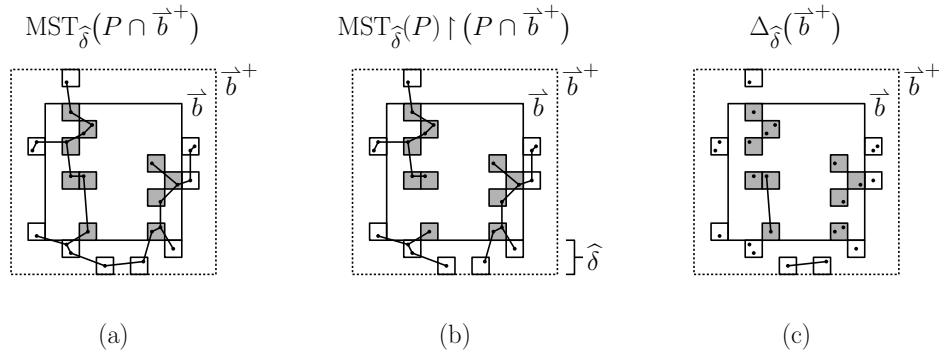


■ **Figure 5** An expanded and shifted block.

Recalling the definitions of Section 2.2, our objective is to compute the shifted block \vec{b} to be associated with b . To do so, we will consider a set of $O((\varepsilon/\delta)^d)$ possible shifts of $2b^+$, each of which will contain b and lie within $3b$. Our next lemma shows that for at least one of these shifts (in fact, for a constant fraction of them) the local connection weight $wt(\Delta_{\hat{\delta}}(2b^+ + \mathbf{v}))$ is $O(\varepsilon_w)$ times the weight of the $\hat{\delta}$ -restricted MST induced on $P \cap 3b$.

► **Lemma 7.** Consider a point set P in \mathbb{R}^d , an approximate query Q , and a mini-block $b \in B_\varepsilon(Q)$. For any constant $c'' > 0$, there exists a translate of $2b^+$, denoted \hat{b} , that is nested between b and $3b$, is aligned with the quadtree grid of side length δ , and

$$wt(\Delta_{\hat{\delta}}(\hat{b})) \leq c'' \cdot \varepsilon_w \cdot wt(\text{MST}_{\hat{\delta}}(P) \upharpoonright (P \cap 3b)).$$



■ **Figure 6** A shifted mini-block \vec{b} and its expansion \vec{b}^+ . The micro-blocks $\mu(\vec{b})$ are shown as shaded boxes and $\mu(\vec{b}^+)$ includes the white boxes as well. (a) The restricted MST of $P \cap \vec{b}^+$, (b) the induced global spanning tree on $P \cap \vec{b}^+$, and (c) the local connectors.

Given a mini-block b , define \vec{b}^+ to be the translated box \vec{b} from the above lemma, and define its shifted block, \vec{b} , to be the corresponding translate of $2b$. This information is computed for each node of the quadtree as part of the preprocessing.

To complete the short-edge processing, we need to compute the local connectors (that is, the edges of the $\hat{\delta}$ -restricted MST on $P \cap \vec{b}$ that are not in the $\hat{\delta}$ -restricted MST induced on these points). An obvious approach would be to compute $\text{MST}_{\hat{\delta}}(P \cap \vec{b})$, and then remove from this the edges from the global MST. While this would work fine for an individual shifted block, this is not sufficient to guarantee connectivity across the entire query region (particularly for blocks near the query's boundary). Since the edges involved are all of length at most $\hat{\delta}$, for the purposes of computing connectivity, we will consider micro-blocks that lie slightly (distance at most $\hat{\delta}$) outside the shifted blocks. Once the connected components have been computed, we will discard these extra blocks.

To make this more precise, given a mini-block b , define $\mu(\vec{b})$ to be the micro-blocks that lie within \vec{b} (the shaded small boxes in Fig. 6), and define $\mu(\vec{b}^+)$ similarly for \vec{b}^+ (all the small boxes in Fig. 6). To compute the local connectors, at preprocessing time for each such mini-block b , we compute \vec{b}^+ (by the previous lemma) and $\text{MST}_{\hat{\delta}}(P \cap \vec{b}^+)$ (see Fig. 6(a)). We assume that the global MST has already been computed. The local connectors consist of the edges that are not already in the global spanning tree induced on these points, that is,

$$\Delta_{\hat{\delta}}(\vec{b}^+) = \text{MST}_{\hat{\delta}}(P \cap \vec{b}^+) \setminus \text{MST}_{\hat{\delta}}(P) \upharpoonright (P \cap \vec{b}^+)$$

(see Figs. 6(b) and (c)).

We cannot deal with structures like $\Delta_{\hat{\delta}}(\vec{b}^+)$ at query time, since they involve individual points. Instead, we will deal with graphs that they induce on the micro-blocks. At preprocessing time, we compute an induced (weighted) graph on the micro-blocks of $\mu(\vec{b}^+)$ from the local connectors as follows. For each edge (p, p') in $\Delta_{\hat{\delta}}(\vec{b}^+)$, create an edge between the respective micro-blocks b and b' that contain them. Set the weight of this edge to be the total length of all such edges. Because each edge of $\Delta_{\hat{\delta}}(\vec{b}^+)$ is of length at most $\hat{\delta}$, the neighbors of each micro-block (whose side length is δ) lie within distance at most $\hat{\delta}$. The number of such neighbors is $O((\hat{\delta}/\delta)^d) = O(d^d) = O(1)$. Therefore, this graph has constant degree. Also, as a part of preprocessing, we compute the weight of the edges of the $\hat{\delta}$ -restricted global MST induced on each pair of micro-blocks. (This is done implicitly. See the full version for details.)

When processing a query Q , we combine the aforementioned graphs at the mini-block level to derive two additional global structures on the micro-block level. The first is a structure that encapsulates all the local-connector weight at the micro-block level. Define $\Delta_{\hat{\delta}}(Q)$ to be the union of the graphs $\Delta_{\hat{\delta}}(\vec{b}^+)$ over all mini-blocks $b \in B_{\varepsilon}(Q)$. This is a structure on points, but we can compute its micro-block induced structure by taking the union of the corresponding micro-block structures mentioned above. If there is an edge (b, b') between the same pair of micro-blocks appearing in multiple shifted blocks (which can happen if their shifted blocks overlap), then assign the edge weight to be the sum over all the contributing edges. (We do this because each edge reflects potentially different pairs of locally connected points, and we need to account for the entire weight of these connections. Note that because these involve expanded shifted blocks (\vec{b}^+) , this will implicitly count the weight of edges whose endpoints lie within $P^+(Q)$ but not $P(Q)$.) The total edge weight of this induced graph is the same as $\Delta_{\hat{\delta}}(Q)$.

Our next lemma bounds the weight of this graph in terms of the weight of the $\hat{\delta}$ -restricted MST of a subset of points lying within the outer query range.

► **Lemma 8.** *The weight of $\Delta_{\hat{\delta}}(Q)$ is at most $(\varepsilon_w/2) \cdot wt(\text{MST}_{\hat{\delta}}(P^+(Q)))$.*

The second structure built at query time is the global-connection graph. It consists of the union of the edges of $\Delta_{\hat{\delta}}(Q)$ together with the edges of the $\hat{\delta}$ -restricted global MST induced on the points lying within the union of the expansions of the shifted blocks (that is, the union of $\text{MST}_{\hat{\delta}}(P) \upharpoonright (P \cap \vec{b}^+)$ over all miniblocks b). As with these other graphs, it is defined on points, but it will be represented as an induced graph on micro-blocks. Since this graph is used only for computing connected components, we do not need to assign weights to its edges.

Summarizing the short-edge processing, the data structure consists of the BBD-tree storing the point set P . Each mini-block b is associated with its shifted block \vec{b} (and implicitly its expansion \vec{b}^+) and the graph of local connectors (from $\Delta_{\hat{\delta}}(\vec{b}^+)$) induced on the micro-blocks of $\mu(\vec{b}^+)$. We also store the edges of the global MST so that we can efficiently extract the weight of the $\hat{\delta}$ -restricted MST induced on the micro-blocks (details in the full version). The total space is dominated by the size of the BBD-tree, the storage of the edges of the MST, and the storage of the edges of the local connectors, which is $O(n)$.

Details regarding how these structures are used in the query processing are deferred to the full version. The following lemma summarizes the short-edge phase.

► **Lemma 9 (Short-edge summary).** *Given an n -element point set P in \mathbb{R}^d and an approximation parameter ε_w , there exists a data structure of space $O(n)$ such that given any ε_q -approximate query Q , in time $O(\log n + 1/(\varepsilon_q \varepsilon_w)^d)$ it is possible to compute (implicitly) point sets $P(Q)$ and $P^+(Q)$, a graph $G_s = (P(Q), E_s)$ (which may contain cycles), and a labeling of the connected components of G_s , such that*

- (i) $Q^- \subseteq P(Q) \subseteq P^+(Q) \subseteq Q^+$,
- (ii) any two points of $P(Q)$ that are within distance $\hat{\delta}$ of each other lie in the same connected component of G_s , and
- (iii) the weight of the edges in E_s is at most $wt(\text{MST}_{\hat{\delta}}(P(Q))) + (\varepsilon_w/2) \cdot wt(\text{MST}_{\hat{\delta}}(P^+(Q)))$.

These point sets are represented implicitly by $O(1/(\varepsilon_q \varepsilon_w)^d)$ micro-blocks. The graph G_s is of constant degree, and so is of the same asymptotic size.

3.2 Long-Edge Processing

Given the information from the short-edge processing, as summarized in Lemma 9, let us now consider the long-edge case. Let Ψ denote a well-separated pair decomposition (WSPD) for the point set $P(Q)$ for some suitable constant separation factor (for definitions see [10]). In particular, we require that if (A, A') is a pair of the WSPD, then for all $p, q \in A$ and all $p', q' \in A'$, $\|pp'\| > \max(\|qp\|, \|q'p'\|)$. Drawing on a standard visual analogy, we think of the WSPD as consisting of a collection of *dumbbells*, where each of the sets being separated lies within one of the two *heads* of a dumbbell. Observe that each well-separated pair contributes at most one edge to $\text{MST}(P(Q))$ (because all the points within a dumbbell head will be connected by Kruskal's algorithm before considering any edge between the heads).

Let $\Psi' \subseteq \Psi$ denote the set of dumbbells such that for any pair of points $p, p' \in P(Q)$, where $\|pp'\| > \widehat{\delta}$, there is a dumbbell in Ψ' that separates p and p' . By standard techniques, we can compute Ψ' in time proportional to the number of δ -blocks that cover the points of $P(Q)$, which is $O(1/\delta^d) = O(1/(\varepsilon_q \varepsilon_w)^d)$ (see, e.g., [17]). We assume that every internal node of the BBD-tree contains an arbitrary *representative point* drawn from the points lying within the node's outer box.

Our objective is to compute a suitable approximation to the closest pair of points separated by each dumbbell. Recall from the high-level overview of Section 3 that the classical approach for doing this would involve decomposing each of the dumbbell heads into sufficiently small blocks so that the error committed can be charged against the resulting edge of the MST. Unfortunately, this will result in an unacceptably high running time. In contrast, our approach is sensitive to the weight of the MST in the vicinity of the dumbbell heads. We decompose the blocks in a breadth-first manner until the number of nonempty subblocks in either of the dumbbell heads is roughly $1/\varepsilon_w$. We will exploit the fact that the existence of this many nonempty subblocks implies that the weight of the MST within this dumbbell heads will be sufficient to pay for the approximation error.

More formally, we introduce a parameter α (whose exact value will be specified later but can be thought of as being roughly ε_w). We will process each dumbbell $\psi \in \Psi'$ and compute an edge e_ψ joining a representative point in each head of ψ . We do this as follows. We decompose the two heads of ψ in parallel, always maintaining boxes of equal side length until reaching a total of $\Theta(1/\alpha)$ nonempty quadtree boxes or encountering all the points within the head (whichever occurs first). We then examine the representative points from each pair of boxes and keep the closest pair. This takes time $O(1/\alpha^2)$ by brute-force. We choose an arbitrary point from each box in this pair. The edge joining these two points is selected as the representative edge e_ψ . Let E_s denote the edges of the short-edge graph G_s , and let E_ℓ denote the edges computed above. Let G denote the graph $(P(Q), E_s \cup E_\ell)$.

Just as we did for G_s , we can associate each edge of E_ℓ with the pair of micro-blocks that contain the edge's respective endpoints. This defines a graph on the micro-blocks. To complete the long-edge phase, we first prune this graph. If any edge of this graph joins two micro-blocks in the same short-edge connected component, we ignore this edge. We then collapse all the micro-blocks belonging to the same short-edge component into a single vertex, forming a component graph. For any two components, we keep only the shortest edge between them. Since the number of well-separated pairs is $O(1/(\varepsilon_q \varepsilon_w)^d)$, the number of vertices and edges in this graph is similarly bounded. We then compute the MST of this component graph, using any standard MST algorithm in time $O(1/(\varepsilon_q \varepsilon_w)^d \log 1/(\varepsilon_q \varepsilon_w))$. The output of the long-edge phase is the weight of the edges of E_ℓ that remain in the final MST.

Rather than analyzing G directly, it will be easier to analyze a related graph. Let G' denote the subgraph of G with the same vertex set and the following edges. We keep all the edges of E_s , but only a subset E'_ℓ of the edges of E_ℓ selected as follows. For each edge e of $\text{MST}_{>\delta}(P(Q))$, we select the representative edge e_ψ associated with the dumbbell $\psi \in \Psi'$ that separates the endpoints of e .

In the rest of this section, we will show that G' is connected and satisfies the desired weight bound. Due to space limitations, we will only present the main lemmas upon which the result is based. Details can be found in the full version. Our analysis will employ the following lemma, which bounds the weight of the MST in terms of the number of quadtree boxes (see, e.g., [12]).

► **Lemma 10.** *Given a finite point set $P \in \mathbb{R}^d$ and a hypercube grid of side length s , let $m(P)$ denote the number of cells of the grid that contain a point of P . Then $wt(\text{MST}(P)) \geq (s/2) \cdot ((m(P)/2^d) - 1)$.*

For any dumbbell $\psi \in \Psi'$, define z_ψ to be the distance between the closest pair of points that are separated by ψ . The following lemma bounds the total error incurred in selecting the long edges.

► **Lemma 11.** *There exists a constant c (depending on dimension) such that*

$$\sum_{\psi \in \Psi'} (wt(e_\psi) - z_\psi) \leq c \cdot \alpha \cdot \log(1/(\varepsilon_q \varepsilon_w)) \cdot wt(\text{MST}(P(Q))).$$

Setting $\alpha = \varepsilon_w / (4c \cdot \lg(1/(\varepsilon_q \varepsilon_w)))$, by the above lemma, the long edges satisfy the following property:

$$\sum_{\psi \in \Psi'} (wt(e_\psi) - z_\psi) \leq \frac{\varepsilon_w}{4} \cdot wt(\text{MST}(P(Q))). \tag{1}$$

The connectedness of G' follows from the WSPD separation properties.

By combining Eq. (1) above with our earlier observation that each dumbbell contributes at most one edge to $\text{MST}(P(Q))$, it follows that the weight of the long edges of G' , namely $wt(E'_\ell)$, is at most $wt(\text{MST}_{>\delta}(P(Q))) + (\varepsilon_w/4) \cdot wt(\text{MST}(P(Q)))$. Because G' connects the components of E_s , its weight cannot be smaller than the MST weight of the component graph, which is the output of this phase. Therefore, we have the following.

► **Lemma 12 (Long-edge summary).** *Given the output from the short-edge processing, in time $O((1/(\varepsilon_q^d \varepsilon_w^{d+2})) \log^2(1/(\varepsilon_q \varepsilon_w)))$, we can output a set of edges that connects all the short-edge components and whose total weight is at most $wt(\text{MST}_{>\delta}(P(Q))) + (\varepsilon_w/4) \cdot wt(\text{MST}(P(Q)))$.*

3.3 Combining the Short and Long Edges

Let us now combine the results of the short-edge and long-edge phases. By Lemma 9(iii), the total weight of the short edges E_s is at most

$$wt(\text{MST}_{\delta}(P(Q))) + \frac{\varepsilon_w}{2} \cdot wt(\text{MST}(P^+(Q))).$$

By Lemma 12, the total weight of the long-edge phase is at most

$$wt(\text{MST}_{>\delta}(P(Q))) + \frac{\varepsilon_w}{4} \cdot wt(\text{MST}(P(Q))).$$

Combining the weights of both phases, we find that the total weight $W(Q)$ output is at most

$$W(Q) = wt(\text{MST}(P(Q))) + \frac{\varepsilon_w}{2} \cdot wt(\text{MST}(P^+(Q))) + \frac{\varepsilon_w}{4} \cdot wt(\text{MST}(P(Q))).$$

Since $P(Q) \subseteq P^+(Q)$, we have $wt(\text{MST}(P(Q))) \leq 2 \cdot wt(\text{MST}(P^+(Q)))$. (This follows from the facts that Steiner tree weight increases monotonically as points are added and that the weight of the MST is at most twice the weight of the Steiner tree.) Therefore, we have

$$W(Q) \leq wt(\text{MST}(P(Q))) + \varepsilon_w \cdot wt(\text{MST}(P^+(Q))).$$

If $wt(\text{MST}(P(Q))) \leq wt(\text{MST}(P^+(Q)))$, then $W(Q)$ can be bounded by $(1 + \varepsilon_w) \cdot wt(\text{MST}(P^+(Q)))$. On the other hand, if $wt(\text{MST}(P(Q))) > wt(\text{MST}(P^+(Q)))$, this can be bounded by $(1 + \varepsilon_w) \cdot wt(\text{MST}(P(Q)))$. By defining $P' = P(Q)$ and P'' to be whichever set yields the larger MST weight, we obtain the following bound

$$wt(\text{MST}(P')) \leq W(Q) \leq (1 + \varepsilon_w) \cdot wt(\text{MST}(P'')),$$

where $P \cap Q^- \subseteq P' \subseteq P'' \subseteq P \cap Q^+$. Therefore, this is a valid answer to the $(\varepsilon_q, \varepsilon_w)$ -approximate MST query.

By Lemma 9, the running time of the short-edge phase is $O(\log n + 1/(\varepsilon_q \varepsilon_w)^d)$, and by Lemma 12, the running time of the long-edge phase is $O((1/(\varepsilon_q^d \varepsilon_w^{d+2})) \log^2(1/\varepsilon_q \varepsilon_w))$. Thus, the overall query time is $O(\log n + (1/(\varepsilon_q^d \varepsilon_w^{d+2})) \log^2(1/\varepsilon_q \varepsilon_w))$. In summary, we have the following result, which is stated more concisely in Theorem 1.

► **Theorem 13.** *Given a set P of n points in \mathbb{R}^d and a weight-approximation parameter $\varepsilon_w > 0$, it is possible to preprocess P into a data structure of space $O(n)$ such that given any ε_q -approximate query Q , it is possible to answer $(\varepsilon_q, \varepsilon_w)$ -approximate MST queries in time $O(\log n + (1/(\varepsilon_q^d \varepsilon_w^{d+2})) \log^2(1/\varepsilon_q \varepsilon_w))$.*

4 Conclusions

We have demonstrated an efficient data structure for answering approximate MST range queries. Although our query processing focused only on returning the approximate weight, our data structure implicitly provides much more information. In particular, the weight returned is an accumulation of three disjoint edge sets, the global MST edges induced on the approximate query range, a set of local connecting edges, and the long edges. All of these edges (not just their weights) are stored within the data structure. Thus, unlike sublinear time algorithms for the MST, which provide just an approximation to the weight, our data structure implicitly provides a certificate for its answer. This certificate could be output, which would result in a data structure for approximate MST range reporting queries. Alternatively, the edges of the certificate could be randomly sampled, which would allow a user to compute statistics about this graph, such as the distribution of its edge weights.

There are two obvious shortcomings with our approach. First, our answer is the weight of a graph on a set of points within the approximate query region, which spans these points but may contain cycles. An obvious open problem is whether it is possible to efficiently compute the exact weight of a graph that is a spanning tree on some subset of points that constitutes a valid answer to the approximate range query. Second, our approximation bounds involve two sets P' and P'' , one for the lower bound and one for the upper bound. It would be nice to relate the result to the weight of the MST on a single point set.

References

- 1 P. K. Agarwal, L. Arge, S. Govindarajan, J. Yanga, and K. Yi. Efficient external memory structures for range-aggregate queries. *Comput. Geom. Theory Appl.*, 46:358–370, 2013.
- 2 A. Andoni, A. Nikolov, K. Onak, and G. Yaroslavtsev. Parallel algorithms for geometric graph problems. In *Proc. 46th Annu. ACM Sympos. Theory Comput.*, pages 574–583, 2014.
- 3 S. Arora. Polynomial time approximation schemes for Euclidean traveling salesman and other geometric problems. *J. Assoc. Comput. Mach.*, 45:753–782, 1998.
- 4 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.
- 5 S. Arya and D. M. Mount. Approximate range searching. *Comput. Geom. Theory Appl.*, 17:135–163, 2000.
- 6 S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. *J. Assoc. Comput. Mach.*, 45:891–923, 1998.
- 7 M. J. Bannister, W. E. Devanny, M. T. Goodrich, J. A. Simons, and L. Trott. Windows into geometric events: Data structures for time-windowed querying of temporal point sets. In *Proc. 26th Canad. Conf. Comput. Geom.*, 2014.
- 8 M. J. Bannister, C. DuBois, D. Eppstein, and P. Smyth. Windows into relational events: data structures for contiguous subsequences of edges. In *Proc. 24th Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 856–864, 2013. (arXiv:1209.5791).
- 9 P. Brass, C. Knauer, C.-S. Shin, M. Smid, and I. Vigan. Range-aggregate queries for geometric extent problems. In *Proc. 19th Computing: The Australasian Theory Symposium (CATS)*, pages 3–10, 2013.
- 10 P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point sets with applications to k -nearest-neighbors and n -body potential fields. *J. Assoc. Comput. Mach.*, 42:67–90, 1995.
- 11 P. B. Callahan and S. R. Kosaraju. Faster algorithms for some geometric graph problems in higher dimensions. In *Proc. Eighth Annu. ACM-SIAM Sympos. Discrete Algorithms*, pages 291–300, 1997.
- 12 A. Czumaj, F. Ergün, L. Fortnow, A. Magen, I. Newman, R. Rubinfeld, and C. Sohler. Approximating the weight of the Euclidean minimum spanning tree in sublinear time. *SIAM J. Comput.*, 35:91–109, 2005.
- 13 A. Czumaj and C. Sohler. Estimating the weight of metric minimum spanning trees in sublinear time. *SIAM J. Comput.*, 39:904–922, 2009.
- 14 G. Frahling, P. Indyk, and C. Sohler. Sampling in dynamic data streams and applications. *Internat. J. Comput. Geom. Appl.*, 18:3–28, 2008.
- 15 Y. Nekrich and M. Smid. Approximating range-aggregate queries using coresets. In *Proc. 22nd Canad. Conf. Comput. Geom.*, pages 253–256, 2010.
- 16 D. Papadias, Y. Tao, K. Mouratidis, and K. Hui. Aggregate nearest neighbor queries in spatial databases. *ACM Transactions on Database Systems (TODS)*, 30:529–576, 2005.
- 17 E. Park and D. M. Mount. Output-sensitive well-separated pair decompositions for dynamic point sets. In *Proc. 21st Internat. Conf. on Advances in Geographic Information Systems*, pages 364–373, 2013. (doi: 10.1145/2525314.2525364).
- 18 J. Shan, D. Zhang, and B. Salzberg. On spatial-range closest-pair query. In T. Hadzilacos, Y. Manolopoulos, J. Roddick, and Y. Theodoridis, editors, *Advances in Spatial and Temporal Databases*, volume 2750 of *Lecture Notes in Computer Science*, pages 252–269. Springer, Berlin, 2003.
- 19 Y. Tao and D. Papadias. Range aggregate processing in spatial databases. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 16:1555–1570, 2004.

Maintaining Contour Trees of Dynamic Terrains*

Pankaj K. Agarwal¹, Thomas Mølhave², Morten Revsbæk³,
Issam Safa⁴, Yusu Wang⁵, and Jungwoo Yang³

1 Department of Computer Science, Duke University, USA

2 Scalable Algorithmics – SCALGO, USA

3 MADALGO – Center for Massive Data Algorithmics, Aarhus University,
Denmark

4 Computational Lithography Group, Intel Corporation, USA

5 Department of Computer Science and Engineering, The Ohio State University,
USA

Abstract

We study the problem of maintaining the contour tree \mathbb{T} of a terrain Σ , represented as a triangulated xy -monotone surface, as the heights of its vertices vary continuously with time. We characterize the combinatorial changes in \mathbb{T} and how they relate to topological changes in Σ . We present a kinetic data structure (KDS) for maintaining \mathbb{T} efficiently. It maintains certificates that fail, i.e., an *event* occurs, only when the heights of two adjacent vertices become equal or two saddle vertices appear on the same contour. Assuming that the heights of two vertices of Σ become equal only $O(1)$ times and these instances can be computed in $O(1)$ time, the KDS processes $O(\kappa + n)$ events, where n is the number of vertices in Σ and κ is the number of events at which the combinatorial structure of \mathbb{T} changes, and processes each event in $O(\log n)$ time. The KDS can be extended to maintain an augmented contour tree and a join/split tree.

1998 ACM Subject Classification F.2.2 [Nonnumerical Algorithms and Problems] Geometrical problems and computations

Keywords and phrases Contour tree, dynamic terrain, kinetic data structure

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.796

1 Introduction

Let \mathbb{M} be a triangulation of \mathbb{R}^2 , also known as a *triangulated irregular network* (TIN), and let $h : \mathbb{M} \rightarrow \mathbb{R}$ be a continuous function, often called a *height function*, that is linear within each triangle of \mathbb{M} . The graph of h , denoted by Σ , is a triangulated xy -monotone surface in \mathbb{R}^3 and is called a *terrain*. There has been extensive work in computational geometry, GIS, and spatial databases on the design and analysis of terrain-analysis algorithms such as flood-risk analysis, visibility analysis, and navigation.

Given a height value ℓ , the level set of the height function h is the set of all points in \mathbb{M} whose height values are ℓ . As ℓ varies, the level set continuously deforms and its topology changes at certain heights. Level sets and their topology are often used for the analysis and visualization of height functions. The contour tree of a height function encodes the evolution

* P. A. and T. M. supported by the ARO contract W911NF-13-P-0018; P. A. also supported by NSF under grants CCF-09-40671, CCF-10-12254, and CCF-11-61359, and by Grant 2012/229 from the U.S.–Israel Binational Science Foundation; M. R. and J. Y. supported by Center for Massive Data Algorithmics, a Center of the Danish National Research Foundation; Y. W. supported by NSF under grants CCF-0747082 and CCF-1319406.



of the level sets and succinctly represents the topology of all level sets, and it has found applications in a wide array of data analysis and visualization problems [4, 6, 10, 11, 15].

A variety of applications involve height functions that vary with time. In some cases, the height function may vary continuously with time (e.g., temperature, air pressure, etc.), or it may be updated dynamically at discrete time values (e.g., the height function models the elevation of points on the Earth, and the elevation is updated because of new measurements reflecting the changes due to natural processes or human activity, or a user interacting with a GIS may wish to change the elevation at some places to see its impact on the hydrology, mobility, visibility of the surface). Motivated by these applications we study how the contour tree changes as the height function varies with time. Even if the height function is updated in discrete steps, the only efficient way we know to update the contour tree is to treat such an update as a continuous change from the old value to the new value. We therefore focus on the case when the height function varies continuously with time.

Related work. Van Kreveld et al. [20] gave an $O(n \log n)$ -time algorithm for constructing the contour tree of a piecewise-linear height function on \mathbb{R}^2 , where n is the number of linear pieces in the height function. Their algorithm was extended to \mathbb{R}^3 by Tarasov and Vyalıy [19], and to arbitrary dimensions by Carr et al. [9]. Agarwal et al. [3] gave an I/O-efficient algorithm for constructing the contour tree of a terrain representation that does not fit in main memory.

There has been some work on contour trees of time-varying height functions. Sohn and Bajaj [17] and Szymczak [18] compute a mapping between the contour trees of the height function at successive time steps, but they ignore (possibly many) combinatorial changes in the contour tree between the time steps. Furthermore they compute the contour tree at each time step using a static algorithm. Edelsbrunner et al. [12] study how the Reeb graph (generalization of contour tree for height functions over manifolds with non-zero genus) of a smooth function on three-dimensional space evolves over time. They characterize the combinatorial changes in the Reeb graph, and they describe an algorithm for updating the Reeb graph whenever a combinatorial change occurs. Their algorithm, however, works in an off-line setting, i.e., they assume that the height function over all time values is given in advance, and the algorithm takes $O(n)$ time to update the graph at each event.

There is extensive work in computational geometry on maintaining geometric/topological structures over a set of continuously moving or varying objects mostly under the kinetic data structure (KDS) framework introduced by Basch et al. [5]. See [13] for a survey of this line of work.

Our results. We describe the first KDS for maintaining the contour tree of a time-varying piecewise-linear height function over a simple triangulated (zero-genus) 2-manifold. The KDS can be extended to maintain the augmented contour tree and the join/split tree.

Our first result is a detailed characterization of the combinatorial changes in the contour tree, more refined than what is described in [12], and how they relate to topological changes in the height function (Section 3). This refined characterization, together with the use of two auxiliary trees, enables us to develop an efficient algorithm for updating the contour tree at each combinatorial change. The second result (Section 4) is a linear-size KDS for maintaining the contour tree. Assuming that the height of each vertex, as a function of time, is specified by a polynomial of constant degree and that the roots of these polynomials can be computed in $O(1)$ time, the KDS can update the contour tree in $O(\log n)$ time at each event. The KDS processes a total of $O(\kappa + n)$ events, where κ is the number of combinatorial

changes in the contour tree and n is the number of vertices in the triangulation. Finally we adapt our KDS to maintain the augmented contour tree and the join/split tree (Section 5).

2 Preliminaries

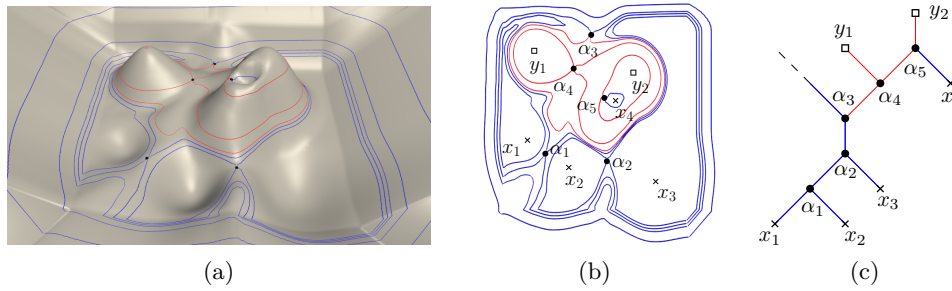
Terrains. Let $\mathbb{M} = (V, E, F)$ be a triangulation of \mathbb{R}^2 , with vertex, edge, and face (triangle) sets V , E , and F , respectively, and let $n = |V|$. For simplicity we focus on \mathbb{M} being a triangulation of \mathbb{R}^2 , but our algorithm works for any genus-zero 2-manifolds. We assume that V contains a vertex v_∞ at infinity, and that each edge $\{u, v_\infty\}$ is a ray emanating from u ; the triangles in \mathbb{M} incident to v_∞ are unbounded. Let $h : \mathbb{M} \rightarrow \mathbb{R}$ be a *height function*. We assume that the restriction of h to each triangle of \mathbb{M} is a linear map, that h approaches $-\infty$ at v_∞ , and that the heights of all vertices are distinct. Given \mathbb{M} and h , the graph of h , called a *terrain* and denoted by Σ , is an xy -monotone triangulated surface whose triangulation is induced by \mathbb{M} . The vertices, edges, and faces of Σ are in one-to-one correspondence with those of \mathbb{M} and with a slight abuse of terminology we refer to V , E , and F , as vertices, edges, and triangles of both Σ and \mathbb{M} .

Critical points. For a vertex v of \mathbb{M} , the *star* of v , denoted by $\text{St}(v)$, consists of all triangles incident on v . The *link* of v , denoted by $\text{Lk}(v)$, is the boundary of $\text{St}(v)$, i.e., the cycle formed by the edges of \mathbb{M} that are not incident on v but belong to the triangles of $\text{St}(v)$. The lower (resp. upper) link of v , $\text{Lk}^-(v)$ (resp. $\text{Lk}^+(v)$), is the subgraph of $\text{Lk}(v)$ induced by vertices u with $h(u) < h(v)$ (resp. $h(u) > h(v)$).

A *minimum* (resp. *maximum*) of \mathbb{M} is a vertex v for which $\text{Lk}^-(v)$ (resp. $\text{Lk}^+(v)$) is empty. A maximum or a minimum vertex is called an *extremal* vertex. A non-extremal vertex v is *regular* if $\text{Lk}^-(v)$ (and also $\text{Lk}^+(v)$) is connected, and *saddle* otherwise. A vertex that is not regular is called a *critical* vertex. For simplicity, we assume that each saddle vertex v is *simple*, meaning that $\text{Lk}^-(v)$ (and also $\text{Lk}^+(v)$) consists of two connected components. If \mathbb{M} contains a non-simple saddle, then we can split it into multiple simple saddles.

Level sets and contours. For $\ell \in \mathbb{R}$, the ℓ -*level set*, ℓ -*sublevel set*, and ℓ -*superlevel set* of \mathbb{M} , denoted by \mathbb{M}_ℓ , $\mathbb{M}_{<\ell}$, $\mathbb{M}_{>\ell}$, respectively, consist of points $x \in \mathbb{M}$, with $h(x) = \ell$, $h(x) < \ell$, and $h(x) > \ell$, respectively. We refer to a level set \mathbb{M}_ℓ where $\ell = h(v)$ for some critical vertex v as a *critical level*. A *contour* of \mathbb{M} is a connected component of a level set of \mathbb{M} . Each vertex $v \in V$ is contained in exactly one contour in $\mathbb{M}_{h(v)}$, which we call *the contour of v* . A contour not passing through a critical vertex is a simple polygonal cycle with non-empty interior. A contour passing through an extremal vertex is a single point, and by our assumption, a contour passing through a saddle consists of two simple cycles with the saddle vertex being their only intersection point. A contour C not passing through a vertex can be represented by the cyclic sequence of edges of \mathbb{M} denoted by $\mathbb{E}(C)$, that it passes through. Two contours are called *combinatorially identical* if their cyclic sequences are the same.

Let $\varepsilon = \varepsilon(\Sigma)$ denote a sufficiently small positive value, in particular, smaller than the height difference between any two vertices of \mathbb{M} . An *up-contour* of a saddle vertex α is any contour of $\mathbb{M}_{h(\alpha)+\varepsilon}$ that intersects an edge incident on α . Similarly, a *down-contour* of α is any contour of $\mathbb{M}_{h(\alpha)-\varepsilon}$ that intersects an edge incident on α . If α has two up-contours and one down-contour it is called a *positive saddle vertex*. If it has two down-contours and one up-contour it is called a *negative saddle vertex*. A simple saddle v is either negative or positive.



■ **Figure 1** (a) (b) An example terrain depicted with contours through saddle vertices and showing the critical vertices of the terrain: α_1 (α_5) is a blue (red) $-ve$ saddle, and α_3 (α_4) is a lue (red) $+ve$ saddle. (c) The contour tree of the terrain in (a).

Following the notation in [1], a contour C of \mathbb{M}_ℓ is called *blue* if points locally in the interior of C belong to $\mathbb{M}_{<\ell}$ and *red* otherwise. A positive (resp. negative) saddle vertex is colored by the color of its unique down-contour (resp. up-contour). Refer to Figure 1 to see the possible saddle colors.

Contour trees. Consider raising ℓ from $-\infty$ to ∞ . The contours continuously deform, but no changes happen to the topology of the level set as long as ℓ varies between two consecutive critical levels. A new contour appears as a single point at a minimum vertex, and an existing contour contracts into a single point and disappears at a maximum vertex. An existing contour (the down-contour of v) splits into two new contours (the up-contours of v) at a positive saddle vertex v , and two contours (the down-contours of v) merge into one contour (the up-contour of v) at a negative saddle vertex v . The *contour tree* \mathbb{T} of h is a tree on the critical vertices of \mathbb{M} that encodes these topological changes of the level set. An edge (v, w) of \mathbb{T} represents the contour that appears at v and disappears at w .

More formally, two contours C_1 and C_2 at levels ℓ_1 and ℓ_2 , respectively, are called *equivalent* if C_1 and C_2 belong to the same connected component of $\Gamma = \{x \in \mathbb{R}^2 \mid \ell_1 \leq h(x) \leq \ell_2\}$ and that component of Γ does not contain any critical vertex. An equivalence class of contours starts and ends at critical vertices. If a class starts at a critical vertex v and ends at w , then (v, w) is an edge in \mathbb{T} . We refer to v as a *down neighbor* of w and to w as an *up neighbor* of v . Equivalently \mathbb{T} is the quotient space in which each contour is represented by a point and connectivity is defined in terms of the quotient topology. Let $\rho : \mathbb{M} \rightarrow \mathbb{T}$ be the associated quotient map, which maps all points of a contour to a single point on an edge of \mathbb{T} . Fix a point p in \mathbb{M} . If p is not a critical vertex, $\rho(p)$ lies in the relative interior of an edge in \mathbb{T} ; if p is an extremal vertex, $\rho(p)$ is a leaf node of \mathbb{T} ; and if p is a saddle vertex then $\rho(p)$ is a non-leaf node of \mathbb{T} . See Figure 1.

We assume that each vertex of \mathbb{T} is labeled with the corresponding critical vertex of \mathbb{M} . The combinatorial description of \mathbb{T} is the set of its vertices along with their labels, and the set of its edges. We also consider augmenting the contour tree with regular vertices to produce the *augmented contour tree* \mathbb{T}_A . For each regular vertex v in \mathbb{M} , we insert a degree two vertex into \mathbb{T} at $\rho(v)$.

Ascent and descent trees. We construct *descent trees* as follows: For each vertex $u \in \mathbb{M}$, if u is not a minimum then we choose a vertex $w \in \text{Lk}^-(u)$ and create the edge (u, w) . This process results in a forest of trees, each rooted at a minimum vertices of \mathbb{M} . We denote the descent tree rooted at x as $\Pi^-(x)$. Similarly, we construct a forest of *ascent trees*, obtained

by creating the edge (v, w) from every vertex v to a single vertex w in $\text{Lk}^+(v)$ unless v is a maximum. Each ascent tree is rooted at a maximum vertex y and is denoted by $\Pi^\uparrow(y)$. Note that the descent tree forest partitions the vertices of \mathbb{M} and the same is true for ascent trees. Lemma 1 below describes how the ascent and descent trees relate to the contour tree.

► **Lemma 1.** *Let v be a regular vertex in \mathbb{M} , and let x (resp. y) be the minimum (resp. maximum) of \mathbb{M} such that $\Pi^\downarrow(x)$ (resp. $\Pi^\uparrow(y)$) contains v . Then the path P in \mathbb{T} between x and y contains $\rho(v)$ and the heights of the vertices on P are monotone.*

3 Time Varying Contour Tree

Suppose the height function varies with time. That is, we have a one parameter family of height functions over \mathbb{M} , $h : \mathbb{M} \times \mathbb{R} \rightarrow \mathbb{R}$, where the extra dimension in the domain is time. In this section, we characterize how and when the contour tree of (\mathbb{M}, h) changes with time. In particular, we describe all the combinatorial changes in the contour tree. For simplicity, we assume h to be *generic* in the sense: (i) at any given time t , at most two vertices have the same height; (ii) the heights of any two vertices become equal at a finite set of time values; and (iii) if $h(u, t_0) = h(v, t_0)$ then the function $h(u, t) - h(v, t)$ changes sign at $t = t_0$.

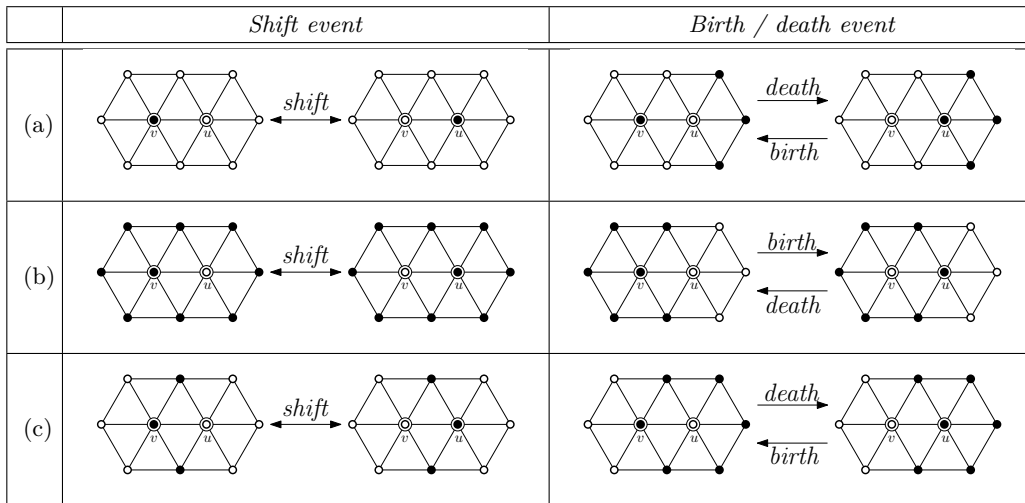
Edelsbrunner et al. [12] showed that if h is a smooth function, then the topology of \mathbb{T} changes at time t if (i) a critical point u becomes degenerate (i.e., the Hessian at u becomes singular), or (ii) two saddle points u and v lie on the same contour. In either case, an edge (u, v) of \mathbb{T} is degenerate in the sense that the interval $[h(u, t), h(v, t)]$ is a single point. In our setting, where $h(\cdot, t)$ is a piecewise-linear function, the two corresponding events are: (i) two adjacent vertices u and v with $h(u, t) = h(v, t)$, and one of them is a saddle and the other is an extremal vertex just before or after the event; (ii) two saddle vertices lie on the same contour. The former event is called a *birth* or a *death* event, and the latter is called an *interchange* event.

Besides these two events, there is another event in the piecewise-linear case, namely, a critical point shifts from one vertex to its neighbor — no new critical points are created, none is destroyed, and the topology of \mathbb{T} does not change. Only the label of a node in \mathbb{T} changes. We refer to this event as a *shift* event. We will refer to events occurring when the heights of two neighboring vertices become equal as *local* events. We note that an interchange event can also occur when the height of two adjacent vertices becomes equal, so a local event may correspond to an interchange event as well.

If an event occurs at time t , then we refer to t^- (resp. t^+) as the time $t - \varepsilon$ (resp. $t + \varepsilon$) for some arbitrarily small $\varepsilon > 0$.

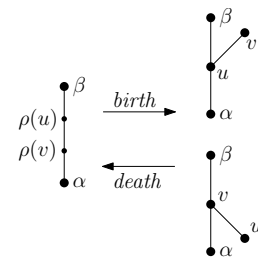
Local events. Suppose a local event occurs at time t_0 at which $h(v, t_0) = h(u, t_0)$, where u is a neighbor of v . For simplicity, we assume that $h(v, t_0^-) < h(u, t_0^-)$ and $h(v, t_0^+) > h(u, t_0^+)$. The case when $h(v, t_0^-) > h(u, t_0^-)$ is symmetric. In the following sections we describe in detail the changes that occur in \mathbb{T} during the three kinds of local events. We assume the interval $[t_0^-, t_0^+]$ to be sufficiently small so that there is no other vertex whose height lies between those of u and v during this interval.

Shift event. A shift event occurs at t_0 if one of u and v , say v , was a critical vertex and the other vertex, u , was a regular vertex at t_0^- , and the critical vertex shifts from v to u at t_0^+ . This event does not cause any change in the topology of \mathbb{T} but the node of \mathbb{T} that was labeled v changes its label to u .



■ **Figure 3** Illustration of local events: (a) v is a minimum vertex; (b) v is a regular vertex; (c) v is a saddle vertex. Hollow vertices are higher than u and v , and filled vertices are lower than u and v . In all examples v is raised, i.e., $h(v, t_0^-) < h(u, t_0^-)$ and $h(v, t_0^+) > h(u, t_0^+)$.

Birth/death event. A *birth event* occurs at time t_0 if both u and v were regular vertices at t_0^- , and they become critical vertices at t_0^+ . A *death event* occurs when both u and v were critical vertices at t_0^- and become regular vertices at t_0^+ . See Figure 2 for the change in the topology of \mathbb{T} . We now describe in detail how \mathbb{T} changes at a birth event; a death event is similar. There are two possibilities: (i) v becomes a negative saddle and u a minimum, or (ii) v becomes a maximum and u a positive saddle. Suppose $\rho(u), \rho(v)$ lie on the edge (α, β) of \mathbb{T} . Then we split (α, β) into two edges by adding a node corresponding to the new saddle and creating a new edge incident on this node whose other endpoint is a leaf. In case (i), v is the node added on the edge (α, β) and u is the new leaf, and in (ii) u is the node on (α, β) and v is the new leaf.



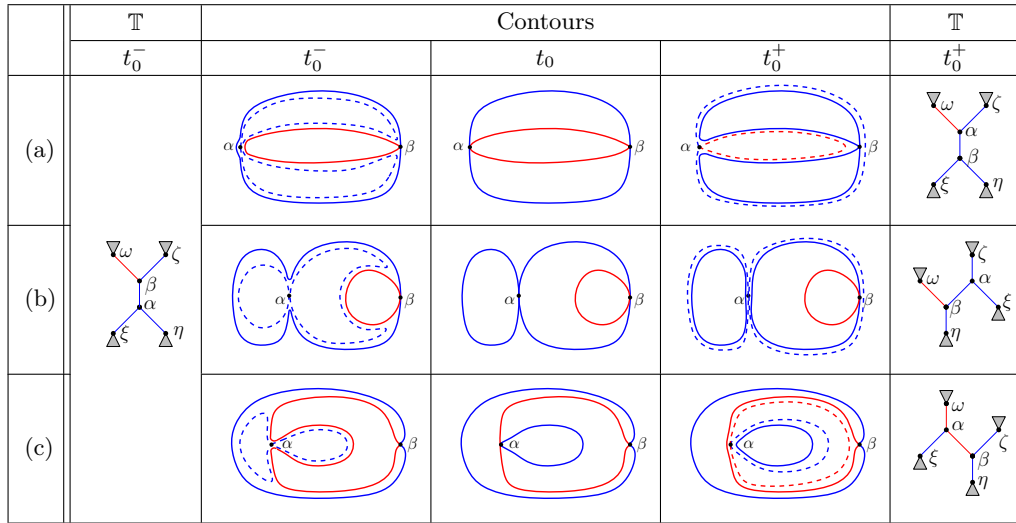
■ **Figure 2** Illustration of the change in the topology of the contour tree in birth/death events.

A local event that corresponds to an interchange event is described in the next subsection. Using an exhaustive case analysis (omitted from this version), it can be shown that the above description includes all possible changes in the contour tree that may be caused by a local event. See Figure 3.

3.1 Interchange events

An interchange event occurs at time t_0 if two saddle vertices α, β lie on the same contour, i.e., $\rho(\alpha) = \rho(\beta)$, at time t_0 . Suppose $h(\alpha, t_0^-) < h(\beta, t_0^-)$. There are four cases, depending on whether α and β are positive or negative saddles. Let us assume that α is a negative saddle. (As we will see below the case when α is positive can be reduced to this case by reversing the z -axis and/or time axis.) Then there are two cases: (i) α is negative and β is positive, and (ii) both α and β are negative saddles. We refer to them as *mixed* and *negative* interchange events, respectively. We need a few notations to describe the interchange events.

At time t_0^- all contours in the equivalence class (α, β) are combinatorially identical because $\rho(w)$, for any vertex w of \mathbb{M} , does not lie on the interior of the edge (α, β) of \mathbb{T} .



■ **Figure 4** Illustration of topological changes in \mathbb{M}_t and contour tree transitions during a mixed interchange event. Dashed contour lines are contours of α and solid contour lines are contours of β at time t_0^- and t_0^+ ; each of them consists of two simple cycles. The two contours merge into one contour at time t_0 (middle column). (a) a sign change event, (b) a blue event, (c) a red event.

With a slight abuse of notation, we will therefore simply refer to all contours in the class (α, β) as the contour C^- without specifying a height. Similarly, at time t_0^+ , all contours in the class (β, α) are combinatorially identical, and we refer to them as C^+ . We label the vertices of C^- and of C^+ that lie on edges incident to α (resp. β) with α (resp. β).

Mixed interchange event. In this case, α is negative and β is positive. We assume that the edge (α, β) is blue at t_0^- , so both α and β are blue at t_0^- (see Section 2). The case when (α, β) is red reduces to this case by reversing the direction of the z -axis. Let ξ and η be the two down neighbors of α , and let ζ, ω be the up neighbors of β at time t_0^- . Since α is blue, both (ξ, α) and (η, α) are blue edges. Since β is blue, one of (β, ζ) and (β, ω) is red and the other is blue (See Figure 1). Assume that the edge (β, ζ) is blue and (β, ω) is red, i.e., the up-contour C_ω^- of β is red and the up-contour C_ζ^- of β is blue. Refer to Figure 4.

Since α is a negative saddle, β is a positive saddle, and C^- intersects both components of $\text{Lk}^+(\alpha)$ and of $\text{Lk}^-(\beta)$, the vertices of C^- labeled with α form two disconnected intervals in C^- , and the same is true for vertices labeled with β .¹ Since β is the only vertex between C^- and the up-contours C_ζ^- and C_ω^- of β , every vertex of C^- either lies on an edge of $\mathbb{E}(C_\zeta^-) \cup \mathbb{E}(C_\omega^-)$ or is labeled with β . We mark the vertices of C^- that lie on the edges of $\mathbb{E}(C_\omega^-)$ as red and the vertices that lie on the edges of $\mathbb{E}(C_\zeta^-)$ as blue, in accordance with the colors of the contours C_ω^- and C_ζ^- , respectively. Refer to Figure 5. Similarly, let C_ξ^- and C_η^- be the down-contours of α , then every vertex of C^- either lies on an edge of $\mathbb{E}(C_\xi^-) \cup \mathbb{E}(C_\eta^-)$ or is labeled with α . There are three types of mixed interchange events depending on the relative positions of the vertices of C^- that are marked α or β . See Figure 5.

¹ If α and β are adjacent in \mathbb{M} , there is a vertex of C^- (the intersection point of C^- with the edge $\alpha\beta$ of \mathbb{M}) that is marked both α and β . In this case, we simply consider vertices marked only α (resp. β). It can be shown that β is an endpoint of a connected component of $\text{Lk}^+(\alpha)$ and that component contains more than one vertex. Therefore the vertices of C^- marked only α form two disconnected intervals in C^- , and a similar argument applies to β .

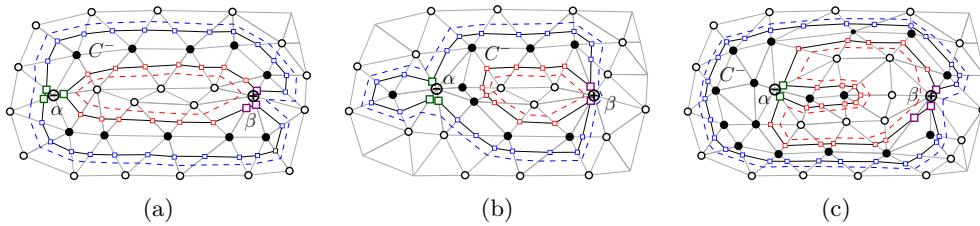


Figure 5 Illustration of the possible colorings of C^- for mixed interchange events. α is a negative saddle and β is positive. Green (resp. purple) contour vertices represent the vertices marked with α (resp. β). Hollow vertices of \mathbb{M} have height higher than both α and β , and filled vertices have lower height. Blue dashed lines indicate C_ξ^- , and red dash lines C_ω^- . (a) Vertices marked α in C^- lie on the edges of both $\mathbb{E}(C_\xi^-)$ and $\mathbb{E}(C_\omega^-)$, (b) vertices marked α lie on the edges of $\mathbb{E}(C_\xi^-)$, (c) vertices marked α lie on the edges of $\mathbb{E}(C_\omega^-)$.

- (i) Vertices marked α and β in C^- are interleaved, i.e., vertices marked α are both red and blue; we refer to this as a *sign-interchange* event.
- (ii) All vertices marked α are blue; we refer to this as a *blue* event.
- (iii) All vertices marked α are red; we refer to this as a *red* event.

In case (ii) and (iii), all vertices marked β lie in one of $\mathbb{E}(C_\xi^-)$ and $\mathbb{E}(C_\eta^-)$. Without loss of generality, assume that they lie on the edges of $\mathbb{E}(C_\eta^-)$. The following lemma then characterizes the change of \mathbb{T} at a mixed interchange event.

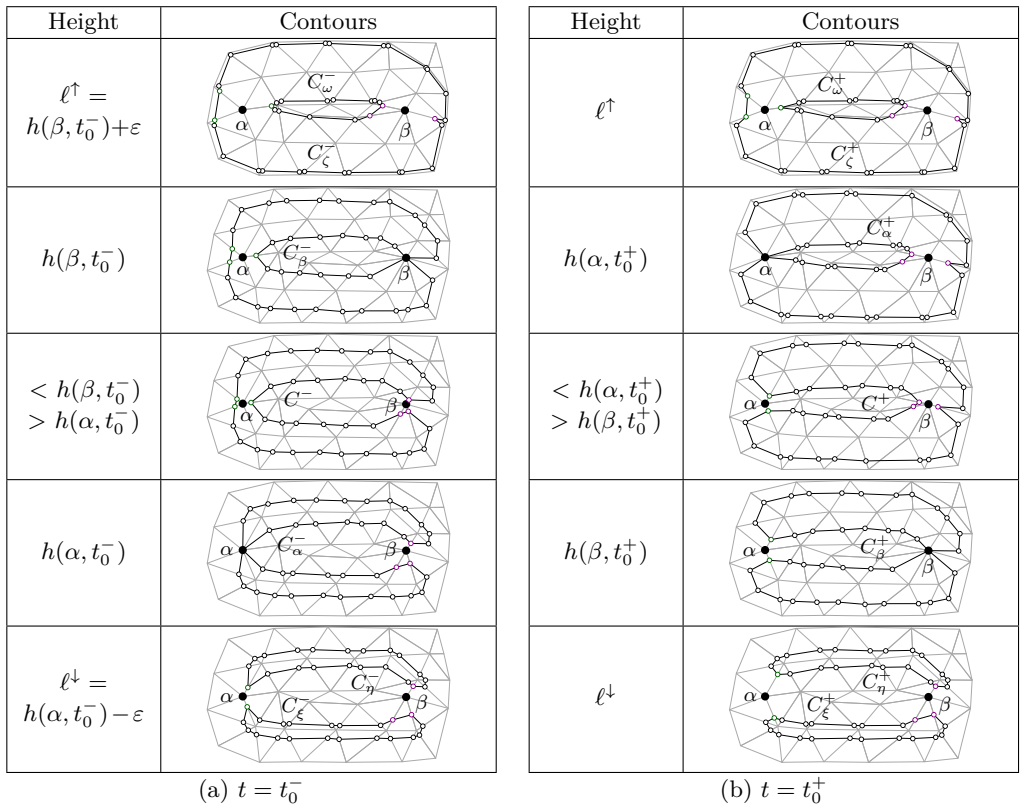
► **Lemma 2.** *Suppose a mixed interchange event occurs at time t_0 involving an edge (α, β) of \mathbb{T} . If the edge (α, β) is blue at t_0^- and $h(\alpha, t_0^-) < h(\beta, t_0^-)$, then the following change occurs in \mathbb{T} at t_0 :*

- (i) *Sign-interchange event: The topology of \mathbb{T} does not change. The only change in \mathbb{T} is that the label α and β of \mathbb{T} get swapped, i.e., the edge (α, β) becomes (β, α) , and α (resp. β) becomes a positive (resp. negative) saddle at t_0^+ ; Figure 4 (a).*
- (ii) *Blue event: The edge (α, β) becomes (β, α) but the color of (β, α) and of the saddles α, β remain blue. At time t_0^+ , η is the down neighbor of β , α and ω are the up neighbors of β , ξ and β are the down neighbors of α , and ζ is the up neighbor of α ; Figure 4 (b).*
- (iii) *Red event: The edge (α, β) becomes (β, α) , and its color becomes red and so does the saddle α . At time t_0^+ , η is the down neighbor of β , α and ζ are the up neighbors of β , ξ and β are the down neighbors of α , and ω is the up neighbor of α ; Figure 4 (c).*

Proof. Consider the sign-interchange event. Let C_ξ^-, C_η^- and C_ζ^-, C_ω^- be the down-contours of α and the up-contours of β at time t_0^- , respectively. Let $\ell^\downarrow = h(\alpha, t_0^-) - \varepsilon$ and $\ell^\uparrow = h(\beta, t_0^-) + \varepsilon$ be the levels of the down-contours of α and of the up-contours of β , respectively. We can choose the values of t_0^+ and ε carefully so that $\ell^\uparrow > h(\alpha, t_0^+)$.

First we fix the time t_0^- and monitor the contour C^- as we decrease the level toward $h(\alpha, t_0^-)$. The vertices of C^- marked α converge to the vertex α at height $h(\alpha, t_0^-)$, and we obtain the contour C_α^- of α . Since the vertices marked α and β in C^- are interleaved, each of the two simple cycles of C_α^- contains the vertices marked β (i.e., each of them intersects the edges incident on β), and therefore so do C_ξ^- and C_η^- . Similarly, each of C_ζ^- and C_ω^- contains vertices marked α . Refer to Figure 6 (a).

Next we fix the heights ℓ^\downarrow and ℓ^\uparrow and move forward in time from t_0^- to t_0^+ . As time moves toward t_0^+ , $C_\xi^-, C_\eta^-, C_\zeta^-$ and C_ω^- continuously deform but no topological changes occur to any of them. Let us consider the deformation of C_ξ^- . The deformation can be represented by a continuous function $F : \mathbb{S} \times [t_0^-, t_0^+] \rightarrow \mathbb{R}^2$, where $F(\cdot, t_0^-) = C_\xi^-$ and $F(\cdot, t)$ denote the



■ **Figure 6** Illustration of a sign-change event. The vertices marked α are illustrated as green vertices and the vertices marked β as purple vertices.

deformation of C_ξ^- at time $t \in [t_0^-, t_0^+]$. By construction, $\mathbb{E}(F(\cdot, t))$ remains the same for all $t \in [t_0^-, t_0^+]$. Let $C_\xi^+ = F(\cdot, t_0^+)$. Similarly define C_η^+ , C_ζ^+ , and C_ω^+ . See Figure 6 (b).

Now we fix the time to t_0^+ and increase the height from ℓ^\ddagger to ℓ^\dagger and monitor how the contours C_ξ^+ and C_η^+ deform. Recall that $h(\beta, t_0^+) < h(\alpha, t_0^+)$, so as we increase the height, we first encounter β and the vertices of C_ξ^+ and C_η^+ marked β converge to the vertex β of \mathbb{M} . Hence, β is now a negative saddle. Since both C_ξ^+ and C_η^+ are blue, so is the saddle β at t_0^+ . C^+ , the up-contour C^+ of β at time t_0^+ , is therefore a blue contour. $\mathbb{E}(C^+)$ contains all edges of $\mathbb{E}(C_\xi^+) \cup \mathbb{E}(C_\eta^+)$ that are not incident on β plus the edges whose lower endpoint is β (at time t_0^+). Furthermore the vertices marked α and β in C^+ are interleaved (as in C^-). If we continue to increase the height, the vertices of C^+ marked α converge to α as we reach $h(\alpha, t_0^+)$, and C^+ splits into two contours at α . Since C^+ is a blue contour, α is a blue positive saddle at t_0^+ . The up-contours of α at time t_0^+ will be C_ζ^+ and C_ω^+ , so ζ and ω will be up-neighbors of α . See Figure 6 (b). This completes the proof for the sign-interchange case.

Next, suppose the red or blue event occurs at time t_0 . The proof proceeds along the same lines as for the previous case. Let ℓ^\ddagger , ℓ^\dagger , C_ξ^+ , C_η^+ , C_ζ^+ , and C_ω^+ be the same as above. If the vertices marked α and β are not interleaved in C^- , then by our assumption the vertices marked β lie on the edges of $\mathbb{E}(C_\eta^+)$, these vertices converge to β at height $h(\beta, t_0^+)$ and C_η^+ splits into two contours at β . Note that as we increase the height C_ξ^+ also deforms but does not meet β , as it has no vertices marked β . Hence β is a blue positive saddle, with η as the down neighbor of β . Let C_β^+ be the contour of β at time t_0^+ .

C_β^+ consists of two simple cycles, one with vertices lying on edges of $\mathbb{E}(C_\omega^+)$ and the other with vertices lying on edges of $\mathbb{E}(C_\zeta^+)$. As we increase the height to $h(\alpha, t_0^+)$, the vertices

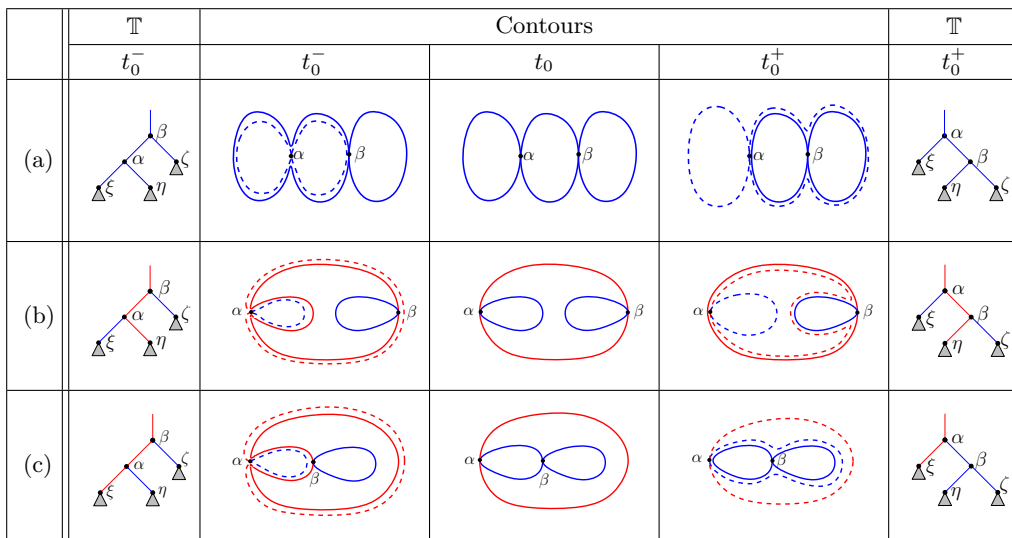


Figure 7 Illustration of topological changes in \mathbb{M}_ℓ and contour tree transitions during negative interchange events. Dashed lines are contours of α and solid lines are contours of β . (a) is when β is blue. (b) and (c) are when β is red.

marked α in C^+ and C_ξ^+ converge to α , and the two contours merge into a single contour C_α^+ at α . If the blue (resp. red) event occurs at t_0 , i.e., vertices marked α lie on the edges of $\mathbb{E}(C_\zeta)$ (resp. $\mathbb{E}(C_\omega)$), then α is a blue (resp. red) negative saddle at time t_0^+ , the up-contour of α is C_ζ^+ (resp. C_ω^+), and the contour C_ω^+ (resp. C_ζ^+) remains an up-contour of β . See Figure 4 (b) and (c). ◀

The above lemma characterizes the changes in the contour tree at a mixed interchange event under the assumption that α was a blue negative saddle at t_0^- . As mentioned above, the other cases can be reduced to the above case. In particular, if α is a red negative saddle at time t_0^- , reverse the direction of the z -axis. Now β becomes a blue negative saddle, α a blue positive saddle, and β lies below α , which is precisely the case described in Lemma 2. If α is a blue positive saddle, reverse the direction of time and swap β and α (see Figure 4 (b,c)). Finally, if α is red positive saddle, then by reversing the direction of time as well as that of the z -axis, we reduce this case to that described in Lemma 2.

Negative interchange event. Let ξ, η be the two down neighbors of α at t_0^- , and let ζ be the other down neighbor of β (α is a down neighbor of β at t_0^-). See Figure 7. The change in topology of \mathbb{T} at a negative interchange event is similar to performing a rotation at node β . That is, the edge (α, β) becomes the edge (β, α) at time t_0^+ , and one of the down subtrees of α (rooted in ξ and η) becomes a down subtree of β . Next we describe the change of \mathbb{T} in more detail.

Let C_ξ^- and C_η^- be the down-contours of α at time t_0^- . Since both α and β are negative saddles and C^- is the up-contour of α and one of the down contour of β (at time t_0^-), the vertices of C^- labeled α form two disconnected intervals and those labeled β form a single connected interval. Since α is the only vertex between C^- and the down-contours of α , vertices of C^- either lie in the interior of an edge in $\mathbb{E}(C_\xi^-) \cup \mathbb{E}(C_\eta^-)$, or is labeled with α . Furthermore the vertices of C^- marked β form a connected interval, the edges incident on β that intersect C^- belong to one of $\mathbb{E}(C_\xi^-)$ and $\mathbb{E}(C_\eta^-)$ but not both. Using these observations,

Figure 7 illustrates different possible cases at a negative interchange event. Lemma 3 below describes how the topology of \mathbb{T} changes. Its proof is similar to that of Lemma 2 and is omitted from this version.

► **Lemma 3.** *If the vertices of C^- labeled β lie on the edges of $\mathbb{E}(C_\eta^-)$, then at time t_0^+ , η becomes a down neighbor of β , ζ the other down neighbor of β , and ξ and β become the down neighbors of α .*

Figure 7 also illustrates how the colors of α and β change at t_0^+ . Indeed, if β is blue at t_0^- then α is also blue at t_0^- , and both of them remain blue at t_0^+ (Figure 7 (a)). If β is red, then α is blue or red. Suppose α is red. If η is red at t_0^- (recall we assume that the edges incident to β that intersect C^- belong to $\mathbb{E}(C_\eta^-)$), then both α and β remain red at t_0^+ (Figure 7 (b)). But if η is blue at t_0^- then α remains red but β becomes blue at t_0^+ (Figure 7 (c)). Finally, if β is red and α is blue at t_0^- , then this corresponds to third case if we reverse the direction of the time axis (see Figure 7 (c), with the roles of α and β being swapped).

4 KDS for \mathbb{T}

In this section, we describe an algorithm for maintaining the contour tree as the height function varies over time, using the KDS framework by Basch et al. [5]. The KDS framework maintains a set of *certificates* that ensure the correctness of the configuration. As the objects move, the certificates fail at certain time instances, called *events*. For each certificate, the closest time that the certificate fails is computed and maintained in an *event queue*. When the current time reaches the time of the next event in the event queue, a repair mechanism is invoked to update the configuration, the set of certificates, and the event queue.

We represent the time-varying height function by specifying the height of each vertex $v \in V$ as a function $f_v : \mathbb{R} \rightarrow \mathbb{R}$ of time. We assume that each f_v is a polynomial and that the maximum degree of these polynomials is bounded by a constant. We also assume that the set $\{f_v \mid v \in V\}$ is generic in the sense described in the beginning of Section 3. We describe the algorithm in the real-RAM model [8], in which various operations on constant-degree polynomials (e.g., finding the roots) can be performed in $O(1)$ time.

Data structure. To maintain \mathbb{T} efficiently, we also maintain: (i) a descent tree $\Pi^\downarrow(x)$ for every minimum x ; (ii) an ascent tree $\Pi^\uparrow(y)$ for every maximum y ; (iii) for every non-extremal vertex v , a *link pointer* to the first vertex of each connected component of $\text{Lk}^-(v)$ and $\text{Lk}^+(v)$, in clockwise order. We store \mathbb{M} using a standard triangulation data structure such as DCEL [7]. We maintain an *event queue* as a priority queue so that the next event can be computed efficiently. We represent the contour tree, the ascent trees, and the descent trees as *link-cut trees* [16], which support each of the following operations in $O(\log n)$ time.

- **LINK**(v, w): Given two vertices v, w , connect the trees containing v and w by inserting the edge (v, w) .
- **CUT**(v, w): Given v, w , split the tree containing v and w by removing the edge (v, w) .
- **ROOT**(v): Return the root of the tree containing v .
- **EXPOSE**(v, w): Return a pointer to a balanced binary search tree storing vertices on the path P from v to w in the order as they appear on P .

These operations enable the following operations on \mathbb{T} : (a) **NEXTTO**(v, w): Given two critical vertices v and w , return the vertex adjacent to v in \mathbb{T} on the path from v to w . (b) **FINDEDGE**(v): For a regular vertex v , return the edge (α, β) of \mathbb{T} that contains $\rho(v)$.

NEXTTO is straightforward and takes $O(\log n)$ time. $\text{FINDEDGE}(v)$ is implemented as follows. Using $\text{ROOT}(v)$ on the ascent trees and the descent trees, we find the minimum x and the maximum y such that $v \in \Pi^\downarrow(x)$ and $v \in \Pi^\uparrow(y)$. By Lemma 1, (α, β) is on the path P between x and y in \mathbb{T} , and the heights of the vertices on P are monotone. We obtain P using $\text{EXPOSE}(x, y)$ and then find (α, β) on P . This procedure takes $O(\log n)$ time.

Certificates. Each edge in an ascent/descent tree is an edge of \mathbb{M} and the link pointers also correspond to the edges of \mathbb{M} , so these auxiliary structures change only when the heights of two endpoints of an edge become equal. The time instance at which an ascent/descent tree or a link pointer needs to be updated is referred to as *auxiliary* event. As described in Section 3, \mathbb{T} changes only when the heights of the endpoints of an edge of \mathbb{M} or of \mathbb{T} become equal. Hence, our KDS maintains the following two sets of certificates to certify the correctness of the structures at any given time: (i) a certificate for each edge (u, v) of \mathbb{M} , that fails when $h(v, t) = h(u, t)$, corresponding to a local event; and (ii) a certificate for each edge (α, β) of \mathbb{T} that fails when $h(\alpha, t) = h(\beta, t)$, corresponding to an interchange event.

Initialization. We initialize by constructing \mathbb{T} at time zero, using the static algorithm [9], in $O(n \log n)$ time. The ascent/descent trees and the link pointers can be initialized in $O(n \log n)$ time. Finally, we initialize the event queue with the certificates associated with edges of \mathbb{M} and \mathbb{T} .

Repair mechanism. We now describe how we update the KDS at each event. Besides \mathbb{T} , we also have to update the auxiliary structures and the event queue. First consider a local event at time t_0 at which $h(v, t_0) = h(u, t_0)$, where (u, v) is an edge of \mathbb{M} . Assume that $h(v, t_0^-) < h(u, t_0^-)$ and $h(v, t_0^+) > h(u, t_0^+)$. This event may be an auxiliary, shift, birth/death, or an interchange event (or multiple of them). Processing a shift event is straightforward as it only involves updating the label of a node of \mathbb{T} (see Section 3) and the certificates, along with their failure times, for the edges of \mathbb{T} adjacent to that node of \mathbb{T} . We will describe the processing of an interchange event later, so we briefly sketch the processing of auxiliary and birth/death events.

Birth/death event. Suppose u, v were regular vertices at t_0^- and become critical vertices at t_0^+ . We find the edge (α, β) of \mathbb{T} that contains u, v using the $\text{FINDEDGE}(u)$ operation. Once we know (α, β) , \mathbb{T} can be updated in $O(\log n)$ time, using LINK and CUT operations. Finally, we update the certificates and their failure times corresponding to the new edges in \mathbb{T} . Processing a death event is similar but simpler since u, v are vertices of \mathbb{T} at t_0^- .

Auxiliary event. Updating the link pointers of u and v is straightforward. Next suppose (v, u) is an edge of an ascent tree at t_0 . We remove the edge (v, u) . If v becomes a maximum at t_0^+ , then v becomes the root of an ascent tree; otherwise we choose another vertex w from $\text{Lk}^+(v)$ and add the edge (v, w) . If u was a maximum at t_0^- , i.e., u was the root of an ascent tree, we add the edge (u, v) . These changes in the ascent tree can be performed in $O(\log n)$ time using LINK , CUT , and ROOT operations. A descent tree is updated in an analogous manner.

Interchange event. We describe how to process a mixed interchange event; other interchange events can be processed similarly. Following the set up in Section 3, suppose a mixed interchange event occurs at t_0 at which $h(\alpha, t_0) = h(\beta, t_0)$ with $h(\alpha, t_0^-) < h(\beta, t_0^-)$, α being a blue negative saddle at t_0^- , and β being a blue positive saddle at t_0^- . Once we know whether this event is sign-interchange, blue, or red event, we can update \mathbb{T} in $O(\log n)$ time in accordance with the characterization of events in Lemma 2.

Borrowing the notation from Section 3, let ζ, ω be the two up neighbors of β at t_0^- with (β, ζ) being blue and (β, ω) being red. Similarly let C^- be a contour of the edge (α, β) , C_ζ^- (resp. C_ω^-) the up-contour of β corresponding to the edge (β, ζ) (resp. (β, ω)). The vertices of the contour C^- labeled α consist of two disconnected intervals, say, I_1 and I_2 , each corresponding to one connected component of $\text{Lk}^+(\alpha)$ at t_0^- . We also note that all edges of $\mathbb{E}(C^-)$ that contain the one connected interval of vertices of C^- marked α belong to $\mathbb{E}(C_\zeta^-)$ or $\mathbb{E}(C_\omega^-)$ but not both. For each of I_1 and I_2 , we determine which of the two sets, $\mathbb{E}(C_\zeta^-)$ or $\mathbb{E}(C_\omega^-)$, these edges belong to. Consider I_1 and choose an arbitrary vertex from this interval. Suppose this vertex lies on the edge $(\alpha, u_1) \in \mathbb{M}$; u_1 can be identified using the link pointers of α . Using the $\text{ROOT}(u_1)$ procedure, we determine the maximum y such that $u_1 \in \Pi^\uparrow(y)$. Consider the path $P_{\beta y}$ from β to y in \mathbb{T} . Let $\gamma_1 \in \{\zeta, \omega\}$ be the vertex next to β in $P_{\beta y}$; γ_1 can be determined by calling $\text{NEXTTO}(\beta, y)$. Then the edge $(\alpha, u_1) \in \mathbb{E}(C_{\gamma_1}^-)$. Similarly we choose a vertex u_2 from the other component of $\text{Lk}^+(\alpha)$ and find the vertex $\gamma_2 \in \{\zeta, \omega\}$ such that $(\alpha, u_2) \in \mathbb{E}(C_{\gamma_2}^-)$. If $\gamma_1 \neq \gamma_2$, then the event at hand is a sign-interchange event, if $\gamma_1 = \gamma_2 = \zeta$ then it is a blue event, and if $\gamma_1 = \gamma_2 = \omega$ then it is a red event. The total time spent in computing γ_1 and γ_2 is $O(\log n)$.

Hence, a mixed interchange event can be processed in $O(\log n)$ time. Similarly other interchange events can be processed in $O(\log n)$ time.

KDS analysis. The shift, birth/death, and interchange events are *external* events as they change \mathbb{T} , and the auxiliary events are *internal* events as they only change auxiliary data structures and do not affect \mathbb{T} . Since auxiliary events are local events, the number of internal events is $O(n)$. Hence, the KDS processes $O(\kappa + n)$ events, where κ is the number of external events. Each certificate of the KDS is associated with an edge of \mathbb{M} or \mathbb{T} , so it maintains $O(n)$ certificates at any time. Each event will cause the update of $O(1)$ number of certificates. The maximum number of certificates in which any one vertex can ever appear is bounded by the degree of the vertex in \mathbb{M} . Although the degree of a vertex in \mathbb{M} can be $\Omega(n)$ in the worst-case, it is often a small constant in practice.

► **Theorem 4.** *Let \mathbb{M} be a triangulation of \mathbb{R}^2 with n vertices, and $h : \mathbb{M} \times \mathbb{R} \rightarrow \mathbb{R}$ a time-varying height function such that the height of each vertex of \mathbb{M} , as a function of time, is specified by a polynomial whose maximum degree is bounded by a constant. The contour tree of h can be maintained by a linear-size KDS that processes $O(\kappa + n)$ events, where κ is the number of external events. The KDS processes each event in $O(\log n)$ time in the real-RAM model.*

► **Remark.** (i) It is easy to construct a height function so that the number of combinatorial changes in \mathbb{T} is $\Omega(n^2)$. As the worst-case number of internal events is much smaller than the worst-case number of external events, our KDS is *weakly efficient* in the terminology of KDS [14]. (ii) If the maximum degree of a vertex is large, our KDS can be modified so that a vertex appears in $O(\log n)$ certificates, but the number of internal events increases to $O(\lambda_s(n) \log n)$, where s is a constant and $\lambda_s(\cdot)$ is the maximum length of an order s Davenport-Schinzel sequence and is almost linear [2]. Each event now takes $O(\log^2 n)$ time to process.

5 Extensions

In this section, we describe kinetic data structures for maintaining the augmented contour tree and join/split tree.

Augmented contour tree. Recall that the augmented contour tree \mathbb{T}_A is obtained by adding to the edges of \mathbb{T} the images of regular vertices of \mathbb{M} under the quotient map ρ . We refer to these newly added degree two vertices in \mathbb{T}_A as *regular vertices* as well, while the original vertices of \mathbb{T} are called *critical vertices*. We can easily modify the KDS framework described in Section 4 to maintain this augmented contour tree \mathbb{T}_A . We provide a brief description below.

Specifically, the only new events that we need to handle are (*regular-regular event*): when two regular vertices u, v lie on the same contour, and (*regular-critical event*): when a regular vertex u and a critical vertex v lie on the same contour. Note that (u, v) is an edge of \mathbb{T}_A in both cases. When a regular-regular event happens, there are two possibilities: (i) either u and v swap their order along the edge (u, v) ; or (ii) it causes a birth event, in which case (u, v) is also necessarily an edge in \mathbb{M} .

At a regular-critical event at time t_0 , there are two cases: (i) u is a regular vertex and v is an extremal vertex. In this case, it is easy to verify that (u, v) has to also be an edge in \mathbb{M} , and it corresponds to a shift event. (ii) u is a regular vertex and v is a saddle vertex. If it does not correspond to a shift event, we need to identify which edge incident on v in \mathbb{T}_A contains u after the event. Assume $h(u, t_0^-) < h(v, t_0^-)$, and let w be a vertex in $\text{Lk}^+(u)$. It is easy to see that w and u are in the same connected component in $\mathbb{M}_{>(h(v, t_0^+)})$ at time t_0^+ . Thus, u is moved onto the incident edge of v whose endpoint is the one obtained by $\text{NEXTTO}(v, y)$, where $\Pi^\uparrow(y)$ contains w . If $h(u, t_0^-) > h(v, t_0^-)$, we consider w as a vertex in $\text{Lk}^-(u)$ and use the descent trees.

The additional events can be clearly handled in $O(\log n)$ time. Note that when a local event associated with an edge (u, v) of \mathbb{M} occurs, there is an edge (u, v) in \mathbb{T}_A and thus certificates associated with edges of \mathbb{M} are unnecessary. Each certificate failure makes a combinatorial change in \mathbb{T}_A . Therefore, this data structure is strongly efficient.

► **Theorem 5.** *Let \mathbb{M} be a triangulation of \mathbb{R}^2 with n vertices, and $h : \mathbb{M} \times \mathbb{R} \rightarrow \mathbb{R}$ a time-varying height function such that the height of each vertex of \mathbb{M} , as a function of time, is specified by a polynomial whose maximum degree is bounded by a constant. The augmented contour tree of h can be maintained by a linear-size KDS that processes $O(\kappa)$ events, where κ is the number of external events. The KDS processes each event in $O(\log n)$ time in the real-RAM model.*

Join and split trees. As described, the contour tree \mathbb{T} encodes the topological changes in \mathbb{M}_ℓ as we increase ℓ from $-\infty$ to ∞ . The *join tree* encodes the topological changes in $\mathbb{M}_{<\ell}$. As we increase ℓ from $-\infty$ to ∞ , connected components in $\mathbb{M}_{<\ell}$ appear at minimum vertices and merge at negative saddle vertices, but they never split or disappear as components never shrink. In the join tree, only minimum vertices and negative saddle vertices appear as nodes. Similarly, the *split tree* encodes the topological changes in $\mathbb{M}_{>\ell}$, in which only maxima and positive saddles appear as nodes. As the join and split trees contain a subset of nodes in the contour tree, the events that occur on these trees of time-varying height functions are a subset of the events that occur on the contour tree. For example, the events that occur on the join tree are the events that involve the minima and negative saddle vertices, i.e., the negative interchange events, sign-interchange change events and a subset of the local events. The algorithm described above can therefore also be used to maintain the join and split trees, the only difference being the reduced set of events. Note that to be able to detect the sign-interchange events efficiently, we need to maintain \mathbb{T} simultaneously with the join tree \mathbb{J} or augment the positive saddle vertices to the edges of \mathbb{J} .

► **Theorem 6.** *Let \mathbb{M} be a triangulation of \mathbb{R}^2 with n vertices, and $h : \mathbb{M} \times \mathbb{R} \rightarrow \mathbb{R}$ a time-varying height function such that the height of each vertex of \mathbb{M} , as a function of time, is specified by a polynomial whose maximum degree is bounded by a constant. The join tree and the split tree of h can be maintained by a linear-size KDS that processes $O(\kappa + n)$ events, where κ is the number of events at which the contour tree of h changes. The KDS processes each event in $O(\log n)$ time in the real-RAM model.*

6 Conclusion

In this paper, we characterized the combinatorial changes in the evolution of the contour tree of a time-varying piecewise linear height function over \mathbb{R}^2 , and described the first KDS for maintaining the contour tree that efficiently handled such changes. Adapting this KDS, we also presented how to maintain the augmented contour tree and join/split tree.

Our analysis of events and our KDS are limited to a height function on simple 2-manifolds. A natural question of course is to extend this analysis and the KDS to a height function on higher dimensional space, say, a simple 3-manifold. A more challenging question is to develop an efficient KDS for maintaining the Reeb graph of a function over a 2-manifold with non-zero genus. Finally, another interesting question is to design a KDS for maintaining the contour tree when not only the height function changes with time but \mathbb{M} itself changes over time.

References

- 1 P. K. Agarwal, L. Arge, T. M. Murali, Kasturi R. Varadarajan, and J. S. Vitter. I/O-efficient algorithms for contour-line extraction and planar graph blocking. In *Proc. 9th ACM-SIAM Sympos. Discrete Algorithms*, pages 117–126, 1998.
- 2 P. K. Agarwal and M. Sharir. Davenport-Schinzel sequences and their geometric applications. In Jörg-Rüdiger Sack and Jorge Urrutia, editors, *Handbook of Computational Geometry*, pages 1–47. Elsevier Science Publishers, 2000.
- 3 Pankaj K. Agarwal, Lars Arge, and Ke Yi. I/O-efficient batched union-find and its applications to terrain analysis. In *Proc. 22nd Annu. Sympos. Comput. Geom.*, pages 167–176, 2006.
- 4 Lars Arge, Morten Revsbæk, and Norbert Zeh. I/O-efficient computation of water flow across a terrain. In *Proc. 26th Annu. Sympos. Comput. Geom.*, pages 403–412, 2010.
- 5 J. Basch, L. J. Guibas, and J. Hershberger. Data structures for mobile data. *J. Algorithms*, 31(1):1–28, 1999.
- 6 K.G. Bemis, D. Silver, P.A. Rona, and C. Feng. Case study: a methodology for plume visualization with application to real-time acquisition and navigation. In *Proc. IEEE Conf. Visualization*, pages 481–494, 2000.
- 7 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.
- 8 Vasco Brattka and Peter Hertling. Feasible real random access machines. *J. Complexity*, 14(4):490–526, December 1998.
- 9 Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing contour trees in all dimensions. *Comput. Geom.*, 24(2):75–94, 2003.
- 10 Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Comput. Geom.*, 43(1):42–58, 2010.
- 11 A. Danner, T. Mølhave, K. Yi, P. K. Agarwal, L. Arge, and H. Mitásová. Terrastream: From elevation data to watershed hierarchies. In *Proc. ACM Sympos. Advances in Geographic Information Systems*, page 28, 2007.

- 12 Herbert Edelsbrunner, John Harer, Ajith Mascarenhas, Valerio Pascucci, and Jack Snoeyink. Time-varying Reeb graphs for continuous space-time data. *Comput. Geom.*, 41(3):149–166, 2008.
- 13 L. Guibas. Modeling motion. In J. Goodman and J. O’Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 1117–1134. Chapman and Hall/CRC, 2nd edition, 2004.
- 14 Leonidas J. Guibas. Kinetic data structures – a state of the art report. In *Proc. Workshop Algorithmic Found. Robot.*, pages 191–209, 1998.
- 15 N. Max, R. Crawfis, and D. Williams. Visualization for climate modeling. *IEEE Comput. Graphics and Appl.*, 13:34–40, 1993.
- 16 Daniel D. Sleator and Robert E. Tarjan. A data structure for dynamic trees. *J. Comput. Sys. Sci.*, 26(3):362–391, 1983.
- 17 B. S. Sohn and Bajaj C. L. Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):14–25, 2006.
- 18 A. Szymczak. Subdomain aware contour trees and contour evolution in time-dependent scalar fields. In *Proc. Conf. Shape Model. and Appl.*, pages 136–144, 2005.
- 19 S. P. Tarasov and M. N. Vyalii. Construction of contour trees in 3D in $O(n \log n)$ steps. In *Proc. 14th Annu. Sympos. Comput. Geom.*, pages 68–75, 1998.
- 20 M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *Proc. 13th Annu. Sympos. Comput. Geom.*, pages 212–220, 1997.

Hyperorthogonal Well-Folded Hilbert Curves

Arie Bos and Herman J. Haverkort

Department of Mathematics and Computer Science
Eindhoven University of Technology, The Netherlands
arie_bos@online.nl, cs.herman@haverkort.net

Abstract

R-trees can be used to store and query sets of point data in two or more dimensions. An easy way to construct and maintain R-trees for two-dimensional points, due to Kamel and Faloutsos, is to keep the points in the order in which they appear along the Hilbert curve. The R-tree will then store bounding boxes of points along contiguous sections of the curve, and the efficiency of the R-tree depends on the size of the bounding boxes—smaller is better. Since there are many different ways to generalize the Hilbert curve to higher dimensions, this raises the question which generalization results in the smallest bounding boxes. Familiar methods, such as the one by Butz, can result in curve sections whose bounding boxes are a factor $\Omega(2^{d/2})$ larger than the volume traversed by that section of the curve. Most of the volume bounded by such bounding boxes would not contain any data points. In this paper we present a new way of generalizing Hilbert's curve to higher dimensions, which results in much tighter bounding boxes: they have at most 4 times the volume of the part of the curve covered, independent of the number of dimensions. Moreover, we prove that a factor 4 is asymptotically optimal.

1998 ACM Subject Classification E.1 Data Structures, F.2.2 Geometrical problems and computations, H.3.1. Indexing methods, H.3.2 File organization

Keywords and phrases space-filling curve, Hilbert curve, multi-dimensional, range query, R-tree

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.812

1 Introduction

1.1 Space-filling curves and spatial index structures

A d -dimensional space-filling curve is a continuous, surjective mapping from \mathbb{R} to \mathbb{R}^d . In the late 19th century Peano [14] described such mappings for $d = 2$ and $d = 3$. Since then, various other space-filling curves have been found, and they have been applied in diverse areas such as spatial databases, load balancing in parallel computing, improving cache utilization in computations on large matrices, finite element methods, image compression, and combinatorial optimization [3, 7, 15]. In this paper we present new space-filling curves for $d > 2$ that have favourable properties for use in spatial data structures.

In particular, we consider data structures for d -dimensional points such as R-trees [12]. In such data structures, data points are organised in blocks, often stored in external memory. Each block contains at most B points, for some parameter B , and each point is stored in exactly one block. For each block we maintain a bounding box, which is the smallest axis-aligned d -dimensional box that contains all points stored in the block. The bounding boxes of the blocks are stored in an index structure, which may often be kept in main memory. To find all points intersecting a given query window Q , we can now query the index structure for all bounding boxes that intersect Q ; then we retrieve the corresponding blocks, and check the points in those blocks for answers to our query. We may also use the index structure to find the nearest neighbour to a query point q : if we search blocks in order of increasing



© Arie Bos and Herman J. Haverkort;

licensed under Creative Commons License CC-BY

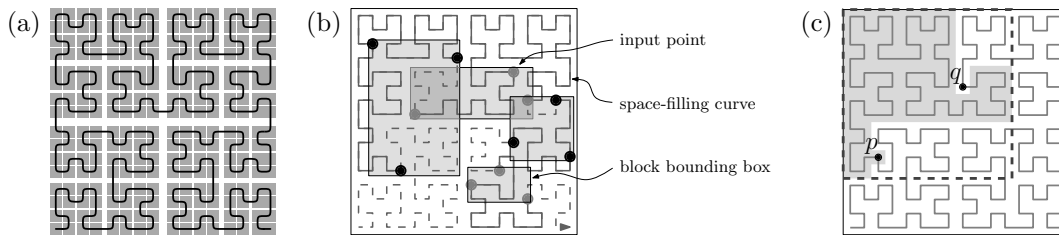
31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 812–826



Leibniz International Proceedings in Informatics

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



■ **Figure 1** (a) Sketch of Hilbert's space-filling curve. (b) Blocks of an R-tree or similar data structure with $B = 3$. (c) Box-to-curve ratio of the section between p and $q = \text{area of the bounding box of the curve section } S \text{ between } p \text{ and } q, \text{ divided by the area covered by } S: 12 \cdot 12/87 \approx 1.66$.

distance from q , we will retrieve exactly the blocks whose bounding boxes intersect the largest empty sphere around q . The grouping of points into blocks determines what block bounding boxes are stored in the index structure, and in practice, retrieving these blocks is what determines the query response time [7].

If we store n points in d dimensions with B points in a block, $\Theta((n/B)^{1-1/d})$ blocks may need to be visited in the worst case if the query window is a rectangular box with no points inside [11], and $\Theta(n/B)$ blocks may need to be visited if the query window is an empty sphere. The *Priority-R-tree* achieves these bounds [2], whereas a heuristic solution by Kamel and Faloutsos [10], which is explained below, may result in visiting $\Theta(n/B)$ blocks even if the query window is a rectangular box with no points inside [2]. However, experimental results for (near-)point data and query ranges with few points inside [8] indicate that the approach by Kamel and Faloutsos seems to be more effective in practice for such settings. Moreover, regardless of the type of data and query ranges, a structure based on the ideas of Kamel and Faloutsos is much easier to build and maintain than a *Priority-R-tree* [2].

Kamel and Faloutsos proposed to determine the grouping of points into blocks as follows: we order the input points along a space-filling curve and then put each next group of B points together in a block (see Figure 1(b)). Note that the number of blocks retrieved to answer a query is simply the number of bounding boxes intersected. Therefore it is important that the ordering induced by the space-filling curve makes us fill each block with points that lie close to each other and thus have a small bounding box.

Kamel and Faloutsos proposed to use the Hilbert curve [9] for this purpose. One way to describe the two-dimensional Hilbert curve is as a recursive construction that maps the unit interval $[0, 1]$ to the unit square $[0, 1]^2$. We subdivide the square into a grid of 2×2 square cells, and simultaneously subdivide the unit interval into four subintervals. Each subinterval is then matched to a cell; thus Hilbert's curve traverses the cells one by one in a particular order. The mapping from unit interval to unit square is refined by applying the procedure recursively to each subinterval-cell pair, so that within each cell, the curve makes a similar traversal. The traversals within these cells are rotated and/or reflected so that the traversal remains continuous from one cell to another (see Figure 1(a)). The result is a fully-specified mapping $f : [0, 1] \rightarrow [0, 1]^2$ from the unit interval to the unit square. The mapping is easily reversed, and thanks to the fact that the curve is based on recursive subdivision of a square in quadrants, the reversed mapping can be implemented very efficiently with coordinates represented as binary numbers. This gives us a way to decide which of any two points in the unit square is the first along the curve.

We can sketch the shape of the curve by drawing, for the n -th level of recursion, a polygonal curve, an *approximating curve* A_n , that connects the centres of the 4^n squares in the order in which they are visited. In fact, the mapping f can also be described as

the limit of the approximating curves A_n as n goes to infinity. Explicit descriptions of the approximating curves help us to reason about the shapes of curve sections, and thus, about the extents of their bounding boxes. For ease of notation, in this paper we scale the approximating curve for any level n by a factor 2^n and translate it so that its vertices are exactly the points $\{0, \dots, 2^n - 1\}^2$.

A d -dimensional version of Hilbert's curve could now be described by a series of curves A_n for increasing n , each visiting the points $\{0, \dots, 2^n - 1\}^d$. For $d \geq 3$, there are many ways to define such a series of curves [1, 5, 6], but their distinctive properties and their differences in suitability for our purposes are largely unexplored.

1.2 Our results

In this paper we present a family of space-filling curves, for any number of dimensions $d \geq 3$, with two properties which we call *well-foldedness* and *hyperorthogonality*—Hilbert's two-dimensional curve also has these properties. We show that these properties imply that the curves have good *bounding-box quality* as defined by Haverkort and Van Walderveen [7].

More precisely, for any $0 \leq a \leq b \leq 1$, let $f([a, b])$ denote the section of the space-filling curve f from $f(a)$ to $f(b)$, that is, $\bigcup_{a \leq t \leq b} f(t)$. The *box-to-curve ratio (BCR)* of a section $f([a, b])$ is the volume of the minimum axis-aligned bounding box of $f([a, b])$ divided by the volume (d -dimensional Lebesgue measure) of $f([a, b])$, see Figure 1(c). The worst-case BCR of a space-filling curve f is the maximum BCR over all sections of f . We show that the worst-case BCR of a well-folded, hyperorthogonal space-filling curve is at most 4, independent of the number of dimensions. Moreover, we show that this is asymptotically optimal: we prove that any d -dimensional space-filling curve that is described by a series of curves A_n as defined above, has a section with BCR at least $4 - O(1/2^d)$. In contrast, the d -dimensional "Hilbert" curves of Butz [4], as implemented by Moore [13], have sections with BCR in $\Omega(2^{d/2})$.

In Section 1.3 we introduce basic nomenclature and notation. Section 2 defines the concept of well-foldedness, and presents sufficient and necessary conditions for approximating curves of well-folded space-filling curves. Section 3 introduces the concept of hyperorthogonality. We present sufficient and necessary conditions for approximating curves of well-folded space-filling curves to be hyperorthogonal. The necessity of these conditions is then used to prove that any section of a hyperorthogonal well-folded space-filling curve has good box-to-curve ratio. Our next task is to show that hyperorthogonal well-folded curves actually exist, and this is the topic of Section 4. We combine the conditions from the previous sections to learn more about the shape of hyperorthogonal well-folded curves, and in particular about self-similar curves (Section 5). It turns out that in two, three, and four dimensions, there are actually very few self-similar, well-folded, hyperorthogonal curves; in five and more dimensions, more such curves exist. In Section 6, we make a few remarks about how to implement a comparison operator based on self-similar, well-folded, hyperorthogonal curves in any number of dimensions greater than two. Finally, in Section 7, we compare the bounding box quality of hyperorthogonal well-folded curves to lower bounds and to the bounding box quality of Butz's generalization of Hilbert curves, and we discuss directions for further research.

In this extended abstract we omit the proofs of most theorems, lemmas, and observations, as well as many details of the comparison operator discussed in Section 6. We intend to publish the proofs and further details (including pseudocode) of a non-recursive implementation of the comparison operator in a more comprehensive version of this paper. Until that is published, the interested reader is welcome to contact the authors for a version of this abstract that includes an appendix with the proofs and the pseudocode.

1.3 Nomenclature and notation

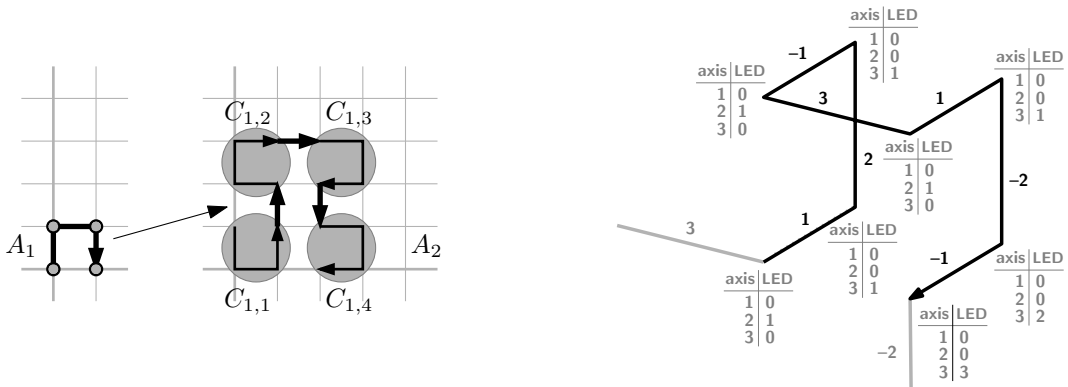
General notation. By D we denote 2^d . By $\text{sign}(i)$ we denote the sign of i , that is, $\text{sign}(i) = -1$ if $i < 0$; $\text{sign}(i) = 0$ if $i = 0$, and $\text{sign}(i) = 1$ if $i > 0$. By $\text{isneg}(i)$ we denote the function defined by $\text{isneg}(i) = 1$ if $i < 0$, and $\text{isneg}(i) = 0$ if $i \geq 0$.

Vertices, edges, directions and axes. The universe in this article is the integer grid in d dimensions \mathbb{Z}^d . A *vertex* is a point $v = (v[1], v[2], \dots, v[d]) \in \mathbb{Z}^d$. An *edge* $e = (v, w)$ is an ordered pair of vertices (v, w) with distance $\|w - v\| = 1$. The *direction* of an edge $e = (v, w)$ is the number $i \in \{-d, \dots, d\} \setminus \{0\}$ such that $w[|i|] - v[|i|] = \text{sign}(i)$ and $w[j] = v[j]$ if $j \neq |i|$. The *axis* of an edge is the absolute value of its direction. Note that the edges (v, w) and (w, v) have opposite directions, but the same axis. By $\langle e_1, e_2, \dots \rangle$ we denote a sequence of edges with directions e_1, e_2, \dots .

Curves, length, volume, entry and exit. For the purposes of this paper, a *curve on the grid* is an ordered set of unique vertices where each subsequent pair of vertices forms an edge as defined above. Note that a curve on the grid never visits the same vertex more than once. Henceforth, a *space-filling curve* is always a mapping $f : [0, 1] \rightarrow [0, 1]^d$, while any other curve discussed in this paper will be assumed to be a curve on the grid. Since a vertex and a direction determine an edge, a curve can alternatively be described by specifying the starting point and listing the directions of its edges in order. A *free curve* is a curve with a specified shape and orientation but with unspecified location; it is described by the directions of its edges. Note that curves are directed. The *reverse* \overleftarrow{C} of a free curve C is obtained by reversing the order of the edge directions *and* reversing the directions themselves, which means negating them. The *length* of a curve is the number of edges, the *volume* of a subset of the grid is the number of vertices. So the volume $\text{vol}(C)$ of a curve C is its length + 1. The first vertex of a curve is called the *entry*; the last vertex is called the *exit*.

k -Curves and k -cubes. A *k -curve* is a Hamiltonian path on the integer grid in a d -dimensional cube with $2^{d \cdot k}$ points, so with a side of length $2^k - 1$. Such a cube is called a *k -cube*. Since each of the (integer) points of the cube is visited by the curve exactly once, the length of a k -curve is $2^{d \cdot k} - 1$ and its volume is $2^{d \cdot k}$.

Approximating curves. The space-filling curves under study in this paper will be approximated by curves on the grid as just defined. By A_0, A_1, \dots we will denote a sequence of curves that approximates a d -dimensional space-filling curve, where A_0 is a single vertex and A_k is a k -curve. By $v_{k,1}, v_{k,2}, \dots, v_{k,K}$, where $K = 2^{d \cdot k}$, we denote the vertices of A_k in order, and by $e_{k,i}$ we denote the direction of the edge $(v_{k,i}, v_{k,i+1})$. Recall that each vertex $v_{k,i}$ of A_k represents a d -dimensional hypercube $H_{k,i}$ of width $1/2^k$ that is visited by the space-filling curve approximated by A_k , and the vertices $v_{k+1,D \cdot i - D + 1}, \dots, v_{k+1,D \cdot i}$ model the order in which the space-filling curve traverses the d -dimensional hypercubes of width $1/2^{k+1}$ whose union is $H_{k,i}$. Therefore it must be possible to construct A_{k+1} from A_k , which we call the *parent curve*, by *inflation*: we replace each vertex $v_{k,i}$ of the parent curve with a 1-curve $C_{k,i}$ (a *child curve*), whose vertices are those of the unit cube, translated by $2 \cdot v_{k,i}$. Each edge $(v_{k,i}, v_{k,i+1})$ of the parent curve is replaced by an edge $(v_{k+1,D \cdot i}, v_{k+1,D \cdot i+1})$ of the same direction, connecting the exit of $C_{k,i}$ to the entry of $C_{k,i+1}$, see Figure 2 (left). Note that not just any choice of child curves results in a valid $(k+1)$ -curve. The 1-curves that replace the vertices have to be chosen carefully such that for each edge $(v_{k,i}, v_{k,i+1})$ of the parent curve, there is indeed an edge in the grid from the exit of $C_{k,i}$ to the entry of



■ **Figure 2** Left: A parent curve A_1 is inflated to create A_2 , which is composed of the child curves $C_{1,1}$, $C_{1,2}$, $C_{1,3}$ and $C_{1,4}$, and edges of A_1 which are translated such that they connect the child curves to each other at their end points. Right: $G(3)$ (in black) with the directions of its edges; in grey: $G(3)$ extended with an entry edge $\langle d \rangle$ and an exit edge $\langle -(d-1) \rangle$, with the edge distance table according to Definition 11 for each vertex.

$C_{k,i+1}$. In Section 2 we will discuss how the 1-curves should be constructed so that they match up.

Observe that our definition of curves on the grid restricts the generalizations of Hilbert curves under study to *face-continuous curves*, that is, each pair of consecutive d -dimensional hypercubes along the curve must share a $(d-1)$ -dimensional face. In Section 7, we will discuss why, in the context of this paper, this restriction is justified.

2 Well-folded curves

In the process of inflating, we will restrict ourselves in this paper to replacing vertices with isometric images (translations, rotations and reflections) of one particular 1-curve, namely the free curve $G(d)$ that follows the so-called *binary reflected Gray code*:

► **Definition 1.** The free curve $G(d)$ is defined recursively as follows: $G(0)$ is empty; $G(d)$ is the concatenation of $G(d-1)$, $\langle d \rangle$, and $\overleftarrow{G}(d-1)$.

For example, $G(2)$ is the free curve $\langle 1, 2, -1 \rangle$, $G(3)$ is shown in Figure 2 (right), and $G(4)$ is the free curve $\langle 1, 2, -1, 3, 1, -2, -1, 4, 1, 2, -1, -3, 1, -2, -1 \rangle$. The length of $G(d)$ is, by induction, $2^d - 1$, which is the maximum length of a Hamiltonian path on the unit cube in \mathbb{Z}^d . Notice that in $G(d)$, each edge $\langle a \rangle$ is preceded by an edge $\langle 1 \rangle$ and followed by an edge $\langle -1 \rangle$ if $|a| = 2$, and it is preceded by an edge $\langle -1 \rangle$ and followed by an edge $\langle 1 \rangle$ if $|a| > 2$.

► **Definition 2.** A curve is *well-folded* if it is a single vertex, or if it is obtained by inflating a well-folded curve by replacing its vertices by isometric images of $G(d)$. A space-filling curve is well-folded if its approximating curves are well-folded.

Note that in two dimensions, all possible 1-curves are in fact isometric images of $G(2)$, so any face-continuous space-filling curve based on recursive subdivision of a square into four squares must be well-folded (for example, Hilbert’s curve or the $\beta\Omega$ -curve [17]). In higher dimensions, the most common generalizations of the Hilbert curve are well-folded as well, but there are also face-continuous curves based on recursive subdivision of a cube into eight cubes that are not well-folded (using generators of types B and C from Alber and

Niedermeier [1, 5]). In Section 7, we will briefly get back to non-well-folded curves; until then, we will focus on well-folded curves.

The following lemma will prove useful later:

► **Lemma 3.** *The axes of the first (and last) n edges of $G(d)$ constitute the set $\{1, \dots, m\}$, where $m = 1 + \lfloor \log_2(n) \rfloor = \lceil \log_2(n + 1) \rceil$.*

The isometric transformations of 1-curves which we need in this paper are those of the hyperoctahedral group of symmetries of the hypercube. This group is the product of the symmetric group S_d (the group of all permutations of the d coordinate axes) and the group of 2^d reflections formed by all combinations of reflections in hyperplanes orthogonal to the coordinate axes. Thus there are $d! \cdot 2^d$ such transformations.

To distinguish these transformations, we will use *signed permutations*. A signed permutation π working on $\{-d, \dots, d\} \setminus \{0\}$ is denoted by $[\pi[1], \pi[2], \dots, \pi[d]]$, where π is the bijection from $\{-d, \dots, d\} \setminus \{0\}$ to itself defined by $\pi(k) = \pi[k]$ and $\pi(-k) = -\pi[k]$ for $k \in \{1, \dots, d\}$. Given a k -cube H , a signed permutation π specifies the isometry that maps H onto itself and maps the direction k to the direction $\pi(k)$. If $\pi = [\pi[1], \pi[2], \dots, \pi[d]]$ is a signed permutation, then $\pi(\mathcal{X})$ denotes the application of π to all elements of the vector, set, or sequence \mathcal{X} ; $|\pi|$ denotes the permutation $[|\pi_1|, |\pi_2|, \dots, |\pi_d|]$; and π^{-1} denotes the inverse of π , that is, $\pi^{-1}(x) = y$ if and only if $\pi(y) = x$.

The *orientation* of a 1-curve C , denoted by $\text{or}(C)$, is the direction of the vector from entry to exit. Note that $\text{or}(G(d)) = d$, the direction of the middle edge of $G(d)$. Hence, $\text{or}(\pi(G(d))) = \pi(d)$.

Consider a sequence of well-folded approximating curves A_0, A_1, \dots . Given a particular level k , let K be 2^{d-k} , and let $\sigma_{k,i}$, for $i \in \{1, \dots, K\}$, be the transformation (modulo translation) that is applied to $G(d)$ to obtain the 1-curve $C_{k,i}$ that replaces $v_{k,i}$ in the inflation of A_k to A_{k+1} . For example, for the curves in Figure 2 (left) we have $\sigma_{0,1} = [1, 2]$; $\sigma_{1,1} = [-1, 2]$; $\sigma_{1,2} = [-2, 1]$; $\sigma_{1,3} = \sigma_{1,4} = [2, -1]$. As observed before, the 1-curves that replace the vertices have to be chosen carefully such that there is an edge with direction $e_{k,i}$ from the exit of $C_{k,i}$ to the entry of $C_{k,i+1}$. This leads to the following conditions:

► **Theorem 4.** *The permutations σ result in a sequence of well-folded approximating curves if and only if, for each k and for each $1 \leq i < 2^{d-k}$, we have:*

- for $j \in \{1, \dots, d\}$, we have $\text{sign}(\sigma_{k,i}^{-1}(j)) = \text{sign}(\sigma_{k,i+1}^{-1}(j))$ if and only if j equals neither or both of $|\sigma_{k,i}(d)|$ and $|e_{k,i}|$;
- $\text{sign}(\sigma_{k,i+1}^{-1}(e_{k,i})) = 1$.

Given the edges and the signs of the inverse permutations, Theorem 4 allows us to determine the last elements of each permutation. Conversely, given the edges and the last elements of each permutation, Theorem 4 allows us to determine the signs of each permutation. Note that this leaves $d - 1$ elements of each $|\sigma_{k,i}|$ unspecified and without consequence: any permutation of those elements will do.

► **Observation 5.** *Let f be a well-folded space-filling curve approximated by A_0, A_1, \dots , and let $x = f(0)$ be the starting point of f . Then $x[j] = \sum_{k=0}^{\infty} \text{isneg}(\sigma_{k,1}^{-1}(j)) / 2^{k+1}$. In other words, the digits of the binary representation of $x[j]$ behind the fractional point are $\text{isneg}(\sigma_{0,1}^{-1}(j)), \text{isneg}(\sigma_{1,1}^{-1}(j)), \text{isneg}(\sigma_{2,1}^{-1}(j)), \dots$*

3 Hyperorthogonal well-folded curves

So far, we have been defining and discussing properties of curves that are in fact common to the best-known previous generalizations of Hilbert’s curve to higher dimensions. We will

now introduce a new property that is *not* satisfied by these curves, and will prove useful in designing novel curves with good box-to-curve ratios.

3.1 Definition and characterization

► **Definition 6.** We call a curve *hyperorthogonal* if and only if, for any $n \in \{0, \dots, d - 2\}$, each sequence of 2^n consecutive edges have exactly $n + 1$ different axes. A space-filling curve is hyperorthogonal if its approximating curves are hyperorthogonal.

Notice that an n -dimensional 1-cube can hold at most $2^n - 1$ consecutive edges of a curve, so any curve constructed by inflation must contain sets of 2^n edges that have at least $n + 1$ different axes, for each $n \leq d - 1$. Hyperorthogonality requires that this holds for *every* set of 2^n edges, provided¹ $n \leq d - 2$. For $d = 2$, hyperorthogonality requires only that each single edge spans a one-dimensional space, which is obvious. So all two-dimensional curves are hyperorthogonal. For $d = 3$ each two consecutive edges must span a two-dimensional space, so each pair of consecutive edges must be orthogonal. (For that reason the property is called ‘hyperorthogonal’ for higher dimensions as well.) Note that $G(d)$ is hyperorthogonal. As can be seen by inspecting familiar generalizations of Hilbert curves to three dimensions, if we construct a sequence of curves A_0, \dots, A_k in three or more dimensions by inflation, using isometric images of $G(d)$ to inflate vertices, then A_k is not necessarily hyperorthogonal, even though $G(d)$ is (see, for example, the Butz-Moore curve in Section 5, Figure 3, right, where there are two collinear edges along the top right edge of the cube). The following theorem states what conditions the isometries should fulfill in order to obtain hyperorthogonal curves:

► **Definition 7.** The *depth* of a direction a in a signed permutation π , denoted $\text{depth}(\pi, a)$, is defined as follows: if $|a| \in \{|\pi_d|, |\pi_{d-1}|\}$, then $\text{depth}(\pi, a) = 0$, otherwise $\text{depth}(\pi, a)$ is the number j such that $|\pi_{d-1-j}| = |a|$.

► **Theorem 8.** Let K be 2^{d-k} , and let A_0, \dots, A_{k+1} be a sequence of well-folded curves constructed by inflation (with all the associated notation introduced in the previous sections). Suppose A_0, \dots, A_k are hyperorthogonal. Then A_{k+1} is hyperorthogonal as well if and only if the following conditions are satisfied:

1. for each $i \in \{1, \dots, K - 1\}$: $\text{depth}(\sigma_{k,i}, e_{k,i}) = \text{depth}(\sigma_{k,i+1}, e_{k,i}) = 0$;
2. for each $i \in \{1, \dots, K - 1\}$ and each $a \in \{-d, \dots, d\} \setminus \{0\}$, we have $|\text{depth}(\sigma_{k,i}, a) - \text{depth}(\sigma_{k,i+1}, a)| \leq 1$.

3.2 Box-to-curve ratio ≤ 4

To bound the box-to-curve ratio (BCR) of sections of hyperorthogonal well-folded space-filling curves, we will make use of the following lemma:

► **Lemma 9.** For any $n \in \{0, \dots, d - 2\}$, each sequence of 2^n consecutive edges of a well-folded, hyperorthogonal curve lies inside an $(n + 1)$ -dimensional unit cube.

► **Theorem 10.** The box-to-curve ratio of any section of a hyperorthogonal well-folded space-filling curve is at most 4.

¹ The definition leaves little room for being made more strict: raising the bound to $n \leq d - 1$ would render hyperorthogonal curves non-existent, at least when $d = 2$.

■ **Table 1** Cases distinguished in the proof of Theorem 10.

Case	<i>MaxBoxVol</i>	<i>MinCrvVol</i>
A: $2^{d-1} + 2 \leq \text{vol}(E_k) \leq 2^{d+1}$	2^{d+1}	2^{d-1}
B: $2^{d-2} + 2 \leq \text{vol}(E_k) \leq 2^{d-1} + 1$; $\text{vol}(Y) \leq \text{vol}(X) \leq 2^{d-2}$	2^d	2^{d-2}
C: $2^{d-2} + 2 \leq \text{vol}(E_k) \leq 2^{d-1} + 1$; $2^{d-3} < \text{vol}(Y) \leq 2^{d-2} < \text{vol}(X)$	$\frac{3}{2} \cdot 2^d$	$\frac{3}{2} \cdot 2^{d-2}$
D: $2^{d-2} + 2 \leq \text{vol}(E_k) \leq 2^{d-1} + 1$; $1 \leq \text{vol}(Y) \leq 2^{d-3}$	2^d	2^{d-2}
E: $2^{d-2} + 2 \leq \text{vol}(E_k) \leq 2^{d-1} + 1$; $\text{vol}(Y) = 0$	2^d	2^{d-2}
F: $3 \leq \text{vol}(E_k) \leq 2^{d-2} + 1$	$4(\text{vol}(E_k) - 2)$	$\text{vol}(E_k) - 2$
G: $\text{vol}(E_k) \leq 2$	2	1

Proof. Consider a section s of a hyperorthogonal well-folded space-filling curve f , approximated by a series of curves A_0, A_1, \dots . Let E_k be the subcurve of A_k that contains all vertices $v_{k,i}$ that represent hypercubes $H_{k,i}$ of width $1/2^k$ that are intersected by s . More specifically, let k be the smallest k such that A_k contains at least one vertex $v_{k,i}$ such that $H_{k,i}$ is fully covered by s . The bounding box of a subcurve $v_{k,h}, \dots, v_{k,j}$ of A_k is the smallest axis-aligned box that fully contains all hypercubes $H_{k,h}, \dots, H_{k,j}$.

By our choice of k , E_k contains vertices from at most two (consecutive) child curves $C_{k-1,x}$ and $C_{k-1,y}$ of A_{k-1} , because otherwise one child curve of A_{k-1} would be completely covered by s , contradicting our choice of k . Note that this implies that the bounding box of E_k has at most the volume of two 1-cubes, that is, 2^{d+1} . Without loss of generality, let $C_{k-1,x}$ be the child curve of A_{k-1} that contains the largest part of E_k and call this part X , let Y be the remaining part of E_k (if any), and let $c = |e_{k,\min(x,y)}|$ be the axis of the connecting edge of X and Y . By definition, $\text{vol}(Y) = \text{vol}(E_k) - \text{vol}(X) \leq \text{vol}(X)$.

A number of cases with smartly chosen boundaries for $\text{vol}(E_k)$, $\text{vol}(X)$ and $\text{vol}(Y)$ can now be distinguished, as shown in Table 1. In each case, we derive an upper bound *MaxBoxVol* on the bounding box volume, and a lower bound *MinCrvVol* on the number of vertices of E_k that represent hypercubes completely covered by s (this is usually all of E_k except for the first and last vertex). From this we can derive that the box-to-curve ratio is less than $\text{MaxBoxVol}/\text{MinCrvVol} \leq 4$. Note that cases B, C, D, and E are subcases for the same bounds on $\text{vol}(E_k)$, where case B is the case of having small X , and cases C, D, E are the cases of large X with various bounds on the size of Y . For cases A, E, and G the bounds on the bounding box volume are trivial; cases B, C, D, and F require a more careful analysis.

Case B: By Theorem 8, for the axis c of the connecting edge between X and Y we have $\text{depth}(\sigma_{k-1,x}, c) = 0$. Since $\text{vol}(X) \leq 2^{d-2}$, Lemma 3 now tells us that the edges of X have axes from $|\sigma_{k-1,x}|(\{1, \dots, d-2\})$, hence not including c . Therefore X is included in the half of $C_{k-1,x}$ that lies closest to $C_{k-1,y}$. Likewise, Y is included in the half of $C_{k-1,y}$ that lies closest to $C_{k-1,x}$. These two halves together constitute a unit cube of volume 2^d .

Case C: As in case B, Y is included in the half of $C_{k-1,y}$ that lies closest to $C_{k-1,x}$. This half, together with $C_{k-1,x}$, has a bounding box of volume $\frac{3}{2} \cdot 2^d$. The minimum curve volume *MinCrvVol* is at least $\text{vol}(E_k) - 2 = \text{vol}(X) + \text{vol}(Y) - 2 \geq 2^{d-2} + 2^{d-3} = \frac{3}{2} \cdot 2^{d-2}$.
Case D: Given the bounds on $\text{vol}(X)$ and $\text{vol}(Y)$, Lemma 3 tells us that the edges of X have axes from $|\sigma_{k-1,x}|(\{1, \dots, d-1\})$, and the edges of Y have axes from $|\sigma_{k-1,y}|(\{1, \dots, d-3\})$. Now let a be $|\sigma_{k-1,x}(d)|$. By Theorem 8, $\text{depth}(\sigma_{k-1,y}, a) \leq \text{depth}(\sigma_{k-1,x}, a) + 1 = 1$ and therefore a is not included in $|\sigma_{k-1,y}|(\{1, \dots, d-3\})$. If $a = c$, it follows that X

and Y lie in half-cubes that together constitute a unit cube of volume 2^d , as in case B. Otherwise, if $a \neq c$, it follows that E_k may contain multiple edges of direction c but does not include any edge with direction a . Therefore E_k lies completely in a box that spans two 1-cubes in dimension c , half a 1-cube in dimension a , and one 1-cube in the remaining dimensions. The volume of this box is 2^d .

Case F: By Lemma 9, each set of $\text{vol}(E_k) - 1$ edges of A_k is contained in a unit cube of $\lceil \log(\text{vol}(E_k) - 1) \rceil + 1 = \lfloor \log(\text{vol}(E_k) - 2) \rfloor + 2$ dimensions, of volume at most $4(\text{vol}(E_k) - 2)$. ◀

4 General construction method

In Section 2, Theorem 4, we learned about sufficient and necessary conditions for well-folded curves in general, and in Section 3, Theorem 8, we learned about specific conditions for hyperorthogonal well-folded curves. It remains to show that curves satisfying both the general and the specific conditions actually exist. In this section we will combine the conditions of Theorems 4 and 8 to derive conditions on the entry and exit points and isometries used in the construction of hyperorthogonal well-folded curves. We will show how to construct curves that satisfy all conditions, for any $d \geq 3$ (recall that for $d = 2$, we have Hilbert’s curve).

► **Definition 11.** The *edge distance* of the axis a to the vertex v within the curve C , denoted $\text{dist}(C, v, a)$, is the distance along C between v and the closest edge with axis a ; more precisely, $\text{dist}(C, v, a)$ is one less than the length of the smallest subcurve of C that includes v and an edge with axis a . (For a small example, see Figure 2, right.)

Theorem 8 has a remarkable consequence:

► **Lemma 12.** *In well-folded hyperorthogonal curves, $\text{depth}(\sigma_{k,i}, a) \leq \text{dist}(A_k, v_{k,i}, a)$.*

Lemma 12 gives us the following idea for an algorithm to specify the permutations $|\sigma_{k,i}|$, except for the order of the last two elements: simply sort all axes a by order of decreasing edge distance $\text{dist}(A_k, v_{k,i}, a)$. In fact, as we will show now, a version of this algorithm that only considers edge distances within small subcurves suffices. We choose an *entry direction* and an *exit direction* and denote these by $e_{k,0} = e_{0,0}$ and $e_{k,K} = e_{0,1}$, respectively, for any k and $K = 2^{d-k}$.

► **Definition 13.** Define the *extended child curve* $C'_{k-1,j}$ as the concatenation of an edge $\langle e_{k-1,j-1} \rangle$, the curve $C_{k-1,j}$, and an edge $\langle e_{k-1,j} \rangle$. We define the *local edge distance* $\text{ldist}_{k,i}(a)$ as $\text{dist}(C'_{k-1,j}, v_{k,i}, a)$, where $j = \lceil i/D \rceil$ and $C_{k-1,j}$ is the child curve that contains $v_{k,i}$.

► **Lemma 14.** *Suppose that, for $k \in \{1, 2, \dots\}$, we construct the permutations $\sigma_{k,i}$ of a well-folded space-filling curve such that the elements of $|\sigma_{k,i}|$ are sorted by order of decreasing local edge distance $\text{ldist}_{k,i}$. Then each curve A_k satisfies the conditions of Theorem 8.*

The above lemma still leaves the order of the last two elements of each $|\sigma_{k,i}|$ undetermined, since these are always the two axes with edge distance zero. To prove that hyperorthogonal well-folded curves exist, it now suffices to show that we can order the last two elements and choose the signs of each $\sigma_{k,i}$ such that the conditions of Theorem 4 are satisfied. We obtain:

► **Theorem 15.** *For each choice of $e_{0,0}$ and $e_{0,1}$ and for each choice for the signs of $\sigma_{k,1}^{-1}(j)$ for all k and j , satisfying $\text{sign}(\sigma_{k,1}^{-1}(e_{k,0})) = 1$ for all k , there is a unique hyperorthogonal, well-folded space-filling curve f approximated by A_0, A_1, \dots in which the elements of each permutation $|\sigma_{k,i}|$ are sorted by order of decreasing local edge distance $\text{ldist}_{k,i}$.*

Proof. For each level k , we generate A_k as follows. We loop over all $i \in \{1, \dots, K-1\}$, where $K = 2^{d \cdot k}$, and proceed as follows. The conditions of Theorem 4 require $\text{sign}(\sigma_{k,i+1}^{-1}(e_{k,i})) = 1$. We now choose $|\sigma_{k,i}(d)|$ such that $|\sigma_{k,i}(d)| = |e_{k,i}|$ if and only if $\text{sign}(\sigma_{k,i}^{-1}(e_{k,i})) = 1$: this is always possible since $|e_{k,i}|$ is among the last two elements of $|\sigma_{k,i}|$ whose order was undetermined. Thus we satisfy the first condition of Theorem 4 for $j = |e_{k,i}|$. With $|\sigma_{k,i}|$ completely determined, we can now fill in the remaining signs of $\sigma_{k,i+1}$ such that they fulfill the first condition of Theorem 4. Finally, we determine $|\sigma_{k,K}(d)|$ as dictated by the exit direction $e_{k,K}$ in the same way as we determined $|\sigma_{k,i}(d)|$ for $i < K$. ◀

5 Self-similar curves

By Observation 5, a choice of signs of $\sigma_{k,1}^{-1}(j)$ for all k and j specifies the starting point $f(0)$ of the space-filling curve f in Theorem 15. Thus, the proof of Theorem 15 is a constructive proof that a hyperorthogonal, well-folded space-filling curve exists for any choice of starting point on the boundary of the unit hypercube.

In a practical setting, such as described in Section 1.1, one may want to sort points in the order in which they appear along the curve. To this end we need a comparison operator that decides which of any two given points p and q comes first along the curve. We can do so by determining the largest k such that there is a hypercube $H_{k,i}$, corresponding to a vertex $v_{k,i}$, which contains both points. Then we can use $\sigma_{k,i}$ to determine in which order the 2^d subcubes of this hypercube are traversed, and in particular, in which order this traversal visits the two subcubes containing p and q . The efficiency of the comparison operator now depends on how efficiently we can determine $\sigma_{k,i}$ for any k and i . Unfortunately, straightforward application of Theorem 15 would require us to traverse all of A_k from $v_{k,1}$ to $v_{k,i}$ to determine $\sigma_{k,i}$.

To enable us to determine $\sigma_{k,i}$ more efficiently, we will, in this section, restrict the curves to be *self-similar*, that is, A_{k+1} is the concatenation of 2^d isometric and/or reversed copies of A_k . We will analyse how the choice of the entry of $C_{1,1}$ propagates to the other child curves of A_1 , and derive conditions that starting points of self-similar, hyperorthogonal, well-folded space-filling curves should fulfill. It turns out that for any $d \geq 3$, only two different starting points (modulo rotation and reflection) exist for such curves.

For the purposes of this section, the following notation will be helpful.

▶ **Definition 16.** The relative coordinate vector of a vertex v is the vector r such that $r[j] = 0$ if $x[j] \bmod 4 \in \{0, 3\}$, and $r[j] = 1$ if $x[j] \bmod 4 \in \{1, 2\}$.

Note that the relative coordinates of a vertex $v_{k+1,i}$ tell us, for each dimension, whether the vertex is on the outside (0) or on the inside (1) with respect to the 2-cube of A_{k+1} corresponding to the vertex $v_{k-1,j}$ of A_{k-1} , where $j = \lfloor i/D^2 \rfloor$.

Let $\text{ent}_{k,i}, \text{ext}_{k,i} : \{1, \dots, d\} \rightarrow \{0, \dots, 2^{k+1} - 1\}$ be functions that give the coordinates of the entry and exit point of $C_{k,i}$, that is, the entry point of $C_{k,i}$ has coordinates $(\text{ent}_{k,i}(1), \dots, \text{ent}_{k,i}(d))$ and the exit point has coordinates $(\text{ext}_{k,i}(1), \dots, \text{ext}_{k,i}(d))$. Note that $\text{ent}_{k,i}(j) = \text{isneg}(\sigma_{k,i}^{-1}(j)) \pmod 2$, and $\text{ext}_{k,i}(j) = \text{ent}_{k,i}(j) \pmod 2$ if and only if $|\sigma_{k,i}^{-1}(j)| \neq d$. Similarly, let $\text{rlent}_{k,i}, \text{rlext}_{k,i} : \{1, \dots, d\} \rightarrow \{0, 1\}$ be functions that give us the *relative* coordinates of the entry and exit point of $C_{k,i}$. Note that we have $\text{rlent}_{k,i}(j) = (\text{ent}_{k,i}(j) + v_{k,i}[j]) \pmod 2$, and $\text{rlext}_{k,i}(j) = (\text{ext}_{k,i}(j) + v_{k,i}) \pmod 2$. Observe that if $\text{rlent}_{k,i}$ and $v_{k,i}$ are given, this determines $\text{ent}_{k,i}$ and hence, the signs of $\sigma_{k,i}^{-1}$.

▶ **Lemma 17.** If A_0, A_1, \dots approximate a self-similar, well-folded, hyperorthogonal space-filling curve f , then each extended child curve $C_{k,i}$, according to Definition 13, is an isometry of either:

- the concatenation of $\langle d \rangle$, $G(d)$, and $\langle -(d-1) \rangle$ (henceforth called type 0);
- the concatenation of $\langle d-1 \rangle$, $G(d)$, and $\langle d \rangle$ (henceforth called type 1).

We will denote the type of the child curve $C_{k,i}$ by $T_{k,i}$. A direct consequence of Lemma 17 is that we may assume, without loss of generality (modulo reflection, rotation and reversal), that $C_{0,1} = A_1 = G(d)$ with entry direction d and exit direction $-(d-1)$, with $T_{0,1} = 0$. Moreover, we should have $v_{2,1}[d] = 0$ and $v_{2,K}[d-1] = 0$, where $K = D^2 = (2^d)^2$, so that the child curves $C_{1,1}$ and $C_{1,D}$ can be extended with, respectively, the same entry edge $\langle d \rangle$ and the same exit edge $\langle -(d-1) \rangle$ as $C_{0,1}$. By tracing the relative coordinates of the entry and exit points through the child curves of A_1 , using the conditions of Theorems 4 and 8, we now find the following:

► **Lemma 18.** $\text{rlent}_{1,D} = \text{rlent}_{1,1} \circ \omega$, where $\omega = [d-1, 2, \dots, d-2, d, 1]$.

When we inflate A_2 to obtain A_3 , so that a 2-curve replaces each vertex of A_1 , the relative coordinates of each 2-curve’s exit point should equal the relative coordinates of the next 2-curve’s entry point. Because of self-similarity, the 2-curve replacing $v_{1,i}$ must itself be an isometry of either A_2 (if $T_{1,i} = 0$) or $\overleftarrow{A_2}$ (if $T_{1,i} = 1$). As a result of the transformation $\sigma_{1,i-1}$, the relative coordinates of the exit point of the 2-curve replacing $v_{1,i-1}$ are given by the function $\text{rlent}_{1,1} \circ \omega \circ |\sigma_{1,i-1}^{-1}|$ if $T_{1,i-1} = 0$, and by $\text{rlent}_{1,1} \circ |\sigma_{1,i-1}^{-1}|$ if $T_{1,i-1} = 1$. The relative coordinates of the entry point of the 2-curve replacing $v_{1,i}$ are given by the function $\text{rlent}_{1,1} \circ |\sigma_{1,i}^{-1}|$ if $T_{1,i} = 0$, and by $\text{rlent}_{1,1} \circ \omega \circ |\sigma_{1,i}^{-1}|$ if $T_{1,i} = 1$. Thus we get:

► **Lemma 19.**

If $T_{1,i-1} = 0$ and $T_{1,i} = 0$, we have $\text{rlent}_{1,1} \circ \omega \circ |\sigma_{1,i-1}^{-1}| = \text{rlent}_{1,1} \circ |\sigma_{1,i}^{-1}|$.

If $T_{1,i-1} = 0$ and $T_{1,i} = 1$, we have $\text{rlent}_{1,1} \circ \omega \circ |\sigma_{1,i-1}^{-1}| = \text{rlent}_{1,1} \circ \omega \circ |\sigma_{1,i}^{-1}|$.

If $T_{1,i-1} = 1$ and $T_{1,i} = 0$, we have $\text{rlent}_{1,1} \circ |\sigma_{1,i-1}^{-1}| = \text{rlent}_{1,1} \circ |\sigma_{1,i}^{-1}|$.

If $T_{1,i-1} = 1$ and $T_{1,i} = 1$, we have $\text{rlent}_{1,1} \circ |\sigma_{1,i-1}^{-1}| = \text{rlent}_{1,1} \circ \omega \circ |\sigma_{1,i}^{-1}|$.

We can now analyse the possible successions of types $T_{1,i}$ and permutations $\sigma_{1,i}$ for $i \in \{1, \dots, 2^d\}$ and prove that Lemma 19 can only be true if:

► **Lemma 20.** $\text{rlent}_{1,1}(j) = \text{rlent}_{1,1}(j-1)$ for all $j \in \{2, \dots, d-1\}$.

By exploiting self-similarity recursively, we now find:

► **Lemma 21.** $\text{rlent}_{k,1} = \text{rlent}_{1,1}$ for all $k \geq 1$.

This leads almost directly to:

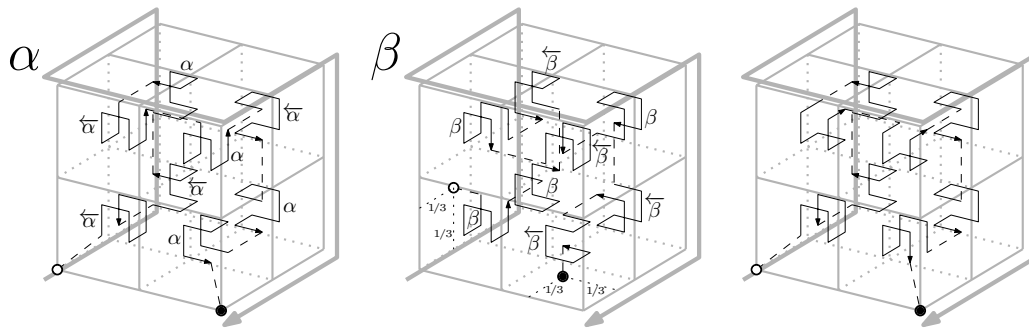
► **Theorem 22.** If f is a hyperorthogonal well-folded space-filling curve mapping $[0, 1]$ to $[0, 1]^d$, then, modulo reflection, reversal and rotation, $f(0)$ is either $(0, \dots, 0, 0)$ or $(\frac{1}{3}, \dots, \frac{1}{3}, 0)$.

In fact, such curves exist for any $d \geq 3$:

► **Theorem 23.** For any $d \geq 3$, there are self-similar, hyperorthogonal, well-folded d -dimensional space-filling curves starting at $(0, \dots, 0, 0)$ and $(\frac{1}{3}, \dots, \frac{1}{3}, 0)$.

It turns out that there are actually very few such curves for $d = 3$ and $d = 4$:

► **Observation 24.** If $d = 3$ or $d = 4$, Lemma 12 leaves no choice with respect to the last two elements, the third-last element, and the first element of the permutations $|\sigma_{k,i}|$ in a self-similar curve.



■ **Figure 3** The three-dimensional, self-similar, hyperorthogonal, well-folded space-filling curve with starting points $(0, 0, 0)$ (left) and $(\frac{1}{3}, \frac{1}{3}, 0)$ (centre), and the three-dimensional curve by Butz and Moore (right). The bold grey curve shows A_1 . The solid black curves depict the child curves of A_1 , the dashed lines between them indicate how they are connected. The symbols next to the child curves indicate whether they are type 0 (without arrow), or its reverse, type 1 (with arrow). For the Butz-Moore curve, no such indications are given, because the curve is symmetric and there is no need to distinguish between reflections and reversals. The white and black dots on the outer cube indicate the location of $f(0)$ and $f(1)$.

► **Corollary 25.** *If $d = 3$ or $d = 4$, there are exactly two self-similar, hyperorthogonal, well-folded d -dimensional space-filling curves.*

Proof. For self-similar curves, we may assume the entry and exit direction to be fixed at $\langle d \rangle$ and $\langle -(d - 1) \rangle$, respectively. For the starting point, that is, the signs of $\sigma_{k,1}^{-1}(j)$ for all k and j , only two combinations are possible (Theorem 22). Theorem 15 states that this leads to two unique hyperorthogonal, well-folded space-filling curves in which the elements of each $|\sigma_{k,i}|$ are sorted by order of decreasing local edge distance $\text{ldist}_{k,i}$. By Observation 24, for $d = 3$ and $d = 4$, there is no other way to order the elements of each $|\sigma_{k,i}|$. ◀

The two three-dimensional self-similar, hyperorthogonal, well-folded space-filling curves are illustrated in Figure 3, left and centre.

6 Implementation in software

It is relatively easy to implement an efficient comparison operator that decides which of any two given points comes first along a d -dimensional self-similar hyperorthogonal well-folded space-filling curve. For a fixed choice of space-filling curve f , a recursive implementation would take as input two points $p, q \in [0, 1]^d$ that need to be compared, along with a signed permutation σ that specifies how the given curve is placed in the unit cube, and the direction of the curve (forward or reversed). Let $S(p)$ and $S(q)$ be the subcubes of width $1/2$ that contain p and q , respectively.

If $p = q$, one point does not precede the other. Otherwise, if $S(p) \neq S(q)$, one can decide immediately which point comes first, based on the relative order of the vertices that represent $S(p)$ and $S(q)$ along the approximating 1-curve $\sigma(G(d))$. Finally, if $S(p) = S(q)$, that is, p and q lie in the same subcube of width $1/2$, then their relative order can be decided by a recursive call with:

- the points p and q , scaled and translated according to the transformation that maps $S(p)$ to the unit cube;
- the signed permutation and direction that specifies how the space-filling curve traverses $S(p)$.

In fact, thanks to the structure of the approximating curve $\sigma(G(d))$, one can examine the coordinates of p and q one by one, from the coordinate in dimension $|\sigma|(d)$ to the coordinate in dimension $|\sigma|(1)$: as soon as a coordinate is found in which the binary representations of the fractional parts of p and q differ in the first bit, one can decide which of the two points precedes the other. Only if p and q are equal in the first bits of all coordinates, the algorithm needs to go in recursion.

To be able to make the recursive call, the algorithm needs to determine the permutation to use in recursion, that is, the transformation that maps the complete space-filling curve f to the section within $S(p)$, modulo scaling and translation. For the curves described by the constructions of Lemma 14 and Theorem 23 this is relatively straightforward. To determine the unsigned permutation to be used in recursion, we sort the d coordinate axes by decreasing local edge distance from $S(p)$. This sorted list of axes can be constructed on the fly in $\Theta(d)$ time while examining the d coordinates of p and q to decide in which subcube they lie. By Lemma 14, the sorted list of axes gives us the (unsigned) permutation to use in recursion. The signs of the permutation to use in recursion now follow from applying the observations on relative entry points and permutation signs calculated in the previous section.

If the binary representations of the coordinates of p and q consist of k bits per coordinate, then the complete comparison operator runs in $O(d \cdot k)$ time.

7 Evaluation

7.1 Comparing to the Butz-Moore curves

The generalization of Hilbert's curve to d dimensions by Butz [4], as implemented by Moore [13], is a self-similar well-folded curve with starting point in the origin, in which the orientations (and therefore, the signs of the inverse permutations) of the child curves of A_1 are the same as in our hyperorthogonal well-folded curve. Concretely, $|\sigma_{1,i}[d]| = 1$ for $i \in \{1, 2^d\}$, and $|\sigma_{1,i}[d]| = \max(|e_{1,i-1}|, |e_{1,i}|)$ for $1 < i < 2^d$. However, otherwise the permutations are different: all permutations in the Butz-Moore curves are rotations (in the permutation sense of the word), so $|\sigma_{1,i}[j]| = |\sigma_{1,i}[d]| + j \pmod{d}$. For a graphical description of the 3-dimensional curve, see Figure 3 (right).

Assuming $d \geq 3$, the curve $G(d)$ contains a sequence $\langle 1, 2, -1, (2 + \lfloor d/2 \rfloor), 1 \rangle$ or $\langle 1, -2, -1, (2 + \lfloor d/2 \rfloor), 1 \rangle$, so there is an i such that $|\sigma_{1,i}(d)| = 2$, $|e_{1,i}| = 1$, and $|\sigma_{1,i+1}(d)| = 2 + \lfloor d/2 \rfloor$. We can now calculate that the curve through the last $2^{\lfloor d/2 \rfloor - 1} + 1$ vertices of $C_{1,i}$ and the first $2^{\lfloor d/2 \rfloor - 1} + 1$ vertices of $C_{1,i+1}$ has box-to-curve ratio at least $2^{d-1}/(2^{d/2} + 2)$, and thus:

► **Theorem 26.** *The Butz-Moore curve contains sections with box-to-curve ratio $\Omega(2^{d/2})$.*

The worst-case box-to-curve ratio of the Butz-Moore curves is thus in sharp contrast with the worst-case box-to-curve ratio of our hyperorthogonal, well-folded curves, which have BCR at most 4 for any d . For verification we also calculated the actual worst-case BCR values for $d \in \{2, 3, 4, 5, 6\}$ with the software from Sasburg [16] (Table 2). Further investigations may be done into average BCR values over curve sections of a given size, both for the hyperorthogonal and the Butz curves.

It should be noted, however, that BCR may not be the only relevant measure of bounding-box quality. Haverkort and Van Walderveen [7] argued that, at least for $d = 2$, the size of the *boundary* of a bounding box may be as important as its volume—although volume and boundary size are usually correlated. Using Sasburg's software with a generalization of the worst-case bounding box perimeter ratio from Haverkort and Van Walderveen to higher

■ **Table 2** Worst-case box-to-curve ratios for various curves in up to 6 dimensions.

curve	$d = 2$	$= 3$	$= 4$	$= 5$	$= 6$	≥ 7
<i>lower bound face-continuous</i>	2.00	2.54	3.15	3.54	3.76	$4 - 16/(2^d + 3)$
best claimed non-self-sim.	2.22 ^a	2.89 ^b				
self-sim. hyperorth. well-fdd. $f(0) = (0, \dots, 0, 0)$	2.40 ^c	3.11	3.53	3.76	3.88	≤ 4
self-sim. hyperorth. well-fdd. $f(0) = (\frac{1}{3}, \dots, \frac{1}{3}, 0)$		3.14	3.67	3.83	3.92	≤ 4
<i>lower bound non-face-continuous</i>	3.00	3.50	3.75	3.87	3.93	$4 - 4/2^d$
Butz-Moore	2.40 ^c	3.11	4.74	7.08	10.65	$\Omega(2^{d/2})$

^a $\beta\Omega$ -curve [17] analysed by H&vW [7]; ^b Iupiter [5]; ^c Hilbert’s curve [9]

dimensions, we found that already for $d = 3$, the self-similar hyperorthogonal well-folded curve with starting point $(\frac{1}{3}, \frac{1}{3}, 0)$ outperforms the Butz curve.

7.2 What about other curves?

In this work we study space-filling curves that can be described by a series of approximating curves A_0, A_1, \dots, A_n , where A_k is a curve on the k -cube. Within this context, we restricted our search for curves with good worst-case BCR first to face-continuous curves; then, more specifically, to well-folded curves; then to hyperorthogonal well-folded curves; and finally to self-similar, hyperorthogonal, well-folded curves. We found that if $d = 3$ or $d = 4$, there are only two self-similar hyperorthogonal well-folded space-filling curves. For $d = 5$ and up, there are many more, as Lemma 12 then starts to leave room for swaps among the first elements of the permutations $\sigma_{k,i}$. We will now address the question of how much room for further improvement there is within these restrictions or if some of these restrictions are dropped.

For $d = 2$, Haverkort and Van Walderveen [7] report that the BCR of any section of the well-folded, non-self-similar $\beta\Omega$ -curve [17] is 2.22 in the worst case, and for $d = 3$, Haverkort [5] claims a fairly complicated, non-self-similar, face-continuous curve with a worst-case BCR of 2.89. These constructions, which do not easily generalize to higher dimensions, constitute improvements of less than 10% with respect to the self-similar hyperorthogonal well-folded curves. For larger values of d , no face-continuous curve can be much better than any hyperorthogonal well-folded curve, since the first is subject to a lower bound that quickly approaches the upper bound of the latter as d grows:

► **Theorem 27.** *If f is a space-filling curve approximated by a series of curves A_0, \dots, A_k within the framework of Section 1.3, then f has a section with BCR at least $4 - 16/(2^d + 3)$.*

The proof is based on the fact that any such curve must contain a sequence of at most $2^{d-2} + 1$ edges that have all axes $\{1, \dots, d\}$. For the specific case of $d = 2$, Haverkort and Van Walderveen [7] prove a stronger lower bound of 2.

Now suppose we drop the restriction to face-continuous curves. More precisely, suppose we have a space-filling curve approximated by a sequence of curves on the grid A_0, A_1, \dots , where we allow our curves on the grid to have diagonal edges, that is, we allow any edge (v, w) such that $w \neq v$ and $|w[j] - v[j]| \leq 1$ for all $j \in \{1, \dots, d\}$. In that case, the lower bound becomes even worse:

► **Theorem 28.** *If there is a k and i such that $v_{k,i}$ and $v_{k,i+1}$ differ in at least two coordinates (in other words: if there is a diagonal edge), then f has a section with BCR at least $4 - 4/2^d$.*

Note that, as Table 2 shows, at least for d up to 6 the lower bound of Theorem 28 for curves with “diagonal edges” is greater than the worst-case BCR of the best hyperorthogonal,

well-folded curves, and for higher dimensions the difference between the lower bound and the upper bound is less than 1%. Therefore, in terms of worst-case BCR, little is to be expected from non-face-continuous curves based on inflation of k -cubes for increasing k .

The question remains whether there are hyperorthogonal curves that are not well-folded, and if so, whether such curves would also have good bounds on the box-to-curve ratio. In other words: is well-foldedness really required in Theorem 10? However, Theorem 27 shows that in any case, there is not much room for finding curves with a better worst-case BCR.

References

- 1 J. Alber and R. Niedermeier. On multidimensional curves with Hilbert property. *Theory of Computing Systems*, 33(4):295–312, 2000.
- 2 L. Arge, M. de Berg, H. Haverkort, and K. Yi. The Priority R-tree: a practically efficient and worst-case optimal R-tree. *ACM Tr. Algorithms*, 4(1):9, 2008.
- 3 M. Bader. *Space-filling curves: an introduction with applications in scientific computing*. Springer, 2013.
- 4 A. R. Butz. Alternative algorithm for Hilbert’s space-filling curve. *IEEE Trans. Comp.*, 20(4):424–426, 1971.
- 5 H. Haverkort. An inventory of three-dimensional Hilbert space-filling curves. *CoRR*, abs/1109.2323, 2011.
- 6 H. Haverkort. Harmonious Hilbert curves and other extradimensional space-filling curves. *CoRR*, abs/1211.0175, 2012.
- 7 H. Haverkort and F. van Walderveen. Locality and bounding-box quality of two-dimensional space-filling curves. *Computational Geometry*, 43(2):131–147, 2010.
- 8 H. Haverkort and F. van Walderveen. Four-dimensional Hilbert curves for R-trees. *ACM J. Experimental Algorithmics*, 16:3.4, 2011.
- 9 D. Hilbert. Über die stetige Abbildung einer Linie auf ein Flächenstück. *Math. Ann.*, 38(3):459–460, 1891.
- 10 I. Kamel and C. Faloutsos. On packing R-trees. In *Conf. on Information and Knowledge Management*, pages 490–499, 1993.
- 11 K. V. R. Kanth and A. K. Singh. Optimal dynamic range searching in non-replicating index structures. In *Int. Conf. Database Theory, LNCS 154*, pages 257–276, 1999.
- 12 Y. Manolopoulos, A. Nanopoulos, A. N. Papadopoulos, and Y. Theodoridis. *R-trees: Theory and Applications*. Springer, 2005.
- 13 D. Moore. Fast Hilbert curve generation, sorting, and range queries. <http://web.archive.org/web/www.caam.rice.edu/~dougmtwiddle/Hilbert/>, 2000, retrieved 23 March 2015.
- 14 G. Peano. Sur une courbe, qui remplit toute une aire plane. *Math. Ann.*, 36(1):157–160, 1890.
- 15 H. Sagan. *Space-Filling Curves*. Universitext. Springer, 1994.
- 16 S. Sasburg. Approximating average and worst-case quality measure values for d -dimensional space-filling curves. Master’s thesis, Eindhoven University of Technology, 2011.
- 17 J.-M. Wierum. Definition of a new circular space-filling curve: $\beta\Omega$ -indexing. Technical Report TR-001-02, Paderborn Center for Parallel Computing PC², 2002.

Topological Analysis of Scalar Fields with Outliers*

Mickaël Buchet¹, Frédéric Chazal¹, Tamal K. Dey², Fengtao Fan²,
Steve Y. Oudot¹, and Yusu Wang²

1 Inria Saclay Île-de-France, Palaiseau, France

mickael.buchet@m4x.org, frederic.chazal@inria.fr, steve.oudot@inria.fr

2 Department of Computer Science and Engineering, The Ohio State University,
Columbus, OH 43210, USA

tamaldey@cse.ohio-state.edu, fan.171@osu.edu, yusu@cse.ohio-state.edu

Abstract

Given a real-valued function f defined over a manifold M embedded in \mathbb{R}^d , we are interested in recovering structural information about f from the sole information of its values on a finite sample P . Existing methods provide approximation to the persistence diagram of f when geometric noise and functional noise are bounded. However, they fail in the presence of aberrant values, also called outliers, both in theory and practice.

We propose a new algorithm that deals with outliers. We handle aberrant functional values with a method inspired from the k -nearest neighbors regression and the local median filtering, while the geometric outliers are handled using the distance to a measure. Combined with topological results on nested filtrations, our algorithm performs robust topological analysis of scalar fields in a wider range of noise models than handled by current methods. We provide theoretical guarantees and experimental results on the quality of our approximation of the sampled scalar field.

1998 ACM Subject Classification I.3.5 Computational Geometry and Object Modeling

Keywords and phrases Persistent Homology, Topological Data Analysis, Scalar Field Analysis, Nested Rips Filtration, Distance to a Measure

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.827

1 Introduction

Consider a network of sensors measuring a quantity such as the temperature, the humidity, or the elevation. These sensors also compute their positions and communicate these data to others. However, they are not perfect and can make mistakes such as providing some aberrant values. Can we still recover topological structure from the measured quantity?

This is an instance of a scalar field analysis problem. Given a manifold M embedded in \mathbb{R}^d and a scalar field $f : M \rightarrow \mathbb{R}$, we want to extract topological information about f , knowing only its values on a finite set of points P . The critical points of a function, that is, peaks (local maxima), pits (local minima), and passes (saddle points) constitute important topological features of the function. In addition, the prominence of these features also contains valuable information, which the geographers use to distinguish between a summit and a local maximum in its shadow. Such information can be captured by the so-called *topological persistence*, which studies the *sub-level sets* $f^{-1}((-\infty, \alpha])$ of a function f and the way their topology evolves as parameter α increases. In the case of geography, we can use

* See [1] for the full version of this paper.



the negated elevation as a function to study the topography. Peaks will appear depending on their altitude and will merge into other topological features at saddle points. This provides a *persistence diagram* describing the lifespan of features where the peaks with more prominence have longer lifespans.

When the domain M of the function f is triangulated, one classical way of computing this diagram is to linearly interpolate the function f on each simplex and then apply the standard persistence algorithm to this piecewise-linear function [16]. For cases where we only have pairwise distances between input points, one can build a family of simplicial complexes and infer the persistent homology of the input function f from them [6] (this construction will be detailed in Section 2).

Both of these approaches can provably approximate persistent homology when the input points admit a bounded noise, i.e., when the Hausdorff distance between P and M is bounded and the L_∞ -error on the observed value of f is also bounded. What happens if the noise is unbounded? A faulty sensor can provide completely wrong information or a bad position. Previous methods no longer work in this setting. Moreover, a sensor with a good functional value but a bad position can become an outlier in function value at its measured position (see Section 3.1 for an example). In this paper, we study the problem of analyzing scalar fields in the presence of unbounded noise both in the geometry and in the functional values. To the best of our knowledge, there is no other method to handle such combined unbounded geometric and functional noise with theoretical guarantees.

Contributions

We consider a general sampling condition. Intuitively, a sample (P, \tilde{f}) of a function $f : M \rightarrow \mathbb{R}$ respects our condition if: (i) the domain M is sampled densely and there is no cluster of noisy samples outside M (roughly speaking, no area outside M has a higher sampling density than on M), and (ii) for any point of P , at least half of its k nearest neighbors have a functional value with an error less than a threshold s . This condition allows functional outliers that may have a value arbitrarily far away from the true one. It encompasses the previous bounded sampling conditions as well as other sampling conditions such as bounded Wasserstein distance for geometry, or generative models like an additive Gaussian noise. Connection to some of these classical sampling conditions can be found in the full version of the paper [1].

We show how to approximate the persistence diagram of f knowing only its observed value \tilde{f} on the set P . We achieve this goal through three main steps:

1. Using the observations \tilde{f} , we provide a new estimator \hat{f} to approximate f . This estimator is inspired by the k -nearest neighbours regression technique but differs from it in an essential way.
2. We filter geometric outliers using a distance to a measure function.
3. We combine both techniques in a unified framework to estimate the persistence diagram of f .

The two sources of noise, geometric and functional, are not independent. The interdependency is first identified by assuming appropriate sampling conditions, and then untangled by separate steps in our algorithm.

Related work

A framework for scalar field topology inference with theoretical guarantees has been previously proposed in [6]. However, it is limited to a bounded noise assumption, which we aim to relax.

For handling the functional noise only, the traditional non-parametric regression mostly uses kernel-based or k -NN estimators. The k -NN methods are more versatile [13]. Nevertheless, the kernel-based estimators are preferred when there is structure in the data. However, the functional outliers destroy the structure on which kernel-based estimators rely. These functional outliers can arise as a result of geometric outliers (see Section 3.1). Thus, in a way, it is essential to be able to handle functional outliers when the input has geometric noise. Functional outliers can also introduce a bias that hampers the robustness of a k -NN regression. For example, if all outliers' values are greater than the actual value, a k -NN regression will shift towards a larger value. Our approach leverages the k -NN regression idea while trying to avoid the sensitivity to this bias.

Various methods for geometric denoising have also been proposed in the literature. If the generative model for noise is known a priori, one can use de-convolution to remove noise. Some methods have been specifically adapted to use topological information for such denoising [14]. In our case where the generative model is unknown, we use a filtering by the value of the distance to a measure, which has been successfully applied to infer the topology of a domain under unbounded noise [4].

2 Preliminaries for Scalar Field Analysis

In [6], Chazal et al. presented an algorithm to analyze the scalar field topology using persistent homology which can handle bounded Hausdorff noise both in geometry and in observed function values. Our approach follows the same high level framework. Hence in this section, we introduce necessary preliminaries along with some of the results from [6].

Riemannian manifold and its sampling.

Consider a compact Riemannian manifold M . Let d_M denote the geodesic metric on M . Consider the open Riemannian ball $B_M(x, r) := \{y \in M \mid d_M(x, y) < r\}$ centered at $x \in M$. $B_M(x, r)$ is *strongly convex* if for any pair (y, y') in the closure of $B_M(x, r)$, there exists a unique minimizing geodesic between y and y' whose interior is contained in $B_M(x, r)$. Given any $x \in M$, let $\varrho(x)$ denote the supremum of the value of r such that $B_M(x, r)$ is strongly convex. As M is compact, the infimum of all $\varrho(x)$ is positive and we denote it by $\varrho(M)$, which is called the *strong convexity radius* of M .

A point set $P \subseteq M$ is a *geodesic ε -sample* of M if for every point x of M , the distance from x to P is less than ε in the metric d_M . Given a c -Lipschitz scalar function $f : M \rightarrow \mathbb{R}$, we aim to study the persistent homology of f . However, the scalar field $f : M \rightarrow \mathbb{R}$ is only approximated by a discrete set of sample points P and a function $\tilde{f} : P \rightarrow \mathbb{R}$. The goal of this paper is to retrieve the topological structure of f from \tilde{f} when some forms of noise are present both in the positions of P and in the function values of \tilde{f} .

Persistent homology.

As in [6], we infer the persistent homology of f using well-chosen *persistence modules*. A *filtration* $\{F_\alpha\}_{\alpha \in \mathbb{R}}$ is a family of sets F_α totally ordered by inclusions $F_\alpha \subseteq F_\beta$. Following [3], a persistence module is a family of vector spaces $\{\Phi_\alpha\}_{\alpha \in \mathbb{R}}$ with a family of homomorphisms $\phi_\alpha^\beta : \Phi_\alpha \rightarrow \Phi_\beta$ such that for all $\alpha \leq \beta \leq \gamma$, $\phi_\alpha^\gamma = \phi_\beta^\gamma \circ \phi_\alpha^\beta$. Given a filtration $\mathcal{F} = \{F_\alpha\}_{\alpha \in \mathbb{R}}$ and $\alpha \leq \beta$, the canonical inclusion $F_\alpha \hookrightarrow F_\beta$ induces a homomorphism at the homology level $H_*(F_\alpha) \rightarrow H_*(F_\beta)$. These homomorphisms and the homology groups of F_α form the so-called *persistence module* of \mathcal{F} .

The persistence module of the filtration $\mathcal{F} = \{F_\alpha\}_{\alpha \in \mathbb{R}}$ is said to be *q-tame* when all the homomorphisms $H_*(F_\alpha) \rightarrow H_*(F_\beta)$ have finite rank [5]. Its algebraic structure can then be described by the *persistence diagram* $\text{Dgm}(\mathcal{F})$, which is a multiset of points in \mathbb{R}^2 describing the lifespan of the homological features in the filtration \mathcal{F} . For technical reasons, $\text{Dgm}(\mathcal{F})$ also contains every point of the diagonal $y = x$ with countably infinite multiplicity. See [10] for a more formal discussion of the persistence diagrams.

Persistence diagrams can be compared using the *bottleneck distance* d_B [8]. Given two multisets with the same cardinality, possibly infinite, D and E in \mathbb{R}^2 , we consider the set \mathcal{B} of all bijections between D and E . The bottleneck distance (under L_∞ -norm) is then defined as:

$$d_B(D, E) = \inf_{b \in \mathcal{B}} \sup_{x \in D} \|x - b(x)\|_\infty. \tag{1}$$

Two filtrations $\{U_\alpha\}$ and $\{V_\alpha\}$ are said to be ε -*interleaved* if, for any α , we have $U_\alpha \subset V_{\alpha+\varepsilon} \subset U_{\alpha+2\varepsilon}$. Recent work in [3, 5] shows that two interleaved filtrations induce close persistence diagrams in the bottleneck distance.

► **Theorem 2.1.** *Let U and V be two q -tame and ε -interleaved filtrations. Then the persistence diagrams of these filtrations verify $d_B(\text{Dgm}(U), \text{Dgm}(V)) \leq \varepsilon$.*

Nested filtrations

The scalar field topology of $f : M \rightarrow \mathbb{R}$ is studied via the topological structure of the sub-level sets filtration of f . More precisely, the sub-level sets of f are defined as $F_\alpha = f^{-1}((-\infty, \alpha])$ for any $\alpha \in \mathbb{R}$. The collection of sub-level sets forms a filtration $\mathcal{F} = \{F_\alpha\}_{\alpha \in \mathbb{R}}$ connected by natural inclusions $F_\alpha \subseteq F_\beta$ for any $\alpha \leq \beta$. Our goal is to approximate the persistence diagram $\text{Dgm}(\mathcal{F})$ from the observed scalar field $\tilde{f} : P \rightarrow \mathbb{R}$. We now describe the results of [6] for approximating $\text{Dgm}(\mathcal{F})$ when P is a geodesic ε -sample of M . These results will later be useful for our approach.

To simulate the sub-level sets filtration $\{F_\alpha\}$ of f , we introduce $P_\alpha = \tilde{f}^{-1}((-\infty, \alpha]) \subseteq P$ for any $\alpha \in \mathbb{R}$. The points in P_α intuitively sample the sub-level set F_α . To estimate the topology of F_α from these discrete samples P_α , we consider the δ -*offset* P^δ of the point set P , i.e., we grow geodesic balls of radius δ around the points of P . This gives us a union of balls that serves as a proxy for $f^{-1}((-\infty, \alpha])$. The nerve of this collection of balls, also known as the *Čech complex*, $C_\delta(P)$, has many interesting properties but is difficult to compute in high dimensions. We consider an alternate complex called the *Vietoris-Rips complex* $R_\delta(P)$ that is easier to compute. It is defined as the maximal simplicial complex with the same 1-skeleton as the Čech complex. The Čech and Rips complexes are related in any metric space: $\forall \delta > 0, C_\delta(P) \subset R_\delta(P) \subset C_{2\delta}(P)$.

Even though a single Vietoris-Rips complex may not capture the homology of the manifold M , a pair of nested complexes can recover it using the inclusions $R_\delta(P_\alpha) \hookrightarrow R_{2\delta}(P_\alpha)$ [7]. Specifically, for a fixed $\delta > 0$, consider the following commutative diagram induced by inclusions, for $\alpha \leq \beta$:

$$\begin{array}{ccc} H_*(R_{2\delta}(P_\alpha)) & \xrightarrow{\phi_\alpha^\beta} & H_*(R_{2\delta}(P_\beta)) \\ i_\alpha \uparrow & & \uparrow i_\beta \\ H_*(R_\delta(P_\alpha)) & \longrightarrow & H_*(R_\delta(P_\beta)) \end{array}$$

As the diagram commutes for all $\alpha \leq \beta$, $\{Im(i_\alpha), \phi_\alpha^\beta|_{Im(i_\alpha)}\}$ defines a persistence module. We call it the persistent homology module of the filtration of nested pairs $\{R_\delta(P_\alpha) \hookrightarrow$

$R_{2\delta}(P_\alpha)\}_{\alpha \in \mathbb{R}}$. This construction can also be done for any filtration of nested pairs. Using this construction, one of the main results of [6] is:

► **Theorem 2.2** (Theorems 2 and 6 of [6]). *Let M be a compact Riemannian manifold and let $f : M \rightarrow \mathbb{R}$ be a c -Lipschitz function. Let P be a geodesic ε -sample of M . If $\varepsilon < \frac{1}{4}\varrho(M)$, then for any $\delta \in [2\varepsilon, \frac{1}{2}\varrho(M))$, the persistent homology modules of f and of the filtration of nested pairs $\{R_\delta(P_\alpha) \hookrightarrow R_{2\delta}(P_\alpha)\}$ are $2c\delta$ -interleaved. Therefore, the bottleneck distance between their persistence diagrams is at most $2c\delta$.*

Furthermore, the k -dimensional persistence diagram for the filtrations of nested pairs $\{R_\delta(P_\alpha) \hookrightarrow R_{2\delta}(P_\alpha)\}$ can be computed in $O(|P|kN + N \log N + N^3)$ time, where N is the number of simplices of $\{R_{2\delta}(P_\infty)\}$, and $|P|$ denotes the cardinality of the sample set P .

It has been observed that, in practice, the persistence algorithm often has a running time linear in the number of simplices, which reduces the above complexity to $O(|P| + N \log N)$ in a practical setting.

We say that \tilde{f} has a precision of ξ over P if $|\tilde{f}(p) - f(p)| \leq \xi$ for any $p \in P$. We then have the following result for the case when we only have this functional noise:

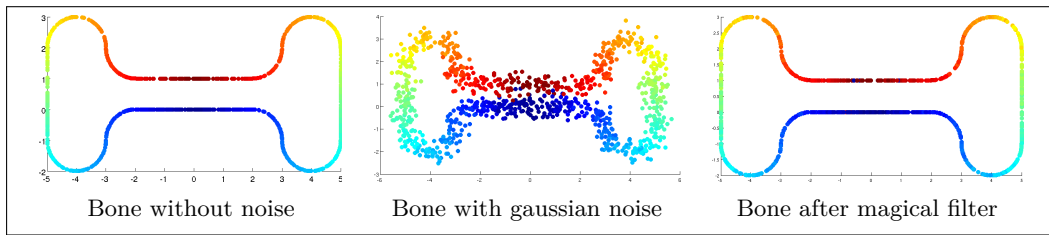
► **Theorem 2.3** (Theorem 3 of [6]). *Let M be a compact Riemannian manifold and let $f : M \rightarrow \mathbb{R}$ be a c -Lipschitz function. Let P be a geodesic ε -sample of M such that the values of f on P are known with precision ξ . If $\varepsilon < \frac{1}{4}\varrho(M)$, then for any $\delta \in [2\varepsilon, \frac{1}{2}\varrho(M))$, the persistent homology modules of f and of the filtration of nested pairs $\{R_\delta(P_\alpha) \hookrightarrow R_{2\delta}(P_\alpha)\}$ are $(2c\delta + \xi)$ -interleaved. Therefore, the bottleneck distance between their persistence diagrams is at most $2c\delta + \xi$.*

Geometric noise was considered in the form of bounded noise in the estimate of the geodesic distances between points in P . It translated into a relation between the measured pairwise distances and the real ones. With only geometric noise, one has the following stability result. It was stated in this form in the conference version of the paper.

► **Theorem 2.4** (Theorem 4 of [6]). *Let M, f be defined as previously and P be an ε -sample of M in its Riemannian metric. Assume that, for a parameter $\delta > 0$, the Rips complexes $R_\delta(\cdot)$ are defined with respect to a metric $\tilde{d}(\cdot, \cdot)$ which satisfies $\forall x, y \in P, \frac{d_M(x, y)}{\lambda} \leq \tilde{d}(x, y) \leq \nu + \mu \frac{d_M(x, y)}{\lambda}$, where $\lambda \geq 1$ is a scaling factor, $\mu \geq 1$ is a relative error and $\nu \geq 0$ an additive error. Then, for any $\delta \geq \nu + 2\mu\frac{\varepsilon}{\lambda}$ and any $\delta' \in [\nu + 2\mu\delta, \frac{1}{\lambda}\varrho(M)]$, the persistent homology modules of f and of the filtration of nested pairs $\{R_\delta(P_\alpha) \hookrightarrow R_{\delta'}(P_\alpha)\}$ are $c\lambda\delta'$ -interleaved. Therefore, the bottleneck distance between their persistence diagrams is at most $c\lambda\delta'$.*

3 Functional Noise

In this section, we focus on the case where we have only functional noise in the observed function \tilde{f} . Suppose we have a scalar function f defined on a Riemannian manifold M embedded in \mathbb{R}^d . Note that the results of section 3 hold if \mathbb{R}^d is replaced by a metric space \mathbb{X} . We are given a geodesic ε -sample $P \subset M$, and a noisy observed function $\tilde{f} : P \rightarrow \mathbb{R}$. Our goal is to approximate the persistence diagram $\text{Dgm}(\mathcal{F})$ of the sub-level set filtration $\mathcal{F} = \{F_\alpha = f^{-1}((-\infty, \alpha])\}_\alpha$ from \tilde{f} . We assume that f is c -Lipschitz with respect to the intrinsic metric of the Riemannian manifold M . Note that this does not imply a Lipschitz condition on \tilde{f} .



■ **Figure 1** Bone example after applying Gaussian perturbation and magical filter

3.1 Functional sampling condition

Previous work on functional noise focused on bounded noise (e.g, [6]) or noise with zero-mean (e.g, [15]). However, there are many practical scenarios where the observed function \tilde{f} may contain these previously considered types of noise combined with *aberrant function values* in \tilde{f} . Hence, we propose below a more general sampling condition that allows such combinations.

Motivating examples

First, we provide some motivating examples for the need of handling *aberrant* function values in \tilde{f} , where $\tilde{f}(p)$ at some sample point p can be totally unrelated to the true value $f(p)$. Consider a sensor network, where each node returns some measures. Such measurements can be imprecise, and in addition to that, a sensor may experience failure and return a completely wrong measure that has no relation with the true value of f . Similarly, an image could be corrupted with impulse noise where there are random pixels with aberrant function values, such as random white or black dots.

More interestingly, outliers in function values can naturally appear as a result of (extrinsic) geometric noise present in the discrete samples. For example, imagine that we have a process that can measure the function value $f : M \rightarrow \mathbb{R}$ with *no error*. However, the geometric location \tilde{p} of a point $p \in M$ can be wrong. In particular, \tilde{p} can be close to other parts of the manifold, thereby although \tilde{p} has the correct function value $f(p)$, it becomes a functional outlier among its neighbors (due to the wrong location of \tilde{p}). See Figure 1 for an illustration. The function defined on this bone-like curve is the geodesic distance to a base point. The two sides of the narrow neck have very different function values. Now, suppose that the points are sampled uniformly on M and their position is then perturbed by an additive Gaussian noise. Then, points from one side of this neck can be sent closer to the other side, causing aberrant values in the observed function.

In fact, even if we assume that we have a “magic filter” that can project each sample back to the closest point on the underlying manifold M , the result is a new set of samples where all points are on the manifold and thus can be seen as having **no** geometric noise; however, this point set now contains functional noise which is actually caused by the original geometric noise. Note that such a magic filter is the goal of many geometric denoising methods. A perfect algorithm in this sense cannot remove or may even cause more aberrant functional noise. This motivates the need for handling functional outliers (in addition to traditional functional noise) as well as processing noise that combines geometric and functional noise together and that does not necessarily have zero-mean.

Another case where our approach is useful concerns with missing data. Assuming that some of the functional values are missing, we can replace them by anything and act as if they were outliers. Without modifying the algorithm, we obtain a way to handle the local loss of information.

Functional sampling condition

To allow both aberrant and more traditional functional noise, we introduce the following sampling condition. Let $P \subset M$ be a geodesic ε -sample of the underlying manifold M . Intuitively, our sampling condition requires that for every point $p \in P$, locally there is a sufficient number of sample points with reasonably good function values. Specifically, we fix two parameters k and k' with the condition that $k \geq k' > \frac{1}{2}k$. Let $NN_P^k(p)$ denote the set of the k -nearest neighbors of p in P in the *extrinsic metric*. We say that a discrete scalar field $\tilde{f} : P \rightarrow \mathbb{R}$ is a (k, k', Δ) -functional-sample of $f : M \rightarrow \mathbb{R}$ if the following holds:

$$\forall p \in P, \left| \left\{ q \in NN_P^k(p) \mid |\tilde{f}(q) - f(p)| \leq \Delta \right\} \right| \geq k' \tag{2}$$

Intuitively, this sampling condition allows up to $k - k'$ samples around a point p to be outliers (whose function values deviates from $f(p)$ by at least Δ). In the full version [1], we consider two standard functional sampling conditions used in the statistical learning community and look at what they correspond to in our setting.

3.2 Functional Denoising

Given a scalar field $\tilde{f} : P \rightarrow \mathbb{R}$ which is a (k, k', Δ) -functional-sample of $f : M \rightarrow \mathbb{R}$, we now aim to compute a denoised function $\hat{f} : P \rightarrow \mathbb{R}$ from the observed function \tilde{f} , and we will later use \hat{f} to infer the topology of $f : M \rightarrow \mathbb{R}$. Below we describe two ways to denoise the noisy observation \tilde{f} : one of which is well-known, and the other one is new. As we will see later, these two treatments lead to similar theoretical guarantees in terms of topology inference. However, they have different characteristics in practice, which are discussed in the full version [1].

***k*-median denoising**

In the *k*-median treatment, we simply perform the following: given any point $p \in P$, we set $\hat{f}(p)$ to be the median value of the set of \tilde{f} values for the k -nearest neighbors $NN_P^k(p) \subseteq P$ of p . We call \hat{f} the *k*-median denoising of \tilde{f} . The following observation is straightforward:

► **Observation 1.** *If $\tilde{f} : P \rightarrow \mathbb{R}$ is a (k, k', Δ) -functional-sample of $f : M \rightarrow \mathbb{R}$ with $k' \geq k/2$, then we have $|\hat{f}(p) - f(p)| \leq \Delta$ for any $p \in P$, where \hat{f} is the *k*-median denoising of \tilde{f} .*

Disparity-based denoising

In the *k*-median treatment, we choose a single value from the k -nearest neighbors of a sample point p and set it to be the denoised value $\hat{f}(p)$. This value, while within Δ distance to the true value $f(p)$ for $k' \geq k/2$, tends to have greater variability among neighboring sample points. Intuitively, taking the average (such as *k*-means) makes the function $\hat{f}(p)$ smoother, but it is sensitive to outliers. We combine these ideas together, and use the following concept of disparity to help us identify a subset of points from the *k*-nearest neighbors of a sample point p to estimate $\hat{f}(p)$.

Given a set $Y = \{x_1, \dots, x_l\}$ of l sample points from P , we define its disparity w.r.t. \tilde{f} as:

$$\phi(Y) = \frac{1}{l} \sum_{i=1}^l (\tilde{f}(x_i) - \mu(Y))^2, \quad \text{where } \mu(Y) = \frac{1}{l} \sum_{i=1}^l \tilde{f}(x_i).$$

$\mu(Y)$ and $\phi(Y)$ are respectively the average and the variance of the observed function values for points from Y . Intuitively, $\phi(Y)$ measures how tight the function values ($\tilde{f}(x_i)$) are clustered. Now, given a point $p \in P$, we define

$$\widehat{Y}_p = \underset{Y \subseteq \text{NN}_P^k(p), |Y|=k'}{\text{argmin}} \phi(Y), \quad \text{and} \quad \widehat{z}_p = \mu(\widehat{Y}_p).$$

That is, \widehat{Y}_p is the subset of k' points from the k -nearest neighbors of p that has the smallest disparity and \widehat{z}_p is its mass center. It turns out that \widehat{Y}_p and \widehat{z}_p can be computed by the following *sliding-window* procedure: (i) Sort $\text{NN}_P^k(p) = \{x_1, \dots, x_k\}$ according to $\tilde{f}(x_i)$. (ii) For every k' consecutive points $Y_i = \{x_i, \dots, x_{i+k'-1}\}$ with $i \in [1, k - k' + 1]$, compute its disparity $\phi(Y_i)$. (iii) Set $\widehat{Y}_p = \underset{Y_i, i \in [1, k - k']}{\text{argmin}} \phi(Y_i)$, and return $\mu(\widehat{Y}_p)$ as \widehat{z}_p . In the *disparity-based denoising* approach, we simply set $\widehat{f}(p) := \widehat{z}_p$ as computed above. The approximation guarantee of \widehat{f} for the function f is given by the following Lemma.

► **Lemma 3.1.** *If $\tilde{f} : P \rightarrow \mathbb{R}$ is a (k, k', Δ) -functional-sample of $f : M \rightarrow \mathbb{R}$ with $k' \geq \frac{k}{2}$, then we have $|\widehat{f}(p) - f(p)| \leq \left(1 + 2\sqrt{\frac{k-k'}{2k'-k}}\right) \Delta$ for every $p \in P$, where \widehat{f} is the disparity-based denoising of \tilde{f} . In particular, if $k' \geq \frac{2}{3}k$, then $|\widehat{f}(p) - f(p)| \leq 3\Delta$ for every $p \in P$.*

Proof. Let $Y_\Delta = \{x \in \text{NN}_P^k(p) : |\tilde{f}(x) - f(p)| \leq \Delta\}$ be the set of points in $\text{NN}_P^k(p)$ whose observed function values are within distance Δ from $f(p)$. Since \tilde{f} is a (k, k', Δ) -functional-sample of f , it is clear that $|Y_\Delta| \geq k'$. Let $Y'_\Delta \subset Y_\Delta$ be a subset with k' elements, $Y'_\Delta = \{x'_i\}_{i=1}^{k'}$. By the definitions of Y_Δ and Y'_Δ , one can immediately check that $|\tilde{f}(x'_i) - \mu(Y'_\Delta)| \leq 2\Delta$ where $\mu(Y'_\Delta) = \frac{1}{k'} \sum_{i=1}^{k'} \tilde{f}(x'_i)$. This inequality then gives an upper bound of the disparity $\phi(Y'_\Delta)$,

$$\begin{aligned} \phi(Y'_\Delta) &= \frac{1}{k'} \sum_{i=1}^{k'} (\tilde{f}(x'_i) - \mu(Y'_\Delta))^2 \\ &\leq \frac{1}{k'} \sum_{i=1}^{k'} (2\Delta)^2 \\ &= 4\Delta^2 \end{aligned}$$

Recall from the sliding window procedure that $\widehat{Y}_p = \underset{Y_i, i \in [1, k - k']}{\text{argmin}} \phi(Y_i)$ and $\widehat{z}_p = \mu(\widehat{Y}_p)$. Denote $A_1 = \widehat{Y}_p \cap Y_\Delta$ and $A_2 = \widehat{Y}_p \setminus A_1$. Since \tilde{f} is a (k, k', Δ) -functional-sample of f , the size of A_2 is at most $k - k'$ and $|A_1| \geq 2k' - k$. If $|\widehat{z}_p - f(p)| \leq \Delta$, nothing needs to be proved. Without loss of generality, one can assume that $f(p) + \Delta \leq \widehat{z}_p$. Denote $\delta = \widehat{z}_p - (f(p) + \Delta)$. The disparity of $\phi(\widehat{Y}_p)$ can then be estimated.

$$\begin{aligned} \phi(\widehat{Y}_p) &= \frac{1}{k'} \left(\sum_{x \in A_1} (\tilde{f}(x) - \widehat{z}_p)^2 + \sum_{x \in A_2} (\tilde{f}(x) - \widehat{z}_p)^2 \right) \\ &\geq \frac{1}{k'} \left(|A_1| \delta^2 + \sum_{x \in A_2} (\tilde{f}(x) - \widehat{z}_p)^2 \right) \\ &\geq \frac{1}{k'} \left(|A_1| \delta^2 + \frac{1}{|A_2|} \left(\sum_{x \in A_2} \tilde{f}(x) - |A_2| \widehat{z}_p \right)^2 \right) \\ &= \frac{1}{k'} \left(|A_1| \delta^2 + \frac{1}{|A_2|} \left(\sum_{x \in A_1} \tilde{f}(x) - |A_1| \widehat{z}_p \right)^2 \right) \\ &\geq \frac{1}{k'} \left(|A_1| \delta^2 + \frac{1}{|A_2|} (|A_1| \delta)^2 \right) \\ &= \frac{1}{k'} \delta^2 \left(\frac{|A_1|}{|A_2|} (|A_1| + |A_2|) \right) \\ &\geq \frac{1}{k'} \delta^2 \left(\frac{k'|A_1|}{|A_2|} \right) \\ &\geq \frac{2k' - k}{k - k'} \delta^2 \end{aligned}$$

where the third line uses the inequality $\sum_{i=1}^n a_i^2 \geq \frac{1}{n} (\sum_{i=1}^n a_i)^2$, and the fourth line uses the fact that $(|A_1| + |A_2|) \widehat{z}_p = \sum_{x \in \widehat{Y}_p} \tilde{f}(x)$. Since $\widehat{Y}_p = \underset{Y_i, i \in [1, k - k']}{\text{argmin}} \phi(Y_i)$, it holds that

$\phi(\widehat{Y}_p) \leq \phi(Y'_\Delta)$. Therefore,

$$\frac{2k' - k}{k - k'} \delta^2 \leq 4\Delta^2.$$

It then follows that $\delta \leq 2\sqrt{\frac{k-k'}{2k'-k}}\Delta$ and $|\widehat{f}(p) - f(p)| \leq \left(1 + 2\sqrt{\frac{k-k'}{2k'-k}}\right)\Delta$ since $\widehat{z}_p = \widehat{f}(p)$. If $k' \geq \frac{2}{3}k$, then $1 + 2\sqrt{\frac{k-k'}{2k'-k}} \leq 1 + 2 = 3$, meaning that $|\widehat{f}(p) - f(p)| \leq 3\Delta$ in this case. ◀

► **Corollary 3.2.** *Given a (k, k', Δ) -functional-sample of $f : M \rightarrow \mathbb{R}$ with $k' \geq k/2$, we can compute a new function $\widehat{f} : P \rightarrow \mathbb{R}$ such that $|\widehat{f}(p) - f(p)| \leq \xi\Delta$ for any $p \in P$, where $\xi = 1$ under k -median denoising, and $\xi = \left(1 + 2\sqrt{\frac{k-k'}{2k'-k}}\right)$ under the disparity-based denoising.*

Hence after the k -median denoising or the disparity-based denoising, we obtain a new function \widehat{f} whose value at each sample point is within $\xi\Delta$ precision to the true function value. We can now apply the scalar field topology inference framework from [6] (as introduced in Section 2) using \widehat{f} as input. In particular, set $L_\alpha = \{p \in P \mid \widehat{f}(p) \leq \alpha\}$, and let $R_\delta(X)$ denote the Rips complex over points in X with parameter δ . We approximate the persistence diagram induced by the sub-level sets filtration of $f : M \rightarrow \mathbb{R}$ from the filtrations of nested pairs $\{R_\delta(L_\alpha) \hookrightarrow R_{2\delta}(L_\alpha)\}_\alpha$. It follows from Theorem 2.3 that:

► **Theorem 3.3.** *Let M be a compact Riemannian manifold and let $f : M \rightarrow \mathbb{R}$ be a c -Lipschitz function. Let P be a geodesic ε -sample of M , and $\widehat{f} : P \rightarrow \mathbb{R}$ a (k, k', Δ) -functional-sample of f . Set $\xi = 1$ if P_α is obtained via k -median denoising, and $\xi = \left(1 + 2\sqrt{\frac{k-k'}{2k'-k}}\right)$ if P_α is obtained via disparity-based denoising. If $\varepsilon < \frac{1}{4}\varrho(M)$, then for any $\delta \in [2\varepsilon, \frac{1}{2}\varrho(M))$, the persistent homology modules of f and the filtration of nested pairs $\{R_\delta(P_\alpha) \hookrightarrow R_{2\delta}(P_\alpha)\}$ are $(2c\delta + \xi\Delta)$ -interleaved. Therefore, the bottleneck distance between their persistence diagrams is at most $2c\delta + \xi\Delta$.*

The above theoretical results are similar for k -median and disparity-based methods with a slight advantage for the k -median. However, interesting experimental results can be obtained when the Lipschitz condition on the function is removed, for example with images, where the disparity based method appears to be more resilient to large amounts of noise than the k -median denoising method. Illustrating examples can be found in the full version [1].

4 Geometric noise

In the previous section, we assumed that we have no geometric noise in the input. In this section, we deal with the case where there is only geometric noise in the input, but no functional noise of any kind. Specifically, for any point $p \in P$, we assume that the observed value $\tilde{f}(p)$ is equal to the true function value $f(\pi(p))$ where $\pi(p)$ is the nearest point projection of p to the manifold. If p is on the medial axis of M , the projection π is arbitrary to one of the nearest points. As we have alluded before, general geometric noise implicitly introduces functional noise because the point p may have become a functional aberration of its orthogonal projection $\pi(p) \in M$. This error will be ultimately dealt with in Section 5 when we combine the results on purely functional noise from the previous section with the results on purely geometric noise in this section.

4.1 Sampling condition

Distance to a measure

The distance to a measure is a tool introduced to deal with geometrically noisy datasets, which are modelled as probability measures [4]. Given a probability measure μ on a metric space \mathbb{X} , we define the *pseudo-distance* $\delta_m(x)$ for any point $x \in \mathbb{R}^d$ and a mass parameter $m \in (0, 1]$ as $\delta_m(x) = \inf\{r \in \mathbb{R} \mid \mu(B(x, r)) \geq m\}$. The distance to a measure is then defined by averaging this quantity:

$$d_{\mu,m}(x) = \sqrt{\frac{1}{m} \int_0^m \delta_l(x)^2 dl}.$$

The *Wasserstein distance* is a standard tool to compare two measures. Given two probability measures μ and ν on a metric space \mathbb{X} , a *transport plan* π is a probability measure over $\mathbb{X} \times \mathbb{X}$ such that for any $A \times B \subset \mathbb{X} \times \mathbb{X}$, $\pi(A \times \mathbb{X}) = \mu(A)$ and $\pi(\mathbb{X} \times B) = \nu(B)$. Let $\Gamma(\mu, \nu)$ be the set of all transport plans between measures μ and ν . The Wasserstein distance is then defined as the minimum transport cost over $\Gamma(\mu, \nu)$:

$$W_2(\mu, \nu) = \sqrt{\min_{\pi \in \Gamma(\mu, \nu)} \int_{\mathbb{X} \times \mathbb{X}} d_{\mathbb{X}}(x, y)^2 d\pi(x, y)},$$

where $d_{\mathbb{X}}(x, y)$ is the distance between x and y in the metric space \mathbb{X} . The distance to a measure is stable with respect to the Wasserstein distance as shown in [4]:

► **Theorem 4.1** (Theorem 3.5 of [4], Theorem 3.2 of [2]). *Let μ and ν be two probability measures on \mathbb{X} and $m \in (0, 1]$. Then, $\|d_{\mu,m} - d_{\nu,m}\|_{\infty} \leq \frac{1}{\sqrt{m}} W_2(\mu, \nu)$.*

We will mainly use the distance to empirical measures in this paper. (See [2, 4, 12] for more details on distance to a measure and its approximation.) Given a finite point set P , its associated *empirical measure* μ_P is defined as the sum of Dirac masses: $\mu_P = \frac{1}{|P|} \sum_{p \in P} \delta_p$. The distance to this empirical measure for a point x can then be expressed as an average of its distances to the $k = m|P|$ nearest neighbors where m is the mass parameter. For the sake of simplicity, k will be assumed to be an integer. The results also hold for other values of k . However, a non integer k introduces unnecessary technical difficulties. Denoting by $p_i(x)$ the i -th nearest neighbors of x in P , one can write:

$$d_{\mu_P,m}(x) = \sqrt{\frac{1}{k} \sum_{i=1}^k d(p_i(x), x)^2}.$$

Geometric sampling condition

Our sampling condition treats the input point data as a measure and relates it to the manifold (where input points are sampled from) via distance-to-measures with the help of two parameters.

► **Definition 4.2.** Let $P \subset \mathbb{R}^n$ be a discrete sample and $M \subset \mathbb{R}^n$ a smooth manifold. Let μ_P denote the empirical measure of P . For a fixed mass parameter $m > 0$, we say that P is an (ε, r) -sample of M if the following holds:

$$\forall x \in M, d_{\mu_P,m}(x) \leq \varepsilon; \quad \text{and} \tag{3}$$

$$\forall x \in \mathbb{R}^n, d_{\mu_P,m}(x) \leq r \implies d(x, M) \leq d_{\mu_P,m}(x) + \varepsilon. \tag{4}$$

The parameter ε captures the distance to the empirical measure for points in M and intuitively tells us how dense P is in relation to the manifold M . The parameter r intuitively indicates how far away we can deviate from the manifold, while keeping the noise sparse enough so as not to be mistaken for signal. We remark that if a point set is an (ε, r) -sample of M then it is an (ε', r') -sample of M for any $\varepsilon' \geq \varepsilon$ and $r' \leq r$. In general, the smaller ε is and the bigger r is, the better an (ε, r) -sample is.

For convenience, denote the distance function to the manifold M by $d_\pi : \mathbb{R}^n \rightarrow \mathbb{R}$, $x \mapsto d(x, M)$. We have the following interleaving relation:

$$\forall \alpha < r - \varepsilon, d_\pi^{-1}([-\infty, \alpha]) \subset d_{\mu_P, m}^{-1}([-\infty, \alpha + \varepsilon]) \subset d_\pi^{-1}([-\infty, \alpha + 2\varepsilon]) \tag{5}$$

To see why this interleaving relation holds, let x be a point such that $d(x, M) \leq \alpha$. Thus $d(\pi(x), x) \leq \alpha$. Using the hypothesis (3), we get that $d_{\mu_P, m}(\pi(x)) \leq \varepsilon$. Given that the distance to a measure is a 1-Lipschitz function we then obtain that $d_{\mu_P, m}(x) \leq \varepsilon + \alpha$.

Now let x be a point such that $d_{\mu_P, m}(x) \leq \alpha + \varepsilon \leq r$. Using the condition on r in (4) we get that $d(x, M) \leq d_{\mu_P, m}(x) + \varepsilon \leq \alpha + 2\varepsilon$ which concludes the proof of Eqn (5).

Eqn (5) gives an interleaving between the sub-level sets of the distance to the measure μ and the offsets of the manifold M . By Theorem 2.1, this implies the proximity between the persistence modules of their respective sub-level sets filtrations. Observe that this relation is in some sense analogous to the one obtained when two compact sets A and B have Hausdorff distance of at most ε :

$$\forall \alpha, d_A^{-1}([-\infty, \alpha]) \subset d_B^{-1}([-\infty, \alpha + \varepsilon]) \subset d_A^{-1}([-\infty, \alpha + 2\varepsilon]). \tag{6}$$

Relation to other sampling conditions

Our sampling condition encompasses several other existing sampling conditions. While the parameter ε is natural, the parameter r may appear to be artificial. It bounds the distances at which we can observe the manifold through the scope of the distance to a measure. In most classical sampling conditions, r is equal to ∞ and thus we obtain a similar relation as for the classical Hausdorff sampling condition in Eqn (6).

One notable noise model where $r \neq \infty$ is when there is a uniform background noise in the ambient space \mathbb{R}^d , sometimes called *clutter noise*. In this case, r depends on the difference between the density of the relevant data and the density of the noise. For other sampling conditions like Wasserstein, Gaussian, Hausdorff sampling conditions, $r = \infty$. Detailed relations and proofs for the Wasserstein and Gaussian sampling conditions can be found in the full version [1].

4.2 Scalar field analysis under geometric noise

In the rest of the paper, we assume that M is a manifold with positive reach ρ_M (minimum distance between M and its medial axis) and whose curvature is bounded by c_M . Assume that the input P is an (ε, r) -sample of M for a given $m \in (0, 1]$, where

$$\varepsilon \leq \frac{\rho_M}{6}, \text{ and } r > 2\varepsilon. \tag{7}$$

As discussed at the beginning of this section, we assume that there is no intrinsic functional noise, that is, for every $p \in P$, the observed function value $\tilde{f}(p) = f(\pi(p))$ is the same as the true value for the projection $\pi(p) \in M$ of this point. Our goal now is to show how to recover the persistence diagram induced by $f : M \rightarrow \mathbb{R}$ from its observations $\tilde{f} : P \rightarrow \mathbb{R}$ on P .

Taking advantage of the interleaving (5), we can use the distance to the empirical measure to filter the points of P to remove geometric noise. In particular, we consider the set

$$L = P \cap d_{\mu_{P,m}}^{-1}(\cdot - \infty, \eta] \text{ where } \eta \geq 2\varepsilon. \quad (8)$$

We will then use a similar approach as the one from [6] for this set L . The optimal choice for the parameter η is 2ε . However, any value with $\eta \leq r$ and $\eta + \varepsilon < \rho_M$ works as long as there exist δ and δ' satisfying the conditions stated in Theorem 2.4.

Let $\bar{L} = \{\pi(x) | x \in L\}$ denote the orthogonal projection of L onto M . To simulate sub-level sets $f^{-1}(\cdot - \infty, \alpha]$ of $f : M \rightarrow \mathbb{R}$, consider the restricted sets $L_\alpha := L \cap (f \circ \pi)^{-1}(\cdot - \infty, \alpha]$ and let $\bar{L}_\alpha = \pi(L_\alpha)$. By our assumption on the observed function $\tilde{f} : P \rightarrow \mathbb{R}$, we have: $L_\alpha = \{x \in L | \tilde{f}(x) \leq \alpha\}$.

Let us first recall a result about the relation between Riemannian and Euclidian metrics (e.g. [9]). For any two points $x, y \in M$ with $d(x, y) \leq \frac{\rho_M}{2}$ one has:

$$d(x, y) \leq d_M(x, y) \leq \left(1 + \frac{4d(x, y)^2}{3\rho_M^2}\right) d(x, y) \leq \frac{4}{3}d(x, y). \quad (9)$$

As a direct consequence of our sampling condition, for each point $x \in M$, there exists a point $p \in L$ at distance less than 2ε : Indeed, for each $x \in M$, since $d_{\mu_{P,m}}(x) \leq \varepsilon$, there must exist a point $p \in P$ such that $d(x, p) \leq \varepsilon$. On the other hand, since the distance to measure is 1-Lipschitz, we have $d_{\mu_{P,m}}(p) \leq d_{\mu_{P,m}}(x) + d(x, p) \leq 2\varepsilon$. Hence $p \in L$ as long as $\eta \geq 2\varepsilon$. We will use the *extrinsic* Vietoris-Rips complex built on top of points from L to infer the scalar field topology. Using the previous relation Eqn (9), we obtain the following result which states that the Euclidean distance for nearby points in L approximates the geodesic distance on M .

► **Proposition 4.3.** *Let $\lambda = \frac{4}{3} \frac{\rho_M}{\rho_M - (\eta + \varepsilon)}$, and assume that $2\varepsilon \leq \eta \leq r$ and $\varepsilon + \eta < \rho_M$. Let $x, y \in L$ be two points from L such that $d(x, y) \leq \frac{\rho_M}{2} - \frac{\eta + \varepsilon}{2}$. Then,*

$$\frac{d_M(\pi(y), \pi(x))}{\lambda} \leq d(x, y) \leq 2(\eta + \varepsilon) + d_M(\pi(x), \pi(y)).$$

Proof. Let x and y be two points of L such that $d(x, y) \leq \frac{\rho_M}{2} - \frac{\eta + \varepsilon}{2}$. As $d_{\mu_{P,m}}(x) \leq \eta \leq r$, Eqn (4) implies $d(\pi(x), x) \leq \eta + \varepsilon$. Therefore, $d(\pi(x), \pi(y)) \leq \frac{\rho_M}{\rho_M - (\eta + \varepsilon)} d(x, y)$ [11, Theorem 4.8,(8)]. This implies $d(\pi(x), \pi(y)) \leq \frac{\rho_M}{2}$ and following (9), $d_M(\pi(x), \pi(y)) \leq \frac{4}{3}d(\pi(x), \pi(y))$.

This proves the left inequality in the Proposition. The right inequality follows from

$$d(x, y) \leq d(\pi(x), x) + d(\pi(y), y) + d_M(\pi(x), \pi(y)) \leq 2(\eta + \varepsilon) + d_M(\pi(x), \pi(y)).$$

◀

► **Theorem 4.4.** *Let M be a compact Riemannian manifold and let $f : M \rightarrow \mathbb{R}$ be a c -Lipschitz function. Let P be an (ε, r) -sample of M , and L be as introduced in Eqn (8). Assume $\varepsilon \leq \frac{\rho_M}{6}$, $r > 2\varepsilon$, and $2\varepsilon \leq \eta \leq r$. Then, for any $\delta \geq 2\eta + 6\varepsilon$ and any $\delta' \in \left[2\eta + 2\varepsilon + \frac{8}{3} \frac{\rho_M}{\rho_M - (\eta + \varepsilon)} \delta, \frac{3}{4} \frac{\rho_M - (\eta + \varepsilon)}{\rho_M} \varrho(M)\right]$, $H_*(f)$ and $H_*(R_\delta(L_\alpha) \hookrightarrow R_{\delta'}(L_\alpha))$ are $\frac{4}{3} \frac{c\rho_M\delta'}{\rho_M - (\eta + \varepsilon)}$ -interleaved.*

Proof. First, note that \bar{L} is a 2ε -sample of M in its geodesic metric. It follows from the definition of $d_{\mu_{P,m}}$ that, for any point $x \in M$, the nearest point $p \in L$ to x satisfies

$d(x, p) \leq d_{\mu_P, m}(x) \leq \varepsilon$. Hence $d(x, \pi(p)) \leq d(x, p) + d(p, \pi(p)) \leq 2d(x, p) \leq 2\varepsilon$. Now we apply Theorem 2.4 to \bar{L} by using $\tilde{d}(\pi(x), \pi(y)) := d(x, y)$; and setting $\lambda = \mu = \frac{4}{3} \frac{\rho_M}{\rho_M - (\eta + \varepsilon)}$, $\nu = 2(\eta + \varepsilon)$: the requirement on the distance function \tilde{d} in Theorem 2.4 is satisfied due to Proposition 4.3. The claim then follows. \blacktriangleleft

Since M is compact, f is bounded due to the Lipschitz condition. We can look at the limit when $\alpha \rightarrow \infty$. There exists a value T such that for any $\alpha \geq T$, $L_\alpha = L$ and $f^{-1}((-\infty, \alpha]) = M$. The above interleaving means that $H_*(M)$ and $H_*(R_\delta(L)) \hookrightarrow R_{\delta'}(L)$ are interleaved. However, both objects do not depend on α and this gives the following inference result:

► **Corollary 4.5.** $H_*(M)$ and $H_*(R_\delta(L)) \hookrightarrow R_{\delta'}(L)$ are isomorphic under conditions specified in Theorem 4.4.

5 Scalar Field Topology Inference under Geometric and Functional Noise

Our constructions can be combined to analyze scalar fields in a more realistic setting. Our *combined sampling condition* follows conditions (3) and (4) for the geometry. We adapt condition (2) to take into account the geometry and introduce the following conditions: we assume that there exist $\eta \geq 2\varepsilon$ and s such that:

$$\forall p \in d_{\mu, m}^{-1}((-\infty, \eta]), |\{q \in NN_k(p) \mid |\tilde{f}(q) - f(\pi(p))| \leq s\}| \geq k' \tag{10}$$

Note that in (10), we are using $f(\pi(p))$ as the “true” function value at a sample p which may be off the manifold M . The condition on the functional noise is only for points close to the manifold (under the distance to a measure). Combining the methods from the previous two sections, we obtain the *combined noise algorithm* where η is a parameter greater than 2ε .

We propose the following 3-steps algorithm. It starts by handling outliers in the geometry then it makes a regression on the function values to obtain a smoothed function \hat{f} before running the existing algorithm for scalar field analysis [6] on the filtration $\hat{L}_\alpha = \{p \in L \mid \hat{f}(p) \leq \alpha\}$.

Combined noise algorithm

1. Compute $L = P \cap d_{\mu, m}^{-1}((-\infty, \eta])$.
 2. Replace functional values \tilde{f} by \hat{f} for points in L using either k-median or disparity based method.
 3. Run the scalar field analysis algorithm from [6] on (L, \hat{f}) .
-

► **Theorem 5.1.** Let M be a compact smooth manifold embedded in \mathbb{R}^d and f a c -Lipschitz function on M . Let $P \subset \mathbb{R}^d$ be a point set and $\tilde{f} : P \rightarrow \mathbb{R}$ be observed function values such that hypotheses (3), (4), (7) and (10) are satisfied. For $\eta \geq 2\varepsilon$, the combined noise algorithm has the following guarantees:

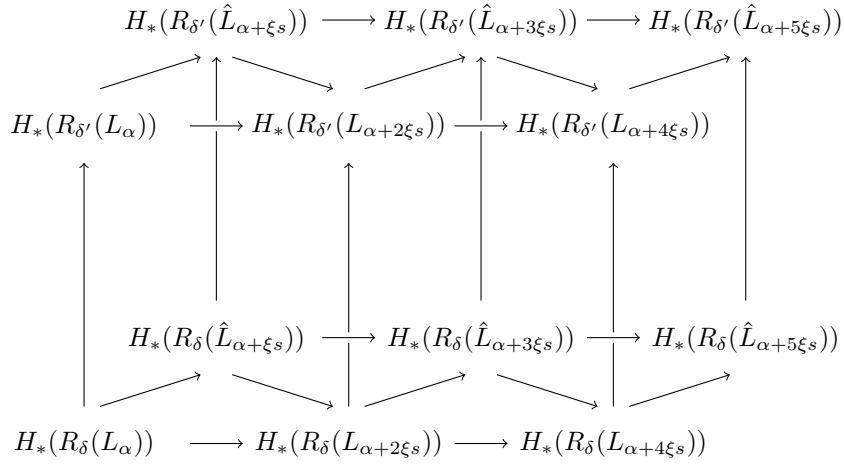
For any $\delta \in \left[2\eta + 6\varepsilon, \frac{\varrho(M)}{2}\right]$ and any $\delta' \in \left[2\eta + 2\varepsilon + \frac{8}{3} \frac{\rho_M}{\rho_M - (\eta + \varepsilon)} \delta, \frac{3}{4} \frac{\rho_M - (\eta + \varepsilon)}{\rho_M} \varrho(M)\right]$, $H_*(f)$ and $H_*(R_\delta(\hat{L}_\alpha) \hookrightarrow R_{\delta'}(\hat{L}_\alpha))$ are $\left(\frac{4}{3} \frac{c\rho_M \delta'}{\rho_M - (\eta + \varepsilon)} + \xi s\right)$ -interleaved where $\xi = 1$ if we use the k -median and $\xi = \left(1 + 2\sqrt{\frac{k - k'}{2k' - k}}\right)$ if we use the disparity method for Step 2.

Proof. First, consider the filtration induced by $L_\alpha = \{x \in L | f(\pi(x)) \leq \alpha\}$; that is, we first imagine that all points in L have correct function values (equals to the true value of their projection on M). By Theorem 4.4, for

$$\delta \in \left[2\eta + 6\varepsilon, \frac{\varrho(M)}{2} \right] \text{ and } \delta' \in \left[2\eta + 2\varepsilon + \frac{8}{3} \frac{\rho_M}{\rho_M - (\eta + \varepsilon)} \delta, \frac{3}{4} \frac{\rho_M - (\eta + \varepsilon)}{\rho_M} \varrho(M) \right],$$

$H_*(f)$ and $H_*(R_\delta(L_\alpha) \hookrightarrow R_{\delta'}(L_\alpha))$ are $\frac{4}{3} \frac{c\rho_M\delta'}{\rho_M - (\eta + \varepsilon)}$ -interleaved.

Next, consider $\hat{L}_\alpha = \{p \in L | \hat{f}(p) \leq \alpha\}$, which leads to a filtration based on the smoothed function values \hat{f} (not observed values). Recall that our algorithm returns $H_*(R_\delta(\hat{L}_\alpha) \hookrightarrow R_{\delta'}(\hat{L}_\alpha))$. We aim to relate this persistence module with $H_*(R_\delta(L_\alpha) \hookrightarrow R_{\delta'}(L_\alpha))$. Specifically, fix α and let (x, y) be an edge of $R_\delta(L_\alpha)$. This means that $d(x, y) \leq 2\delta$, $f(\pi(x)) \leq \alpha$, $f(\pi(y)) \leq \alpha$. Corollary 3.2 can be applied to the function $f \circ \pi$ due to hypothesis (10). Hence $|\hat{f}(x) - f(\pi(x))| \leq \xi s$ and $|\hat{f}(y) - f(\pi(y))| \leq \xi s$. Thus $(x, y) \in R_\delta(\hat{L}_{\alpha + \xi s})$. One can reverse the role of \hat{f} and f and get an ξs -interleaving of $\{R_\delta(L_\alpha)\}$ and $\{R_\delta(\hat{L}_\alpha)\}$. This gives rise to the following commutative diagram since all arrows are induced by inclusions.



Thus the two persistence modules induced by filtrations of nested pairs $\{R_\delta(L_\alpha) \hookrightarrow R_{\delta'}(L_\alpha)\}$ and $\{R_\delta(\hat{L}_\alpha) \hookrightarrow R_{\delta'}(\hat{L}_\alpha)\}$ are ξs -interleaved. Combining this with the interleaving between $H_*(R_\delta(L_\alpha) \hookrightarrow R_{\delta'}(L_\alpha))$ and $H_*(f)$, we obtain the stated results. ◀

We note that, while this theorem assumes a setting where we can ensure theoretical guarantees, the algorithm can be applied in a more general setting still producing good results.

Acknowledgments. This work was supported by the ANR project TopData 13-BS01-008, the ERC project Gudhi 339025 and the NSF grants CCF-1064416, CCF-1116258, CCF-1319406 and CCF-1318595.

References

- 1 M. Buchet, F. Chazal, T. K. Dey, F. Fan, S. Y. Oudot, and Y. Wang. Topological analysis of scalar fields with outliers. *arXiv preprint arXiv:1412.1680*, 2014.
- 2 M. Buchet, F. Chazal, S. Oudot, and D. R. Sheehy. Efficient and robust persistent homology for measures. In *Proceedings of the 26th ACM-SIAM symposium on Discrete algorithms*. SIAM, 2015.

- 3 F. Chazal, D. Cohen-Steiner, M. Glisse, L. J. Guibas, and S. Oudot. Proximity of persistence modules and their diagrams. In *Proc. 25th ACM Sympos. on Comput. Geom.*, pages 237–246, 2009.
- 4 F. Chazal, D. Cohen-Steiner, and Q. Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.
- 5 F. Chazal, V. de Silva, M. Glisse, and S. Oudot. The structure and stability of persistence modules, 2013. arXiv:1207.3674.
- 6 F. Chazal, L. J. Guibas, S. Y. Oudot, and P. Skraba. Scalar field analysis over point cloud data. *Discrete & Computational Geometry*, 46(4):743–775, 2011.
- 7 F. Chazal and S. Y. Oudot. Towards persistence-based reconstruction in euclidean spaces. In *Proceedings of the twenty-fourth annual symposium on Computational geometry*, pages 232–241. ACM, 2008.
- 8 D. Cohen-Steiner, H. Edelsbrunner, and J. Harer. Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1):103–120, 2007.
- 9 T. K. Dey, J. Sun, and Y. Wang. Approximating cycles in a shortest basis of the first homology group from point data. *Inverse Problems*, 27(12):124004, 2011.
- 10 H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction*. Amer. Math. Soc., Providence, Rhode Island, 2009.
- 11 H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, pages 418–491, 1959.
- 12 L. Guibas, D. Morozov, and Q. Mérigot. Witnessed k-distance. *Discrete & Computational Geometry*, 49(1):22–45, 2013.
- 13 L. Györfi. *A distribution-free theory of nonparametric regression*. Springer, 2002.
- 14 J. Kloke and G. Carlsson. Topological de-noising: Strengthening the topological signal. *arXiv preprint arXiv:0910.5947*, 2009.
- 15 S. Kpotufe. k-nn regression adapts to local intrinsic dimension. *arXiv preprint arXiv:1110.4300*, 2011.
- 16 A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete & Computational Geometry*, 33(2):249–274, 2005.

On Computability and Triviality of Well Groups*

Peter Franek¹ and Marek Krčál²

1 Institute of Computer Science, Academy of Sciences, Prague, Czech Republic
franek@cs.cas.cz

2 IST Austria, Am Campus 1 3400 Klosterneuburg, Austria
marek.krcal@ist.ac.at

Abstract

The concept of *well group* in a special but important case captures homological properties of the zero set of a continuous map $f: K \rightarrow \mathbb{R}^n$ on a compact space K that are invariant with respect to perturbations of f . The perturbations are arbitrary continuous maps within L_∞ distance r from f for a given $r > 0$. The main drawback of the approach is that the computability of well groups was shown only when $\dim K = n$ or $n = 1$.

Our contribution to the theory of well groups is twofold: on the one hand we improve on the computability issue, but on the other hand we present a range of examples where the well groups are incomplete invariants, that is, fail to capture certain important robust properties of the zero set.

For the first part, we identify a computable subgroup of the well group that is obtained by cap product with the pullback of the orientation of \mathbb{R}^n by f . In other words, well groups can be algorithmically approximated from below. When f is smooth and $\dim K < 2n - 2$, our approximation of the $(\dim K - n)$ th well group is exact.

For the second part, we find examples of maps $f, f': K \rightarrow \mathbb{R}^n$ with all well groups isomorphic but whose perturbations have different zero sets. We discuss on a possible replacement of the well groups of vector valued maps by an invariant of a better descriptive power and computability status.

1998 ACM Subject Classification G.1.5 Roots of Nonlinear Equations, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases nonlinear equations, robustness, well groups, computation, homotopy theory

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.842

1 Introduction

In many engineering and scientific solutions, a highly desired property is the resistance against noise or perturbations. We can only name a fraction of the instances: stability in data analysis [4], robust optimization [2], image processing [14], or stability of numerical methods [16]. Some very important tools for robust design come from topology, which can capture stable properties of spaces and maps.

In this paper, we take the robustness perspective on the study of the solution set of systems of nonlinear equations, a fundamental problem in mathematics and computer science. Equations arising in mathematical modeling of real problems are usually inferred

* This research was supported by institutional support RVO:67985807 and by the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no [291734].



from observations, measurements or previous computations. We want to extract maximal information about the solution set, if an estimate of the error in the input data is given.

More formally, for a continuous map $f : K \rightarrow \mathbb{R}^n$ on a compact Hausdorff space K and $r > 0$ we want to study properties of the family of zero sets

$$Z_r(f) := \{g^{-1}(0) : \|f - g\| \leq r\},$$

where $\|\cdot\|$ is the max-norm with respect to some fixed norm $|\cdot|$ in \mathbb{R}^n . The functions g with $\|f - g\| \leq r$ (or $\|f - g\| < r$) will be referred to as r -perturbations of f (or *strict r -perturbations of f* , respectively). Quite notably, we are not restricted to *parameterized* perturbations but allow arbitrary continuous functions at most (or less than) r far from f in the max-norm.

Well groups. Recently, the concept of well groups was developed to measure “robustness of intersection” of a map $f : K \rightarrow Y$ with a subspace $Y' \subseteq Y$ [8].

In the special but very important case when $Y = \mathbb{R}^n$ and $Y' = \{0\}$ it is a property of $Z_r(f)$ that, informally speaking, captures “homological properties” that are common to all zero sets in $Z_r(f)$. We enhance the theory to include a *relative case*¹ that is especially convenient in the case when K is a manifold with boundary. Let $B \subseteq K$ be a pair of compact Hausdorff spaces and $f : K \rightarrow \mathbb{R}^n$ continuous. Let $X := |f|^{-1}[0, r]$ where $|f|$ denotes the function $x \mapsto |f(x)|$; this is the smallest space containing zero sets of all r -perturbations g of f . In the rest of the paper, for any space $Y \subseteq K$ we will abbreviate the pair $(Y, Y \cap B)$ by (Y, B) and, similarly for homology, $H_*(Y, Y \cap B)$ by $H_*(Y, B)$. Everywhere in the paper we use homology and cohomology groups with coefficients in \mathbb{Z} unless explicitly stated otherwise. For brevity we omit the coefficients from the notation.

The j th well group $U_j(f, r)$ of f at radius r is the subgroup of $H_j(X, B)$ defined by

$$U_j(f, r) := \bigcap_{Z \in Z_r(f)} \text{Im}(H_j(Z, B) \xrightarrow{i_*} H_j(X, B)),$$

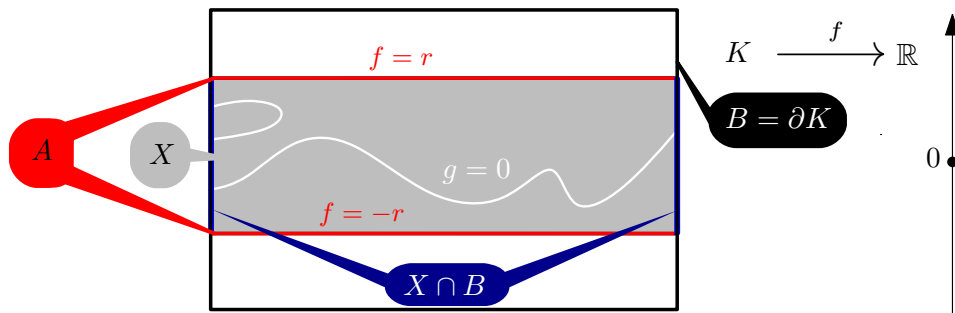
where i_* is induced by the inclusion $i : g^{-1}(0) \hookrightarrow X$ and H refers to a convenient homology theory of compact metrizable spaces that we describe below.² For a simple example of a map f with $U_1(f, r)$ nontrivial see Figure 1.

Significance of well groups. We only mention a few of many interesting things mostly related to our setting. The well group in dimension zero characterizes robustness of solutions of a system of equations $f(x) = 0$. Namely, $\emptyset \in Z_r(f)$ if and only if $U_0(f, r) \cong 0$. Higher well groups capture additional robust topological properties of the zero set such as in Figure 1. Perhaps the most important is their ability to form *well diagrams* [8] – a kind of measure for robustness of the zero set (or more generally, robustness of the intersection of f with other subspace $Y' \subseteq Y$). The well diagrams are stable with respect to taking perturbations of f .³

¹ Authors of [3] develop a different notion of relativity that is based on considering a pair of spaces (Y', Y'_0) instead of the single space Y' . This direction is rather orthogonal to the matters of this paper.

² In [8, 3], well groups were defined by means of singular homology. But then, once we allow arbitrary continuous perturbations, to the best of our knowledge, no $f : K \rightarrow \mathbb{R}^n$ with nontrivial $U_j(f, r)$ for $j > 0$ would be known. In particular, the main result of [3] would not hold. The correction via means of Steenrod homology was independently identified by the authors of [3].

³ Namely, so called *bottleneck distance* between a well diagrams of f and f' is bounded by $\|f - f'\|$. The stability does not say how well the well diagrams describe the zero set. This question is also addressed in this paper.



■ **Figure 1** For the projection $f(x, y) = y$ to the vertical axis defined on a box K , the zero set of every r -perturbation is contained in $X = |f|^{-1}[0, r]$ and ∂X consists of A (upper and lower side) where $|f| = r$, and $X \cap B \subseteq \partial K$. The zero set always separates the two components of A . On the homological level, the zero set “connects” the two components of $X \cap B$ and the image of $H_1(g^{-1}(0), B)$ in $H_1(X, B)$ is always surjective and thus $U_1(f, r) \cong H_1(X, B)$. Note that the well group would be trivial with $B = \emptyset$.

Homology theory. For the foundation of well groups we need a homology theory on compact Hausdorff spaces that satisfies some additional properties that we specify later in Section 2. Roughly speaking, we want that the homology theory behaves well with respect to infinite intersections. Without these properties we would have to consider only “well behaved” perturbations of a given f in order to be able to obtain some nontrivial well groups in dimension greater than zero. We explain this in more detail also in Section 2. For the moment it is enough to say that the Čech homology can be used and that for any computational purposes it behaves like simplicial homology. In Section 2 we explain why using singular homology would make the notion of well groups trivial.

A basic ingredient of our methods is the notion of *cap product*

$$\frown: H^n(X, A) \otimes H_k(X, A \cup B) \rightarrow H_{k-n}(X, B)$$

between cohomology and homology. We refer the reader to [21, Section 2.2] and [15, p. 239] for its properties and to [11, Appendix E] for its construction in Čech (co)homology. Again, it behaves like the simplicial cap product when applied to simplicial complexes. For an algorithmic implementation, one can use its simplicial definition from [21].

1.1 Computability results

Computer representation. To speak about computability, we need to fix some computer representation of the input. Here we assume the simple but general setting of [10], namely, K is a finite simplicial complex, $B \subseteq K$ a subcomplex, f is simplexwise linear with rational values on vertices⁴ and the norm $|\cdot|$ in \mathbb{R}^n can be (but is not restricted to) ℓ_1, ℓ_2 or ℓ_∞ norm.

Previous results. The algorithm for the computation of well groups was developed only in the particular cases of $n = 1$ [3] or $\dim K = n$ [5]. In [10] we settled the computational complexity of the well group $U_0(f, r)$. The complexity is essentially identical to deciding

⁴ We emphasize that the considered r -perturbations of f need not be neither simplexwise linear nor have rational values on the vertices.

whether the restriction $f|_A: A \rightarrow S^{n-1}$ can be extended to $X \rightarrow S^{n-1}$ for $A = |f|^{-1}(r)$, or equivalently, $A = f^{-1}(S^{n-1})$. The extendability problem can be decided as long as $\dim K \leq 2n - 3$ or $n = 1, 2$ or n is even. On the contrary, the extendability of maps into a sphere – as well as triviality of $U_0(f, r)$ – cannot be decided for $\dim K \geq 2n - 2$ and n odd, see [10].⁵ In this paper we shift our attention to higher well groups.

Higher well groups – extendability revisited. The main idea of our study of well groups is based on the following. We try to find r -perturbations of f with as small zero set as possible, that is, avoiding zero on X' for $X' \subseteq X$ as large as possible. It is shown in [11, Lemma D.1] that for each strict r -perturbation g of f we can find an extension $e: X \rightarrow \mathbb{R}^n$ of $f|_A$ with $g^{-1}(0) = e^{-1}(0)$ and vice versa. Thus equivalently, we try to extend $f|_A$ to a map $X' \rightarrow S^{n-1}$ for X' as large as possible. The higher skeleton⁶ of X we cover, the more well groups we kill.

► **Observation 1.1.** *Let $f: K \rightarrow \mathbb{R}^n$ be a map on a compact space. Assume that the pair of spaces $A \subseteq X$ defined as $|f|^{-1}(r) \subseteq |f|^{-1}[0, r]$, respectively, can be triangulated and $\dim X = m$. If the map $f|_A$ can be extended to a map $A \cup X^{(i-1)} \rightarrow S^{n-1}$ then $U_j(f, r)$ is trivial for $j > m - i$.*

Assume, in addition, that there is no extension $A \cup X^{(i)} \rightarrow S^{n-1}$. By the connectivity of the sphere S^{n-1} , we have $i \geq n$. Does the lack of extendability to $X^{(i)}$ relate to higher well groups, especially $U_{m-i}(f, r)$? The answer is *yes* when $i = n$ as we show in our computability results below. On the other hand, when $i > n$, the lack of extendability is *not* necessarily reflected by $U_{m-i}(f, r)$. This leads to the incompleteness results we show in the second part of the paper.

The first obstruction. The lack of extendability of $f|_A$ to the n -skeleton is measured by the so called *first obstruction* that is defined in terms of cohomology theory as follows. We can view f as a map of pairs $(X, A) \rightarrow (B^n, S^{n-1})$ where B^n is the ball bounded by the sphere $S^{n-1} := \{x: |x| = r\}$. Then the first obstruction ϕ_f is equal to the pullback $f^*(\xi) \in H^n(X, A)$ of the fundamental cohomology class $\xi \in H^n(B^n, S^{n-1})$.⁷

► **Theorem 1.2.** *Let $B \subseteq K$ be compact spaces and let $f: K \rightarrow \mathbb{R}^n$ be continuous. Let $|f|^{-1}[0, r]$ and $|f|^{-1}(r)$ be denoted by X and A , respectively, and ϕ_f be the first obstruction. Then $\phi_f \frown H_k(X, A \cup B)$ is a subgroup of $U_{k-n}(f, r)$ for each $k \geq n$.*

Our assumptions on computer representation allow for simplicial approximation of X, A and f . The pullback of $\xi \in H^n(B^n, S^{n-1})$ and the cap product can be computed by the standard formulas. This together with more details worked out in the proof in Section 2 gives the following.

⁵ We cannot even approximate the “robustness of roots”: it is undecidable, given a simplicial complex K and a simplexwise linear map $f: K \rightarrow \mathbb{R}^n$, whether there exists $\epsilon > 0$ such that $U_0(f, \epsilon)$ is nontrivial or whether $U_0(f, 1)$ is trivial. The extendability can always be decided for n even, however, the problem is less likely tractable for $\dim K > 2n - 2$.

⁶ The i -skeleton $X^{(i)}$ of a simplicial (cell) complex X is the subspace of X containing all simplices (cells) of dimension at most i .

⁷ This is the global description of the first obstruction as presented in [25]. It can be shown that ϕ_f depends on the homotopy class of $f|_A$ only. Another way of defining the first obstruction is the following. It is represented by the so-called *obstruction cocycle* $z_f \in Z^n(X, A)$ that assigns to each n -simplex $\sigma \in X$ the element $[f]_{\partial\sigma} \in \pi_{n-1}(S^{n-1}) \cong \mathbb{Z}$ [21, Chap. 3]. Through this definition it is not difficult to derive that the map $f|_A$ can be extended to $X^{(n)} \rightarrow S^{n-1}$ if and only if $\phi_f = 0$, see also [21, Chap. 3].

► **Theorem 1.3.** *Under the assumption on computer representation of K, B and f as above, the subgroup $\phi_f \frown H_k(X, A \cup B)$ of $U_{k-n}(f, r)$ (as in Theorem 1.2) can be computed.*

The gap between U_{k-n} and $\phi_f \frown H_k(X, A \cup B)$. There are maps f with ϕ_f trivial but nontrivial $U_0(f, r)$.⁸ But this can be detected by the above mentioned extendability criterion. We do not present an example where $U_{k-n}(f, r) \neq \phi_f \frown H_k(X, A \cup B)$ for $k - n > 0$, although the inequality is possible in general. In the rest of the paper we work in the other direction to show that there is no gap in various cases and various dimensions.

An important instance of Theorem 1.2 is the case when X can be equipped with the structure of a smooth orientable manifold.

► **Theorem 1.4.** *Let $f: K \rightarrow \mathbb{R}^n$ and X, A be as above. Assume that X can be equipped with a smooth orientable manifold structure, $A = \partial X, B = \emptyset$ and $n + 1 \leq m \leq 2n - 3$ for $m = \dim X$. Then*

$$U_{m-n}(f, r) = \phi_f \frown H_m(X, \partial X).$$

When $m = n$, the well group $U_0(f, r)$ can be strictly larger than $\phi_f \frown H_n(X, \partial X)$ but it can be computed.

We believe that the same claim holds when X is an orientable PL manifold. It remains open whether the last equation holds also for $m > 2n - 3$. Throughout the proof of Theorem 1.4, we will show that if $g: K \rightarrow \mathbb{R}^n$ is a smooth r -perturbation of f transverse to 0, then the fundamental class of $g^{-1}(0)$ is mapped to the Poincaré dual of the first obstruction. This also holds if $B \neq \emptyset$ and in all dimensions.

1.2 Well groups $U_*(f, r)$ are incomplete as an invariant of $Z_r(f)$

A simple example illustrating Theorem 1.4 is the map $f: S^2 \times B^3 \rightarrow \mathbb{R}^3$ defined by $f(x, y) := y$ with B^3 considered as the unit ball in \mathbb{R}^3 . It is easy to show that

$$\text{for every 1-perturbation } g \text{ of } f \text{ and every } x \in S^2 \text{ there is a root of } g \text{ in } \{x\} \times B^3. \quad (1)$$

This robust property is nicely captured by (and can be also derived from) the fact $U_2(f, 1) \cong \mathbb{Z}$.

The main question of Section 3 is what happens, when the first obstruction ϕ_f is trivial – and thus $f|_A$ can be extended to $X^{(n)}$ – but the map $f|_A$ cannot be extended to whole of X . The zero set of f can still have various robust properties such as (1). It is the case of $f: S^2 \times B^4 \rightarrow \mathbb{R}^3$ defined by $f(x, y) := |y|\eta(y/|y|)$ where $\eta: S^3 \rightarrow S^2$ is a homotopically nontrivial map such as the Hopf map. The zero set of each r -perturbation g of f intersects each section $\{x\} \times B_4$, but unlike in the example before, well groups do not capture this property. All well groups $U_j(f, r)$ are trivial for $j > 0$ and,⁹ consequently, they cannot distinguish f from another f' having only a single robust root in X . We will describe the construction of such f' for a wider range examples.

In the following, B_q^i will denote the i -dimensional ball of radius q , that is, $B_q^i = \{y \in \mathbb{R}^i: |y| \leq q\}$. We also emphasize that this and the following theorem hold for arbitrary coefficient group of the homology theory H_* .

⁸ This is the case for $f: \mathbb{R}^4 \rightarrow \mathbb{R}^3$ given by $f(x) := |x|\eta(x/|x|)$ where $\eta: S^3 \rightarrow S^2$ is the Hopf map.

⁹ Namely $U_2(f, r) \cong 0$ as is shown by the r -perturbation $g(x, y) = f(x, y) - rx$ with the zero set homeomorphic to the 3-sphere.

► **Theorem 1.5.** *Let $i, m, n \in \mathbb{N}$ be such that $m - i < n < i < (m + n + 1)/2$ and both $\pi_{i-1}(S^{n-1})$ and $\pi_{m-1}(S^{n-1})$ are nontrivial. Then on $K = S^{m-i} \times B_1^i$ we can define two maps $f, f' : K \rightarrow \mathbb{R}^n$ such that for all $r \in (0, 1]$*

- *f, f' induce the same $X = S^{m-i} \times B_r^i$ and $A = \partial X$ and have the same well groups for any coefficient group of the homology theory H_* defining the well groups,*
- *but $Z_r(f) \neq Z_r(f')$.*

In particular, the property

$$\text{for each } Z \in Z_r(\cdot) \text{ and } x \in S^{m-i} \text{ there exists } y \in B_r^i \text{ such that } (x, y) \in Z$$

is satisfied for f but not for f' . Namely, $Z_\epsilon(f')$ contains a singleton for each $\epsilon > 0$.

The lack of extendability not reflected by $U_{m-i}(f, r)$. The key property of the example of Theorem 1.5 is that the maps $f|_A$ and $f'|_A$ can be extended to the $(i - 1)$ -skeleton $X^{(i-1)}$ of X , for $i > n$. The difference between the maps lies in the extendability to $X^{(i)}$. Unlike in the case when $i = n$, the lack of extendability is not reflected by the well groups. The crucial part is the triviality of the well groups in dimension $m - i$ and¹⁰ this triviality holds in greater generality:

► **Theorem 1.6.** *Let $f : K \rightarrow \mathbb{R}^n$, $B \subseteq K$, $X := |f|^{-1}[0, r]$ and $A := |f|^{-1}\{r\}$. Assume that the pair (X, A) can be finitely triangulated.¹¹ Further assume that $f|_A$ can be extended to a map $h : A \cup X^{(i-1)} \rightarrow S^{n-1}$ for some i such that $m - i < n < i < (m + n)/2$ for $m := \dim X$. Then $U_{m-i}(f, r) = 0$ for any coefficient group of the homology theory H_* .*

The whole proof is in [11, Appendix C] but its core idea is already contained in the proof of Theorem 1.5. There we also comment on the possibility of finding pairs of maps f and f' with the same well groups but different robust properties of their zero sets in this more general situation.

Our subjective judgment on well groups of \mathbb{R}^n -valued maps. We find the problem of the computability of well groups interesting and challenging with connections to homotopy theory (see also Proposition 1.7 below). Moreover, we acknowledge that well groups may be accessible for non-topologists: they are based on the language of homology theory that is relatively intuitive and easy to understand. On the other hand, well groups may not have sufficient descriptive power for various situations (Theorems 1.5 and 1.6). Furthermore, despite all the effort, the computability of well groups seems far from being solved. In the following paragraphs, we propose an alternative based on homotopy and obstruction theory that addresses these drawbacks.

1.3 Related work

A replacement of well groups of \mathbb{R}^n -valued maps. In a companion paper [20], we find a complete invariant for an enriched version of $Z_r(f)$. The starting point is the surprising claim that $Z_r(f)$ – an object of a geometric nature – is determined by terms of homotopy theory.

¹⁰This dimension is somewhat important as all higher well groups are trivial by [11, Lemma C.2] and all lower homology groups of X may be trivial as is the case in Theorem 1.5. On the other hand, $H_{m-i}(X, \pi_{i-1}(S^{n-1}))$ has to be nontrivial in the case when X is a manifold for the reasons following from obstruction theory and Poincaré duality.

¹¹ That is, there exist finite simplicial complexes $A^\Delta \subseteq X^\Delta$ and a homeomorphism $(X^\Delta, A^\Delta) \rightarrow (X, A)$.

► **Proposition 1.7** ([20]). *Let $f: K \rightarrow \mathbb{R}^n$ be a continuous map on a compact Hausdorff domain, $r > 0$, and let us denote the space $|f|^{-1}[r, \infty]$ by A_r . Then the set $Z_r(f) := \{g^{-1}(0) : \|g - f\| \leq r\}$ is determined by the pair (K, A_r) and the homotopy class of $f|_{A_r}$ in $[A_r, \{x \in \mathbb{R}^n : \|x\| \geq r\}] \cong [A_r, S^{n-1}]$.¹²*

The complete proof can be found in [11, Appendix D] and will also appear in [20].

Note that since the well groups is a property of $Z_r(f)$, they are determined by the pair (K, A_r) and the homotopy class $[f|_{A_r}]$. Thus the homotopy class has a greater descriptive power and the examples from the previous section show that this inequality is strict. If K is a simplicial complex, f is simplexwise linear and $\dim A_r \leq 2n - 4$ then $[A_r, S^{n-1}]$ has a natural structure of an Abelian group denoted by $\pi^{n-1}(A_r)$. The restriction $\dim A_r \leq 2n - 4$ does not apply when $n = 1, 2$ and¹³ otherwise we could replace $[A_r, S^{n-1}]$ with $[A_r^{(2n-4)}, S^{n-1}]$ which contains less information but is computable. The isomorphism type of $\pi^{n-1}(A_r)$ together with the distinguished element $[f|_{A_r}]$ can be computed essentially by [23, Thm 1.1]. Moreover, the inclusions $A_s \subseteq A_r$ for $s \geq r$ induce computable homomorphisms between the corresponding pointed Abelian groups. Thus for a given f we obtain a sequence of pointed Abelian groups $\pi^{n-1}(A_r), r > 0$ and it can be easily shown that the interleaving distance of the sequences $\pi^{n-1}(A_*(f))$ and $\pi^{n-1}(A_*(g))$ is bounded by $\|g - f\|$. Thus after tensoring the groups by an arbitrary field, we get persistence diagrams (with a distinguished bar) that will be stable with respect to the bottleneck distance and the L_∞ norm. The construction will be detailed in [20].

The computation of the cohomotopy group $\pi^{n-1}(A)$ is naturally segmented into a hierarchy of approximations of growing computational complexity. Therefore our proposal allows for compromise between the running time and the descriptive power of the outcome. The first level of this hierarchy is the primary obstruction ϕ_f . One could form similar modules of cohomology groups with a distinguished element as we did with the cohomotopy groups above. However, in this paper we passed to homology via cap product in order to relate it to the established well groups. In the “generic” case when X is a manifold no information is lost as from the Poincaré dual $\phi_f \frown [X]$ we can reconstruct the primary obstruction ϕ_f back.

The cap-image groups. The groups $\phi_f \frown H_k(X, A)$ (with $B = \emptyset$) has been studied by Amit K. Patel under the name *cap-image groups*. In fact, his setting is slightly more complex with \mathbb{R}^n replaced by arbitrary manifold Y . Instead of the zero sets, he considers preimages of all points of Y simultaneously in some sense. Although his ideas have not been published yet, they influenced our research; the application of the cap product in the context of well groups should be attributed to Patel.¹⁴

Verification of zeros. An important topic in the interval computation community is the verification of the (non)existence of zeros of a given function [19]. While the nonexistence can be often verified by interval arithmetic alone, a proof of existence requires additional

¹² Here $[A_r, S^{n-1}]$ denotes the set of all homotopy classes of maps from A_r to S^{n-1} , that is, the cohomotopy group $\pi^{n-1}(A_r)$ when $\dim A_r \leq 2n - 4$.

¹³ Note that for $n = 1$ the structure of the set $[A, S^{n-1}]$ is very simple and for $n = 2$ we have $[A, S^{n-1}] \cong H^1(A; \mathbb{Z})$ no matter what the dimension of A_r is.

¹⁴ We originally proved that when K is a triangulated orientable manifold, the Poincaré dual of ϕ_f is contained in $U_{m-n}(f, r)$. Expanding the proof was not difficult, but the preceding inspiration of replacing the Poincaré duality by cap product came from Patel. The cap product provides a nice generalization to an arbitrary simplicial complex K .

methods which often include topological considerations. In the case of continuous maps $f : B^n \rightarrow \mathbb{R}^n$, Miranda's or Borsuk's theorem can be used for zero verification [13, 1], or the computation of the topological degree [17, 6, 12]. Fulfilled assumptions of these tests not only yield a zero in B^n but also a "robust" zero and a nontrivial 0th well group $U_0(f, r)$ for some $r > 0$. Recently, topological degree has been used for simplification of vector fields [22].

The first obstruction ϕ_f is the analog of the degree for underdetermined systems, that is, when $\dim K > n$ in our setting. To the best of our knowledge, this tool has not been algorithmically utilized.

2 Computing lower bounds on well groups

Homology theory behind the well groups. For computing the approximation $\phi_f \frown H_k(X, A \cup B)$ of well group $U_{k-n}(f)$ we only have to work with simplicial complexes and simplicial maps for which all homology theories satisfying the Eilenberg–Steenrod axioms are naturally equivalent. Hence, regardless of the homology theory H_* used, we can do the computations in simplicial homology. Therefore the standard algorithms of computational topology [7] and the formula for the cap product of a simplicial cycle and cocycle [21, Section 2.2] will do the job.

The need for a carefully chosen homology theory stems from the courageous claim that the zero set Z of arbitrary continuous perturbation supports $\phi_f \frown \beta$ for any $\beta \in H_*(X, A \cup B)$, i.e. some element of $H_*(Z, B)$ is mapped by the inclusion-induced map to $\phi_f \frown \beta$. Without more restrictions on the perturbations, the zero sets can be "wild" non-triangulable topological spaces that can fool singular homology and render this claim false and – to the best of our knowledge – make well groups trivial. See an example after the proof of Theorem 1.2.

For the purpose of the work with the general zero sets, we will require that our homology theory satisfies the Eilenberg–Steenrod axioms with a possible exception of the exactness axiom, and these additional properties:

1. *Weak continuity property:* for an inverse sequence of compact pairs $(X_0, B_0) \supset (X_1, B_1) \supset \dots$ the homomorphism $H_* \varprojlim (X_i, B_i) \rightarrow \varprojlim H_*(X_i, B_i)$ induced by the family of inclusion $\varprojlim (X_i, B_i) = \bigcap (X_i, B_i) \hookrightarrow (X_j, B_j)$ is surjective.
2. *Strong excision:* Let $f : (X, X') \rightarrow (Y, Y')$ be a map of compact pairs that maps $X \setminus X'$ homeomorphically onto $Y \setminus Y'$. Then $f_* : H_*(X, X') \rightarrow H_*(Y, Y')$ is an isomorphism.

Čech homology theory satisfies these properties as well as the Eilenberg–Steenrod axioms with the exception of the exactness axiom, and coincides with simplicial homology for triangulable spaces [24, Chapter 6].

In addition, we need a cohomology theory H^* that satisfies the Eilenberg–Steenrod axioms and is paired with H_* via a cap product $H^n(X, A) \otimes H_k(X, A \cup B) \xrightarrow{\frown} H_{k-n}(X, B)$ that is natural¹⁵ and coincides with the simplicial cap product when applied to simplicial complexes. We have not found any reference for the definition of cap product in Čech (co)homology, so we present our own construction in [11, Appendix E].

Proof of Theorem 1.2. We need to show that for any map g with $\|g - f\| \leq r$, the image of the inclusion-induced map

$$H_*(g^{-1}(0), B) \rightarrow H_*(X, B)$$

¹⁵ Naturality of the cap product means that if $f : (X, A \cup B, A) \rightarrow (X; A' \cup B', A')$ is continuous, then $f_*(f^*(\tilde{\alpha}) \frown \beta) = \tilde{\alpha} \frown f_*(\beta)$ for any $\beta \in H_*(X, A \cup B)$ and $\tilde{\alpha} \in H^*(X', A')$.

contains the cap product of the first obstruction $\phi_f := f^*(\xi)$ with all relative homology classes of $(X, A \cup B)$. Let us first restrict to the less technical case of g being a strict r -perturbation, that is, $\|g - f\| < r$.

Let us denote $X_0 := X = |f|^{-1}[0, r]$ and $A_0 := A = |f|^{-1}(r)$. Next we choose a decreasing positive sequence $\epsilon_1 > \epsilon_2 > \dots$ with $\lim_{i \rightarrow \infty} \epsilon_i = 0$ and with $\epsilon_1 < r - \|f - g\|$. Thus $X_1 := |g|^{-1}[0, \epsilon_1] \subseteq X_0$ and $A'_0 := |g|^{-1}[\epsilon_2, \infty] \cap X_0 \supseteq |g|^{-1}[\epsilon_2, \epsilon_1]$. Then we for each $i > 0$ we define

- $X_i := |g|^{-1}[0, \epsilon_i]$,
- and its subspaces $A_i := |g|^{-1}[\epsilon_{i+1}, \epsilon_i]$, $A'_i := |g|^{-1}[\epsilon_{i+2}, \epsilon_i]$ and $B_i := B \cap X_i$.

Note that $\bigcap_i X_i = g^{-1}(0)$, and consequently, $\bigcap_i B_i = g^{-1}(0) \cap B$. For any given $\beta \in H_k(X, A \cup B)$, our strategy is to find homology classes $\alpha_i \in H_{k-n}(X_i, B_i)$, with $\alpha_0 = \phi_f \frown \beta$, that fit into the sequence of maps $H_{k-n}(X_0, B_0) \leftarrow H_{k-n}(X_1, B_1) \leftarrow \dots$ induced by inclusions. This gives an element in $\varprojlim H_{k-n}(X_i, B_i)$, and consequently by the weak continuity property (requirement 1 above), we get the desired element $\alpha \in H_{k-n}(g^{-1}(0), B)$.

The elements α_i will be constructed as cap products. To that end, we need to obtain “analogs” of β and for that we will need a more complicated sequence of maps. It is the zig-zag sequence

$$X_0 \xrightarrow{\text{id}} X_0 \xleftarrow{\text{incl}} X_1 \xrightarrow{\text{id}} X_1 \xleftarrow{\text{incl}} X_2 \xrightarrow{\text{id}} \dots \tag{2}$$

that restricts to the zig-zags

$$A_0 \xrightarrow{\text{incl}} A'_0 \xleftarrow{\text{incl}} A_1 \xrightarrow{\text{incl}} A'_1 \xleftarrow{\text{incl}} A_2 \xrightarrow{\text{incl}} \dots \tag{3}$$

and

$$A_0 \cup B_0 \xrightarrow{\text{incl}} A'_0 \cup B_0 \xleftarrow{\text{incl}} A_1 \cup B_1 \xrightarrow{\text{incl}} A'_1 \cup B_1 \xleftarrow{\text{incl}} A_2 \cup B_2 \xrightarrow{\text{incl}} \dots \tag{4}$$

The pair $(X_{i+1}, A_{i+1} \cup B_{i+1})$ is obtained from $(X_i, A'_i \cup B_i)$ by excision of $|g|^{-1}(\epsilon_{i+1}, \epsilon_i]$, that is, $X_{i+1} = X_i \setminus |g|^{-1}(\epsilon_{i+1}, \epsilon_i]$ and $A_{i+1} \cup B_{i+1} = (A'_i \cup B_i) \setminus |g|^{-1}(\epsilon_{i+1}, \epsilon_i]$. Hence by excision,¹⁶ each inclusion of the pairs $(X_i, A'_i \cup B_i) \hookrightarrow (X_{i+1}, A_{i+1} \cup B_{i+1})$ induces isomorphism on relative homology groups. Therefore the zig-zag sequences (2) and (4) induce a sequence

$$\begin{array}{ccccccc} H_k(X_0, A_0 \cup B_0) & \rightarrow & H_k(X_0, A'_0 \cup B_0) & \cong & H_k(X_1, A_1 \cup B_1) & \rightarrow & H_k(X_1, A'_1 \cup B_1) & \cong & \dots \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \\ \beta_0 := \beta & & \beta'_0 & & \beta_1 & & \beta'_1 & & \dots \end{array}$$

that can be made pointed by choosing the distinguished homology classes $\beta_i \in H_k(X_i, A_i \cup B_i)$ and $\beta'_i \in H_k(X_i, A'_i \cup B_i)$ that are the images of $\beta_0 := \beta \in H_k(X, A \cup B)$ in this sequence.

Similarly, we want to construct a pointed zig-zag sequence in cohomology induced by (2) and (3). The distinguished elements $\phi_i \in H^n(X_i, A_i)$ and $\phi'_i \in H^n(X_i, A'_i)$ are defined as the pullbacks of the fundamental cohomology class $\xi \in H^n(\mathbb{R}^n, \mathbb{R}^n \setminus \{0\})$ by the restrictions of g . Because of the functoriality of cohomology, ϕ_i and ϕ'_i fit into the sequence induced by (2) and (3):

$$\begin{array}{ccccccc} H^n(X_0, A_0) & \leftarrow & H^n(X_0, A'_0) & \rightarrow & H^n(X_1, A_1) & \leftarrow & H^n(X_1, A'_1) & \rightarrow & \dots \\ \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \\ \phi_0 & & \phi'_0 & & \phi_1 & & \phi'_1 & & \dots \end{array}$$

¹⁶ Because of our careful choice of the spaces A_i and A'_i we do not need the strong excision here. However, we do not know how to avoid it in the case when $\|g - f\| = r$.

Since g is an r -perturbation of f and thus $g|_{(X,A)}$ is homotopic to $f|_{(X,A)}$ via the straight line homotopy, we have that $\phi_0 = \phi_f \in H^n(X, A)$.

From the naturality of the cap product we get that the elements $\phi_i \frown \beta_i$ and $\phi'_i \frown \beta'_i$ fit into the sequence

$$\begin{array}{ccccccc} H_{k-n}(X_0, B_0) & \xrightarrow{\text{id}} & H_{k-n}(X_0, B_0) & \leftarrow & H_{k-n}(X_1, B_1) & \xrightarrow{\text{id}} & H_{k-n}(X_1, B_1) & \leftarrow & \dots \\ \cup & & \cup & & \cup & & \cup & & \\ \phi_0 \frown \beta_0 & & \phi'_0 \frown \beta'_0 & & \phi_1 \frown \beta_1 & & \phi'_1 \frown \beta'_1 & & \dots \\ \parallel & & & & & & & & \\ \phi_f \frown \beta & & & & & & & & \end{array}$$

that is induced by (2), that is, each $H_{k-n}(X_i, B_i) \xrightarrow{\text{id}} H_{k-n}(X_i, B_i)$ is induced by the identity $X_i \xrightarrow{\cong} X_i$ and each map $H_{k-n}(X_i, B_i) \leftarrow H_{k-n}(X_{i+1}, B_{i+1})$ is induced by the inclusion $X_i \hookrightarrow X_{i+1}$. Hence $\alpha_i := \phi_i \frown \beta_i$ are the desired elements and thus there is an element $\tilde{\alpha} := (\alpha_0, \alpha_1, \dots)$ in $\varprojlim H_{k-n}(X_i, B_i)$.

We recall that the weak continuity property of the homology theory H_* assures the surjectivity of the the map

$$(\iota_i)_{i \geq 0}: H_{k-n}\left(\bigcap_i X_i, B\right) \rightarrow \varprojlim H_{k-n}(X_i, B) \tag{5}$$

where each component ι_i is induced by the inclusion $\bigcap_i X_i \hookrightarrow X_i$. Let $\alpha \in H_{k-n}(g^{-1}(0), B)$ be arbitrary preimage of $\tilde{\alpha}$ under the surjection (5). By construction, α is mapped to $\alpha_0 = \phi_f \frown \beta$ by the map ι_0 .

It remains to prove the theorem in the case when $\|g - f\| = r$. The proof goes along the same lines with only the following differences:

- For arbitrary decreasing sequence $1 = \epsilon_0 > \epsilon_1 > \epsilon_2 > \dots$ with $\lim \epsilon_i = 0$ we define $h_i := \epsilon_i f + (1 - \epsilon_i)g$ for $i \geq 0$. We will furthermore need that $2\epsilon_{i+1} > \epsilon_i$ for every $i \geq 0$. Let

$$\begin{aligned} X_i &:= |h_i|^{-1}[0, \epsilon_i r], \\ \cup \\ A'_i &:= \{x \in X : |h_i(x)| \leq \epsilon_i r \text{ and } |h_{i+1}(x)| \geq \epsilon_{i+1} r\} \text{ and} \\ \cup \\ A_i &:= |h_i|^{-1}(\epsilon_i r). \end{aligned}$$

We have $A_i \subseteq A'_i$ because by definition $\|h_i - h_{i+1}\| \leq (\epsilon_i - \epsilon_{i+1})r$ and thus $|h_i(x)| = \epsilon_i r$ implies $|h_{i+1}(x)| \geq \epsilon_{i+1} r$. Similarly $A_{i+1} \subseteq A'_i$ and $X_{i+1} \subseteq X_i$. Therefore as before, the zig-zag sequence (2) restricts to (3) and (4).

- The homology classes β_i and β'_i are defined as above. We only need to use the strong excision for the inclusion $(X_i, A'_i \cup B_i) \hookrightarrow (X_{i+1}, A_{i+1} \cup B_{i+1})$.
- We define the cohomology classes $\phi_i := h_i^*(\xi)$ and $\phi'_i := h_{i+1}^*(\xi)$. We only need to check that h_i is homotopic to h_{i+1} as a map of pairs $(X_i, A'_i) \rightarrow (\mathbb{R}^n, \mathbb{R}^n \setminus \{0\})$. Indeed, they are homotopic via the straight-line homotopy since $|h_{i+1}(x)| \geq \epsilon_{i+1} r$ implies $|h_i(x)| \geq \epsilon_{i+1} r - (\epsilon_i - \epsilon_{i+1})r = (2\epsilon_{i+1} - \epsilon_i)r > 0$. We used the inequality $2\epsilon_{i+1} > \epsilon_i$ which was our requirement on the sequence $(\epsilon_i)_{i > 0}$. We also have $\phi_0 = \phi_f$ as $h_0 = f$ and $(X_0, A_0) = (X, A)$.
- We continue by defining cap products α_i , their limit $\tilde{\alpha}$ and its preimage α under the surjection $H_{k-n}(\bigcap_i X_i, B) \rightarrow \varprojlim H_{k-n}(X_i, B)$. To finish the proof we claim that $\bigcap_i X_i = g^{-1}(0)$. Indeed, $g(x) = 0$ implies $h_i(x) \leq \|h_i - g\| = \epsilon_i r$ for each i and $g(x) > 0$ implies $h_i(x) > 0$ for i such that $2\epsilon_i r < |g(x)|$.

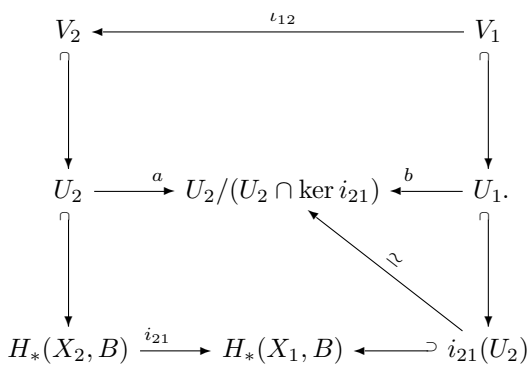


The surjectivity of (5) and the strong excision is not only a crucial step for Theorem 1.2 but implicitly also for the results stated in [3, p. 16]. If we defined well groups by means of singular homology, then even in a basic example $f(x, y) = x^2 + y^2 - 2$ and $r = 1$, the first well group $U_1(f, r)$ would be trivial. The zero set of any 1-perturbation g is contained in the annulus $X := \{(x, y) : 1 \leq x^2 + y^2 \leq 3\}$ and the two components of ∂X are not in the same connected components of $\{x \in X : g(x) \neq 0\}$. However, we could construct a “wild” 1-perturbation g of f such that $g^{-1}(0)$ is a Warsaw circle [18] which is, roughly speaking, a circle with infinite length, trivial first singular homology, but nontrivial Čech homology. Thus Čech homology serves as a better theoretical basis for the well groups. Another solution to avoid problems with wild zero sets would be to restrict ourselves to “nice” perturbations, for example piecewise linear or smooth and transverse to 0. Such approach would lead, to the best of our knowledge, to identical results.

Proof of Theorem 1.3. Under the assumption on computer representation of K and f , the pair (X, A) is homeomorphic to a computable simplicial pair (X', A') such that X' is a subcomplex of a subdivision K' of K [10, Lemma 3.4]. Therefore, the induced triangulation B' of $B \cap X'$ is a subcomplex of X' . Furthermore, a simplicial approximation $f' : A' \rightarrow S'$ of $f|_A : A \rightarrow S^{n-1}$ can be computed. The computation is implicit in the proof of Theorem 1.2 in [10] where the sphere S^{n-1} is approximated by the boundary S' of the n -dimensional cross polytope B' . The simplicial approximation $(X', A') \rightarrow (B', S')$ of $f|_X$ can be constructed consequently by sending each vertex of $X \setminus A$ to an arbitrary point in the interior of the cross polytope, say $0 \in \mathbb{R}^n$. The pullback of a cohomology class can be computed by standard algorithms. Therefore ϕ_f and $H_*(X, B)$ can be computed and the explicit formula for the cap product in [21, Section 2.1] yields the computation of $\phi_f \frown H_*(X, B)$. All this can be done without any restriction on the dimensions of the considered simplicial complexes. ◀

Well diagram associated with $\phi \frown H_*(X, A \cup B)$. Let $r_1 > r_2 > 0$ and let X_1, X_2, A_1, A_2 be $|f|^{-1}[0, r_1], |f|^{-1}[0, r_2], |f|^{-1}\{r_1\}, |f|^{-1}\{r_2\}$ respectively, ϕ_1, ϕ_2 be the respective obstructions. Further, let $A'_1 := |f|^{-1}[r_2, r_1]$ and $\phi'_1 = f^*(\xi) \in H^n(X_1, A'_1)$ be the pullback of the fundamental class $\xi \in H^n(\mathbb{R}^n, \mathbb{R}^n \setminus \{0\})$. The inclusions $(X_1, A_1) \subseteq (X_1, A'_1) \supseteq (X_2, A_2)$ induce cohomology maps that take ϕ'_1 to ϕ_1 resp. ϕ_2 . Let us denote, for simplicity, by V_1 the group $\phi_1 \frown H_*(X_1, A_1 \cup B)$, $V_2 := \phi_2 \frown H_*(X_2, A_2 \cup B)$ and $V'_1 := \phi'_1 \frown H_*(X_1, A'_1 \cup B)$. Further, let U_1 resp. U_2 be the well groups $U(f, r_1)$ resp. $U(f, r_2)$.

In this section, we analyze the relation between V_1 and V_2 . First let i_1 be a map from V_1 to V'_1 that maps $\phi_1 \frown \beta_1$ to $\phi'_1 \frown i_*(\beta_1)$. By the naturality of cap product, $\phi_1 \frown \beta_1 = \phi'_1 \frown i_*(\beta_1)$, so i_1 is an inclusion.



By excision, there is an inclusion-induced isomorphisms $i'_1 : H_*(X_2, A_2 \cup B) \xrightarrow{\sim} H_*(X_1, A'_1 \cup B)$ and its inverse induces an isomorphism $i_2 : V'_1 \xrightarrow{\sim} V_2$ by mapping $\phi'_1 \frown \beta'_1$ to $\phi_2 \frown (i'_1)^{-1}(\beta'_1)$. The composition $i_2 \circ i_1 =: \iota_{12}$ is a homomorphism from V_1 to V_2 . Being the composition of an inclusion and an isomorphism, ι_{12} is an injection and one easily verifies that the inclusion-induced map $i_{21} : H_*(X_2, B) \rightarrow H_*(X_1, B)$ satisfies $i_{21} \circ \iota_{12} = \text{id}_{|V_1}$. It follows that $\{V(r_i), \iota_{i, i+1}\}_{r_i > r_{i+1}}$ is

a persistence module consisting of shrinking abelian groups and injections $V_i \rightarrow V_{i+1}$ for $r_i > r_{i+1}$. The relation between ι and well diagrams described in [9] is reflected by the commutative diagram above.

The idea behind the proof of Theorem 1.4. In the special case when X is a smooth m -manifold with $A = \partial X$, the zero set of any smooth r -perturbation g transverse to 0 is an $(m - n)$ -dimensional smooth submanifold of X . It is not so difficult to show that its fundamental class $[g^{-1}(0)]$ is mapped by the inclusion-induced map to $\phi_f \frown [X]$, where $[X] \in H_m(X, \partial X)$ is the fundamental class of X . If $g^{-1}(0)$ is connected, then $H_{m-n}(g^{-1}(0))$ is generated by its fundamental class and we immediately obtain the reverse inclusion $\phi_f \frown H_m(X, A) \supseteq U_{m-n}(f, r)$. The nontrivial part in the proof of Theorem 1.4 is to show that in the indicated dimension range, we can find a perturbation g so that $g^{-1}(0)$ is connected. The full proof is in [11, Appendix B].

3 Incompleteness of well groups

In this section, we study the case when the first obstruction ϕ_f is trivial and thus the map $f|_A$ can be extended to a map $f^{(n)}: X^{(n)} \rightarrow S^{n-1}$ on the n -skeleton $X^{(n)}$ of X . Observation 1.1 (proved in [11, Appendix C]) implies that the only possibly nontrivial well groups are $U_j(f, r)$ for $j \leq m - n - 1$.

The following lemma summarizes the necessary tools for the constructions of this section. They directly follow from Lemma D.1 in [11, Appendix D] and from [10, Lemma 3.3].

► **Lemma 3.1.** *Let $f: K \rightarrow \mathbb{R}^n$ be a map on a compact Hausdorff space, $r > 0$, and let us denote the pair of spaces $|f|^{-1}[0, r]$ and $|f|^{-1}\{r\}$ by X and A , respectively. Then*

1. *for each extension $e: X \rightarrow \mathbb{R}^n$ of $f|_A$ we can find a strict r -perturbation g of f with $g^{-1}(0) = e^{-1}(0)$;*
2. *for each r -perturbation g of f without a root there is an extension $e: X \rightarrow \mathbb{R}^n \setminus \{0\}$ of $f|_A$ (without a root).*

In the following we want to show that well groups can fail to distinguish between maps with intrinsically different families of zero sets. Namely, in the following examples we present maps f and f' with $U_0(f, r) = U_0(f', r) = \mathbb{Z}$ for each $r \leq 1$ and $U_i(f, r) = U_i(f', r) = 0$ for each $r \leq 1$ and $i > 0$. However, $Z_r(f)$ will be significantly different from $Z_r(f')$.

Proof of Theorem 1.5. We have that $B = \emptyset$ and $K = S^j \times B^i$, where B^i is represented by the unit ball in \mathbb{R}^i and $j = m - i$. Let the maps $f, f': K \rightarrow \mathbb{R}^n$ be defined by

$$f(x, y) := |y|\varphi(x, y/|y|) \quad \text{and} \quad f'(x, y) := |y|\varphi'(x, y/|y|)$$

where $\varphi, \varphi': S^j \times S^{i-1} \rightarrow S^{n-1} \subseteq \mathbb{R}^n$ are defined by

- $\varphi(x, y) := \mu(y)$ where $\mu: S^{i-1} \rightarrow S^{n-1}$ is an arbitrary nontrivial map.
- φ' is defined as the composition $S^j \times S^{i-1} \rightarrow S^{m-1} \xrightarrow{\nu} S^{n-1}$ where the first map is the quotient map $S^j \times S^{i-1} \rightarrow S^j \wedge S^{i-1} \cong S^{m-1}$ and ν is an arbitrary nontrivial map. In other words, we require that the composition $\varphi'\Phi$ – where Φ denotes the characteristic map of the $(m - 1)$ -cell of $S^j \times S^{i-1}$ – is equal to the composition νq , where q is the quotient map $B^{m-1} \rightarrow B^{m-1}/(\partial B^{m-1}) \cong S^{m-1}$.

Well groups computation. Next we prove that the well groups of $U_*(f, r)$ and $U_*(f', r)$ are the same for $r \in (0, 1]$, namely, nonzero only in dimension 0, where they are isomorphic to \mathbb{Z} . We obviously have $X = S^j \times \{y \in \mathbb{R}^i : |y| \leq r\} \simeq S^j \times B^i$ and $A = \partial X$ for both maps. The restriction $f|_A$ and $f'|_A$ are equal to φ and φ' (after normalization). We first prove that $U_0(f, 1) \cong U_0(f', 1) \cong \mathbb{Z}$. This fact follows from $H_0(X) \cong \mathbb{Z}$, from non-extendability of φ and φ' and from Lemma 3.1 part 2 (or [10, Lemma 3.3]).

► **Lemma 3.2.** *The map φ' cannot be extended to a map $X \rightarrow S^{n-1}$.*

The proof can be found in [11, Appendix A]. Since the map $\mu: S^{i-1} \rightarrow S^{n-1}$ cannot be extended to $B^i \supset S^{i-1}$, also φ cannot be extended to X .

Since then only the j th homology group of X is nontrivial, the remaining task is to show that $U_j(f, 1) \cong U_j(f', 1) \cong 0$. We do so by presenting two r -perturbations g and g' of f and f' , respectively:

- $g(x, y) := f(x, y) - rx = |y|\mu(y/|y|) - rx$ where we consider $S^j \subseteq \mathbb{R}^{j+1}$ as a subset of \mathbb{R}^n naturally embedded in the first $j + 1$ coordinates (here we need that $j = m - i < n$).
- We first construct an extension $e': X \rightarrow \mathbb{R}^n$ of $\varphi' = f'|_A$ and then the r -perturbation g' is obtained by Lemma 3.1 part 1. The extension e' is defined as constant on the single i -cell of X , that is, $e'(x_0, y)$ is put equal to the basepoint of $S^{n-1} \subseteq \mathbb{R}^n$. On the remaining m -cell $B^m \cong \{z \in \mathbb{R}^m : |z| \leq 1\}$ of X we define $e'(z) := |z|e'(z/|z|)$, where each point z is identified with a point of X via the characteristic map $\Psi_1: B^m \rightarrow X$ of the m -cell B^m .¹⁷

By definition the only root of g' is the single point $\Psi_1(0)$ of the interior of X . Therefore $U_j(f, 1) \cong 0$. Note that the role of $\Psi_1(0)$ could be played by an arbitrary point in the interior of X .¹⁸

The zero set $g^{-1}(0) = \{(x, y) : |y| = r \text{ and } \mu(y/|y|) = x\}$ is by definition homeomorphic to the pullback (i.e., a limit) of the diagram

$$\begin{array}{ccc}
 & S^{i-1} & \\
 & \downarrow \mu & \\
 S^j & \xrightarrow{\iota} & S^{n-1}
 \end{array} \tag{6}$$

where ι is the equatorial embedding, i.e., sends each element x to $(x, 0, 0, \dots)$. In plain words, the zero set is the μ -preimage of the equatorial j -subsphere of S^{n-1} . We will prove that under our assumptions on dimensions, this is the $(m - n)$ -sphere S^{m-n} . Then from $m - n > m - i = j$ it will follow that $H_j(g^{-1}(0)) \cong 0$ which proves Theorem 1.5.

The topology of the pullback is particularly easy to see in the case when $j = n - 1$ and ι is the identity. There it is simply the domain of μ , that is, S^{i-1} where $i - 1 = m - j - 1 = m - n$.

In the general case, the only additional tool we use to identify the pullback is the Freudenthal suspension theorem. The pullback is homeomorphic to the μ -preimage of the equatorial subsphere $S^{m-i} \subseteq S^{n-1}$. By Freudenthal suspension theorem μ is homotopic to an iterated suspension $\Sigma^a \eta$ for some $\eta: S^{i-1-a} \rightarrow S^{n-1-a}$ assuming $i - 1 - a \leq 2(n - 1 - a) - 1$. We want to choose a so that $n - 1 - a = m - i$ and thus images $\text{Im}(\eta) = S^{n-1-a}$ and $\text{Im}(\iota) = S^j \subseteq S^{n-1}$ coincide (since $j = m - i$ by definition). The last inequality with the choice $a = n - 1 - m + i$ is equivalent to the bound $i \leq (m + n - 1)/2$ from the hypotheses of

¹⁷ Thus the formal definition is $e'(\Psi_1(z)) := |z|e'(\Psi_1(z)/|z|)$.

¹⁸ With more effort we could show that for any point z of X there is an r -perturbation of f' with z being its only zero point.

the theorem. In our example, we may have chosen f in such a way that $\mu = \Sigma^a \eta$. But even for the choices of μ only homotopic to $\Sigma^a \eta$ we could have changed f on a neighborhood of ∂K by a suitable homotopy. To finish the proof we use the fact that, by the definition of suspension, the μ -preimage of $S^{m-i} \subseteq S^{m-1}$ is identical to the η -preimage of S^{m-i} , that is $S^{i-1-j} = S^{m-n}$.

Difference between $Z_r(f)$ and $Z_r(f')$. Because the map μ is homotopically nontrivial, the zero set of each extension $e: X \rightarrow \mathbb{R}^n$ of $f|_A$ intersects each “section” $\{x\} \times B^i$ of X . By Lemma 3.1 part 2 (or [10, Lemma 3.3]) applied to each restriction $f|_{\{x\} \times B^i}$, the same holds for r -perturbations g of f as well. In other words, the formula “for each $x \in S^j$ there is $y \in B^i$ such that $f(x, y) = 0$ ” is *satisfied robustly*, that is

$$\forall Z \in Z_r(f) : \forall x \in S^j : \exists y \in B^i : (x, y) \in Z$$

is satisfied. The above formula is obviously not true for f' as can be seen on the r -perturbations g' . In particular, for every $r \in (0, 1]$ the family $Z_r(f')$ contains a singleton. ◀

As an example of another relevant property of $Z_r(f)$ not captured by the well groups, we mention the following. For any given $u: K \rightarrow \mathbb{R}$, we may want to know what is the *r-robust maximum of u over the zero set of f* , i.e., $\inf_{Z \in Z_r(f)} \max_{z \in Z} u(z)$. Let, for instance, $u(x, y) = u(x)$ depend on the first coordinate only. Then the r -robust maximum for f is equal to $\max_{x \in S^j} u(x)$ as follows from the discussion in the previous paragraph. On the other hand, the r -robust maximum for f' is equal to $\min_x u(x)$ and is attained in g' when we set the value $\Psi_1(0) := (\arg \min_{x \in S^j} u(x), 0)$ from the proof above. This holds for r arbitrarily small. The robust optima constitutes another and, in our opinion, practically relevant quantity whose approximation cannot be derived from well groups.

Acknowledgements. We are grateful to Ryan Budnay, Martin Čadek, Marek Filakovský, Tom Goodwillie, Amit Patel, Martin Tancer, Lukáš Vokřínek and Uli Wagner for useful discussions.

References

- 1 G. E. Alefeld, F. A. Potra, and Z. Shen. On the existence theorems of kantorovich, moore and miranda. Technical Report 01/04, Institut für Wissenschaftliches Rechnen und Mathematische Modellbildung, 2001.
- 2 A. Ben-Tal, L.E. Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.
- 3 P. Bendich, H. Edelsbrunner, D. Morozov, and A. Patel. Homology and robustness of level and interlevel sets. *Homology, Homotopy and Applications*, 15(1):51–72, 2013.
- 4 G. Carlsson. Topology and data. *Bull. Amer. Math. Soc. (N.S.)*, 46(2):255–308, 2009.
- 5 F. Chazal, A. Patel, and P. Škraba. Computing the robustness of roots. *Applied Mathematics Letters*, 25(11):1725 — 1728, November 2012.
- 6 P. Collins. Computability and representations of the zero set. *Electron. Notes Theor. Comput. Sci.*, 221:37–43, December 2008.
- 7 H. Edelsbrunner and J. L. Harer. *Computational topology*. American Mathematical Society, Providence, RI, 2010.
- 8 H. Edelsbrunner, D. Morozov, and A. Patel. Quantifying transversality by measuring the robustness of intersections. *Foundations of Computational Mathematics*, 11(3):345–361, 2011.

- 9 Herbert Edelsbrunner, Dmitriy Morozov, and Amit Patel. Quantifying transversality by measuring the robustness of intersections. *Foundations of Computational Mathematics*, 11(3):345–361, 2011.
- 10 P. Franek and M. Krčál. Robust satisfiability of systems of equations. In *Proc. Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA)*, 2014. Extended version accepted to Journal of ACM. Preprint in arXiv:1402.0858.
- 11 P. Franek and M. Krčál. On computability and triviality of well groups, 2015. Preprint arXiv:1501.03641v2.
- 12 P. Franek, S. Ratschan, and P. Zgliczynski. Quasi-decidability of a fragment of the analytic first-order theory of real numbers, 2012. Preprint in arXiv:1309.6280.
- 13 A. Frommer and B. Lang. Existence tests for solutions of nonlinear equations using Borsuk’s theorem. *SIAM Journal on Numerical Analysis*, 43(3):1348–1361, 2005.
- 14 F. Goudail and P. Réfrégier. *Statistical Image Processing Techniques for Noisy Images: An Application-Oriented Approach*. Kluwer Academic / Plenum Publishers, 2004.
- 15 A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, 2001.
- 16 N.J. Higham. *Accuracy and Stability of Numerical Algorithms: Second Edition*. Society for Industrial and Applied Mathematics, 2002.
- 17 R. B. Kearfott. On existence and uniqueness verification for non-smooth functions. *Reliable Computing*, 8(4):267–282, 2002.
- 18 S. Mardešić. Thirty years of shape theory. *Mathematical Communications*, 2(1):1–12, 1997.
- 19 A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge Univ. Press, Cambridge, 1990.
- 20 P. Franek, M. Krčál. Cohomotopy groups capture robust properties of zero sets. Manuscript in preparation, 2014.
- 21 V. V. Prasolov. *Elements of Homology Theory*. Graduate Studies in Mathematics. American Mathematical Society, 2007.
- 22 P. Škraba, B. Wang, Ch. Guoning, and P. Rosen. 2D vector field simplification based on robustness, 2014. to appear in IEEE Pacific Visualization (PacificVis).
- 23 M. Čadek, M. Krčál, J. Matoušek, F. Sergeraert, L. Vokřínek, and U. Wagner. Computing all maps into a sphere. *J. ACM*, 61(3):17:1–17:44, June 2014.
- 24 A.H. Wallace. *Algebraic Topology: Homology and Cohomology*. Dover Books on Mathematics Series. Dover Publications, 2007.
- 25 J.H.C. Whitehead. On the theory of obstructions. *Annals of Mathematics*, pages 68–84, 1951.

Geometric Inference on Kernel Density Estimates*

Jeff M. Phillips¹, Bei Wang², and Yan Zheng¹

¹ School of Computing, University of Utah, USA

² Scientific Computing and Imaging Institute, University of Utah, USA

Abstract

We show that geometric inference of a point cloud can be calculated by examining its kernel density estimate with a Gaussian kernel. This allows one to consider kernel density estimates, which are robust to spatial noise, subsampling, and approximate computation in comparison to raw point sets. This is achieved by examining the sublevel sets of the *kernel distance*, which isomorphically map to superlevel sets of the kernel density estimate. We prove new properties about the kernel distance, demonstrating stability results and allowing it to inherit reconstruction results from recent advances in distance-based topological reconstruction. Moreover, we provide an algorithm to estimate its topology using weighted Vietoris-Rips complexes.

1998 ACM Subject Classification F.2.2: Nonnumerical Algorithms and Problems

Keywords and phrases topological data analysis, kernel density estimate, kernel distance

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.857

1 Introduction

Geometry and topology have become essential tools in modern data analysis: geometry to handle spatial noise and topology to identify the core structure. Topological data analysis (TDA) has found applications spanning protein structure analysis [24, 40] to heart modeling [32] to leaf science [49], and is the central tool of identifying quantities like connectedness, cyclic structure, and intersections at various scales. Yet it can suffer from spatial noise in data, particularly outliers.

When analyzing point cloud data, classically these approaches consider α -shapes [23], where each point is replaced with a ball of radius α , and the union of these balls is analyzed. More recently a distance function interpretation [8] has become more prevalent where the union of α -radius balls can be replaced by the sublevel set (at value α) of the Hausdorff distance to the point set. Moreover, the theory can be extended to other distance functions to the point sets, including the *distance-to-a-measure* [12] which is more robust to noise.

This has more recently led to statistical analysis of TDA. These results show not only robustness in the function reconstruction, but also in the topology it implies about the underlying dataset. This work often operates on persistence diagrams which summarize the persistence (difference in function values between appearance and disappearance) of all homological features in single diagram. A variety of work has developed metrics on these diagrams and probability distributions over them [43, 55], and robustness and confidence intervals on their landscapes [6, 30, 15, 16]). It is now more clear than ever, that these works are most appropriate when the underlying function is robust to noise, e.g., the distance-to-a-measure [12].

* Thanks to supported to JMP by NSF CCF-1350888, IIS-1251019, and ACI-1443046, and for BW by INL 00115847 via DE-AC07ID14517, DOE NETL DEEE0004449, DOE DEFC0206ER25781, DOE DE-SC0007446, and NSF 0904631.



© Jeff M. Phillips, Bei Wang, and Yan Zheng;
licensed under Creative Commons License CC-BY

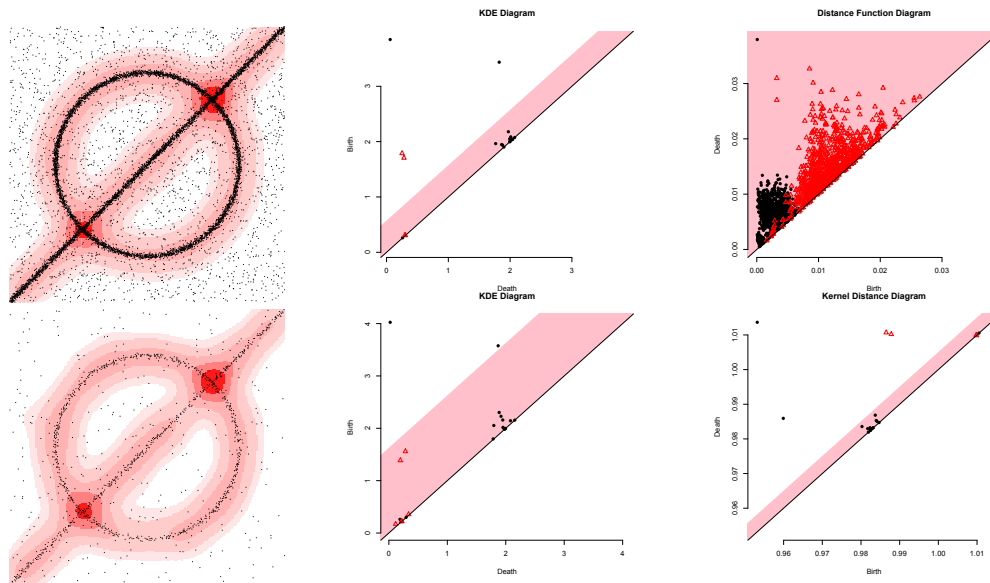
31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 857–871



Leibniz International Proceedings in Informatics

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



■ **Figure 1** Example with 10,000 points in $[0, 1]^2$ generated on a circle or line with $N(0, 0.005)$ noise; 25% of points are uniform background noise. The generating function is reconstructed with KDE with $\sigma = 0.05$ (upper left), and its persistence diagram based on the superlevel set filtration is shown (upper middle). A coreset [58] of the same dataset with only 1,384 points (lower left) and persistence diagram (lower middle) are shown, again using KDE. This associated confidence interval contains the dimension 1 homology features (red triangles) suggesting they are noise; this is because it models data as iid – but the coreset data is not iid, it subsamples more intelligently. We also show persistence diagrams of the original data based on the sublevel set filtration of the standard distance function (upper right, with no useful features due to noise) and the kernel distance (lower right).

A very recent addition to this progression is the new TDA package for R [29]; it includes built in functions to analyze point sets using Hausdorff distance, distance-to-a-measure, k -nearest neighbor density estimators, kernel density estimates, and kernel distance. The example in Figure 1 used this package to generate persistence diagrams. While, the stability of the Hausdorff distance is classic [8, 23], and the distance-to-a-measure [12] and k -nearest neighbor distances have been shown robust to various degrees [4], this paper is the first to analyze the stability of kernel density estimates and the kernel distance in the context of geometric inference. Some recent manuscripts show related results. Bobrowski *et al.* [5] consider kernels with finite support, and describe approximate confidence intervals on the superlevel sets, which recover approximate persistence diagrams. Chazal *et al.* [14] explore the robustness of the kernel distance in bootstrapping-based analysis.

In particular, we show that the kernel distance and kernel density estimates, using the Gaussian kernel, inherit some reconstruction properties of distance-to-a-measure, that these functions can also be approximately reconstructed using weighted (Vietoris-)Rips complexes [7], and that under certain regimes can infer homotopy of compact sets. Moreover, we show further robustness advantages of the kernel distance and kernel density estimates, including that they possess small coresets [45, 58] for persistence diagrams and inference.

1.1 Kernels, Kernel Density Estimates, and Kernel Distance

A *kernel* is a non-negative similarity measure $K : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^+$; more similar points have higher value. For any fixed $p \in \mathbb{R}^d$, a kernel $K(p, \cdot)$ can be normalized to be a probability

distribution; that is $\int_{x \in \mathbb{R}^d} K(p, x) dx = 1$. For the purposes of this article we focus on the Gaussian kernel defined as $K(p, x) = \sigma^2 \exp(-\|p - x\|^2 / 2\sigma^2)$.¹

A *kernel density estimate* [53, 50, 21, 22] is a way to estimate a continuous distribution function over \mathbb{R}^d for a finite point set $P \subset \mathbb{R}^d$; they have been studied and applied in a variety of contexts, for instance, under subsampling [45, 58, 2], motion planning [48], multimodality [52, 25], and surveillance [28], road reconstruction [3]. Specifically,

$$\text{KDE}_P(x) = \frac{1}{|P|} \sum_{p \in P} K(p, x).$$

The *kernel distance* [37, 33, 38, 46] (also called *current distance* or *maximum mean discrepancy*) is a metric [44, 54] between two point sets P, Q (as long as the kernel used is characteristic [54], a slight restriction of being positive definite [1, 57], this includes the Gaussian and Laplace kernels). Define a similarity between the two point sets as

$$\kappa(P, Q) = \frac{1}{|P|} \frac{1}{|Q|} \sum_{p \in P} \sum_{q \in Q} K(p, q).$$

Then the kernel distance between two point sets is defined as

$$D_K(P, Q) = \sqrt{\kappa(P, P) + \kappa(Q, Q) - 2\kappa(P, Q)}.$$

When we let point set Q be a single point x , then $\kappa(P, x) = \text{KDE}_P(x)$.

Kernel density estimates applies to any measure μ (on \mathbb{R}^d) as $\text{KDE}_\mu(x) = \int_{p \in \mathbb{R}^d} K(p, x) d\mu(p)$. The similarity between two measures is $\kappa(\mu, \nu) = \int_{(p, q) \in \mathbb{R}^d \times \mathbb{R}^d} K(p, q) d\mathbf{m}_{\mu, \nu}(p, q)$, where $\mathbf{m}_{\mu, \nu}$ is the product measure of μ and ν ($\mathbf{m}_{\mu, \nu} := \mu \otimes \nu$), and then the kernel distance between two measures μ and ν is still a metric, defined as $D_K(\mu, \nu) = \sqrt{\kappa(\mu, \mu) + \kappa(\nu, \nu) - 2\kappa(\mu, \nu)}$. When the measure ν is a Dirac measure at x ($\nu(q) = 0$ for $x \neq q$, but integrates to 1), then $\kappa(\mu, x) = \text{KDE}_\mu(x)$. Given a finite point set $P \subset \mathbb{R}^d$, we can work with the empirical measure μ_P defined as $\mu_P = \frac{1}{|P|} \sum_{p \in P} \delta_p$, where δ_p is the Dirac measure on p , and $D_K(\mu_P, \mu_Q) = D_K(P, Q)$.

If K is positive definite, it is said to have the reproducing property [1, 57]. This implies that $K(p, x)$ is an inner product in some reproducing kernel Hilbert space (RKHS) \mathcal{H}_K . Specifically, there is a lifting map $\phi : \mathbb{R}^d \rightarrow \mathcal{H}_K$ so that $K(p, x) = \langle \phi(p), \phi(x) \rangle_{\mathcal{H}_K}$, and moreover the entire set P can be represented as $\Phi(P) = \sum_{p \in P} \phi(p)$, which is a single element of \mathcal{H}_K and has a norm $\|\Phi(P)\|_{\mathcal{H}_K} = \sqrt{\kappa(P, P)}$. A single point $x \in \mathbb{R}^d$ also has a norm $\|\phi(x)\|_{\mathcal{H}_K} = \sqrt{K(x, x)}$ in this space.

1.2 Geometric Inference and Distance to a Measure: A Review

Given an unknown compact set $S \subset \mathbb{R}^d$ and a finite point cloud $P \subset \mathbb{R}^d$ that comes from S under some process, geometric inference aims to recover topological and geometric properties of S from P . The offset-based (and more generally, the distance function-based) approach for geometric inference reconstructs a geometric and topological approximation of S by offsets from P (e.g. [10, 11, 12, 17, 18]).

Given a compact set $S \subset \mathbb{R}^d$, we can define a *distance function* f_S to S ; a common example is $f_S(x) = \inf_{y \in S} \|x - y\|$. The offsets of S are the sublevel sets of f_S , denoted $(S)^r = f_S^{-1}([0, r])$. Now an approximation of S by another compact set $P \subset \mathbb{R}^d$ (e.g. a

¹ The choice of coefficient σ^2 is not the standard normalization, but it is perfectly valid as it scales everything by a constant. It has the property that $\sigma^2 - K(p, x) \approx \|p - x\|^2 / 2$ for $\|p - x\|$ small.

finite point cloud) can be quantified by the Hausdorff distance $d_H(S, P) := \|f_S - f_P\|_\infty = \inf_{x \in \mathbb{R}^d} |f_S(x) - f_P(x)|$ of their distance functions. The intuition behind the inference of topology is that if $d_H(S, P)$ is small, thus f_S and f_P are close, and subsequently, S , $(S)^r$ and $(P)^r$ carry the same topology for an appropriate scale r . In other words, to compare the topology of offsets $(S)^r$ and $(P)^r$, we require Hausdorff stability with respect to their distance functions f_S and f_P . An example of an offset-based topological inference result is formally stated as follows (as a particular version of the reconstruction Theorem 4.6 in [11]), where the *reach* of a compact set S , $\text{reach}(S)$, is defined as the minimum distance between S and its medial axis [42].

► **Theorem 1** (Reconstruction from f_P [11]). *Let $S, P \subset \mathbb{R}^d$ be compact sets such that $\text{reach}(S) > R$ and $\varepsilon := d_H(S, P) < R/17$. Then $(S)^\eta$ and $(P)^r$ are homotopy equivalent for sufficiently small η (e.g., $0 < \eta < R$) if $4\varepsilon \leq r < R - 3\varepsilon$.*

Here $\eta < R$ ensures that the topological properties of $(S)^\eta$ and $(S)^r$ are the same, and the ε parameter ensures $(S)^r$ and $(P)^r$ are close. Typically ε is tied to the density with which a point cloud P is sampled from S .

For function $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^+$ to be *distance-like* it should satisfy the following properties:

- (D1) ϕ is 1-Lipschitz: For all $x, y \in \mathbb{R}^d$, $|\phi(x) - \phi(y)| \leq \|x - y\|$.
 - (D2) ϕ^2 is 1-semiconcave: The map $x \in \mathbb{R}^d \mapsto (\phi(x))^2 - \|x\|^2$ is concave.
 - (D3) ϕ is proper: $\phi(x)$ tends to the infimum of its domain (e.g., ∞) as x tends to infinity.
- In addition to the Hausdorff stability property stated above, as explained in [12], f_S is distance-like. These three properties are paramount for geometric inference (e.g. [11, 41]).

(D1) ensures that f_S is differentiable almost everywhere and the medial axis of S has zero d -volume [12]; and (D2) is a crucial technical tool, e.g., in proving the existence of the flow of the gradient of the distance function for topological inference [11].

Distance to a measure. Given a probability measure μ on \mathbb{R}^d and a parameter $m_0 > 0$ smaller than the total mass of μ , the *distance to a measure* $d_{\mu, m_0}^{\text{CCM}} : \mathbb{R}^n \rightarrow \mathbb{R}^+$ [12] is defined for any point $x \in \mathbb{R}^d$ as

$$d_{\mu, m_0}^{\text{CCM}}(x) = \left(\frac{1}{m_0} \int_{m=0}^{m_0} (\delta_{\mu, m}(x))^2 dm \right)^{1/2}, \quad \text{where } \delta_{\mu, m}(x) = \inf \{ r > 0 : \mu(\bar{B}_r(x)) \geq m \},$$

and where $B_r(x)$ is a ball of radius r centered at x and $\bar{B}_r(x)$ is its closure. It has been shown in [12] that $d_{\mu, m_0}^{\text{CCM}}$ is a distance-like function (satisfying (D1), (D2), and (D3)), and:

- (M4) [Stability] For probability measures μ and ν on \mathbb{R}^d and $m_0 > 0$, then $\|d_{\mu, m_0}^{\text{CCM}} - d_{\nu, m_0}^{\text{CCM}}\|_\infty \leq \frac{1}{\sqrt{m_0}} W_2(\mu, \nu)$, where W_2 is the *Wasserstein distance* [56].

Given a point set P , the sublevel sets of $d_{\mu_P, m_0}^{\text{CCM}}$ can be described as the union of balls [35], and then one can algorithmically estimate the topology (e.g., persistence diagram) with weighted alpha-shapes [35] and weighted Rips complexes [7].

1.3 Our Results

We show how to estimate the topology (e.g., approximate persistence diagrams, infer homotopy of compact sets) using superlevel sets of the kernel density estimate of a point set P . We accomplish this by showing that a similar set of properties hold for the kernel distance with respect to a measure μ , (in place of distance to a measure $d_{\mu, m_0}^{\text{CCM}}$), defined as

$$d_\mu^K(x) = D_K(\mu, x) = \sqrt{\kappa(\mu, \mu) + \kappa(x, x) - 2\kappa(\mu, x)}.$$

This treats x as a probability measure represented by a Dirac mass at x . Specifically, we show d_μ^K is distance-like (it satisfies (D1), (D2), and (D3)), so it inherits reconstruction properties of d_{μ, m_0}^{CCM} . Moreover, it is stable with respect to the kernel distance:

- (K4) [Stability] If μ and ν are two measures on \mathbb{R}^d , then $\|d_\mu^K - d_\nu^K\|_\infty \leq D_K(\mu, \nu)$.

In addition, we show how to construct these topological estimates for d_μ^K using weighted Rips complexes, following power distance machinery introduced in [7].

We also describe further advantages of the kernel distance. (i) Its sublevel sets conveniently map to the superlevel sets of a kernel density estimate. (ii) It is Lipschitz with respect to the smoothing parameter σ when the input x is fixed. (iii) As σ tends to ∞ for any two probability measures μ, ν , the kernel distance is bounded by the Wasserstein distance: $\lim_{\sigma \rightarrow \infty} D_K(\mu, \nu) \leq W_2(\mu, \nu)$. (iv) It has a small coreset representation, which allows for sparse representation and efficient, scalable computation. In particular, an ε -kernel sample [38, 45, 58] Q of μ is a finite point set whose size only depends on $\varepsilon > 0$ and such that $\max_{x \in \mathbb{R}^d} |\text{KDE}_\mu(x) - \text{KDE}_{\mu_Q}(x)| = \max_{x \in \mathbb{R}^d} |\kappa(\mu, x) - \kappa(\mu_Q, x)| \leq \varepsilon$. These coresets preserve inference results and persistence diagrams.

2 Kernel Distance is Distance-Like

We prove d_μ^K satisfies (D1), (D2), and (D3); hence it is distance-like. Recall we use the σ^2 -normalized Gaussian kernel $K_\sigma(p, x) = \sigma^2 \exp(-\|p - x\|^2/2\sigma^2)$. For ease of exposition, unless otherwise noted, we will assume σ is fixed and write K instead of K_σ .

2.1 Semiconcave Property for d_μ^K

► **Lemma 2** (D2). $(d_\mu^K)^2$ is 1-semiconcave: the map $x \mapsto (d_\mu^K(x))^2 - \|x\|^2$ is concave.

Proof. Let $T(x) = (d_\mu^K(x))^2 - \|x\|^2$. The proof will show that the second derivative of T along any direction is nonpositive. We can rewrite

$$\begin{aligned} T(x) &= \kappa(\mu, \mu) + \kappa(x, x) - 2\kappa(\mu, x) - \|x\|^2 \\ &= \kappa(\mu, \mu) + \kappa(x, x) - \int_{p \in \mathbb{R}^d} (2K(p, x) + \|x\|^2) d\mu(p). \end{aligned}$$

Note that both $\kappa(\mu, \mu)$ and $\kappa(x, x)$ are absolute constants, so we can ignore them in the second derivative. Furthermore, by setting $t(p, x) = -2K(p, x) - \|x\|^2$, the second derivative of $T(x)$ is nonpositive if the second derivative of $t(p, x)$ is nonpositive for all $p, x \in \mathbb{R}^d$. First note that the second derivative of $-\|x\|^2$ is a constant -2 in every direction. The second derivative of $K(p, x)$ is symmetric about p , so we can consider the second derivative along any vector $u = x - p$,

$$\frac{d^2}{du^2} t(p, x) = 2 \left(\frac{\|u\|^2}{\sigma^2} - 1 \right) \exp \left(-\frac{\|u\|^2}{2\sigma^2} \right) - 2.$$

This reaches its maximum value at $\|u\| = \|x - p\| = \sqrt{3}\sigma$ where it is $4 \exp(-3/2) - 2 \approx -1.1$; this follows by setting the derivative of $s(y) = 2(y - 1) \exp(-y/2) - 2$ to 0, ($\frac{d}{dy} s(y) = (1/2)(3 - y) \exp(-y/2)$), substituting $y = \|u\|^2/\sigma^2$. ◀

2.2 Lipschitz Property for d_μ^K

We generalize a (folklore, see [12]) relation between semiconcave and Lipschitz functions. A function f is ℓ -semiconcave if the function $T(x) = (f(x))^2 - \ell\|x\|^2$ is concave.

► **Lemma 3.** Consider a twice-differentiable function g and a parameter $\ell \geq 1$. If $(g(x))^2$ is ℓ -semiconcave, then $g(x)$ is ℓ -Lipschitz.

We can now state the following lemma as a corollary of Lemma 2 and Lemma 3.

► **Lemma 4 (D1).** d_μ^K is 1-Lipschitz on its input.

2.3 Properness of d_μ^K

Finally, for d_μ^K to be distance-like, we need to show it is proper when its range is restricted to be less than $c_\mu := \sqrt{\kappa(\mu, \mu) + \kappa(x, x)}$. This is required for a distance-like version ([12], Proposition 4.2) of the Isotopy Lemma ([34], Proposition 1.8). Here, the value of c_μ depends on μ not on x since $\kappa(x, x) = K(x, x) = \sigma^2$.

► **Lemma 5 (D3).** d_μ^K is proper.

We delay the proof to the full version [47]. The main technical difficulty comes in mapping standard definitions and approaches for distance functions to our function d_μ^K with a restricted range. We use two more general, but equivalent definitions of a proper map and the notion of *escape to infinity*. Specifically, a sequence $\{p_i\}$ in \mathbb{X} *escapes to infinity* if for every compact set $G \subset \mathbb{X}$, there are at most finitely many values of i for which $p_i \in G$ ([39], page 46).

By the definition of properness, Lemma 5 implies that it is a closed map and its levelset at any value $a \in [0, c_\mu)$ is compact. This also means that the sublevel set of d_μ^K (for ranges $[0, a) \subset [0, c_\mu)$) is compact. Since the levelset (sublevel set) of d_μ^K corresponds to the levelset (superlevel set) of KDE_μ , we have the following corollary.

► **Corollary 6.** The superlevel sets of KDE_μ for all ranges with threshold $a > 0$, are compact.

The result in [25] shows that given a measure μ_P defined by a point set P of size n , the KDE_{μ_P} has polynomial in n modes; hence the superlevel sets of KDE_{μ_P} are compact in this setting. The above corollary is a more general statement as it holds for any measure.

3 Power Distance using Kernel Distance

A *power distance* using d_μ^K is defined with a point set $P \subset \mathbb{R}^d$ and a metric $d(\cdot, \cdot)$ on \mathbb{R}^d ,

$$f_P(\mu, x) = \sqrt{\min_{p \in P} (d(p, x)^2 + d_\mu^K(p)^2)}.$$

A point $x \in \mathbb{R}^d$ takes the distance under $d(p, x)$ to the closest $p \in P$, plus a weight from $d_\mu^K(p)$; thus a sublevel set of $f_P(\mu, \cdot)$ is defined by a union of balls. We consider a particular choice of the distance $d(p, x) := D_K(p, x)$ which leads to a kernel version of power distance

$$f_P^K(\mu, x) = \sqrt{\min_{p \in P} (D_K(p, x)^2 + d_\mu^K(p)^2)}.$$

In Section 4.2 we use $f_P^K(\mu, x)$ to adapt the construction introduced in [7] to approximate the persistence diagram of the sublevel sets of d_μ^K , using a weighted Rips filtration of $f_P^K(\mu, x)$.

Given a measure μ , let $p_+ = \arg \max_{q \in \mathbb{R}^d} \kappa(\mu, q)$, and let $P_+ \subset \mathbb{R}^d$ be a point set that contains p_+ . We show below, in Theorem 11 and Theorem 8, that $\frac{1}{\sqrt{2}} d_\mu^K(x) \leq f_{P_+}^K(\mu, x) \leq \sqrt{14} d_\mu^K(x)$. However, constructing p_+ exactly seems quite difficult.

Now consider an empirical measure μ_P defined by a point set P . We show (in the full version [47]) how to construct a point \hat{p}_+ (that approximates p_+) such that $D_K(P, \hat{p}_+) \leq (1 + \delta)D_K(P, p_+)$ for any $\delta > 0$. For a point set P , the *median concentration* Λ_P is a radius such that no point $p \in P$ has more than half of the points of P within Λ_P , and the *spread* β_P is the ratio between the longest and shortest pairwise distances. The runtime is polynomial in n and $1/\delta$ assuming β_P is bounded, and that σ/Λ_P and d are constants.

Then we consider $\hat{P}_+ = P \cup \{\hat{p}_+\}$, where \hat{p}_+ is found with $\delta = 1/2$ in the above construction. Then we can provide the following multiplicative bound, proven in Theorem 12. The lower bound holds independent of the choice of P as shown in Theorem 8.

► **Theorem 7.** For any point set $P \subset \mathbb{R}^d$ and point $x \in \mathbb{R}^d$, with empirical measure μ_P defined by P , then $\frac{1}{\sqrt{2}}d_{\mu_P}^K(x) \leq f_{\hat{P}_+}^K(\mu_P, x) \leq \sqrt{71}d_{\mu_P}^K(x)$.

3.1 Kernel Power Distance for a Measure μ

First consider the case for a kernel power distance $f_P^K(\mu, x)$ where μ is an arbitrary measure.

► **Theorem 8.** For measure μ , point set $P \subset \mathbb{R}^d$, and $x \in \mathbb{R}^d$, $D_K(\mu, x) \leq \sqrt{2}f_P^K(\mu, x)$.

Proof. Let $p = \arg \min_{q \in P} (D_K(q, x)^2 + D_K(\mu, q)^2)$. Then we can use the triangle inequality and $(D_K(\mu, p) - D_K(p, x))^2 \geq 0$ to show

$$D_K(\mu, x)^2 \leq (D_K(\mu, p) + D_K(p, x))^2 \leq 2(D_K(\mu, p)^2 + D_K(p, x)^2) = 2f_P^K(\mu, x)^2. \quad \blacktriangleleft$$

► **Lemma 9.** For measure μ , point set $P \subset \mathbb{R}^d$, point $p \in P$, and point $x \in \mathbb{R}^d$ then $f_P^K(\mu, x)^2 \leq 2D_K(\mu, x)^2 + 3D_K(p, x)^2$.

Proof. Again, we can reach this result with the triangle inequality.

$$\begin{aligned} f_P^K(\mu, x)^2 &\leq D_K(\mu, p)^2 + D_K(p, x)^2 \\ &\leq (D_K(\mu, x) + D_K(p, x))^2 + D_K(p, x)^2 \\ &\leq 2D_K(\mu, x)^2 + 3D_K(p, x)^2. \end{aligned} \quad \blacktriangleleft$$

Recall the definition of a point $p_+ = \arg \max_{q \in \mathbb{R}^d} \kappa(\mu, q)$.

► **Lemma 10.** For any measure μ and point $x, p_+ \in \mathbb{R}^d$ we have $D_K(p_+, x) \leq 2D_K(\mu, x)$.

Proof. Since x is a point in \mathbb{R}^d , $\kappa(\mu, x) \leq \kappa(\mu, p_+)$ and thus $D_K(\mu, x) \geq D_K(\mu, p_+)$. Then by triangle inequality of D_K to see that $D_K(p_+, x) \leq D_K(\mu, x) + D_K(\mu, p_+) \leq 2D_K(\mu, x)$. ◀

► **Theorem 11.** For any measure μ in \mathbb{R}^d and any point $x \in \mathbb{R}^d$, using the point $p_+ = \arg \max_{q \in \mathbb{R}^d} \kappa(\mu, q)$ then $f_{\{p_+\}}^K(\mu, x) \leq \sqrt{14}D_K(\mu, x)$.

Proof. Combine Lemma 9 and Lemma 10 as

$$f_{\{p_+\}}^K(\mu, x)^2 \leq 2D_K(\mu, x)^2 + 3D_K(p_+, x)^2 \leq 2D_K(\mu, x)^2 + 3(4D_K(\mu, x)^2) = 14D_K(\mu, x)^2. \quad \blacktriangleleft$$

We now need two properties of the point set P to reach our bound, namely, the spread β_P and the median concentration Λ_P . Typically $\log(\beta_P)$ is not too large, and it makes sense to choose σ so $\sigma/\Lambda_P \leq 1$, or at least $\sigma/\Lambda_P = O(1)$.

► **Theorem 12.** Consider any point set $P \subset \mathbb{R}^d$ of size n , with measure μ_P , spread β_P , and median concentration Λ_P . We can construct a point set $\hat{P}_+ = P \cup \hat{p}_+$ in $O(n^2((\sigma/\Lambda_P)\delta)^d + \log(\beta))$ time such that for any point x , $f_{\hat{P}_+}^K(\mu_P, x) \leq \sqrt{71}D_K(\mu_P, x)$.

Proof. We use a result from the full version [47] to find a point \hat{p}_+ such that $D_K(P, \hat{p}_+) \leq (3/2)D_K(P, p_+)$ in the stated runtime. Thus for any $x \in \mathbb{R}^d$, using the triangle inequality

$$\begin{aligned} D_K(\hat{p}_+, x) &\leq D_K(\hat{p}_+, p_+) + D_K(p_+, x) \leq D_K(\mu_P, \hat{p}_+) + D_K(\mu_P, p_+) + D_K(p_+, x) \\ &\leq (5/2)D_K(\mu_P, p_+) + D_K(p_+, x). \end{aligned}$$

Now combine this with Lemma 9 and Lemma 10 as

$$\begin{aligned} f_{\hat{P}_+}^K(\mu_P, x)^2 &\leq 2D_K(\mu_P, x)^2 + 3D_K(\hat{p}_+, x)^2 \\ &\leq 2D_K(\mu_P, x)^2 + 3((5/2)D_K(\mu_P, x) + D_K(p_+, x))^2 \\ &\leq 2D_K(\mu_P, x)^2 + 3((25/4) + (5/2))D_K(\mu_P, x)^2 + (1 + 5/2)D_K(p_+, x)^2 \\ &= (113/4)D_K(\mu_P, x)^2 + (21/2)D_K(p_+, x)^2 \\ &\leq (113/4)D_K(\mu_P, x)^2 + (21/2)(4D_K(\mu_P, x)^2) < 71D_K(\mu_P, x)^2. \quad \blacktriangleleft \end{aligned}$$

4 Reconstruction and Topological Estimation using Kernel Distance

Now applying distance-like properties from Section 2 and the power distance properties of Section 3 we can apply known reconstruction results to the kernel distance.

4.1 Homotopy Equivalent Reconstruction using d_μ^K

We have shown that the kernel distance function d_μ^K is a distance-like function. Therefore the reconstruction theory for a distance-like function [12] holds in the setting of d_μ^K . We state the following two corollaries for completeness, whose proofs follow from the proofs of Proposition 4.2 and Theorem 4.6 in [12]. Before their formal statement, we need some notation adapted from [12] to make these statements precise. Let $\phi : \mathbb{R}^d \rightarrow \mathbb{R}^+$ be a distance-like function. A point $x \in \mathbb{R}^d$ is an α -critical point if $\phi^2(x+h) \leq \phi^2(x) + 2\alpha\|h\|\phi(x) + \|h\|^2$ with $\alpha \in [0, 1]$, $\forall h \in \mathbb{R}^d$. Let $(\phi)^r = \{x \in \mathbb{R}^d \mid \phi(x) \leq r\}$ denote the sublevel set of ϕ , and let $(\phi)^{[r_1, r_2]} = \{x \in \mathbb{R}^d \mid r_1 \leq \phi(x) \leq r_2\}$ denote all points at levels in the range $[r_1, r_2]$. For $\alpha \in [0, 1]$, the α -reach of ϕ is the maximum r such that $(\phi)^r$ has no α -critical point, denoted as $\text{reach}_\alpha(\phi)$. When $\alpha = 1$, reach_1 coincides with reach introduced in [31].

► **Theorem 13** (Isotopy lemma on d_μ^K). *Let $r_1 < r_2$ be two positive numbers such that d_μ^K has no critical points in $(d_\mu^K)^{[r_1, r_2]}$. Then all the sublevel sets $(d_\mu^K)^r$ are isotopic for $r \in [r_1, r_2]$.*

► **Theorem 14** (Reconstruction on d_μ^K). *Let d_μ^K and d_ν^K be two kernel distance functions such that $\|d_\mu^K - d_\nu^K\|_\infty \leq \varepsilon$. Suppose $\text{reach}_\alpha(d_\mu^K) \geq R$ for some $\alpha > 0$. Then $\forall r \in [4\varepsilon/\alpha^2, R - 3\varepsilon]$, and $\forall \eta \in (0, R)$, the sublevel sets $(d_\mu^K)^\eta$ and $(d_\nu^K)^r$ are homotopy equivalent for $\varepsilon \leq R/(5 + 4/\alpha^2)$.*

4.2 Constructing Topological Estimates using d_μ^K

In order to actually construct a topological estimate using the kernel distance d_μ^K , one needs to be able to compute quantities related to its sublevel sets, in particular, to compute the persistence diagram of the sub-level sets filtration of d_μ^K . Now we describe such tools needed for the kernel distance based on machinery recently developed by Buchet et al. [7], which shows how to approximate the persistent homology of distance-to-a-measure for any metric space via a power distance construction. Then using similar constructions, we can use the weighted Rips filtration to approximate the persistence diagram of the kernel distance.

To state our results, first we require some technical notions and assume basic knowledge on persistent homology (see [26, 27] for a readable background). Given a metric space \mathbb{X} with the distance $d_{\mathbb{X}}(\cdot, \cdot)$, a set $P \subseteq \mathbb{X}$ and a function $w : P \rightarrow \mathbb{R}$, the (general) *power distance* f associated with (P, w) is defined as $f(x) = \sqrt{\min_{p \in P} (d_{\mathbb{X}}(p, x)^2 + w(p)^2)}$. Now given the set (P, w) and its corresponding power distance f , one could use the weighted Rips filtration to approximate the persistence diagram of w . Consider the sublevel set of f , $f^{-1}((-\infty, \alpha])$. It is the union of balls centered at points $p \in P$ with radius $r_p(\alpha) = \sqrt{\alpha^2 - w(p)^2}$ for each p . The weighted Čech complex $C_{\alpha}(P, w)$ for parameter α is the union of simplices s such that $\bigcap_{p \in s} B(p, r_p(\alpha)) \neq \emptyset$. The weighted Rips complex $R_{\alpha}(P, w)$ for parameter α is the maximal complex whose 1-skeleton is the same as $C_{\alpha}(P, w)$. The corresponding weighted Rips filtration is denoted as $\{R_{\alpha}(P, w)\}$.

Setting $w := d_{\mu_P}^K$ and given point set \hat{P}_+ described in Section 3, consider the weighted Rips filtration $\{R_{\alpha}(\hat{P}_+, d_{\mu_P}^K)\}$ based on the kernel power distance, $f_{\hat{P}_+}^K$. We view the persistence diagrams on a logarithmic scale, that is, we change coordinates of points following the mapping $(x, y) \mapsto (\ln x, \ln y)$. d_B^{\ln} denotes the corresponding bottleneck distance between persistence diagrams. We show in the full version [47] that persistence diagrams $\text{Dgm}(d_{\mu_P}^K)$ and $\text{Dgm}(\{R_{\alpha}(\hat{P}_+, d_{\mu_P}^K)\})$ follow technical tameness conditions and are well-defined. We now state a corollary of Theorem 7.

► **Corollary 15.** *The weighted Rips filtration $\{R_{\alpha}(\hat{P}_+, d_{\mu_P}^K)\}$ can be used to approximate the persistence diagram of $d_{\mu_P}^K$ such that $d_B^{\ln}(\text{Dgm}(d_{\mu_P}^K), \text{Dgm}(\{R_{\alpha}(\hat{P}_+, d_{\mu_P}^K)\})) \leq \ln(2\sqrt{71})$.*

Proof. To prove that two persistence diagrams are close, one could prove that their filtration are interleaved [9], that is, two filtrations $\{U_{\alpha}\}$ and $\{V_{\alpha}\}$ are ε -interleaved if for any α , $U_{\alpha} \subseteq V_{\alpha+\varepsilon} \subseteq U_{\alpha+2\varepsilon}$. The results of Theorem 7 implies an $\sqrt{71}$ multiplicative interleaving. Therefore for any $\alpha \in \mathbb{R}$,

$$(d_{\mu_P}^K)^{-1}((-\infty, \alpha]) \subset (f_{\hat{P}_+}^K)^{-1}((-\infty, \sqrt{2}\alpha]) \subset (d_{\mu_P}^K)^{-1}((-\infty, \sqrt{71}\sqrt{2}\alpha]).$$

On a logarithmic scale (by taking the natural log of both sides), such interleaving becomes additive,

$$\ln d_{\mu_P}^K - \sqrt{2} \leq \ln f_{\hat{P}_+}^K \leq \ln d_{\mu_P}^K + \sqrt{71}.$$

Theorem 4 of [13] implies

$$d_B^{\ln}(\text{Dgm}(d_{\mu_P}^K), \text{Dgm}(f_{\hat{P}_+}^K)) \leq \sqrt{71}.$$

In addition, by the Persistent Nerve Lemma ([19], Theorem 6 of [51], an extension of the Nerve Theorem [36]), the sublevel sets filtration of d_{μ}^K , which correspond to unions of balls of increasing radius, has the same persistent homology as the nerve filtration of these balls (which, by definition, is the Čech filtration). Finally, there exists a multiplicative interleaving between weighted Rips and Čech complexes (Proposition 31 of [13]), $C_{\alpha} \subseteq R_{\alpha} \subseteq C_{2\alpha}$. We then obtain the following bounds on persistence diagrams,

$$d_B^{\ln}(\text{Dgm}(f_{\hat{P}_+}^K), \text{Dgm}(\{R_{\alpha}(P_+, d_{\mu_P}^K)\})) \leq \ln(2).$$

We use triangle inequality to obtain the final result:

$$d_B^{\ln}(\text{Dgm}(d_{\mu_P}^K), \text{Dgm}(\{R_{\alpha}(P_+, d_{\mu_P}^K)\})) \leq \ln(2\sqrt{71}). \quad \blacktriangleleft$$

Based on Corollary 15, we have an algorithm that approximates the persistent homology of the sublevel set filtration of d_{μ}^K by constructing the weighted Rips filtration corresponding to the kernel-based power distance and computing its persistent homology.

4.3 Distance to the Support of a Measure vs. Kernel Distance

Suppose μ is a uniform measure on a compact set S in \mathbb{R}^d . We now compare the kernel distance d_μ^K with the distance function f_S to the support S of μ . We show how d_μ^K approximates f_S , and thus allows one to infer geometric properties of S from samples from μ .

A generalized gradient and its corresponding flow associated with a distance function are described in [11] and later adapted for distance-like functions in [12]. Let $f_S : \mathbb{R}^d \rightarrow \mathbb{R}$ be a distance function associated with a compact set S of \mathbb{R}^d . It is not differentiable on the medial axis of S . A *generalized gradient function* $\nabla_S : \mathbb{R}^d \rightarrow \mathbb{R}^d$ coincides with the usual gradient of f_S where f_S is differentiable, and is defined everywhere and can be integrated into a continuous flow $\Phi^t : \mathbb{R}^d \rightarrow \mathbb{R}^d$ that points away from S . Let γ be an integral (flow) line. The following technical lemma is proved in the full version [47].

► **Lemma 16.** *Given any flow line γ associated with the generalized gradient function ∇_S , $d_\mu^K(x)$ is strictly monotonically increasing along γ for x sufficiently far away from the medial axis of S , for $\sigma \leq \frac{R}{6\Delta_G}$ and $f_S(x) \in (0.014R, 2\sigma)$. Here $B(\sigma/2)$ denotes a ball of radius $\sigma/2$, $G := \frac{\text{Vol}(B(\sigma/2))}{\text{Vol}(S)}$, $\Delta_G := \sqrt{12 + 3 \ln(4/G)}$ and suppose $R := \min(\text{reach}(S), \text{reach}(\mathbb{R}^d \setminus S)) > 0$.*

The strict monotonicity of d_μ^K along the flow line under the conditions in Lemma 16 makes it possible to define a deformation retract of the sublevel sets of d_μ^K onto sublevel sets of f_S . Such a deformation retract defines a special case of homotopy equivalence between the sublevel sets of d_μ^K and sublevel sets of f_S . Consider a sufficiently large point set $P \in \mathbb{R}^d$ sampled from μ , and its induced measure μ_P . We can then also invoke Theorem 14 and a sampling bound (see Section 6) to show homotopy equivalence between the sublevel sets of f_S and $d_{\mu_P}^K$.

5 Stability Properties for the Kernel Distance to a Measure

► **Lemma 17 (K4).** *For two measures μ and ν on \mathbb{R}^d we have $\|d_\mu^K - d_\nu^K\|_\infty \leq D_K(\mu, \nu)$.*

Proof. Since $D_K(\cdot, \cdot)$ is a metric, then by triangle inequality, for any $x \in \mathbb{R}^d$ we have $D_K(\mu, x) \leq D_K(\mu, \nu) + D_K(\nu, x)$ and $D_K(\nu, x) \leq D_K(\nu, \mu) + D_K(\mu, x)$. Therefore for any $x \in \mathbb{R}^d$ we have $|D_K(\mu, x) - D_K(\nu, x)| \leq D_K(\mu, \nu)$, proving the claim. ◀

Both the Wasserstein and kernel distance are *integral probability metrics* [54], so (M4) and (K4) are both interesting, but not easily comparable. We now attempt to reconcile this.

5.1 Comparing D_K to W_2

► **Lemma 18.** *There is no Lipschitz constant γ such that for any two probability measures μ and ν we have $W_2(\mu, \nu) \leq \gamma D_K(\mu, \nu)$.*

Proof. Consider two measures μ and ν which are almost identical: the only difference is some mass of measure τ is moved from its location in μ a distance n in ν . The Wasserstein distance requires a transportation plan that moves this τ mass in ν back to where it was in μ with cost $\tau \cdot \Omega(n)$ in $W_2(\mu, \nu)$. On the other hand, $D_K(\mu, \nu) = \sqrt{\kappa(\mu, \mu) + \kappa(\nu, \nu) - 2\kappa(\mu, \nu)} \leq \sqrt{\sigma^2 + \sigma^2 - 2 \cdot 0} = \sqrt{2}\sigma$ is bounded. ◀

We conjecture for any two probability measures μ and ν that $D_K(\mu, \nu) \leq W_2(\mu, \nu)$. This would show that d_μ^K is at least as stable as d_{μ, m_0}^{CM} since a bound on $W_2(\mu, \nu)$ would also

bound $D_K(\mu, \nu)$, but not vice versa. We leave much of the technical details from this section to the full version [47]. We start with a special case.

► **Lemma 19.** *Consider two probability measures μ and ν on \mathbb{R}^d where ν is represented by a Dirac mass at a point $x \in \mathbb{R}^d$. Then $d_\mu^K(x) = D_K(\mu, \nu) \leq W_2(\mu, \nu)$ for any $\sigma > 0$, where the equality only holds when μ is also a Dirac mass at x .*

Next we show that if ν is not a unit Dirac, then this inequality holds in the limit as σ goes to infinity. The technical work is making precise how $\sigma^2 - K(p, x) \leq \|x - p\|^2/2$ and how this compares to bounds on $D_K(\mu, \nu)$ and $W_2(\mu, \nu)$.

► **Lemma 20.** *For any $p, q \in \mathbb{R}^d$ we have $K(p, q) = \sigma^2 - \frac{\|p - q\|^2}{2} + \sum_{i=2}^{\infty} \frac{(-\|p - q\|^2)^i}{2^{i+1}\sigma^{2i-2}i!}$.*

Proof. We use the Taylor expansion of $e^x = \sum_{i=0}^{\infty} x^i/i! = 1 + x + \sum_{i=2}^{\infty} x^i/i!$. Then it is easy to see

$$K(p, q) = \sigma^2 \exp\left(-\frac{\|p - q\|^2}{2\sigma^2}\right) = \sigma^2 - \frac{\|p - q\|^2}{2} + \sum_{i=2}^{\infty} \frac{(-\|p - q\|^2)^i}{2^i\sigma^{2i-2}i!}. \quad \blacktriangleleft$$

This lemma illustrates why the choice of coefficient of σ^2 is convenient. Since then $\sigma^2 - K(p, q)$ acts like $\frac{1}{2}\|p - q\|^2$, and becomes closer as σ increases. Define $\bar{\mu} = \int_p p \cdot d\mu(p)$ to represent the mean point of measure μ .

► **Theorem 21.** *For any two probability measures μ and ν defined on \mathbb{R}^d $\lim_{\sigma \rightarrow \infty} D_K(\mu, \nu) = \|\bar{\mu} - \bar{\nu}\|$ and $\|\bar{\mu} - \bar{\nu}\| \leq W_2(\mu, \nu)$. Thus $\lim_{\sigma \rightarrow \infty} D_K(\mu, \nu) \leq W_2(\mu, \nu)$.*

5.2 Kernel Distance Stability with Respect to σ

We now explore the Lipschitz properties of d_μ^K with respect to the noise parameter σ . We argue any distance function that is robust to noise needs some parameter to address how many outliers to ignore or how far away a point is to be considered as an outlier. Such a parameter in d_{μ, m_0}^{CCM} is m_0 which controls the amount of measure μ to be used in the distance.

Here we show that d_μ^K has a particularly nice property, that it is Lipschitz with respect to the choice of σ for any fixed x . Many details are deferred to the full version [47].

► **Lemma 22.** *Let $h(\sigma, z) = \exp(-z^2/2\sigma^2)$. We can bound $h(\sigma, z) \leq 1$, $\frac{d}{d\sigma}h(\sigma, z) \leq (2/e)/\sigma$ and $\frac{d^2}{d\sigma^2}h(\sigma, z) \leq (18/e^3)/\sigma^2$ over any choice of $z > 0$.*

► **Theorem 23.** *For any measure μ defined on \mathbb{R}^d and $x \in \mathbb{R}^d$, $d_\mu^K(x)$ is ℓ -Lipschitz with respect to σ , for $\ell = 18/e^3 + 8/e + 2 < 6$.*

Proof. (Sketch) Recall that $m_{\mu, \nu}$ is the product measure of any μ and ν . Define $M_{\mu, \nu}$ as $M_{\mu, \nu}(p, q) = m_{\mu, \mu}(p, q) + m_{\nu, \nu}(p, q) - 2m_{\mu, \nu}(p, q)$. It is useful to define a function $f_x(\sigma)$ as

$$f_x(\sigma) = \int_{(p, q)} \exp\left(\frac{-\|p - q\|^2}{2\sigma^2}\right) dM_{\mu, \delta_x}(p, q)$$

$$F(\sigma) = (d_\mu^K(x))^2 - \ell\|\sigma\|^2 = \sigma^2 f_x(\sigma) - \ell\sigma^2.$$

Now $d_\mu^K(x) = \sigma\sqrt{f_x(\sigma)}$. Now to prove $d_\mu^K(x)$ is ℓ -Lipschitz, we can show that $(d_\mu^K)^2$ is ℓ -semiconcave with respect to σ , and apply Lemma 3. This boils down to showing the second derivative of $F(\sigma)$ is always non-positive.

$$\frac{d^2}{d\sigma^2}F(\sigma) = \sigma^2 \frac{d^2}{d\sigma^2}f_x(\sigma) + 4\sigma \frac{d}{d\sigma}f_x(\sigma) + 2f_x(\sigma) - 2\ell.$$

First we note that for any distribution μ and Dirac delta that $\int_{(p,q)} c \cdot dM_{\mu,\delta_x}(p, q) \leq 2c$. Thus since $\exp\left(\frac{-\|p-q\|^2}{2\sigma^2}\right)$ is in $[0, 1]$ for all choices of p, q , and $\sigma > 0$, then $0 \leq f_x(\sigma) \leq 2$ and $2f_x(\sigma) \leq 4$. This bounds the third term in $\frac{d^2}{d\sigma^2}F(\sigma)$, we now need to use a similar approach to bound the first and second terms. Using Lemma 22 to obtain

$$\frac{d^2}{d\sigma^2}F(\sigma) \leq 36/e^3 + 16/e + 4 - 2(18/e^3 + 8/e + 2) = 0. \quad \blacktriangleleft$$

Lipschitz in m_0 for $d_{\mu, m_0}^{\text{CCM}}$. There is no Lipschitz property for $d_{\mu, m_0}^{\text{CCM}}$, with respect to m_0 , independent of μ . Consider a measure μ_P for point set $P \subset \mathbb{R}$ consisting of two points at $a = 0$ and at $b = \Delta$. When $m_0 = 1/2 + \alpha$ for $\alpha > 0$, then $d_{\mu_P, m_0}^{\text{CCM}}(a) = \alpha\Delta/(1/2 + \alpha)$ and $\frac{d}{dm_0}d_{\mu_P, m_0}^{\text{CCM}}(a) = \frac{d}{d\alpha}d_{\mu_P, \frac{1}{2} + \alpha}^{\text{CCM}}(a) = \frac{(1/2 + 2\alpha)\Delta}{(1/2 + \alpha)^2}$, which is maximized as α approaches 0 with an infimum of 2Δ . Hence the Lipschitz constant for $d_{\mu_P, m_0}^{\text{CCM}}$ with respect to m_0 is $2\Delta_P$ where $\Delta_P = \max_{p, p' \in P} \|p - p'\|$.

6 Algorithmic and Approximation Observations

Kernel coresets. The kernel distance is robust under random samples [38]. Specifically, if Q is a point set randomly chosen from μ of size $O((1/\varepsilon^2)(d + \log(1/\delta)))$ then $\|\text{KDE}_\mu - \text{KDE}_Q\|_\infty \leq \varepsilon$ with probability at least $1 - \delta$. We call such a subset Q and ε -kernel sample of (μ, K) . Furthermore, it is also possible to construct ε -kernel samples Q with even smaller size of $|Q| = O(((1/\varepsilon)\sqrt{\log(1/\varepsilon\delta)})^{2d/(d+2)})$ [45]; in particular in \mathbb{R}^2 the required size is $|Q| = O((1/\varepsilon)\sqrt{\log(1/\varepsilon\delta)})$. Exploiting the above constructions, recent work [58] builds a data structure to allow for efficient approximate evaluations of KDE_P where $|P| = 100,000,000$.

These constructions of Q also immediately imply that $\|(d_\mu^K)^2 - (d_Q^K)^2\|_\infty \leq 4\varepsilon$ since $(d_\mu^K(x))^2 = \kappa(\mu, \mu) + \kappa(x, x) - 2\text{KDE}_\mu(x)$, and both the first and third term incur at most 2ε error in converting to $\kappa(Q, Q)$ and $2\text{KDE}_Q(x)$, respectively. Thus, an $(\varepsilon^2/4)$ -kernel sample Q of (μ, K) implies that $\|d_\mu^K - d_Q^K\|_\infty \leq \varepsilon$.

This implies algorithms for geometric inference on enormous noisy data sets, or when input Q is assumed to be drawn iid from an unknown distribution μ .

► **Corollary 24.** Consider a measure μ defined on \mathbb{R}^d , a kernel K , and a parameter $\varepsilon \leq R(5 + 4/\alpha^2)$. We can create a coreset Q of size $|Q| = O(((1/\varepsilon^2)\sqrt{\log(1/\varepsilon\delta)})^{2d/(d+2)})$ or randomly sample $|Q| = O((1/\varepsilon^4)(d + \log(1/\delta)))$ points so, with probability at least $1 - \delta$, any sublevel set $(d_\mu^K)^\eta$ is homotopy equivalent to $(d_Q^K)^r$ for $r \in [4\varepsilon/\alpha^2, R - 3\varepsilon]$ and $\eta \in (0, R)$.

Stability of persistence diagrams. Furthermore, the stability results on persistence diagrams [20] hold for kernel density estimates and kernel distance of μ and Q (where Q is a coreset of μ with the same size bounds as above). If $\|f - g\|_\infty \leq \varepsilon$, then $d_B(\text{Dgm}(f), \text{Dgm}(g)) \leq \varepsilon$, where d_B is the bottleneck distance between persistence diagrams.

► **Corollary 25.** Consider a measure μ defined on \mathbb{R}^d and a kernel K . We can create a coreset Q of size $|Q| = O(((1/\varepsilon)\sqrt{\log(1/\varepsilon\delta)})^{2d/(d+2)})$ or randomly sample $|Q| = O((1/\varepsilon^2)(d + \log(1/\delta)))$ points which will have the following properties with probability at least $1 - \delta$.

- $d_B(\text{Dgm}(\text{KDE}_\mu), \text{Dgm}(\text{KDE}_Q)) \leq \varepsilon$.
- $d_B(\text{Dgm}((d_\mu^K)^2), \text{Dgm}((d_Q^K)^2)) \leq \varepsilon$.

► **Corollary 26.** Consider a measure μ defined on \mathbb{R}^d and a kernel K . We can create a coreset Q of size $|Q| = O(((1/\varepsilon^2)\sqrt{\log(1/\varepsilon\delta)})^{2d/(d+2)})$ or randomly sample $|Q| = O((1/\varepsilon^4)(d + \log(1/\delta)))$ points which will have the following property with probability at least $1 - \delta$.

- $d_B(\text{Dgm}(d_\mu^K), \text{Dgm}(d_Q^K)) \leq \varepsilon$.

Another bound was independently derived to show an upper bound on the size of a random sample Q such that $d_B(\text{Dgm}(\text{KDE}_{\mu_P}), \text{Dgm}(\text{KDE}_Q)) \leq \varepsilon$ in [2]; this can, as above, also be translated into bounds for $\text{Dgm}((d_Q^K)^2)$ and $\text{Dgm}(d_Q^K)$. This result assumes $P \subset [-C, C]^d$ and is parametrized by a bandwidth parameter h that retains that $\int_{x \in \mathbb{R}^d} K_h(x, p) dx = 1$ for all p using that $K_1(\|x - p\|) = K(x, p)$ and $K_h(\|x - p\|) = \frac{1}{h^d} K_1(\|x - p\|^2/h)$. This ensures that $K(\cdot, p)$ is $(1/h^d)$ -Lipschitz and that $K(x, x) = \Theta(1/h^d)$ for any x . Then their bound requires $|Q| = O(\frac{d}{\varepsilon^2 h^d} \log(\frac{Cd}{\varepsilon \delta h}))$ random samples.

To compare directly against the random sampling result we derive from Joshi *et al.* [38], for kernel $K_h(x, p)$ then $\|\text{KDE}_{\mu_P} - \text{KDE}_Q\|_\infty \leq \varepsilon K_h(x, x) = \varepsilon/h^d$. Hence, our analysis requires $|Q| = O((1/\varepsilon^2 h^{2d})(d + \log(1/\delta)))$, and is an improvement when $h = \Omega(1)$ or C is not known or bounded, as well as in some other cases as a function of ε , h , δ , and d .

Acknowledgements. The authors thank Don Sheehy, Frédéric Chazal and the rest of the Geometrica group at INRIA-Saclay for enlightening discussions on geometric and topological reconstruction. We also thank Don Sheehy for personal communications regarding the power distance constructions, and Yusu Wang for ideas towards Lemma 16. Finally, we are also indebted to the anonymous reviewers for many detailed suggestions leading to improvements in results and presentation.

References

- 1 N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, 68:337–404, 1950.
- 2 Sivaraman Balakrishnan, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Statistical inference for persistent homology. Technical report, ArXiv:1303.7117, March 2013.
- 3 James Biagioni and Jakob Eriksson. Map inference in the face of noise and disparity. In *ACM SIGSPATIAL GIS*, 2012.
- 4 Gérard Biau, Frédéric Chazal, David Cohen-Steiner, Luc Devroye, and Carlos Rodriguez. A weighted k -nearest neighbor density estimate for geometric inference. *Electronic Journal of Statistics*, 5:204–237, 2011.
- 5 Omer Bobrowski, Sayan Mukherjee, and Jonathan E. Taylor. Topological consistency via kernel estimation. Technical report, arXiv:1407.5272, 2014.
- 6 Peter Bubenik. Statistical topological data analysis using persistence landscapes. *Journal of Machine Learning Research*, 2014.
- 7 Mickael Buchet, Frederic Chazal, Steve Y. Oudot, and Donald R. Sheehy. Efficient and robust persistent homology for measures. In *SODA*, 2015.
- 8 Frédéric Chazal and David Cohen-Steiner. Geometric inference. *Tessellations in the Sciences*, 2012.
- 9 Frédéric Chazal, David Cohen-Steiner, Marc Glisse, Leonidas J. Guibas, and Steve Y. Oudot. Proximity of persistence modules and their diagrams. In *SOCG*, 2009.
- 10 Frédéric Chazal, David Cohen-Steiner, and André Lieutier. Normal cone approximation and offset shape isotopy. *CGTA*, 42:566–581, 2009.
- 11 Frédéric Chazal, David Cohen-Steiner, and André Lieutier. A sampling theory for compact sets in Euclidean space. *DCG*, 41(3):461–479, 2009.
- 12 Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *FOCM*, 11(6):733–751, 2011.
- 13 Frederic Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. The structure and stability of persistence modules. arXiv:1207.3674, 2013.

- 14 Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: Distance-to-a-measure and kernel distance. Technical report, arXiv:1412.7197, 2014.
- 15 Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. On the bootstrap for persistence diagrams and landscapes. *Modeling and Analysis of Information Systems*, 20:96–105, 2013.
- 16 Frédéric Chazal, Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, and Larry Wasserman. Stochastic convergence of persistence landscapes. In *SOCG*, 2014.
- 17 Frédéric Chazal and André Lieutier. Weak feature size and persistent homology: computing homology of solids in R^n from noisy data samples. In *SOCG*, pages 255–262, 2005.
- 18 Frédéric Chazal and André Lieutier. Topology guaranteeing manifold reconstruction using distance function to noisy data. In *SOCG*, 2006.
- 19 Frédéric Chazal and Steve Oudot. Towards persistence-based reconstruction in euclidean spaces. In *SOCG*, 2008.
- 20 David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *DCG*, 37:103–120, 2007.
- 21 Luc Devroye and László Györfi. *Nonparametric Density Estimation: The L_1 View*. Wiley, 1984.
- 22 Luc Devroye and Gábor Lugosi. *Combinatorial Methods in Density Estimation*. Springer-Verlag, 2001.
- 23 Herbert Edelsbrunner. The union of balls and its dual shape. In *SOCG*, 1993.
- 24 Herbert Edelsbrunner, Michael Facello, Ping Fu, and Jie Liang. Measuring proteins and voids in proteins. In *Proceedings 28th Annual Hawaii International Conference on Systems Science*, 1995.
- 25 Herbert Edelsbrunner, Brittany Terese Fasy, and Günter Rote. Add isotropic Gaussian kernels at own risk: More and more resilient modes in higher dimensions. In *SOCG*, 2012.
- 26 Herbert Edelsbrunner and John Harer. Persistent homology. *Contemporary Mathematics*, 453:257–282, 2008.
- 27 Herbert Edelsbrunner and John Harer. *Computational Topology: An Introduction*. American Mathematical Society, Providence, RI, USA, 2010.
- 28 Ahmed Elgammal, Ramani Duraiswami, David Harwood, and Larry S. Davis. Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. *Proc. IEEE*, 90:1151–1163, 2002.
- 29 Brittany Terese Fasy, Jisu Kim, Fabrizio Lecci, and Clément Maria. Introduction to the R package TDA. Technical report, arXiv:1411.1830, 2014.
- 30 Brittany Terese Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Statistical inference for persistent homology: Confidence sets for persistence diagrams. In *The Annals of Statistics*, volume 42, pages 2301–2339, 2014.
- 31 H. Federer. Curvature measures. *Transactions of the American Mathematical Society*, 93:418–491, 1959.
- 32 Mingchen Gao, Chao Chen, Shaoting Zhang, Zhen Qian, Dimitris Metaxas, and Leon Axel. Segmenting the papillary muscles and the trabeculae from high resolution cardiac CT through restoration of topological handles. In *Proceedings International Conference on Information Processing in Medical Imaging*, 2013.
- 33 Joan Glaunès. *Transport par difféomorphismes de points, de mesures et de courants pour la comparaison de formes et l’anatomie numérique*. PhD thesis, Université Paris 13, 2005.
- 34 Karsten Grove. Critical point theory for distance functions. *Proceedings of Symposia in Pure Mathematics*, 54:357–385, 1993.

- 35 Leonidas Guibas, Quentin Mérigot, and Dmitriy Morozov. Witnessed k -distance. In *SOCG*, 2011.
- 36 Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.
- 37 Matrial Hein and Olivier Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *Proceedings 10th International Workshop on Artificial Intelligence and Statistics*, 2005.
- 38 Sarang Joshi, Raj Varma Kommaraju, Jeff M. Phillips, and Suresh Venkatasubramanian. Comparing distributions and shapes using the kernel distance. In *SOCG*, 2011.
- 39 John M. Lee. *Introduction to smooth manifolds*. Springer, 2003.
- 40 Jie Liang, Herbert Edelsbrunner, Ping Fu, Pamidighantam V. Sudharkar, and Shankar Subramanian. Analytic shape computation of macromolecules: I. molecular area and volume through alpha shape. *Proteins: Structure, Function, and Genetics*, 33:1–17, 1998.
- 41 André Lieutier. Any open bounded subset of \mathbb{R}^n has the same homotopy type as its medial axis. *Computer-Aided Design*, 36:1029–1046, 2004.
- 42 Quentin Mérigot. *Geometric structure detection in point clouds*. PhD thesis, Université de Nice Sophia-Antipolis, 2010.
- 43 Yuriy Mileyko, Sayan Mukherjee, and John Harer. Probability measures on the space of persistence diagrams. *Inverse Problems*, 27(12), 2011.
- 44 A. Müller. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29(2):429–443, 1997.
- 45 Jeff M. Phillips. ε -samples for kernels. *SODA*, 2013.
- 46 Jeff M. Phillips and Suresh Venkatasubramanian. A gentle introduction to the kernel distance. arXiv:1103.1625, March 2011.
- 47 Jeff M. Phillips, Bei Wang, and Yan Zheng. Geometric inference on kernel density estimates. In *arXiv:1307.7760*, 2015.
- 48 Florian T. Pokorny, Carl Henrik, Hedvig Kjellström, and Danica Kragic. Persistent homology for learning densities with bounded support. In *Neural Informations Processing Systems*, 2012.
- 49 Charles A. Price, Olga Symonova, Yuriy Mileyko, Troy Hilley, and Joshua W. Weitz. Leaf gui: Segmenting and analyzing the structure of leaf veins and areoles. *Plant Physiology*, 155:236–245, 2011.
- 50 David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.
- 51 Donald R. Sheehy. A multicover nerve for geometric inference. *CCCG*, 2012.
- 52 Bernard W. Silverman. Using kernel density estimates to investigate multimodality. *J. R. Statistical Society B*, 43:97–99, 1981.
- 53 Bernard W. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall/CRC, 1986.
- 54 Bharath K. Sriperumbudur, Arthur Gretton, Kenji Fukumizu, Bernhard Schölkopf, and Gert R.G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *JMLR*, 11:1517–1561, 2010.
- 55 Kathryn Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet means for distributions of persistence diagrams. *DCG*, 2014.
- 56 Cédric Villani. *Topics in Optimal Transportation*. American Mathematical Society, 2003.
- 57 Grace Wahba. Support vector machines, reproducing kernel Hilbert spaces, and randomization. In *Advances in Kernel Methods – Support Vector Learning*, pages 69–88. The MIT Press, 1999.
- 58 Yan Zheng, Jeffrey Jests, Jeff M. Phillips, and Feifei Li. Quality and efficiency in kernel density estimates for large data. In *SIGMOD*, 2012.

Modeling Real-World Data Sets

Susanne Albers

Department of Computer Science, Technische Universität München
Boltzmannstr. 3, 85748 Garching, Germany
albers@in.tum.de

Abstract

Traditionally, the performance of algorithms is evaluated using worst-case analysis. For a number of problems, this type of analysis gives overly pessimistic results: Worst-case inputs are rather artificial and do not occur in practical applications. In this lecture we review some alternative analysis approaches leading to more realistic and robust performance evaluations.

Specifically, we focus on the approach of modeling real-world data sets. We report on two studies performed by the author for the problems of self-organizing search and paging. In these settings real data sets exhibit locality of reference. We devise mathematical models capturing locality. Furthermore, we present combined theoretical and experimental analyses in which the theoretically proven and experimentally observed performance guarantees match up to very small relative errors.

1998 ACM Subject Classification F.2 Analysis of Algorithms and Problem Complexity, F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases Worst-case analysis, real data sets, locality of reference, paging, self-organizing lists

Digital Object Identifier 10.4230/LIPIcs.SOCG.2015.872

Category Invited Talk



© Susanne Albers;

licensed under Creative Commons License CC-BY

31st International Symposium on Computational Geometry (SoCG'15).

Editors: Lars Arge and János Pach; pp. 872–872



Leibniz International Proceedings in Informatics

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