Report from Dagstuhl Seminar 19352

Computation in Low-Dimensional Geometry and Topology

Edited by

Maarten Löffler[1], Anna Lubiw[2], Saul Schleimer[3], and Erin Moriarty Wolf Chambers[4]

1 **Utrecht University, NL,** `m.loffler@uu.nl`
2 **University of Waterloo, CA,** `alubiw@uwaterloo.ca`
3 **University of Warwick – Coventry, GB,** `s.schleimer@warwick.ac.uk`
4 **St. Louis University, US,** `echambe5@slu.edu`

Abstract

This report documents the program and the outcomes of Dagstuhl Seminar 19352 "Computation in Low-Dimensional Geometry and Topology". The seminar participants investigated problems in: knot theory, trajectory analysis, algorithmic topology, computational geometry of curves, and graph drawing, with an emphasis on how low-dimensional structures change over time.

Seminar August 25–30, 2019 – http://www.dagstuhl.de/19352
2012 ACM Subject Classification Mathematics of computing → Geometric topology, Human-centered computing → Graph drawings, Theory of computation → Computational geometry
Keywords and phrases Geometric topology, Graph Drawing, Computational Geometry
Digital Object Identifier 10.4230/DagRep.9.8.84
Edited in cooperation with Boris Klemz

1 Executive Summary

Erin Moriarty Wolf Chambers (St. Louis University, US)
Maarten Löffler (Utrecht University, NL)
Anna Lubiw (University of Waterloo, CA)
Saul Schleimer (University of Warwick – Coventry, GB)

One-dimensional structures embedded in higher-dimensional spaces are ubiquitous in both the natural and artificial worlds: examples include DNA strands, migration paths, planetary orbits, rocket trajectories, robot motion planning, chip design, and many more. These are studied in different areas of mathematics and computer science, under many names: knots, curves, paths, traces, trajectories, graphs, and others. However, researchers in many areas are just beginning to apply algorithmic techniques to find efficient algorithms for these structures, especially when more fundamental mathematical results are required. Broad examples of such problems include:

- classical algorithms on trajectories like the Fréchet distance as a way to formalize a distance measure as a curve changes;
- morphing between two versions of a common graph, which again tracks a higher dimensional space that corresponds to movement of a one-dimensional object;
- drawing and manipulating objects in three-manifolds, such as graphs, curves, or surfaces; and
- perhaps the simplest problem posed (in different ways) in all these areas, "how does one draw and morph a nice curve on a nice surface?"

This seminar was the second in a series. In the first seminar, the goal was to identify connections and seed new research collaborations along the spectrum from knot theory and topology, through to computational topology and computational geometry, and all the way to graph drawing. After the success of the first seminar, the goal for this second round was to continue and extend prior work, in particular by focussing on how objects change over time.

The seminar began with three overview talks from researchers in different areas (trajectory analysis, algorithmic topology, and graph drawing) to motivate and introduce problems which would fit the theme of changes over time in the representations of low-dimensional objects in higher dimensional spaces. We then invited all participants to describe open problems (most of which were circulated in advance of the meeting) that fit with the topic of the workshop and could benefit from broad expertise. For the remainder of the workshop we split into small working groups each focussed on a particular open problem.

Throughout the workshop we used Coauthor, a tool for collaboration designed by Erik Demaine (MIT), to share progress and updates among all the working groups. This, together with twice-daily progress reports, allowed us to share ideas and expertise among all participants, which was very effective. Another advantage was that we had a record of the work accomplished when the workshop ended.

Below, we (the organizers) describe the main working group topics and how they connected to the overarching theme. The abstracts of talks in the seminar and preliminary results from the working groups are also outlined later in this report.

One group worked on open questions that were motivated by 3-manifolds. In particular, they considered lower bounds for deciding the complexity of a knot or link equivalence, with a goal of finding specific knots that require many simplification moves. Their work involved both designing smaller examples, as well as doing larger scale exhaustive search using the software tool Regina.

Another group considered representations of graphs and hypergraphs by touching polygons in 3-d. They were able to leverage the dual graph of the polyhedral complex in 3d, and make progress on classifying which types of graphs could (or could not) be realized. Their problem was primarily combinatorial, but the techniques used included several interesting topological arguments about embeddings of manifolds into 3d or into 3-manifolds.

Several groups considered problems about flows or morphs of curves in various settings. One question centered on visualizing actual embedded homotopies on a given surface; there is considerable prior work on how to compute such homotopies between curves quickly, but it generally focuses on computing the complexity of the homotopy as opposed to the actual sequence of simplifications needed. The group looked more closely at this algorithm, and was able to outline a proof that in fact an extension of that algorithm would generate the actual homotopy, for a slightly higher time cost. A second group considered curves in the plane, and investigated options for computing a "nice" morph between them. As the question was more vague, the group did quite a bit of background investigation on prior work, and then discussed a new technique based on 3-manifolds and normal surface theory which might lead to a new family of morphs. A third group looked at the problem of preprocessing a given curve so that the Fréchet distance to any other query curve could be efficiently computed, and were able to obtain improved time bounds for several variants of the problem.

In summary, the workshop fostered a highly collaborative environment where combining the expertise and knowledge of researchers from different communities allowed us to solve problems of common interest across those communities. A major theme was how connected the various problems could be; often, a proof technique or piece of literature suggested by a member of a different community proved useful or insightful to a group working in a different domain.

2 Table of Contents

Executive Summary
Erin Moriarty Wolf Chambers, Maarten Löffler, Anna Lubiw, and Saul Schleimer . 84

Overview of Talks

One-Dimensional Structures in Low-Dimensional Algorithmic Topology
Arnaud de Mesmay 87

Curves, Distance Measures, and Homotopies
Carola Wenk 87

Survey on Graph and Hypergraph Drawing
André Schulz and Alexander Wolff 87

Working groups

Frechet Distance Data Structures
Maike Buchin, Tim Ophelders, Lena Schlipf, Rodrigo I. Silveira, Frank Staals, and Ivor van der Hoog 90

Lower Bounds for the Complexity of Knot and Link Equivalence
Benjamin Burton, Hsien-Chih Chang, Arnaud de Mesmay, Francis Lazarus, Maarten Löffler, Clément Maria, Saul Schleimer, Eric Sedgwick, and Jonathan Spreer . . . 91

Combinatorial Homotopies
Hsien-Chih Chang, Vincent Despré, Linda Kleist, Francis Lazarus, Anna Lubiw, Tim Ophelders, Hugo Parlier, Saul Schleimer, Stephan Tillmann, Birgit Vogtenhuber, Carola Wenk, and Erin Moriarty Wolf Chambers 94

Simple Graph Cycle in Homotopy Class
Hsien-Chih Chang, Arnaud de Mesmay, Vincent Despré, Francis Lazarus, and Erin Moriarty Wolf Chambers 96

Beautiful curves on beautiful surfaces
Saul Schleimer, Vincent Despré, Francis Lazarus, Hugo Parlier, Stephan Tillmann, and Erin Moriarty Wolf Chambers 99

Minimum Area Homotopies
Carola Wenk, Hsien-Chih Chang, Vincent Despré, Francis Lazarus, Anna Lubiw, Tim Ophelders, Hugo Parlier, and Erin Moriarty Wolf Chambers 101

Nice Morphs and Isotopic Frechet Distance
Erin Moriarty Wolf Chambers, Vincent Despré, Linda Kleist, Maarten Löffler, Anna Lubiw, Tim Ophelders, Hugo Parlier, Stephan Tillmann, Birgit Vogtenhuber, and Carola Wenk 103

Representing Graphs by Polygons with Edge Contacts in 3D
Alexander Wolff, Elena Arseneva, Arnaud de Mesmay, Linda Kleist, Boris Klemz, Maarten Löffler, André Schulz, and Birgit Vogtenhuber 106

Participants 112

3 Overview of Talks

3.1 One-Dimensional Structures in Low-Dimensional Algorithmic Topology

Arnaud de Mesmay (University of Grenoble, FR)

In this talk, I survey algorithms to test the topological equivalence of one-dimensional objects in dimensions two and three. More precisely, I focus on algorithms to test homotopy of curves on surfaces (and connected topics), and isotopy of knots in $\mathbb{R}^3$. I explain in particular why hyperbolic geometry helps for these problems, how researchers have come up with approaches to combinatorialize it into efficient algorithms in two dimensions, and how this is lacking in 3d. We also foray into implementations.

3.2 Curves, Distance Measures, and Homotopies

Carola Wenk (Tulane University – New Orleans, US)

We study distance measures for curves, their relation to homotopies, and curve simplification. In particular we consider Hausdorff and Fréchet distances, and discuss algorithmic and hardness results for curve simplification under these distances. As a distance measure for metric spaces we consider the Gromov Hausdorff distance in Euclidean space. Finally we study the minimum homotopy area for a closed curve in the plane and as a distance measure for two curves.

3.3 Survey on Graph and Hypergraph Drawing

André Schulz (FernUniversität in Hagen, DE) and Alexander Wolff (Universität Würzburg, DE)

In this talk we review the basic concepts and the methodology of Graph Drawing, which is an active research area in the intersection of Graph Theory, Computational Geometry, and Information Visualization. Graph Drawing is about finding algorithms that map abstract, combinatorial objects (graphs or hypergraphs) to drawings, that is, "tangible" geometric objects. The goal is to find algorithms that guarantee a provable geometric quality measure in the worst case. For example, there are algorithms that draw any planar graph on a grid whose size is quadratic in the number of vertices of the graph. Other than in areas such as Information Visualization, the evaluation is usually *not* task-driven.

The Graph Drawing problem has numerous incarnations: supported graph classes (e.g., trees, outerplanar, planar, or bipartite graphs), drawing styles (e.g., orthogonal, straight-line, Bézier), quality measures (e.g., number of bends, number of crossings, crossing resolution), the embedding space (2D or 3D), and the type of representation (node–link diagrams, contact or intersection representations). Often, optimizing one measure leads to drawings that are bad in other measures. There is a lack of algorithms that are "pretty good" or at least "not too bad" in many aspects.

There is a (surprisingly small) set of standard techniques for drawing graphs. For example, if the graph class for which we want to design a drawing algorithm has a recursive definition, an obvious approach is to construct drawings recursively. A prominent example are orthogonal straight-line drawings of binary trees. It turns out that they can be drawn in a compact way; on a grid of size $O(n \log n)$, where n is the number of vertices of the given tree [4]. Similarly, if a graph class has an inductive definition, we may try to draw the given graph of that class inductively. Such an approach is used to show that every n-vertex planar 3-tree can be drawn using $2n - 4$ segments [6]. Finally, there are two at first glance very different approaches for drawing planar graphs on a grid of quadratic size. One approach, the shift algorithm [5], constructs the drawing incrementally; the other approach [11] counts some combinatorial objects (using a Schnyder wood) and then turns the resulting numbers into coordinates. A more careful analysis, however, reveals structural similarities between the two approaches.

Topics that have received considerable attention over the last few years are simultaneous embedding, morphing of graphs, drawings with large crossing angles, drawings of beyond-planar graphs, and visual complexity. The visual complexity of a drawing is measured by the number of geometric objects needed to compose or cover the drawing. For example, the *segment number* of a planar graph is the smallest number of straight-line segments whose union represents a straight-line drawing of the given graph [6]. The *arc number* [12] is defined accordingly with respect to circular-arc drawings, which often allow for less complex or more compact representations. Another recent generalization [10] are variants of the segment number for nonplanar graphs, either admitting crossings or embedding in 3D. The *plane cover number* [2, 3] asks for the smallest number of planes needed to cover a straight-line drawing of a given graph in 3D. Accordingly, one can define the *line cover number* in 2D (for planar graphs) or in 3D (for arbitrary graphs), which is obviously upperbounded by the corresponding segment number. Also weak versions of these numbers have been studied where only the vertices of a crossing-free straight-line drawing of the graph need to covered. While it is not hard to see that any outerplanar graph has weak line cover number 2 [2], even some cubic, 3-connected, bipartite planar graphs have unbounded weak line cover number (exceeding $\sqrt[3]{n}$, where n is the number of vertices) [7]. On the other hand, every 4-connected plane triangulation on n vertices has weak line cover number at most $\sqrt{2n}$ [9].

This seminar's topic – Computation in Low-Dimensional Geometry and Topology – is in line with a recent effort to better understand the drawing of graphs and hypergraphs in 3D. For example, we now know that every graph has a contact representation in 3D where vertices are represented by pairwise interior-disjoint convex polygons and edges by vertex–vertex contacts between the corresponding polygons [8]. On the other hand, the 3-uniform complete hypergraph with six (or more) vertices does not admit a representation where vertices are represented by points and hyperedges by (pairwise interior-disjoint) triangles connecting the corresponding points [1].

References

1 Johannes Carmesin. Embedding simply connected 2-complexes in 3-space – I. A Kuratowski-type characterisation. ArXiv report, 2019. URL: http://arxiv.org/abs/1709.04642.
2 Steven Chaplick, Krzysztof Fleszar, Fabian Lipp, Alexander Ravsky, Oleg Verbitsky, and Alexander Wolff. Drawing graphs on few lines and few planes. In Yifan Hu and Martin

Nöllenburg, editors, *Proc. 24th Int. Symp. Graph Drawing & Network Vis. (GD'16)*, volume 9801 of *LNCS*, pages 166–180. Springer, 2016. doi:10.1007/978-3-319-50106-2_14.

3 Steven Chaplick, Krzysztof Fleszar, Fabian Lipp, Alexander Ravsky, Oleg Verbitsky, and Alexander Wolff. The complexity of drawing graphs on few lines and few planes. In Faith Ellen, Antonina Kolokolova, and Jörg-Rüdiger Sack, editors, *WADS 2017*, volume 10389 of *LNCS*, pages 265–276. Springer, 2017. URL: http://arxiv.org/abs/1607.06444, doi:10.1007/978-3-319-62127-2_23.

4 Pierluigi Crescenzi, Giuseppe Di Battista, and Adolfo Piperno. A note on optimal area algorithms for upward drawings of binary trees. *Comput. Geom. Theory Appl.*, 2(4):187–200, 1992. doi:10.1016/0925-7721(92)90021-J.

5 Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. doi:10.1007/BF02122694.

6 Vida Dujmović, David Eppstein, Matthew Suderman, and David R. Wood. Drawings of planar graphs with few slopes and segments. *Comput. Geom. Theory Appl.*, 38(3):194–212, 2007. doi:10.1016/j.comgeo.2006.09.002.

7 David Eppstein. Cubic planar graphs that cannot be drawn on few lines. In *Proc. 35th Int. Symp. Comp. Geom. (SoCG'19)*, volume 129 of *LIPIcs*, pages 32:1–32:15, 2019. URL: https://arxiv.org/abs/1903.05256, doi:10.4230/LIPIcs.SoCG.2019.32.

8 William Evans, Paweł Rzążewski, Noushin Saeedi, Chan-Su Shin, and Alexander Wolff. Representing graphs and hypergraphs by touching polygons in 3d. In Daniel Archambault and Csaba D. Tóth, editors, *Proc. 27th Int. Symp. Graph Drawing & Network Vis. (GD'19)*, volume 11904 of *LNCS*. Springer, 2019. To appear. URL: https://arxiv.org/abs/1908.08273, doi:10.1007/978-3-030-35802-0_2.

9 Stefan Felsner. 4-connected triangulations on few lines. In Daniel Archambault and Csaba D. Tóth, editors, *Proc. 24th Int. Symp. Graph Drawing & Network Vis. (GD'16)*, volume 11904 of *LNCS*. Springer, 2019. To appear. URL: https://arxiv.org/abs/1908.04524, doi:10.1007/978-3-030-35802-0_30.

10 Yoshio Okamoto, Alexander Ravsky, and Alexander Wolff. Variants of the segment number of a graph. In Daniel Archambault and Csaba D. Tóth, editors, *Proc. 27th Int. Symp. Graph Drawing & Network Vis. (GD'19)*, volume 11904 of *LNCS*. Springer, 2019. To appear. URL: https://arxiv.org/abs/1908.08871, doi:10.1007/978-3-030-35802-0_33.

11 Walter Schnyder. Embedding planar graphs on the grid. In David S. Johnson, editor, *Proc. 1st ACM-SIAM Symp. Discrete Algorithms (SODA'90)*, pages 138–148, 1990. URL: https://dl.acm.org/citation.cfm?id=320191.

12 André Schulz. Drawing graphs with few arcs. *J. Graph Alg. Appl.*, 19(1):393–412, 2015. doi:10.7155/jgaa.00366.

4 Working groups

4.1 Frechet Distance Data Structures

Maike Buchin (Ruhr-Universität Bochum, DE), Tim Ophelders (Michigan State University, US), Lena Schlipf (FernUniversität in Hagen, DE), Rodrigo I. Silveira (UPC – Barcelona, ES), Frank Staals (Utrecht University, NL), and Ivor van der Hoog (Utrecht University, NL)

The Fréchet distance is a well known distance measure between two polygonal curves P and Q. It is also well known that computing the Fréchet distance can be done in $O(nm \log(nm))$ time, where n and m are the number of vertices in the curves P and Q respectively [1]. Furthermore, it is unlikely that a significantly faster algorithm is possible [2]. We consider a version of the problem in which we are given one of the curves, say P, in advance, and we can preprocess and store it so that for a query curve Q we can more quickly compute the Fréchet distance between P and Q. Moreover, like previous results [3], we focus on the case in which Q is a single line segment. We report on some preliminary results:

- In case of the discrete Fréchet distance we can preprocess P in $O(n \log n)$ time and space so that we can report the discrete Fréchet distance in $O(\log^3 n)$ time.
- In case of the weak Fréchet distance we can preprocess P in $O(n \log n)$ time and space so that we can report the discrete Fréchet distance in $O(\log^2 n)$ time.
- In case the query segments are restricted to be horizontal, we can build a data structure of size $O(n^{3/2})$ that can report the (real) Fréchet distance between P and Q in $O(\log n)$ time. This improves the existing result by de Berg et al. [4] that requires $O(n^2)$ space.
- In case the query segment may have an arbitrary orientation, we can answer queries in $O(\log^2 n)$ time, using $O(n^{4+\varepsilon})$ time. Furthermore, we are hopeful that an extension of our technique for horizontal query segments leads to an improved data structure using only $O(n^{7/2})$ space.

References

1 H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995.
2 K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 661–670, Oct 2014.
3 M. de Berg, J. Gudmundsson, and A. D. Mehrabi. A dynamic data structure for approximate proximity queries in trajectory data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '17, pages 48:1–48:4, New York, NY, USA, 2017. ACM.
4 M. de Berg, A. D. Mehrabi, and T. Ophelders. Data structures for Fréchet queries in trajectory data. In *CCCG*, pages 214–219, 2017.

4.2 Lower Bounds for the Complexity of Knot and Link Equivalence

Benjamin Burton (The University of Queensland – Brisbane, AU), Hsien-Chih Chang (Duke University – Durham, US), Arnaud de Mesmay (University of Grenoble, FR), Francis Lazarus (GIPSA Lab – Grenoble, FR), Maarten Löffler (Utrecht University, NL), Clément Maria (INRIA – Valbonne, FR), Saul Schleimer (University of Warwick – Coventry, GB), Eric Sedgwick (DePaul University – Chicago, US), and Jonathan Spreer (University of Sydney, AU)

A *knot* is a piecewise-linear embedding of S^1 into $\mathbb{R}^3$. A *link* is a piecewise-linear embedding of a disjoint union of S^1s into $\mathbb{R}^3$. Two knots/links are considered *equivalent* if there is a continuous deformation (an *isotopy*) between them that does not induce any crossing.

Open problem: Prove complexity lower bounds for the problem of deciding knot or link equivalence.

A little background: Deciding efficiently the equivalence of two knots or links is arguably one of the biggest problems in knot theory, see for example the survey of Lackenby [8]. All the existing algorithms are terribly inefficient (at least in theory), the only one with an analyzed complexity seems to be brute-forcing Reidemeister moves using the humongous bound of Lackenby and Coward [2]: this algorithm takes time $\exp^{(c^n)}(n)$ Reidemeister moves, where $c = 10^{1000000}$, and the notation $\exp^{(*)}$ denotes a tower of exponential of height $*$.

Finding new and improved algorithms probably requires some new insights and breakthroughs – this open problem aims at proving lower bounds instead. Strikingly *no* computational lower bound at all is known for this problem, even for links. While proving **NP**-hardness is a natural goal, even the seemingly humble result of proving that the link equivalence problem is at least as hard as Graph Isomorphism is open (see Lackenby [10], page 2). Note that testing homeomorphism of 3-manifolds is known to be at least as hard as Graph Isomorphism [5, 10].

Outputs: The objectives of the group were to investigate the complexity of testing knots and links equivalence, from two perspectives. The first one was about algorithmic complexity, and aimed at proving computational lower bounds for the problem. The second one was combinatorial, and aimed at finding instances of hard knots, i.e., knots for which any simplifying sequence of Reidemeister moves incurs a substantial increase in the number of crossings.

Despite our efforts, we could not progress on the first problem, and focused our attention to the second one, after pondering that it is consistent with current knowledge and not implausible that the knot equivalence problem lies in NP and coNP, and thus, under classical complexity hypotheses, would not be NP-hard, and possibly not even Graph Isomorphism-hard.

Hard knot diagrams

We consider the following problem: Are there diagrams of the unknot that require a substantial increase in the number of crossings in order to be simplified by Reidemeister moves to the trivial diagram? One can also consider related simplification problems, such as disconnecting diagrams of split links.

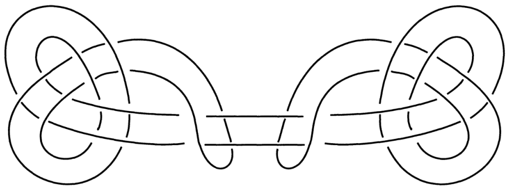

Figure 1 A hard unknot diagram [6].

This question is fundamental in algorithmic knot theory as simplifying diagrams comes as very first step in any computation on knots. The increase in the size of intermediate diagrams measures the difficulty of the search for a diagram with fewer crossings.

Dynnikov's work on arc presentation [4] implies the existence of a super polynomially long sequence of Reidemeister moves on an unknot diagram with n crossings, leading to the trivial diagram, and for which all intermediate diagrams have at most $(n-1)^2/2$ crossings. A similar result holds for disconnecting the diagram of a split link. More recently, Lackenby [9] proved that unknot and split link diagrams can be simplified in polynomially many Reidemeister moves (specifically $O(n^{11})$ moves) without exceeding a quadratic number of crossings ($O(n^2)$).

In practice however, the "hardest" unknot diagrams known require only 1 extra crossing after which they monotonically simplify. The examples in the literature [7] purporting to require two additional crossings were observed to be erroneous. Further claims of the existence of a family of unknots requiring an arbitrarily large number of additional crossings [7] are suspect.

Constructing hard unknots from any knot K

Figure 1 pictures a classical hard unknot diagram, which has been studied in the context of the energy minimization approach to the unknotting problem [6]. It can be constructed and generalized with the following procedure. For a knot K,

1. double the knot K (along the blackboard framing),
2. cut it open. We now have four ends coming in two pairs.
3. Take two mirror symmetric pairs of ends and stretch them out as two parallel strands,
4. take the remaining pair of ends and wrap them around those two strands before connecting them up,
5. mirror a second copy of this tangle and build the connect sum in the obvious way.

Figure 1 shows the construction for the trefoil knot K. We were able to show, by brute force computation with `Regina` [1], that in the case of the trefoil, this construction gives an unknot that requires 3 extra crossings to be unknotted. The constructions for more complex knots K are under investigation.

Generalizing the Goeritz unknot

A classical example of a hard unknot due to Goeritz is pictured in Figure 2 on the top left. It can be thought of (see the rest of Figure 2) as the concatenation of two inverse braids with two flypes inserted in between on both of its strands. Undoing these flypes requires turning the braid which can be shown experimentally to require at least one additional crossing.

We generalized this approach to braids with more strands in the braids, in order to increase the number of additional crossings needed. The framework pictured in Figure 3 seems promising, where Δ_1 and Δ_2 are the analogues of flypes, and can be readily generalized to higher number of strands.

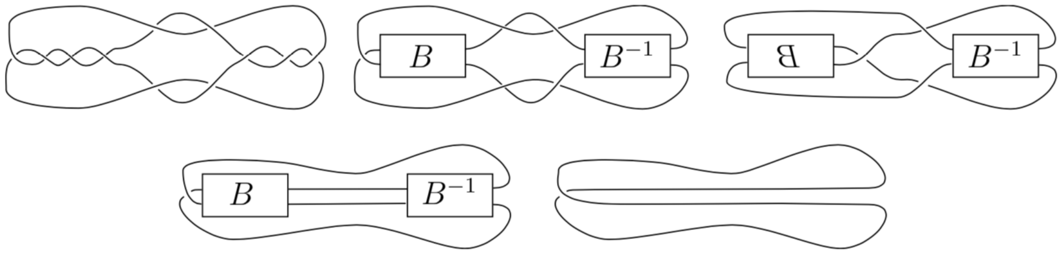

Figure 2 The Goeritz unknot.

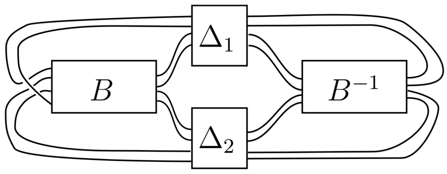

Figure 3 Harder unknots.

An exhaustive computer search with `Regina` [1] shows that for a good choice of a four-strand pseudo-Anosov braid B, more than two additional crossings are required to simplify such unknots. Figure 4 pictures such an example. Proving lower bounds for generalizations of this example as the number of strands goes to infinity is the object of ongoing work.

References

1. B. A. Burton, R. Budney, W. Pettersson, et al., *Regina: Software for low-dimensional topology.* `http://regina-normal.github.io/`, 1999–2017.
2. A. Coward and M. Lackenby, *An upper bound on reidemeister moves*, American Journal of Mathematics, 136 (2014), pp. 1023–1066.
3. Marc Culler, Nathan M. Dunfield, Matthias Goerner, and Jeffrey R. Weeks. SnapPy, a computer program for studying the geometry and topology of three-manifolds. Available at http://snappy.computop.org (2019-11-18).
4. I. A. Dynnikov, *Arc-presentations of links: Monotonic simplification*, Fund. Math., 190 (2006), pp. 29–76.
5. D. Eppstein, *Approaches to gi inspired by knot problem.* Theoretical Computer Science Stack Exchange. URL: https://cstheory.stackexchange.com/q/12792 (version: 2012-10-02).
6. M. H. Freedman, Z.-X. He, and Z. Wang, *Möbius energy of knots and unknots*, Annals of Mathematics, 139 (1994), pp. 1–50.
7. L. H. Kauffman and S. Lambropoulou, *Hard unknots and collapsing tangles*, Introductory lectures on knot theory, Ser. Knots Everything, 46 (2012), pp. 187–247.
8. M. Lackenby, *Elementary knot theory*, Lectures on Geometry, Oxford University Press, (2017).
9. M. Lackenby, *A polynomial upper bound on Reidemeister moves*, Annals of Mathematics, 182 (2015), pp. 491—564.
10. M. Lackenby, *Some conditionally hard problems on links and 3-manifolds*, Discrete & Computational Geometry, 58 (2017), pp. 580–595.

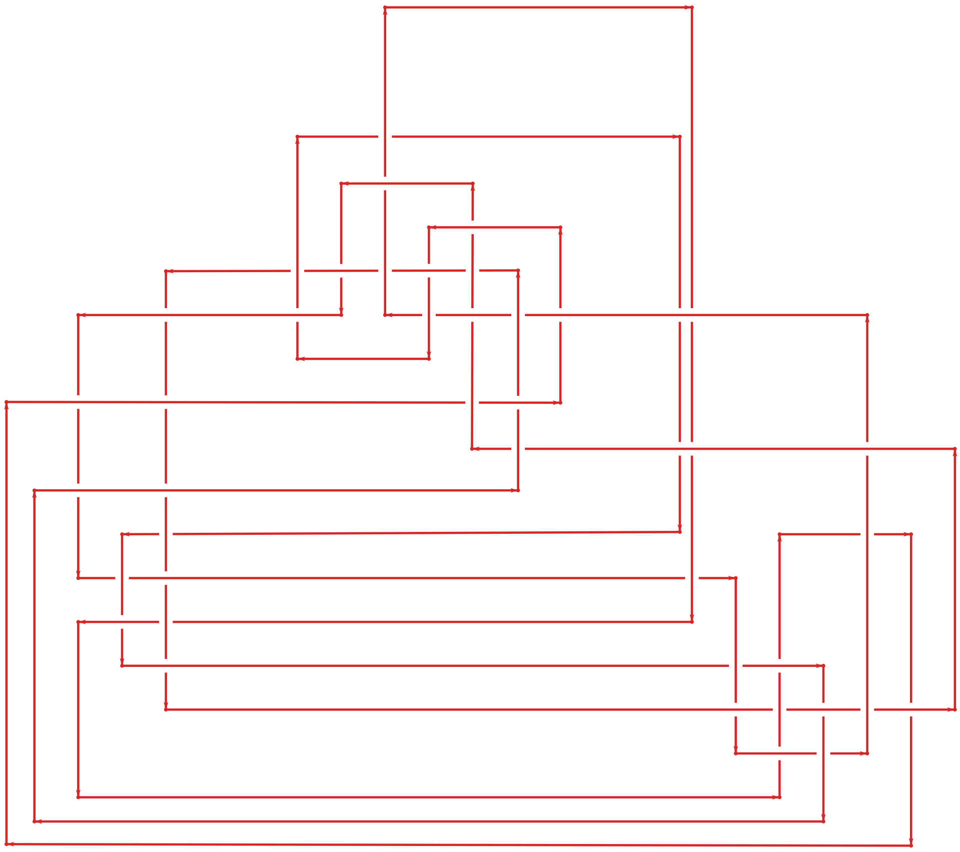

Figure 4 An example of a hard(er) unknot. Figure created with SnapPy [3].

4.3 Combinatorial Homotopies

Hsien-Chih Chang (Duke University – Durham, US), Vincent Despré (INRIA Nancy – Grand Est, FR), Linda Kleist (TU Braunschweig, DE), Francis Lazarus (GIPSA Lab – Grenoble, FR), Anna Lubiw (University of Waterloo, CA), Tim Ophelders (Michigan State University, US), Hugo Parlier (University of Luxembourg, LU), Saul Schleimer (University of Warwick – Coventry, GB), Stephan Tillmann (University of Sydney, AU), Birgit Vogtenhuber (TU Graz, AT), Carola Wenk (Tulane University – New Orleans, US), and Erin Moriarty Wolf Chambers (St. Louis University, US)

Main reference J. Erickson and K. Whittelsey. Transforming curves on surfaces redux. Proc. 24th ACM-SIAM Symp. Discrete Alg. (SODA), 2013
URL https://epubs.siam.org/doi/10.1137/1.9781611973105.118

A fundamental problem in computational topology is to test if a given loop in a space is contractible, or more generally if two loops are homotopic. The problem is known to be intractable in general but has a relatively simple solution when restricting the space to a 2 dimensional surface: there is a linear time algorithm that takes as input a combinatorial description of the surface and one or two closed paths in its 1-skeleton graph [1, 3]. This algorithm was implemented (and will be available in the next release of CGAL) and can be considered as a black box that may be used for more complex algorithms. Suppose that we want to convince an outside observer that two curves are indeed homotopic or contractible, without just relying on the answer returned by the above algorithm. It seems that the most intuitive certificate is to exhibit an actual (combinatorial) homotopy. Such a homotopy can be decomposed into elementary homotopies that consist of

- adding or removing a spur,
- replacing a piece of a facial walk by the complementary piece.

Elementary homotopies directly translate into PL homotopies that can be further visualized.

The problem is thus to design an efficient algorithm that outputs a sequence of elementary homotopies.

During this Dagstuhl seminar, we resolved the problem by providing a simple optimal algorithm. The idea is to simulate the linear time algorithm in [1]. Let S be an input combinatorial surface of genus g and let G be its 1-skeleton (vertex-edge graph). In [1, 3] a preprocesing step is to transform S into a quadrangulation Q composed of 2 vertices and $4g$ edges. Any walk γ in G can then be transformed into a homotopic walk c in Q of length at most $2|\gamma|$. Such a walk c admits a canonical representative in its homotopy class that corresponds to the shortest and rightmost homotopic walk in Q. In [1], this canonical representative is obtained by producing a sequence of elementary homotopies for c in Q. It remains to translate those elementary homotopies in G. We propose the following algorithm.

We first compute a representation of Q in G where each edge of Q is represented by a path in G. For this, we compute a system of loops in $O(gn)$ time ($n = |G|$) using a spanning tree and $2g$ additional chordal edges. One can also compute a shortest system of loops in $O(n \log n + gn)$ time as described in [2]. Note that each edge of G appears at most twice in a loop and so at most $4g$ times in the whole system of loops. We then cut S through the loops. We get a polygonal schema with the same number of faces as G, although the boundary has size $O(gn)$. The basepoint of the loops appears $4g$ times along this boundary. Pick any vertex in this polygonal schema and join it to the $4g$ copies of the basepoint using shortest paths. We make those $4g$ paths correspond to the edges of the system of quads. Let Q' be the union of those paths. Q' cuts S into $2g$ quadrilaterals regions, each comprising $O(n)$ faces of G.

We next "push" the given walk γ into Q', thus getting a homotopic curve γ' (corresponding to the above c) composed of at most $2|\gamma|$ concatenations of the paths in Q'. We finally simulate the moves in Q by sweeping quadrilateral regions. We thus obtain a canonical representative in G that can be further homotoped to a curve δ, applying the reverse sequence of moves that brings δ to the canonical representative.

Analysis: Sweeping a single quad costs $O(n)$ elementary homotopies. We observe by a simple analysis of the algorithm in [1] that the total length (number of quads) swept by the canonisation of c in Q is $O(|c|) = O(|\gamma|)$. Indeed, referring to the terminology in [1], it appears that the shortcut part of a bracket cannot be part of another bracket oriented the same way as this would imply a degree 4 vertex in Q. The linear bound on the number of swept quads easily follows. We thus get a sequence of $O(n|\gamma|)$ elementary homotopies to obtain a "canonical" form for γ in G. This number of elementary homotopies is tight: if the surface S has a subpart with the shape of a big "mushroom" and if γ surrounds its foot many times, we have to sweep over that mushroom that many times, enforcing a sequence of $\Omega(n|\gamma|)$ moves. A similar lower bound can be obtained from two homotopic walks winding around the ends of a long cylinder in S.

The above observation seems to imply that the computation of a canonical form does not require the run length encoding of the curve used in the paper by Erickson and Whittelsey [1]. A simple forward traversal of the curve should allow to shorten it without backtracking more that once when we shortcut brackets.

References

1 J. Erickson and K. Whittelsey, *Transforming curves on surfaces redux.*, Proc. 24th ACM-SIAM Symp. Discrete Alg. (SODA), 2013, pp. 1646–1655.
2 J. Erickson and K. Whittelsey, *Greedy optimal homotopy and homology generators.*, Proc. 16th ACM-SIAM Symp. Discrete Alg. (SODA), 2005, pp. 1646–1655, pp. 1038–1046.
3 F. Lazarus and J. Rivaud, *On the homotopy test on surfaces.*, Proc. 53rd IEEE Symp. Found. Comput. Sci. (FOCS), 2012, pp. 440–449.

4.4 Simple Graph Cycle in Homotopy Class

Hsien-Chih Chang (Duke University – Durham, US), Arnaud de Mesmay (University of Grenoble, FR), Vincent Despré (INRIA Nancy – Grand Est, FR), Francis Lazarus (GIPSA Lab – Grenoble, FR), and Erin Moriarty Wolf Chambers (St. Louis University, US)

Problem definition

Let Σ be a compact surface and G be a graph embedded in Σ. Let $\gamma = \{\gamma_1, \ldots, \gamma_k\}$ be a collection of closed curves on Σ. Is there a polynomial-time algorithm to compute a set of simple closed walks $C = \{C_1, \ldots, C_k\}$ in G, such that each C_i is homotopic to γ_i?

Result

It turns out that Schrijver [1] had studied this problem before and obtained the following characterization, which is conjectured by Lovász and Seymour [2, Section 76.7]. So instead of reporting on our discussions, we present a sketch of Schrijver's proof.

Given an embedded graph G and two closed curves γ and δ on Σ, define $\mathbf{cr}(\boldsymbol{G};\boldsymbol{\delta})$ to be the number of crossings between G and δ, and $\mathbf{mincr}(\boldsymbol{\gamma};\boldsymbol{\delta})$ be the minimum number of crossings between γ' and δ', where γ' is homotopic to γ and δ is homotopic to δ'. A closed curve δ is ***doubly-odd*** with respect to G and multicurve γ if $\delta = \delta_1 \cdot \delta_2$ for some closed curves δ_1 and δ_2, satisfying

$$\mathrm{cr}(G;\delta_1) \not\equiv \sum_i \mathrm{cr}(\gamma_i, \delta_1) \pmod 2, \text{ and}$$
$$\mathrm{cr}(G;\delta_2) \not\equiv \sum_i \mathrm{cr}(\gamma_i, \delta_2) \pmod 2.$$

The ***double point*** of δ is equal to $\delta_1(0) = \delta_2(0)$.

▶ **Theorem 1** (Schrijver [1]). *Let Σ be a compact surface and G be a graph embedded in Σ. Let $\gamma = \{\gamma_1, \ldots, \gamma_k\}$ be a collection of closed curves on Σ. There are simple closed walks $C = \{C_1, \ldots, C_k\}$ in G where each C_i is homotopic to γ_i if and only if the all the following properties hold:*

1. *there are pairwise disjoint simple closed curves* $\gamma'_1, \ldots, \gamma'_k$ *on surface* Σ *(not necessarily in* G*),*
2. *for each closed curve* δ *on* Σ*,*

$$\mathrm{cr}(G;\delta) \geq \sum_i \mathrm{mincr}(\gamma_i, \delta); \textit{ and}$$

3. *for each doubly-odd closed curve* $\delta = \delta_1 \cdot \delta_2$ *whose double point is not on* G*,*

$$\mathrm{cr}(G;\delta) > \sum_i \mathrm{mincr}(\gamma_i, \delta).$$

In the book by Schrijver [2, Section 76.7] it is also claimed that such a characterization leads to a polynomial-time algorithm. However this is not immediately clear from the characterization. We give a short exposition on Schrijver's proof using modern language in discrete and computational topology.

Sketch of proof

Necessity. First we prove the easy direction that the three conditions in Theorem 1 are all necessary. Let $C_1, \ldots, C_k$ be the simple closed walks in G promised by the theorem. Condition 1 is satisfied easily by taking $\gamma'_1, \ldots, \gamma'_k$ to be $C_1, \ldots, C_k$. Condition 2 is also straightforward:

$$\mathrm{cr}(G;\delta) \geq \sum_i \mathrm{cr}(C_i, \delta) \geq \sum_i \mathrm{mincr}(\gamma_i, \delta),$$

where the second inequality follows by the fact that C_i is homotopic to γ_i for all i. As for Condition 3, let $\delta = \delta_1 \cdot \delta_2$ be any doubly-odd closed curve whose double point is not on G. For both $j \in \{1, 2\}$, one has $\mathrm{cr}(G;\delta_j) \geq \sum_i \mathrm{cr}(C_i, \delta_j)$ immediately, and the inequality must be strict because

$$\mathrm{cr}(G;\delta_j) \not\equiv \sum_i \mathrm{cr}(\gamma_i, \delta_j) \equiv \sum_i \mathrm{cr}(C_i, \delta_j) \pmod 2$$

because δ is doubly-odd and parity of number of intersections is unchanged under homotopy. Thus,

$$\begin{aligned}
\mathrm{cr}(G;\delta) &= \mathrm{cr}(G;\delta_1) + \mathrm{cr}(G;\delta_2) \\
&> \sum_i \mathrm{cr}(C_i, \delta_1) + \sum_i \mathrm{cr}(C_i, \delta_2) \\
&= \sum_i \mathrm{cr}(C_i, \delta) \\
&\geq \sum_i \mathrm{mincr}(C_i, \delta).
\end{aligned}$$

Sufficiency. Here we present the proof when γ is a single closed curve and the goal is to find one simple closed walk C; the proof for the general case is a rather straightforward extension. Without loss of generality we can assume that Σ is orientable and not topologically a sphere [1, Claim 2], and G is cellularly embedded in Σ [1, Claim 1]. We can also safely assume that the closed curve δ in conditions of Theorem 1 does not cross G or its dual G^* at the edges. This implies that $\mathrm{cr}(G;\delta) = \mathrm{cr}(G^*;\delta)$. We can recast the problem to the dual graph G^*; the existence of disjoint simple closed walk C in G is equivalent to the following:

There is a simple closed curve C^* not intersecting the vertices of G^*, such that C^* is homotopic to γ and each face of G^* is traversed by C^* at most once.

Now we describe how to construct the set of simple closed walk C^* satisfying the above properties. Let γ be the input closed curve. By Condition 1 we can assume without loss of generality that γ itself is a *simple* closed curve and not intersecting any vertices of G^*; we also assume that γ has transverse intersections with G^*. In other words, γ is a *normal curve* with respect to G^*. Consider a face f of G^* together with the portion of γ intersecting f; we call the collection of arcs from the intersection the ***curve parts*** of f. For each component κ of the curve parts, draw an infinitesimally short segment crossing κ in a way that all short segments are disjoint. We call the collection of endpoints of all short segments as ***terminals*** and denoted as $\boldsymbol{T}$. The two endpoints of the same segment are called a ***terminal pair***.

We are going to set up a linear program with variables $\psi(\cdot)$ indexed by T, and set the constraints properly so that a solution to the linear program gives us hint on how to find the correct homotopy to turn γ into C^*. Consider the following linear program:

$$\boxed{\begin{array}{lll} \min & \displaystyle\sum_{t\in T} |\psi(t)| & \\ \text{s.t.} & \psi(t)+\psi(\bar{t}) = 0 & \text{for each terminal pair } t \text{ and } \bar{t}, \\ & \psi(t)+\psi(t') \le \lambda(tt') & \text{for each pair of } t \text{ and } t' \text{ in } T, \end{array}}$$

where $\boldsymbol{\lambda(tt')} := \min_\pi \mathrm{cr}(G^*;\pi)-1$, with the minimum taking over all paths π on Σ connecting t to t', such that the two endpoints of the lift $\hat{\pi}$ of π in the universal cover $\hat{\Sigma}$ connects different lifts of γ, and $\hat{\pi}$ lies in the same component of $\hat{\Sigma}$ subtracting $\hat{\gamma}$. Schrijver showed that the existence of a solution to the above linear program is equivalent to the properties in Theorem 1 [1, Section 2 and Claim 3].

Now here comes the important definitions. Recall that $\hat{\Sigma}$ is the universal cover of Σ and everything with a hat is a lift of the corresponding object in Σ. Let $\boldsymbol{\mathring{\Sigma}_\gamma}$ be the ***cyclic cover*** of Σ with respect to γ. Lifts $\mathring{G}$ and $\mathring{\gamma}$ are defined accordingly. Notice that $\mathring{\gamma}$ is a simple closed curve in $\mathring{\Sigma}$. The notion of terminals and function $\psi(\cdot)$ associated with the linear program can be lifted and defined with respect to the universal cover and cyclic cover as well.

For any lift ℓ of γ and any lift $\hat{f}$ of a face f of G^*, define

$$\boldsymbol{\Pi_\ell(\hat{f})} := \min_{\hat{\pi}} \big(\mathrm{cr}(\hat{G}^*;\hat{\pi}) - \psi(\pi(0))\big),$$

where $\hat{\pi}$ ranging over all paths in Σ that starts at some lift of a terminal corresponding to a curve part from ℓ that projects to $\pi(0)$ and ends in $\hat{f}$, and crosses ℓ an even number of times. Notice by definition if some $\hat{f}$ intersects ℓ then $\Pi_\ell(\hat{f}) \le 0$. Define ***zero (dual) faces*** $\boldsymbol{F_0(\ell)}$ to be the collection of faces $\hat{f}$ in $\hat{G}^*$ with $\Pi_\ell(\hat{f}) = 0$. Define ***non-positive (dual) vertices*** $\boldsymbol{V_{\le 0}}$ to be the collection of vertices v of $\mathring{G}^*$ that has a path $\mathring{\pi}$ starting from a terminal in $\mathring{\Sigma}$ that projects to $\pi(0)$ and ending at v that crosses $\mathring{\gamma}$ an even number of times, such that

$$\mathrm{cr}(\mathring{G}^*;\mathring{\pi}) - \psi(\pi(0)) \le 0.$$

Based on the upper bound on $\mathrm{cr}(\mathring{G};\mathring{\pi})$, the number of non-positive vertices in $V_{\le 0}$ must be finite.

Equipped with these definitions we are now ready to describe the construction of C. Let $[\mathring{\gamma}]$ be the representative of the *homology class* of $\mathring{\gamma}$ over $\mathbb{Z}_2$; as γ does not intersect vertices of G^*, $\mathring{\gamma}$ can be viewed as a closed walk in $\mathring{G}$. Here we abuse the type and refer to $[\mathring{\gamma}]$ as a subset of edges of $\mathring{G}$. Define $\boldsymbol{E_0}$ to be the symmetric difference between $[\mathring{\gamma}]$ and the (dual) edge cut formed by $V_{\leq 0}$ (which is a collection of cycles, a valid homology class). Among the simple cycles of E_0 in $\mathring{G}$, there is at least one simple cycle $\mathring{\boldsymbol{C}}$ homotopic to $\mathring{\gamma}$ on $\mathring{\Sigma}$ [1, Claim 6]. Project $\mathring{C}$ back to Σ to obtain the desired cycle $\boldsymbol{C}$ in G. Important properties of these definitions are

1. if $\mathring{C}$ passes through some face $\mathring{f}$ in $\mathring{G}^*$, then any lift of $\mathring{f}$ must be in $F_0(\ell)$ [1, Claims 6 and 7]; and
2. each pair of $F_0(\ell)$ and $F_0(\ell')$ are disjoint if $\ell \neq \ell'$ [1, Claim 5].

Finally it is sufficient to show that C is indeed simple (or, vertex-disjoint); in other words, C^* never passes through any dual face in G^* more than once. The construction guarantees that $\mathring{C}$ is simple, so it must be the case that two faces $\mathring{f}$ and $\mathring{g}$ of $\mathring{G}^*$ passed by $\mathring{C}$ projects to the same face in G^*. Now $F_0(\ell)$ must contain both $\hat{f}$ and $\hat{g}$ by Property (1). Consider the deck transformation ϕ on $\hat{\Sigma}$ that maps face $\hat{f}$ to $\hat{g}$. Now $\phi(\ell)$ is another lift of γ. Since $\hat{g} = \phi(\hat{f}) \in \phi(F_0(\ell)) = F_0(\phi(\ell))$ as well, by Property (2) above ℓ and $\phi(\ell)$ must be the same lift of γ. This implies that after projecting to $\mathring{\Sigma}$ the two faces $\mathring{f}$ and $\mathring{g}$ must be identical. Therefore, C^* never passes through any dual face in G^* more than once.

Efficient implementation

To carry out a polynomial-time algorithm, a few questions remain.

- How do we compute every $\lambda(tt')$ in polynomial-time to set up the linear program?
- How to compute cycles $\mathring{C}$ and C? In particular, how to construct non-positive vertices $V_{\leq 0}$?

References

1 Alexander Schrijver. Disjoint circuits of prescribed homotopies in a graph on a compact surface. *Journal of Combinatorial Theory, Series B* 51(1):127–159. Elsevier, 1991.

2 Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency.* Algorithms and Combinatorics 24. Springer-Verlag, 2003.

4.5 Beautiful curves on beautiful surfaces

Saul Schleimer (University of Warwick – Coventry, GB), Vincent Despré (INRIA Nancy – Grand Est, FR), Francis Lazarus (GIPSA Lab – Grenoble, FR), Hugo Parlier (University of Luxembourg, LU), Stephan Tillmann (University of Sydney, AU), and Erin Moriarty Wolf Chambers (St. Louis University, US)

We considered the problem of how to draw "beautiful curves on beautiful surfaces".

That is, suppose we are given a closed connected oriented surface S in the form of a very nice mesh in $\mathbb{R}^3$. Suppose that we are also given a simple closed curve α in S, say as the (circular) sequence of mesh edges it crosses. We wish to find a "beautiful" representative of α.

Curves on the torus

For the torus, we can argue that the Clifford torus (stereographically projected from the three-sphere) is the "best" geometric torus. Now, for any curve α there is an optimal position,

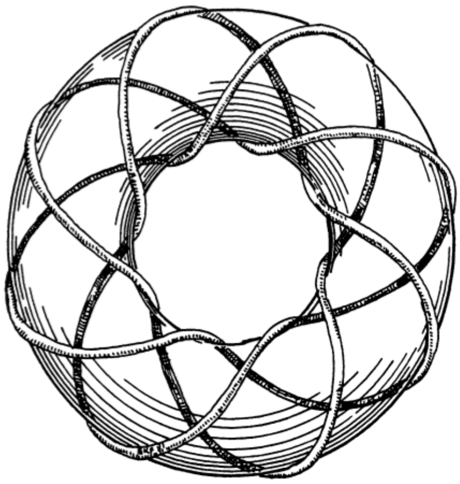

Figure 5 From page 38 of *A topological picturebook* by George Francis [1].

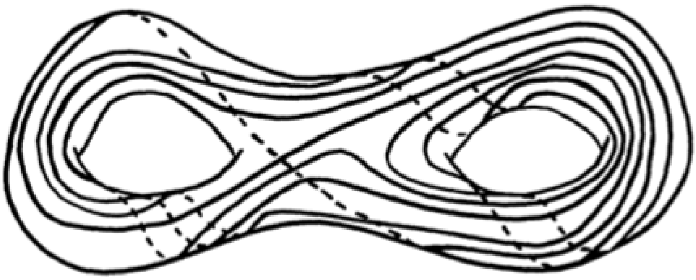

Figure 6 From page 421 of *On the geometry and dynamics of diffeomorphisms of surfaces* by William Thurston [2].

making α both as straight as possible and also as well-spaced as possible. To see this, consider the Clifford torus in S^3 and stereographically project it to $\mathbb{R}^3$. Since the Clifford torus is square, there are "obvious" representatives (straight, going through the origin). These are best both in terms of straightness and well-spacedness.

Higher genus

In higher genus things are less clear-cut. It is much harder to argue for a particular "best" geometric surface. Note that in the picture by Thurston the surface is a bit pinched along the outer two non-separating curves.

It is also now much harder to decide what constitutes a "best" representive of a given isotopy class of curve. Thurston's curve shows some of the issues involved, especially when compared to the standard short curves. Note that in Thurston's curve it is not so easy to "see" the number of components, or the topological type.

Also, Thurston has preferred to make the curve "well-spaces" rather than "straight". This gives a few places where the geodesic curvature is "too high". Thus we see that there is a tension between drawing the curve straight and having good "interarc" spacing. The former takes us towards geodesic laminations, the latter towards quadratic differentials.

Teruaki

One implemented solution (in genus two) is given by Kazushi Ahara as part of his game Teruaki exploring the mapping class group.

http://www.aharalab.sakura.ne.jp/teruaki.html

This is makes for a very interesting game! Note that if you perform large powers of Dehn twists, then the programme gets confused; curves may crash into themselves and may also drift off of the surface.

Plans of attack

We can think about two ways to approach the problem. Fix a good triangular mesh Δ of the surface $S \subset \mathbb{R}^3$. Our first approach concentrates on making the curves straight.

Hyperbolic geometry: Cut open Δ; this gives a fundamental domain D. Lay D out conformally in the hyperbolic plane. Compute the resulting Fuchsian group. Given an element of $\pi_1(S)$ draw all of its intersections D. Push these forward to $\Delta \subset \mathbb{R}^3$.

Our second approach concentrates on making the curves well-spaced.

Quadratic differentials: For any simple closed curve $\alpha \subset S$, the *Strebel differential* $q = q_\alpha$ is the unique quadratic differential in the conformal class of S, of area one, so that all non-singular flowlines are isotopic to α. So the periods of q are given and we must solve for the differential. Now compute the "core-curve" of the maximal cylinder in q.

References

1. George K. Francis. *A topological picturebook*. Springer, New York, 2007. Reprint of the 1987 original.
2. William P. Thurston. On the geometry and dynamics of diffeomorphisms of surfaces. *Bull. Amer. Math. Soc. (N.S.)*, 19(2):417–431, 1988.

4.6 Minimum Area Homotopies

Carola Wenk (Tulane University – New Orleans, US), Hsien-Chih Chang (Duke University – Durham, US), Vincent Despré (INRIA Nancy – Grand Est, FR), Francis Lazarus (GIPSA Lab – Grenoble, FR), Anna Lubiw (University of Waterloo, CA), Tim Ophelders (Michigan State University, US), Hugo Parlier (University of Luxembourg, LU), and Erin Moriarty Wolf Chambers (St. Louis University, US)

Given a closed curve γ embedded in the plane, can we compute a "nice" homotopy in polynomial time that minimizes the homotopy area contracting γ to a point?

Chambers and Wang have introduced the notion of minimum-area homotopies as a distance measure between two curves [1]. They presented a polynomial-time dynamic programming algorithm for the case that the winding numbers, induced by the closed curve composed of the two open curves, are consistent (i.e., all non-negative or all non-positive). For the general case, Nie has presented a polynomial-time algorithm [2] that is algebraic in nature and results in homotopies with degeneracies. Recently it has been shown that one can express Nie's algorithm entirely using such (degenerate) homotopies that either collapse a face or cancel around a face [3]. On the other hand, it has been shown that there is always a minimum-area homotopy that can be represented as a decomposition of self-overlapping curves [4]. Contracting each self-overlapping curve in such a decomposition to a point then results in a "nice" homotopy. However, the algorithm in [4] to compute such a decomposition is exponential. A related paper on combinatorial properties of self-overlapping curves and interior boundaries [5], that introduces the notions of obstinance and wrapping, helps in understanding the relationships between minimum area homotopies and self-overlappingness of curves in the plane.

Some of the insights and related problems discussed during the Dagstuhl seminar are as follows:

1. Can we check whether a curve is self-overlapping in less than $O(N^3)$ time? Here, N is the number of vertices of a polygonal input curve. Shor and Van Wyk's dynamic programming algorithm [6] runs in $O(N^3)$ time.
2. If we know how many times a face is swept in a minimum-area homotopy (e.g., from Nie's algorithm), can we compute an area-optimal self-overlapping curve decomposition? Alternatively, is computing an area-optimal self-overlapping decomposition NP-hard?

Insights:

- Blank's algorithm [7] for computing whether a curve is self-overlapping runs in time quadratic to the depth-sum of the curve, and is therefore faster than Shor and van Wyk's for shallow curves.
- All $2 \to 0$ moves and two of the three $0 \to 2$ moves preserves self-overlappingness; the last $0 \to 2$ move requires the local potentials to be positive.
- Any tree-like self-overlapping curve [8, 9] is simple. A possible strategy: First add a "shell" to the input curve, then use regular homotopy to shrink the curve. With enough shelling all $2 \leftrightarrow 0$ moves can be executed safely.

Other questions that were discussed were how to decide self-overlappingness of a curve that is embedded on a surface, for a example on a sphere or on a higher genus surfaces. For curves in the plane it was discussed whether the casings by Eppstein and Mumford [10] help in determining that a curve is self-overlapping. In this case a 2-move must stay consistent, both over-over or under-under; is there a relation to Morse theory to find a height function?

References

1 E.W. Chambers and Y. Wang. Measuring similarity between curves on 2-manifolds via homotopy area. ACM Sympos. Comput. Geom. (SoCG), 2013.
2 Z. Nie. On the Minimum Area of Null Homotopies of Curves Traced Twice. *arXiv:* 1412.0101, 2014.
3 B.T. Fasy, B. McCoy, D. Millman, C. Wenk. A Geometric Interpretation of the Cancellation Norm. In preparation, 2019.
4 B.T. Fasy, S. Karakoc, C. Wenk. On Minimum Area Homotopies of Normal Curves in the Plane. *arXiv:* 1707.02251, 2017.
5 P. Evans and C. Wenk. Combinatorial Properties of Self-Overlapping Curves Through Minimum Homotopy Area. In submission, 2019.
6 P.W. Shor and C.J. Van Wyk. Detecting and decomposing self-overlapping curves. *Computational Geometry: Theory and Applications* 2(1): 31-50, 1992.
7 Samuel J. Blank. Extending Immersions and regular Homotopies in Codimension 1. *Ph.D. Dissertation, Brandeis University*, 1967.
8 V. Arnold. Plane curves, their invariants, perestroikas and classifications. *Advances in Soviet Mathematics* 21, 1994.

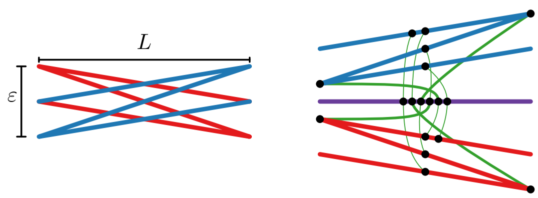

Figure 7 The optimal isotopy for the simple zigzag. (The green trajectories are actually straight-line; they are drawn as curves for better visibility.) For further details see [7].

9 F. Aicardi. Tree-like curves. *Advances in Soviet Mathematics* 21, 1994.
10 D. Eppstein and E. Mumford. Self-overlapping curves revisited. *20th ACM-SIAM Symp. Discrete Algorithms*: 160-169, 2009.

4.7 Nice Morphs and Isotopic Frechet Distance

Erin Moriarty Wolf Chambers (St. Louis University, US), Vincent Despré (INRIA Nancy – Grand Est, FR), Linda Kleist (TU Braunschweig, DE), Maarten Löffler (Utrecht University, NL), Anna Lubiw (University of Waterloo, CA), Tim Ophelders (Michigan State University, US), Hugo Parlier (University of Luxembourg, LU), Stephan Tillmann (University of Sydney, AU), Birgit Vogtenhuber (TU Graz, AT), and Carola Wenk (Tulane University – New Orleans, US)

Introduction

Fréchet distance is a way of measuring the distance between two simple (i.e., non-self-intersecting) curves γ_0 and γ_1 in the plane. Informally, we want the minimum leash length that allows a person to travel along γ_0 while their dog travels along γ_1. Either one may vary their speed but neither may back up. There is a polynomial time algorithm to compute Fréchet distance between two curves [2]. Any travel plan that realizes the minimum leash length yields a *Fréchet matching* between the points of γ_0 and the points of γ_1. This in turn determines a continuous deformation from γ_0 to γ_1, as each point moves along the taut leash (say at uniform speed) between the matched points. However, in the course of this continuous deformation, the curve will not remain simple in general.

Adding the further constraint that the curve should remain simple throughout the deformation yields the notion of *isotopic Fréchet distance*, which can be defined as follows. A *morph* from γ_0 to γ_1 is a continuous family of simple curves γ_t, indexed by time t, $0 \leq t \leq 1$ so that the curve at time $t = 0$ is γ_0 and the curve at time $t = 1$ is γ_1. A morph determines a trajectory $p(t), 0 \leq t \leq 1$ of each point $p = p(0)$ in γ_0 to its destination point $p(1)$ in γ_1. The minimum over all morphs of the maximum trajectory length is the *isotopic Fréchet distance*, first introduced in [3]. An example is shown in Figure 7.

We explored the problem of finding a "nice" morph between γ_0 and γ_1 to realize or approximate the isotopic Fréchet distance.

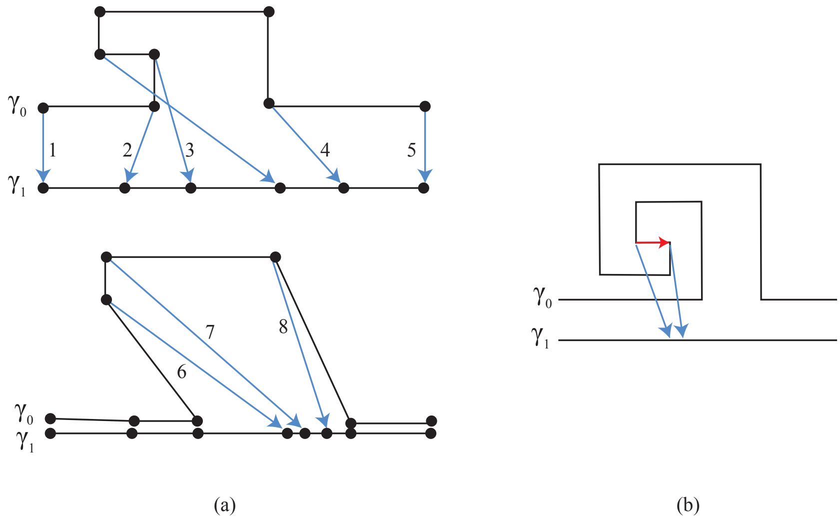

Figure 8 (a) γ_0 morphs to γ_1 as vertices travel on the blue trajectories in the order specified; (b) no morph is possible for these trajectories because the segment of γ_0 drawn in red must rotate 360° degrees but the trajectories do not allow that.

Related Work

When the two curves do not cross each other but share the same start and end points, they determine a simple polygon. Efrat et al. [4] gave an algorithm for the version of Fréchet distance where the leash must remain inside the polygon. In this situation, the leash paths will not cross each other so the Fréchet mapping yields a morph between the curves.

Another relevant related problem is to morph between two simple polygons while preserving edge lengths as much as possible [5]. For results on morphing planar graphs, see [1] and references therein.

Morphing along the Fréchet leashes

One approach we explored was to morph by moving each vertex along its Fréchet "leash" but not necessarily at uniform speed. Given a mapping (perhaps not even a Fréchet mapping) between γ_0 and γ_1, subdivide γ_0 and γ_1 so that each vertex of one maps to a vertex of the other. This gives a set of "leashes", each one a straight line segment, between [the expanded set of] vertices of γ_0 and γ_1. Now we try to move each vertex along its leash path so that the curve remains simple at all times, using the freedom that vertices need not travel at uniform speed.

This is not always possible. However, in the special case when both curves are x-monotone (increasing) then it is possible as shown by Tim Ophelders in his thesis [7]. We examined the case where the two curves do not intersect and found a counter-example—see Figure 8.

It would be interesting to prove NP-hardness for the decision problem (can vertices be moved on the given leash paths so that the curve remains simple). One related result is that it is NP-complete to decide if we can morph some segments from initial to final positions with the restriction that the segments never cross and every vertex moves along a straight line from initial to final position [10, Chapters 6,7]. As above, the freedom is that vertices need not travel at uniform speeds.

A lower bound

Suppose a subpath ab of γ_0 should morph to a subpath $a'b'$ of γ_1. It may need to turn 360° (as in Figure 8), or some multiple $k \cdot 360°$. Imagine morphing the subpath ab to a point x, rotating it by $k \cdot 360°$, and then morphing to $a'b'$. To minimize the maximum trajectory length, we should choose x so that $\max(|ax| + |xa'|, |bx| + |xb'|)$ is minimized. We show that this provides a lower bound on the isotopic Fréchet distance.

Let α and β be the paths from a to a' and from b to b' obtained from an optimal morph, and assume that the path $t \mapsto \beta(t) - \alpha(t)$ winds (clockwise or counterclockwise) around the origin for at least 180°. We claim that the cost of the morph is at least the minimum value of $\max(|ax|+|xa'|, |bx|+|xb'|)$ over all x. The path α lies inside the ellipse A with foci a and a' and a major axis of length $\|\alpha\|$, and β lies inside the ellipse B with foci b and b' and a major axis of length $\|\beta\|$. For any $x \in A \cap B$, we have $\|\alpha\| \geq |ax| + |xa'|$ and $\|\beta\| \geq |bx| + |xb'|$, so it suffices to show that A intersects B. Since the morph is an isotopy, we have $\alpha(t) \neq \beta(t)$, so we can define $\theta(t)$ to be the angle of the (directed) segment from $\alpha(t)$ to $\beta(t)$. Because of the winding, there exist t and t' such that $\theta(t) = \theta(t') + 180°$. This defines a trapezoid with one diagonal connecting $\alpha(t)$ and $\alpha(t')$, and the other diagonal connecting $\beta(t)$ and $\beta(t')$. By convexity of A and B, the first diagonal lies in A and the second lies in B, and since diagonals of a trapezoid intersect, A intersects B.

We believe that there exist mappings between curves that result in an optimal morph that is more expensive than both the lower bound described above, and the (homotopic) Fréchet distance. In particular, the above lower bound essentially requires α to have only a single bend, namely infinitesimally close to the unique point x that minimizes $\max(|ax|+|xa'|, |bx|+|xb'|)$. It is likely that we can simultaneously force the α leash to bend at a unique point $y \neq x$ that minimizes $\max(|ay| + |ya'|, |cy| + |yc'|)$, which violates the lower bound. It is unclear whether this violation can actually be realized in an optimal Fréchet isotopy.

Three-dimensional approaches

If we view time as the third dimension then we have two curves, say γ_0 in the $z = 0$ plane and γ_1 in the $z = 1$ plane, and we wish to find a "nice" surface joining them. The morph is then obtained by slicing the surface in successive planes. For morphs of planar graphs, this interpretation is explicitly discussed by Surazhsky and Gotsman [9]. We discussed soap bubbles (fascinating but tricky) and PL-minimal surfaces [6].

One idea is to triangulate the top and bottom planes with many triangles, and then triangulate the region between them with "nice" tetrahedra. We need to then find a normal surface with boundary equal to the two curves, for which techniques as in [8] might be used. One complication is that the minimal surface that solves this discrete version of the Plateau problem may not be an annulus bounded by the two curves, but have higher genus. Thus, either one needs a triangulation that is closely adapted to the local geometry of the curves, or a more sophisticated clean-up step would be required in order to obtain the desired morph.

A piecewise linear approach

An alternative approach, coming from classical PL topology, is to relate the curves by a sequence of elementary moves. An elementary move replaces two edges of a triangle with the third, or vice-versa. The main problem with this approach is to determine an objective function defining a "nice morph". One possibility is to ask for a minimal sequence of elementary collapses.

Conclusions

It seems that finding an isotopy (a simplicity preserving morph) that minimizes the maximum trajectory (i.e., isotopic Fréchet) is really at odds with finding a “nice” morph. This is because solutions to isotopic Fréchet shrink spiraling parts of the curve infinitesimally small and then unspiral them.

References

1 Soroush Alamdari, Patrizio Angelini, Fidel Barrera-Cruz, Timothy M Chan, Giordano Da Lozzo, Giuseppe Di Battista, Fabrizio Frati, Penny Haxell, Anna Lubiw, Maurizio Patrignani, Vincenzo Roselli, Sahil Singla, and Bryan T. Wilkinson. How to morph planar graph drawings. *SIAM J. Computing*, 46(2):29 pages, 2017.
2 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5(01n02):75–91, 1995.
3 Erin W. Chambers, David Letscher, Tao Ju, and Lu Liu. Isotopic Fréchet distance. In *CCCG*, 2011.
4 Alon Efrat, Leonidas J. Guibas, Sariel Har-Peled, Joseph S. B. Mitchell, and T. M. Murali. New similarity measures between polylines with applications to morphing and polygon sweeping. *Discrete & Computational Geometry*, 28(4):535–569, 2002.
5 Hayley N. Iben, James F O’Brien, and Erik D Demaine. Refolding planar polygons. *Discrete & Computational Geometry*, 41(3):444–460, 2009.
6 William Jaco, J. Hyam Rubinstein. PL minimal surfaces in 3-manifolds. *Journal of Differential Geometry*, 27(3):493–524, 1988.
7 Tim Ophelders. *Continuous Similarity Measures for Curves and Surfaces.* PhD thesis, Eindhoven University of Technology, Netherlands, 2018.
8 Jonathan Spreer. Normal surfaces as combinatorial slicings. *Discrete Mathematics*, 311(14):1295 – 1309, 2011.
9 Vitaly Surazhsky and Craig Gotsman. High quality compatible triangulations. *Engineering with Computers*, 20(2):147–156, 2004.
10 Hamideh Vosoughpour. *Straight Line Movement in Morphing and Pursuit Evasion.* PhD thesis, University of Waterloo, Canada, 2017.

4.8 Representing Graphs by Polygons with Edge Contacts in 3D

Alexander Wolff (Universität Würzburg, DE), Elena Arseneva (St. Petersburg State University, RU), Arnaud de Mesmay (University of Grenoble, FR), Linda Kleist (TU Braunschweig, DE), Boris Klemz (FU Berlin, DE), Maarten Löffler (Utrecht University, NL), André Schulz (FernUniversität in Hagen, DE), and Birgit Vogtenhuber (TU Graz, AT)

Evans et al. [8] showed that every graph has a contact representation in 3D in which each vertex is represented by a (flat) convex polygon, two polygons touch *in a corner* if and only if the corresponding two vertices are adjacent, and the interiors of any two polygons are disjoint. In this short note, we investigate representations where polygons that correspond to adjacent vertices must share an edge, rather than a corner.

We first allow our polygons to be nonconvex. In this case, we can easily represent every graph; see Figure 9.

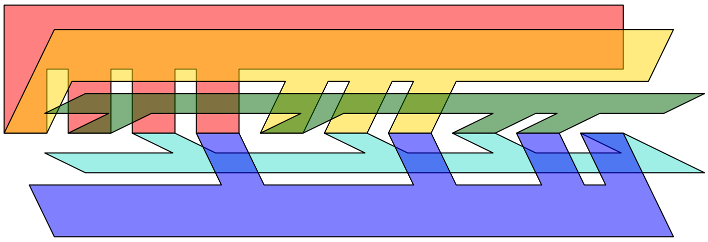

Figure 9 A realization of K_5 by nonconvex polygons with edge contacts.

Figure 10 The Szilassi polyhedron realizes K_7 by nonconvex polygons with edge contacts [3].

▶ **Observation 1.** Every graph can be realized by polygons with edge contacts in 3D.

Proof. We sketch how to obtain a realization. To represent a graph G with n vertices, we start with n rectangles such that the intersection of all these rectangles is a line segment s. We then cut away parts of each rectangle thereby turning it into a comb shaped polygon; see Figure 9. The goal of this step is to ensure that for each pair (P, P') of polygons, there is a subsegment s' of s such that s' is an edge of both P and P' that is disjoint from the remaining polygons. The result is a representation of K_n. To obtain a realization of G, it remains to remove edge contacts that correspond to unwanted adjacencies, which is easy. ◀

If we additionally insist that each polygon shares all of its edges with other polygons, the polygons describe a closed volume. In this model, K_7 can be realized as the Szilassi polyhedron; see Figure 10. The tetrahedron and the Szilassi polyhedron are the only two known polyhedra in which each face shares an edge with each other face [3]. Which graphs other than complete graphs can be represented by edge contacts of non-convex polygons where every edge must be shared, remains an open problem.

We now consider the setting where each vertex of the given graph is represented by a convex polygon in 3D and two vertices of the given graph are adjacent, if and only if their polygons have edge contact. (So far, most people have only insisted that the edge of one polygon is contained in the edge of the adjacent polygon. For example, Duncan et al. [1] showed that in this model every planar graph can be realized by hexagons in the plane and that hexagons are sometimes necessary.) Note that it is allowed to have edges that do not touch other polygons. We start with some simple observations.

▶ **Observation 2.** Every planar graph can be realized by convex polygons with edge contacts in 2D.

Proof. Let G be a planar (embedded) graph. Add to G a new vertex r and connect it to all vertices of some face. Let G' be a triangulation of the resulting graph. Then the dual of G', $G^\star$, is a cubic 3-connected planar graph. Using Tutte's barycentric method, draw $G^\star$ into a regular polygon with $\deg_{G'}(r)$ corners such that the face dual to r becomes the outer face. Note that the interior faces in this drawing are convex polygons; the polygon that corresponds to a vertex v of G' has $\deg_{G'}(v)$ corners. To convert this contact representation of $G' - r$ into a contact representation of G, we may need to remove some edge contacts, which can be easily achieved.

The same can be shown as follows. Using the classical result of Koebe, take a contact representation of the given planar graph by touching disks. For each pair of touching disks, place a very short line segment on their common tangent such that the line segment is centered on the touching point. Then represent every vertex of the given graph by the convex hull of the line segments that touch its disk. If the line segments are short enough, every two of the resulting convex polygons are interior disjoint. By construction, the polygons of adjacent vertices share an edge. Each polygon has twice as many edges as the degree of the corresponding vertex. ◀

So for planar graphs vertex and edge contacts behave similarly. For nonplanar graphs (for which we need the third dimension), the situation is different. Here, edge contacts are more restrictive. We introduce the following notation. In a 3D representation of a graph G by polygons, we denote by p_v the polygon that represents vertex v of G.

▶ **Lemma 1.** *Let G be a graph. Consider a 3D edge-contact representation of G with convex polygons. If G contains a triangle uvw, polygons p_v and p_w lie on the same side of the plane that supports p_u.*

Proof. Due to the convexity of the polygons, p_v and p_w either both lie above or both lie below the plane that supports p_u, otherwise p_v and p_w cannot share an edge. In this case, the edge vw of G would not be represented; a contradiction. ◀

▶ **Observation 3.** For $n \geq 5$, K_n is not realizable by convex polygons with edge contacts in 3D.

Proof. Assume that K_n admits a 3D edge-contact representation. Since every three vertices in K_n are pairwise connected, by Lemma 1, for every polygon of the representation, its supporting plane has the rest of the complex on one side. In other words, the complex we obtain is a subcomplex of a convex polyhedron. Consequently, the dual graph has to be planar, which rules out K_n for $n \geq 5$. ◀

▶ **Observation 4.** $K_{4,4}$ is realizable by convex polygons with edge contacts.

Proof. We sketch how to obtain a realization. Start with a box in 3D and intersect it with two rectangular slabs as indicated in Figure 11 on the left. We can now draw polygons on the faces of this complex such that every vertical face contains a polygon that has an edge contact with a polygon on a horizontal or slanted face. The polygons on the slanted faces lie in the interior of the box and intersect each other. To remove this intersection we pull out one corner of the original box (see Figure 11). ◀

In contrast to Observation 4, we believe that the analogous statement does not hold for all bipartite graphs, i.e., we conjecture the following:

▶ **Conjecture 1.** There exist values n and m such that the complete bipartite graph $K_{m,n}$ is not realizable by convex polygons with edge contacts.

By Observation 2, all planar 3-trees can be realized by convex polygons with edge contacts (even in 2D). When switching to 3D, also nonplanar 3-trees can have a realization by convex polygons with edge contacts; see for example Figure 12. However, this is not the case for all nonplanar 3-trees.

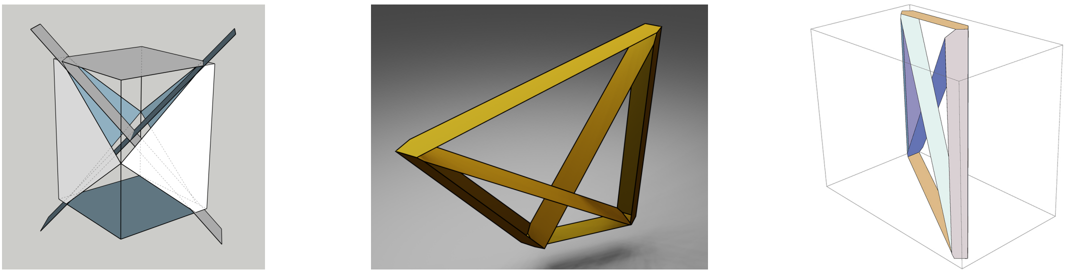

Figure 11 A realization of $K_{4,4}$ by convex polygons with edge contacts.

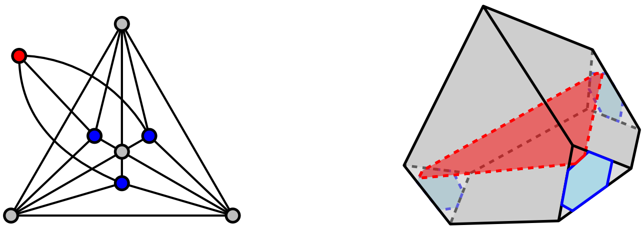

Figure 12 A nonplanar 3-tree with a realization by convex polygons with edge contacts. The gray vertices form a K_4.

▶ **Observation 5.** Not all 3-trees can be realized by convex polygons with edge contacts.

Proof. Consider the 3-tree in Figure 13, which consists of $K_{3,3}$ plus a cycle that connects the gray vertices of one part of the bipartition. The other part of the bipartition consists of three colored vertices (red, green, blue). For the sake of contradiction, assume that there is a representation by convex polygons with edge contacts and distinguish two cases: Either the three polygons are coplanar or not.

If the gray polygons are coplanar, then all edge contacts must lie in the same plane. This, however, contradicts the fact that $K_{3,3}$ is not planar.

If the three edges are not coplanar, the gray polygons form a prism-like shape. Note that every colored vertex together with the gray vertices forms a K_4. Hence, by Lemma 1, all gray polygons must lie on one side of the supporting plane of a colored polygon. Each supporting plane of a colored polygon intersects the gray triangular prism in a triangle. Two of these triangles must intersect (otherwise one of the outer colored polygons would be cut off by the middle colored polygon and would not have any contact with the gray polygons). The two intersecting triangles (say, the red and the green) cross each other exactly twice, and the two points of intersection lie on two distinct sides s_1 and s_2 of the gray prism; see Figure 14. Let s_3 denote the side of the gray prism that does not contain any of these intersection points. Each of the two crossing triangles intersects s_3 in a line segment. These two line segments partition s_3 into three regions. For $i \in \{1, 2, 3\}$, let p_i denote the gray polygon that lies on side s_i. Polygon p_3 lies in the middle (bounded) region of s_3, otherwise it cannot have an edge contact with both the red and the green polygon. The two crossing triangles partition each of the remaining two sides s_1 and s_2 into four regions. Two of these regions are unbounded; the other two are bounded and triangular. To realize edge contacts with p_3, polygons p_1 and p_2 have to be located in the triangular region of s_1 and s_2, respectively, that is adjacent to the middle region of s_3. However, in this case p_1 and p_2 cannot possibly touch; a contradiction. ◀

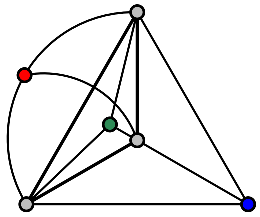

Figure 13 A 3-tree that is not realizable by convex polygons with edge contacts. The gray vertices form a 3-cycle.

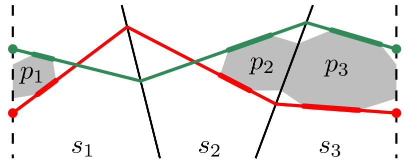

Figure 14 Schematic drawing of a potential realization. Net of the three gray polygons and traces of the planes that contain the red and green polygons, which must touch each of the gray polygons. The line of intersection between two of the gray polygons is drawn twice (dashed).

We can realize n-vertex graphs with $4n - O(1)$ edges by stacking horizontal polygons into a tetrahedron whose bottom face is horizontal and whose bottom corners have been cut off, but we can do better.

▶ **Theorem 2.** *There is a family of graphs $(G_k)_{k \geq 4}$ such that, for each $k \geq 4$, G_k has $n_k = k^2 + k - 3$ vertices and $5n_k - O(\sqrt{n_k})$ edges, and admits a realization by convex polygons with edge contacts.*

Proof. Let $k \geq 4$ be arbitrary but fixed. We construct G_k in several steps:

1. We start with a maximally triangulated planar graph H_k with m vertices, and designate a root vertex r of degree $c = 3$. Note that, for every $n \geq 4$, there exists a triangulation with a vertex of degree 3, e.g., a planar 3-tree.
2. Create a convex polytope P_k with H_k as its dual such that the face of r is *largest*, i.e., there exists a direction in which we can project the polytope such that the image is planar and r is the outer face.
3. Delete the face of r, and scale P_k until it is almost completely flat.
4. Imagine a k-sided cylinder. We will assume the axis of the cylinder is vertical.
5. Create k copies of P_k and place them at the faces of the cylinder, with the hollow side facing the inside of the cylinder. Each copy is rotated such that no edges are horizontal, and they are rotation-symmetric (around the cylinder axis) copies of each other.
6. For every vertex v of P_k not adjacent to r (corresponding to a face of H_k), create a regular k-gon with the k copies of v as its vertices. The k-gon will be horizontal.
7. Since v has (at least) three incident edges, either two go up and one down, or the other way around. In the first case, move the k-gon up by ε. In the second case, move the k-gon down by ε.
8. Cut a corner of each face between the two upgoing or downgoing edges, and glue the resulting horizontal edge to the k-gon.

We now have $n_k = k(m-1) + (2m-c)$ convex polygons with $k(3m-6-c) + k(2m-c) = 5km - (6+2c)k$ adjacencies. Using $c = 3$ and $m = k$ yields $n_k = k(m-1) + (2m-3) = k^2 + k - 3$ and $k(3m-6-3) + k(2m-3) = 5k^2 - 12k = 5n_k - 17k + 15 = 5n_k - O(\sqrt{n_k})$. ◀

Evans et al. [2] also considered representing *hypergraphs* in 3D. In their model, each hyperedge is represented by a (flat) convex polygon, two polygons share a corner if and only if the two hyperedges share a vertex, and any two polygons have disjoint interiors. They showed that the two smallest Steiner triple systems $S(2,3,7)$ and $S(2,3,9)$ admit such a representation (using triangles). In addition, they conjectured that no Steiner quadruple system (SQS) can be realized by using quadrilaterals in 3D.

We show that their conjecture is true for *convex* quadrilaterals. Assume that a SQS can be represented using convex quadrilaterals in 3D. Then the intersection of these quadrilaterals with a small sphere around a vertex is a planar graph. In a SQS $S(3, 4, n)$, each vertex is incident to $(n-1)(n-2)/6$ convex quadrilaterals, which can be split into $(n-1)(n-2)/3$ triangles. This yields a planar graph on $(n-1)$ vertices with $(n-1)(n-2)/3$ edges, which is impossible for $n > 9$. The same argument also precludes arbitrary topological embeddings into arbitrary 3-manifolds. Since Evans et al. [2] showed that $S(3, 4, 8)$ has no realization with quadrilaterals in 3D (and there is no SQS for $n = 9$), no SQS can be realized using convex quadrilaterals.

References

1 Christian A. Duncan, Emden R. Gansner, Y. F. Hu, Michael Kaufmann, and Stephen G. Kobourov. Optimal polygonal representation of planar graphs. *Algorithmica*, 63(3):672–691, 2012. doi:10.1007/s00453-011-9525-2.

2 William Evans, Paweł Rzążewski, Noushin Saeedi, Chan-Su Shin, and Alexander Wolff. Representing graphs and hypergraphs by touching polygons in 3D. In Daniel Archambault and Csaba Tóth, editors, *Proc. Int. Symp. Graph Drawing & Network Vis. (GD'19)*, LNCS, 2019, to appear. URL: http://arxiv.org/abs/1908.08273.

3 Szilassi polyhedron. Wikipedia entry. Accessed 2019-10-08. URL: https://en.wikipedia.org/wiki/Szilassi_polyhedron.

Participants

- Elena Arseneva
St. Petersburg State University, RU
- Maike Buchin
Ruhr-Universität Bochum, DE
- Benjamin Burton
The University of Queensland – Brisbane, AU
- Hsien-Chih Chang
Duke University – Durham, US
- Arnaud de Mesmay
University of Grenoble, FR
- Vincent Despré
INRIA Nancy – Grand Est, FR
- Linda Kleist
TU Braunschweig, DE
- Boris Klemz
FU Berlin, DE
- Francis Lazarus
GIPSA Lab – Grenoble, FR
- Maarten Löffler
Utrecht University, NL
- Anna Lubiw
University of Waterloo, CA
- Clément Maria
INRIA – Valbonne, FR
- Tim Ophelders
Michigan State University, US
- Hugo Parlier
University of Luxembourg, LU
- Saul Schleimer
University of Warwick – Coventry, GB
- Lena Schlipf
FernUniversität in Hagen, DE
- André Schulz
FernUniversität in Hagen, DE
- Eric Sedgwick
DePaul University – Chicago, US
- Rodrigo I. Silveira
UPC – Barcelona, ES
- Jonathan Spreer
University of Sydney, AU
- Frank Staals
Utrecht University, NL
- Stephan Tillmann
University of Sydney, AU
- Ivor van der Hoog
Utrecht University, NL
- Birgit Vogtenhuber
TU Graz, AT
- Carola Wenk
Tulane University – New Orleans, US
- Erin Moriarty Wolf Chambers
St. Louis University, US
- Alexander Wolff
Universität Würzburg, DE