Metric Sketching and Dynamic Algorithms for Geometric and Topological Graphs

Sujoy Bhore[*1], **Jie Gao**[*2], **Hung Le**[*3], **Csaba D. Tóth**[*4], **and Lazar Milenković**[†5]

1   **Indian Institute of Technology Bombay – Mumbai, IN.** `sujoy@cse.iitb.ac.in`
2   **Rutgers University – Piscataway, US.** `jg1555@cs.rutgers.edu`
3   **University of Massachusetts Amherst, US.** `hungle@cs.umass.edu`
4   **California State University Northridge – Los Angeles, US.** `csaba.toth@csun.edu`
5   **Tel Aviv University, IL.** `milenkovic.lazar@gmail.com`

―――― **Abstract** ――――――――――――――――――――――――――――――――――――――

Sketching is a basic technique to handle big data: Compress a big input dataset into a small dataset, called a *sketch*, that (approximately) preserves the important information in the input dataset. A metric space is often given as a distance matrix with $\Omega(n^2)$ entries, and *metric sketching* techniques aim to reduce the space to linear. One goal of this Dagstuhl Seminar was to understand different sketching techniques and metric spaces that admit small sketches. Another common approach to handling big datasets is dynamic algorithms. Typically, large datasets do not arrive in a single batch; instead, they are updated over time in small increments. The objective of dynamic algorithms is to respond to data updates quickly, ideally with an update time that is polylogarithmic in the size of the whole dataset.

In this Dagstuhl Seminar "Metric Sketching and Dynamic Algorithms for Geometric and Topological Graphs" (25212), we considered sketching and dynamic algorithms in the context of geometric intersection graphs and topological graphs. Geometric intersection graphs have been used to model many real-world massive graphs, such as wireless networks. Topological graphs, including planar graphs, have been used in applications such as geographic information systems and motion planning. While geometric intersection graphs and topological graphs are seemingly different, they have common structural properties that allow the transfer of algorithmic techniques between the two domains, which was the motivation of this seminar: Uncovering deeper connections between metric sketching, dynamic algorithms, geometric intersection graphs, and topological graphs. More concretely, we studied: (1) the construction of sketching structures, such as spanners, tree covers, distance oracles, and emulators with optimal parameters for various metrics and graphs, including geometric and topological graphs; (2) dynamic problems in geometric intersections graphs, including connectivity, spanners, shortest paths; and (3) dynamic maintenance of metric sketching structures in topological graphs.

Seminar May 18–23, 2025 – `https://www.dagstuhl.de/25212`

2012 ACM Subject Classification Theory of computation → Computational geometry; Mathematics of computing → Discrete mathematics; Theory of computation → Design and analysis of algorithms

Keywords and phrases geometric spanners, geometric intersection graphs, planar metrics, metric covering, computational geometry

Digital Object Identifier 10.4230/DagRep.15.5.134

―――――――――――――

* Editor / Organizer
† Editorial Assistant / Collector

1 Executive Summary

Sujoy Bhore (Indian Institute of Technology Bombay – Mumbai, IN)
Jie Gao (Rutgers University – Piscataway, US)
Hung Le (University of Massachusetts Amherst, US)
Csaba D. Tóth (California State University – Northridge, US)

Our goal was to bring together world-renowned researchers in Computational Geometry, Graph Theory, Data Structures, and Algorithms and collaboratively advance the research agenda for the study of metric sketching, and dynamic algorithms for geometric and topological graphs. The participants identified and investigated fundamental open problems and research directions. In particular, we worked on specific open problems about metric sketching, geometric intersection graphs, geometry of topological graphs, and their applications to dynamic algorithms.

This Dagstuhl Seminar "Metric Sketching and Dynamic Algorithms for Geometric and Topological Graphs" (25212) brought together researchers from four research domains, and harvested the benefits of the significant interplay between fundamental geometric & topological structures, and modern algorithmic paradigms. The general objectives of the seminar were:

1. To advance knowledge on metric sketching and dynamic algorithms for geometric and topological graphs, and investigate fundamental open problems and research directions.
2. To stimulate inter-disciplinary collaboration between Computer Science (computational geometry, data structure and algorithms) and Mathematics (combinatorics and topological graph theory).
3. To catalog the state of the art, and distill major open problems on metric sketching and dynamic algorithms for geometric and topological graphs.

In addition to identifying fundamental research problems on metric sketching and dynamic algorithms for geometric and topological graphs, we shall studied concrete problems chosen by the participants during the seminar. The specific scientific aims of this seminar were:

1. *Structural properties:* Identify key characteristics of classes of geometric and topological graphs, and leverage those understandings to address fundamental challenges of metric sketching.
2. *Trade-offs:* Provide the best possible trade-offs among fundamental aspects for metric sketching mechanisms, e.g., lightness, sparsity, diameter, degree, and others.
3. *Efficient Algorithms:* Develop efficient and robust algorithms for fundamental problems defined on geometric and topological graphs, in both static and dynamic regimes.

Behind the scientific challenges of this seminar lies the pragmatic need for better understanding the complex network systems, such as, banking, education, transportation, healthcare, among others. The scale of the data in any present day network system is huge. Hence, sparsifcation is a natural mechanism of handing such big networks. Moreover, the data is constantly changing. With a huge amount of data, recomputation is not an option anymore. Recent progress on dynamic/online algorithms led to the development of the current best tools to address some of these problems. Hence, a deeper understanding of geometric data structures helps improve tradeoffs between resource utilization and solution quality.

2   Table of Contents

3   Introduction

Sujoy Bhore (Indian Institute of Technology Bombay – Mumbai, IN)
Jie Gao (Rutgers University – Piscataway, US)
Hung Le (University of Massachusetts Amherst, US)
Csaba D. Tóth (California State University – Northridge, US)

In the last several years multiple new techniques have been developed along separate lines of metric sketching (spanners and tree covers), geometry of topological graphs, and geometric intersection graphs. These new techniques have allowed for new ways to solve problems that were deemed challenging previously. For example, geometric tools such as VC-dimension and additive Voronoi diagram have been used to solve problems in planar graphs or graphs of forbidden minors. These ideas then feed back progress on geometric intersection graphs which were studied mainly in the geometry community. We also believe that these new data structures will inspire further development in handling dynamic settings.

Below we briefly review both recent developments in the four topics and also list directions and open problems we tackled.

Metric Sketching.   The goal of metric sketching is, given a set P of n points in a metric space (X, δ_X), construct a *compact* (or low space) structure to (approximately) represent the distances between the points in P. Ideally, we would like the structure to have a linear space of $O(n)$. There are many sketching structures; here, we highlight the two most important structures: *spanners* and *tree covers*.

A *t-spanner* is a graph G whose vertex set is P and edge set contains pairs of points in P with weight to be the metric distance between the endpoints, and furthermore, $\delta_G(x, y) \leq t \cdot \delta_X(x, y)$ for any two points $x, y \in P$. Spanners with $t = 1 + \epsilon$ have been studied extensively for low dimensional Euclidean and doubling metrics. It was known in the 90s that one can construct Euclidean $(1 + \epsilon)$-spanners which are sparse [1], light [2], and have a small diameter [3]. A recent line of work determines exactly the dependency of sparsity [6], lightness [6], and hop bounds [8] on the distance error parameter ϵ as well as the role of other points in (X, δ_X) in these constructions [4, 6].

Another important sketching structure is a tree cover. Informally, a tree cover for P is a collection of trees such that the distance between any two points in P is preserved in at least *one of the trees* in the cover. Formally, a *t-tree cover* is a collection $\mathcal{T}$ of trees such that for every two points $x, y \in P$, $\delta_X(x, y) \leq \min_{T \in \mathcal{T}} \delta_T(x, y) \leq t \cdot \delta_X(x, y)$. Tree covers are stronger than spanners in the sense that they are not only compact but also allow querying distances quickly (in constant time); in this sense, tree covers are closer to an ideal metric sketching. On the other hand, constructing a small tree cover is a very difficult problem. Small tree covers were constructed for low-dimensional Euclidean point sets in the 90s [3], but only recently, the number of trees needed was determined [9], and the result was extended to doubling metrics [7]. Remarkably, it was recently shown that *metrics induced by shortest distances in topological graphs* have a tree cover with a small number of trees [10, 11, 12].

In the context of the developments above, there are a number of questions we have explored in this Dagstuhl Seminar with the aim of pushing the frontier in this topic much further. Which metrics admit a compact distance sketching? Can we construct a sketching structure for high-dimensional Euclidean spaces? What is the precise dependency between the number of trees in the tree cover for topological metrics on the distance error parameter ϵ? Could the existing distance sketching structures be dynamized efficiently when points are inserted/deleted?

Geometric Intersection Graphs. Given a set $\mathcal{O}$ of n objects in $\mathbb{R}^d$, a geometric intersection graph $G_\times$ for $\mathcal{O}$ has each vertex corresponding to an object in $\mathcal{O}$ and an edge between two vertices if their corresponding objects intersect. Geometric intersection graphs have been successfully used to model many practical networks, such as wireless networks and map labeling. Geometric intersection graphs share certain interesting structures as some other well-studied graph families, such as planar graphs – specifically, planar graphs are geometric intersection graphs of disks by the circle packing theorem (a.k.a. the Koebe–Andreev–Thurston theorem). However, geometric intersection graphs can be dense, which makes them quite different from planar graphs from a computational perspective. To obtain efficient algorithms, typically, one would like to avoid computing all the edges explicitly. Examples of such algorithms include single source shortest paths in such graphs. On the other hand, useful structures such as balanced separators for planar graphs need to be modified/augmented and may lose the nice bound on the small size [13].

Many problems that are easy for point sets in $\mathbb{R}^d$ become extremely hard for $G_\times$ even for simple, uniform, fat objects. One example is the diameter problem: can we compute the diameter of unit disk graphs, i.e., intersection graphs of unit disks, in truly subquadratic time? This problem remains wide open despite many attempts [14, 15]. Related to metric sketching, it is not known if unit disk graphs have a tree cover with a constant number of trees [5] or spanners of small hop diameters [16, 18].

Another interesting perspective of geometric intersection graphs is that the existence of efficient algorithms on such graphs sometimes crucially depends on the shape, size, dimension, and topological properties of the objects. Of particular interest in this family is to understand how far one can push on efficient algorithms with only topological properties of how two objects intersect (e.g., geometric intersection graphs of pseudo-disks versus disk graphs).

We also investigated dynamic algorithms for geometric intersection graphs. In the dynamic settings, objects are inserted and deleted from time to time. The insertion/deletion of an object implicitly induces the insertion/deletion of up to $\Omega(n)$ edges incident to the object. However, the goal of the dynamic algorithm is to handle each update in (amortized) $O(\text{polylog}(n))$ time. The massive number of implicit updates makes the design of dynamic algorithms extremely challenging. We focused on fundamental problems such as dynamic connectivity [17], dynamic shortest paths [19], and dynamic optimization problems, including independent sets [20, 21] and matching [22], and shortest paths [23, 24].

Geometry of Topological Graphs. Topological graphs are graphs equipped with topological properties, for example, planar graphs, surface-embedded graphs, and, more generally, minor-free graphs. Research on topological graphs has largely focused on understanding the structural properties of these graphs and hopes to exploit structural insights to develop efficient algorithms. The separator theorem by Lipton and Tarjan [25] for planar graphs and the graph minor project by Robertson and Seymour are representative examples of this line of research. A recent line of research focuses on understanding the *geometry of topological graphs*, specifically the shortest path metrics of these graphs. A representative question is: To what extent does a planar metric resemble the Euclidean plane? Research on this question has produced deep results, including tree cover of constant size [11, 12], VC dimension [26], and low-treewidth embedding [27], with a large number of algorithmic applications. In this seminar, we sought a better understanding of the geometry of topological graphs by drawing connections to research in metric sketching. For example, techniques in constructing spanners were crucial in developing the first linear-space constant query-time distance oracle in planar graphs [28]. A concrete question was to determine the best trade-offs between the distance error parameter ϵ and the number of trees, the treewidth, or the size of the metric sketches.

Dynamic Algorithms.   In dynamic algorithms, the input is updated over time, often involving insertions and deletions of new "data points", for example, inserting/deleting points from a point set or edges from a graph. The goal is to quickly update the solution of an algorithm task when the input is updated. Handling dynamic updates brings an extra (and often very challenging) dimension to algorithm design in the three aforementioned topics. We have touched upon designing dynamic algorithms for metric sketching and geometric intersection graphs. Designing dynamic algorithms for topological graphs under edge updates is an emerging line of research that has not been duly explored.

On the negative side it is known that for exact problems, such as exact shortest paths and exact all pairs shortest paths, any dynamic algorithm must spend polynomial update time or query time under popular fine-grained hypotheses [29]. However, these lower bounds do not hold when we seek an approximation, which is often good enough for applications. Basic problems we explored include approximate single source shortest paths, approximate distance oracles, and approximate optimization problems, such as independent sets and matching [30].

References

1   J. Mark Keil. Approximating the complete Euclidean graph. In *SWAT*, volume 318 of *Lecture Notes in Computer Science*, pages 208–213. Springer, 1988.

2   Barun Chandra, Gautam Das, Giri Narasimhan, and José Soares. New sparseness results on graph spanners. *Int. J. Comput. Geom. Appl.*, 5:125–144, 1995.

3   Sunil Arya, Gautam Das, David M. Mount, Jeffrey S. Salowe, and Michiel H. M. Smid. Euclidean spanners: Short, thin, and lanky. In *STOC*, pages 489–498. ACM, 1995.

4   Sujoy Bhore and Csaba D. Tóth. Euclidean Steiner spanners: Light and sparse. *SIAM J. Discret. Math.*, 36(3):2411–2444, 2022.

5   Haim Kaplan, Wolfgang Mulzer, Liam Roditty, and Paul Seiferth. Routing in unit disk graphs. *Algorithmica*, 80(3):830–848, 2018.

6   Hung Le and Shay Solomon. Truly optimal Euclidean spanners. In *FOCS*, pages 1078–1100. IEEE Computer Society, 2019.

7   Yair Bartal, Ora Nova Fandina, and Ofer Neiman. Covering metric spaces by few trees. *J. Comput. Syst. Sci.*, 130:26–42, 2022.

8   Hung Le, Lazar Milenkovic, and Shay Solomon. Sparse Euclidean spanners with optimal diameter: A general and robust lower bound via a concave inverse-Ackermann function. In *SoCG*, volume 258 of *LIPIcs*, pages 47:1–47:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

9   Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Optimal Euclidean tree covers. In *SoCG*, volume 293 of *LIPIcs*, pages 37:1–37:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

10   Sujoy Bhore, Balázs Keszegh, Andrey Kupavskii, Hung Le, Alexandre Louvet, Dömötör Pálvölgyi, and Csaba D. Tóth. Spanners in planar domains via Steiner spanners and non-Steiner tree covers. In *SODA*, pages 4292–4326. SIAM, 2025.

11   Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Covering planar metrics (and beyond): O(1) trees suffice. In *FOCS*, pages 2231–2261. IEEE, 2023.

12   Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Shortcut partitions in minor-free graphs: Steiner point removal, distance oracles, tree covers, and more. In *SODA*, pages 5300–5331. SIAM, 2024.

13   Mark de Berg, Sándor Kisfaludi-Bak, Morteza Monemizadeh, and Leonidas Theocharous. Clique-based separators for geometric intersection graphs. *Algorithmica*, 85(6):1652–1678, 2023.

14 Karl Bringmann, Sándor Kisfaludi-Bak, Marvin Künnemann, André Nusser, and Zahra Parsaeian. Towards sub-quadratic diameter computation in geometric intersection graphs. In *SoCG*, volume 224 of *LIPIcs*, pages 21:1–21:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

15 Hsien-Chih Chang, Jie Gao, and Hung Le. Computing diameter+2 in truly-subquadratic time for unit-disk graphs. In *SoCG*, volume 293 of *LIPIcs*, pages 38:1–38:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

16 Timothy M. Chan and Zhengcheng Huang. Constant-hop spanners for more geometric intersection graphs, with even smaller size. In *SoCG*, volume 258 of *LIPIcs*, pages 23:1–23:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

17 Timothy M. Chan and Zhengcheng Huang. Dynamic geometric connectivity in the plane with constant query time. In *SoCG*, volume 293 of *LIPIcs*, pages 36:1–36:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

18 Jonathan B. Conroy and Csaba D. Tóth. Hop-spanners for geometric intersection graphs. *J. Comput. Geom.*, 14(2):26–64, 2022.

19 Timothy M. Chan and Dimitrios Skrepetos. All-pairs shortest paths in geometric intersection graphs. *J. Comput. Geom.*, 10(1):27–41, 2019.

20 Sujoy Bhore, Martin Nöllenburg, Csaba D. Tóth, and Jules Wulms. Fully dynamic maximum independent sets of disks in polylogarithmic update time. In *SoCG*, volume 293 of *LIPIcs*, pages 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

21 Monika Henzinger, Stefan Neumann, and Andreas Wiese. Dynamic approximate maximum independent set of intervals, hypercubes and hyperrectangles. In *SoCG*, volume 164 of *LIPIcs*, pages 51:1–51:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

22 Sujoy Bhore and Timothy M. Chan. Fully dynamic geometric vertex cover and matching. *CoRR*, abs/2402.07441, 2024.

23 Peyman Afshani, Mark de Berg, Kevin Buchin, Jie Gao, Maarten Löffler, Amir Nayyeri, Benjamin Raichel, Rik Sarkar, Haotian Wang, and Hao-Tsung Yang. Approximation algorithms for multi-robot patrol-scheduling with min-max latency. In *WAFR*, volume 17 of *Springer Proceedings in Advanced Robotics*, pages 107–123. Springer, 2021.

24 Jie Gao, Mayank Goswami, Karthik C. S., Meng-Tsung Tsai, Shih-Yu Tsai, and Hao-Tsung Yang. Obtaining approximately optimal and diverse solutions via dispersion. In *LATIN*, volume 13568 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2022.

25 Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

26 Hung Le and Christian Wulff-Nilsen. VC set systems in minor-free (di)graphs and applications. In *SODA*, pages 5332–5360. SIAM, 2024.

27 Vincent Cohen-Addad, Hung Le, Marcin Pilipczuk, and Michal Pilipczuk. Planar and minor-free metrics embed into metrics of polylogarithmic treewidth with expected multiplicative distortion arbitrarily close to 1. In *FOCS*, pages 2262–2277. IEEE, 2023.

28 Hung Le and Christian Wulff-Nilsen. Optimal approximate distance oracle for planar graphs. In *FOCS*, pages 363–374. IEEE, 2021.

29 Amir Abboud and Søren Dahlgaard. Popular conjectures as a barrier for dynamic planar graph algorithms. In *FOCS*, pages 477–486. IEEE Computer Society, 2016.

30 Tuukka Korhonen, Wojciech Nadara, Michal Pilipczuk, and Marek Sokolowski. Fully dynamic approximation schemes on planar and apex-minor-free graphs. In *SODA*, pages 296–313. SIAM, 2024.

4   Overview of Talks

4.1   Metric Embeddings into Trees and its Various Spin-offs

Arnold Filtser (Computer Science Department, Bar Ilan University – Ramat Gan, IL)

Given a metric space (X, d_X), a stochastic metric embedding into trees is a distribution over dominating embeddings of the points X into a tree with polylogarithmic expected distortion. We will begin be describing the classic Bartal'96 embedding, and it's construction of padded decompositions. We will then discuss how this approach can be utilized to obtain online metric embedding (in particular deterministic into Euclidean space). We will then present the padded decomposition of Miller, Peng and Xu, and see how it can be used to obtain stochastic metric embedding into spanning trees. We will finish by mentioning some of the recent works of embedding structured graph families into bounded treewidth graphs.

4.2   Developments in Geometric Intersection Graphs

Édouard Bonnet (Laboratoire de l'Informatique du Parallélisme, ENS de Lyon, FR)

We survey some recent developments in algorithmic graph theory for geometric intersection graphs, and present some conjectures in the novel topic of induced minor theory.

4.3   Dynamic Algorithms in Geometry

Wolfgang Mulzer (Institut für Informatik, Freie Universität Berlin, DE)

I will present some classic results and recent advances in dynamic algorithms for geometric problems. In particular, I will focus on algorithms for dynamic lower envelopes in two and three dimensions, for lines, planes, and more general objects. I will also discuss applications for dynamic geometric intersection graphs.

4.4   Constructing and Routing on Plane Constant Spanners

Prosenjit Bose (School of Computer Science, Carleton University – Ottawa, CA)

Given a weighted graph and a real number t, a t-spanner of G is a spanning subgraph H with the property that for every edge $xy \in G$, there exists a path between x and y in H whose weight is no more than t times the weight of the edge xy. An online routing algorithm is an

algorithm that finds a short path in a graph without having full knowledge of the graph. We review results and present open problems on different variants of the problem of constructing and routing on plane geometric t-spanners.

4.5 Bicriteria Approximation for Minimum Dilation Graph Augmentation

Sampson Wong (University of Copenhagen, DK)

Spanner constructions focus on the initial design of the network. However, networks tend to improve over time. In this paper, we focus on the improvement step. Given a graph and a budget k, which k edges do we add to the graph to minimise its dilation? Gudmundsson and Wong [1] provided the first positive result for this problem, but their approximation factor is linear in k.

Our main result is a $(2\sqrt[r]{2}\ k^{1/r}, 2r)$-bicriteria approximation that runs in $O(n^3 \log n)$ time, for all $r \geq 1$. In other words, if t^* is the minimum dilation after adding any k edges to a graph, then our algorithm adds $O(k^{1+1/r})$ edges to the graph to obtain a dilation of $2rt^*$. Moreover, our analysis of the algorithm is tight under the Erdős girth conjecture.

References
1 Joachim Gudmundsson and Sampson Wong. Improving the dilation of a metric graph by adding edges. *ACM Trans. Algorithms*, 18(3):20:1–20:20, 2022.

4.6 Tree Covers for Euclidean and Planar Metrics and Their Applications

Lazar Milenković (Tel Aviv University, IL)

I will survey some recent results on tree covers for Euclidean and planar graphs [1, 2, 3]. Due to their structural simplicity, tree cover are suitable for various algorithmic applications. In this talk I will highlight some of the applications, such as compact routing schemes [1, 2] and construction of spanners with optimal tradeoff between hop-diameter and other parameters, including sparsity [4, 5] and lightness [6].

References
1 Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Covering planar metrics (and beyond): O(1) trees suffice. In *FOCS*, pages 2231–2261. IEEE, 2023.
2 Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Optimal Euclidean tree covers. In *SoCG*, volume 293 of *LIPIcs*, pages 37:1–37:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
3 Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Shortcut partitions in minor-free graphs: Steiner point removal, distance oracles, tree covers, and more. In *SODA*, pages 5300–5331. SIAM, 2024.

4   Hung Le, Lazar Milenkovic, and Shay Solomon. Sparse Euclidean spanners with tiny diameter: A tight lower bound. In *SoCG*, volume 224 of *LIPIcs*, pages 54:1–54:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.
5   Hung Le, Lazar Milenkovic, and Shay Solomon. Sparse Euclidean spanners with optimal diameter: A general and robust lower bound via a concave inverse-Ackermann function. In *SoCG*, volume 258 of *LIPIcs*, pages 47:1–47:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.
6   Sujoy Bhore and Lazar Milenkovic. Light spanners with small hop-diameter. In *ICALP*, volume 334 of *LIPIcs*, pages 30:1–30:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025.

4.7   VC Dimension and Their Applications in Topological and Geometric Graphs

Da Wei Zheng (University of Illinois Urbana-Champaign, US)

Minor-free graphs and geometric intersection graphs are two classes of graphs that share a surprisingly geometric property: they have very natural set systems that have bounded VC-dimension. We can algorithmically exploit these set systems to design algorithms and data structures not possible in general graphs such as subquadratic time algorithms for diameter and subquadratic space data structures for distance oracles.

4.8   Distance Approximating Structures for Planar and Minor-Free Graphs

Jonathan Conroy (Dartmouth College – Hanover, US)

In this talk, I will discuss several ways in which one can transform planar/minor-free metrics into some simpler object. One helpful tool in this area is the "buffered cop decomposition", a new way of partitioning minor-free graphs which we recently introduced [1]. I will try to convey some intuition about this structural result, and additionally sketch several applications including $O(1)$-distortion Steiner point removal [1] and optimal $O(\log r)$ padded decomposition for K_r-minor-free graphs [2].

References
1   Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Optimal Euclidean tree covers. In *SoCG*, volume 293 of *LIPIcs*, pages 37:1–37:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.
2   Jonathan Conroy and Arnold Filtser. How to protect yourself from threatening skeletons: Optimal padded decompositions for minor-free graphs. In *STOC*, pages 2281–2292. ACM, 2025.

5 Open Problems

5.1 DAG Covers for Directed Graph Classes

Nicole Wein (University of Michigan – Ann Arbor, US)

This is an open-ended problem based on a recent paper of mine with Sepehr Assadi and Gary Hoppenworth [1]. In this paper we define the notion of a DAG Cover, which is a directed analog of a tree cover. Given a directed graph, the goal is to construct a small collection of DAGs so that for all s, t: (1) no DAG in the collection underestimates $\text{dist}(s, t)$, and (2) some DAG in the collection gives a good approximation for $\text{dist}(s, t)$.

In the paper we study the problem when the input is a general directed graph, and get some upper and lower bound results that I will describe below. But first, I will state the problem.

Problem. Can we get interesting/useful DAG covers for special classes of directed graphs? Specifically, since tree covers have been very useful for classes of geometric graphs, it is natural to wonder whether DAG covers could be useful for geometric classes of directed graphs. As for what classes of graphs would be natural, I haven't thought about this at all!

Prior Work: I will outline the results that we get for general graphs, to help contextualize potential questions to ask about special classes of graphs.

First, one could ask: Do the DAGs need to be subgraphs of the original graph? This turns out to be a consequential question. If "yes" then a directed cycle shows that you need n DAGs in the cover even if you just want to preserve reachability. If "no", then there's a simple construction (that I can show you) that achieves only 2 DAGs and preserves exact distances. So, the interesting question is when you allow only a limited number of additional edges. Here are our results.

1. Lower bound: You can't have $O(n^{2-\varepsilon})$ added edges and $O(n^{1-\varepsilon})$ DAGs, and preserve even reachability (i.e. essentially the strongest possible lower bound for when the number of added edges is a function of the number of vertices n. Thus, we turn out attention to the number of edges m.
2. Upper bound: With $O(m)$ added edges, you can get only 2 DAGs and preserve reachability. (This one is simple and I can show it to you.)
3. Main upper bound: With $\tilde{O}(m)$ added edges, you can get polylog n DAGs and polylog n approximation factor. (Similar to the known tree cover result of $\log n$ trees and $\log n$ approximation factor.)
4. Lower bound: With $\tilde{O}(m)$ added edges, for preserving exact distances you can't get $n^{1/6-\varepsilon}$ DAGs.

References

1 Sepehr Assadi, Gary Hoppenworth, and Nicole Wein. Covering approximate shortest paths with DAGs. In *STOC*, pages 2269–2280. ACM, 2025.

5.2   Plane $(1 + \varepsilon)$-spanners in $\mathbb{R}^2$

Csaba D. Tóth (California State University Northridge – Los Angeles, US)

For n points in Euclidean plane, several constructions are known for $(1 + \varepsilon)$-spanners: Θ-graphs, Yao-graphs, Greedy spanners, and WSPD-based spanners. As $\varepsilon \to 0$ an $(1 + \varepsilon)$-spanner will inevitably have many edge crossings. If we allow Steiner points, we can subdivide the edges at crossing points and obtain a plane $(1 + \varepsilon)$-spanner. How many Steiner points do we really need?

Let $s(n, \varepsilon)$ be the minimum integer such that every set of n points in the plane admits a plane $(1 + \varepsilon)$-spanner with at most $s(n, \varepsilon)$ Steiner points.

Problem 1.   Determine $s(n, \varepsilon)$.

We know that for n points in the plane, there are $(1 + \varepsilon)$-spanners with $O(\varepsilon^{-1} n)$ edges; and Steiner $(1 + \varepsilon)$-spanners with $O(\varepsilon^{-1/2} n)$ edges. However, if the edges pairwise cross (naively), this would give $O(\varepsilon^{-1} n^2)$ crossings, hence $s(n, \varepsilon) \leq O(\varepsilon^{-1} \cdot n^2)$. It is not difficult to improve this naive bound and show that $s(n, \varepsilon)$ is closer to linear in n The goal is to find a bound of the form $s(n, \varepsilon) = O(f(\varepsilon) \cdot n)$ for some function $f(.)$ with the best possible function $f(.)$.

One can also require the Steiner $(1 + \varepsilon)$-spanner to minimize the number of Steiner point and meet other optimization criteria (e.g., small weight or small hop diameter).

Problem 2.   Determine the minimum number of Steiner points for a Steiner $(1 + \varepsilon)$-spanner of weight $O(\varepsilon^{-1} \cdot w(MST))$ on n points in the plane.

To show that $s(n, \varepsilon)$ is linear in n one can use the planarization of a greedy $(1+\varepsilon)$-spanner. It is known that the crossing graph is $O(f(\varepsilon))$-degenerate [1]. For a nontrivial lower bound, one might be able to use the so-called degenerate crossing number [2, 3] if Steiner points are modeled as degenerate crossings. But perhaps they can be modeled by bundled crossings [4, 5].

References
1   David Eppstein and Hadi Khodabandeh. On the edge crossings of the greedy spanner. In *SoCG*, volume 189 of *LIPIcs*, pages 33:1–33:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
2   Eyal Ackerman and Rom Pinchasi. On the degenerate crossing number. *Discret. Comput. Geom.*, 49(3):695–702, 2013.
3   János Pach and Géza Tóth. Degenerate crossing numbers. *Discret. Comput. Geom.*, 41(3):376–384, 2009.
4   Md. Jawaherul Alam, Martin Fink, and Sergey Pupyrev. The bundled crossing number. In *GD*, volume 9801 of *Lecture Notes in Computer Science*, pages 399–412. Springer, 2016.
5   Alan Arroyo and Stefan Felsner. Approximating the bundled crossing number. *J. Graph Algorithms Appl.*, 27(6):433–457, 2023.

5.3   Vantage Point Selection

Jie Gao (Rutgers University – Piscataway, US)
Nicole Wein (University of Michigan – Ann Arbor, US)

Problem Setup: We have an undirected graph G. (You can assume the graph is either weighted or unweighted, whichever is more convenient for what you're trying to prove.) Assume that for every pair of vertices there is one unique shortest path. Each edge also has a unique "capacity" that is hidden from us. We are allowed to choose vertices to "query". When we query a vertex s then for every vertex t the capacity of the minimum capacity edge on the shortest st-path is revealed.

Let OPT be the maximum number of edges that could be revealed by performing a single query in G. Note that OPT $< n$ since any shortest paths tree has $< n$ edges.

Question.   How many queries do we need in the worst case to reveal the capacity of $\Omega(\text{OPT})$ edges? It is known that the correct answer is between $\tilde{\Omega}(\sqrt{n})$ and $O(n)$. We think it should be possible to get $O(n^{1-\varepsilon})$

Alternate version of question.   We don't have a proof that this question is equivalent to the original question, but we would be happy to answer either one. Given an instance of the problem, let ALG_1 be the number of edges we can reveal with just one single query (in expectation if the algorithm is randomized). What is the minimum competitive ratio OPT/ALG_1 that we can guarantee? The known bounds are the same as the original problem. I.e., if we can reveal $\geq n^\varepsilon$ edges in expectation with a single query, then we've achieved the desired $n^{1-\varepsilon}$ upper bound.

Motivation.   This problem was recently defined by people who work in the theory and practice of networking, in the context of "internet bandwidth estimation". Direct all of your questions about motivation to Jie.

Prior Work.   Very little is known about this problem. The trivial upper bound is n. There is also a simple lower bound of $\tilde{\Omega}(\sqrt{n})$. Consider $\sqrt{n}$ paths of length $\sqrt{n}$ each. One of the paths (we don't know which) has capacities ordered smallest to largest along the path. All of the other paths have capacities in a random order. In this case, OPT$= \sqrt{n}$ by picking the first vertex of the special path. However, if we query a vertex on a non-special path, in expectation we only reveal roughly $\log n$ capacities. Since we don't know which path is special, we need $\Omega(\sqrt{n})$ queries to find it.

Observations and simplifying assumptions.   Let's say we are trying to achieve an upper bound of $n^{1-\varepsilon}$ for any small constant ε. Notice that if we query a vertex v all of vs incident edges are revealed. So, if G has a vertex of degree at least n^ε then we are done. In fact, we may want to assume that every vertex in G has constant degree, since we don't know how to solve that case.

If OPT $= O(n^{1-\varepsilon})$ then we're done because we can pick $O(n^{1-\varepsilon})$ arbitrary vertices to query, each of which reveals at least one edge. As a simplifying assumption, we may want to assume that OPT reveals $\Omega(n)$ capacities.

Another simplifying assumption under which we don't know the answer: Diameter is $\Theta(\log n)$.

5.4 Fat Minors

Michał Pilipczuk (University of Warsaw, PL)

All graphs are unweighted. For a graph G and a vertex subset $X \subseteq G$ we call X (k,r)-coverable if X can be covered by k balls of radius r in G. Motivated by the talk of Édouard, we pose the following.

Conjecture. For every graph H there exist constants $c, r \in \mathbb{N}$ such that the following holds: Every n-vertex graph G that excludes H as an induced minor has a $2/3$-balanced separator that is $(c\sqrt{n}, r)$-coverable.

More generally, the suspicion is that exclusion of an induced minor could be relaxed to exclusion of a fat minor. We say that G contains H as a d-fat minor, for a parameter $d \in \mathbb{N}$ if there are connected subgraphs $\{B_u \colon u \in V(H)\}$ and $\{B_e \colon e \in E(H)\}$ of G such that

- whenever u is an endpoint of e the subgraphs B_u and B_e intersect; and
- except for the above, all the subgraphs in $\{B_f \colon f \in V(G) \cup E(G)\}$ are pairwise disjoint and at distance more than d in G.

Clearly, if G contains H as a d-fat minor for any $d \geq 1$ then G contains H as an induced minor. Therefore, exclusion of a fat minor is a weaker assumption than exclusion of an induced minor. We conjecture that the statement above should also hold under the assumption that G excludes H as a d-fat minor for some fixed $d \in \mathbb{N}$ then c and r may also depend on d.

To give some background, Georgakopoulos and Papasoglu [1] conjectured that if G excludes H as a d-fat minor for some constant d then G is quasi-isometric to some graph G' that is H'-minor-free, for some H' depending only on H and d. (This is not exactly how they stated it, and their original formulation was disproved. See [2] for a discussion.) If the conjecture of Georgakopoulos and Papasoglu was true, then the conjecture above would follow even for fat minors, because the balanced separator of G' could be pulled back through the quasi-isometry. But the conjecture above can be also attacked without trying to verify the conjecture of Georgakopoulous and Papasoglu in full generality (the feeling is that this conjecture may well be just false).

References
1 Agelos Georgakopoulos and Panos Papasoglu. Graph minors and metric spaces. *Comb.*, 45(3):33, 2025.
2 James Davies, Robert Hickingbotham, Freddie Illingworth, and Rose McCarty. Fat minors cannot be thinned (by quasi-isometries). *CoRR*, abs/2405.09383, 2024.

5.5 Planar Spanners

Prosenjit Bose (Carleton University – Ottawa, CA)

Given a set P of n points in the plane, a geometric graph $G(P, E)$ is a graph whose vertex set is P whose edges are segments joining pairs of vertices and edges are typically weighted by their length. A geometric graph H is a t-spanner of G if H is a subgraph of G and $\forall xy \in E, d_H(x, y) \leq t|xy|$ for some $t \geq 1$, where $d_H(x, y)$ is the length of the shortest path

from x to y in H. The smallest value of t that satisfies this property is called the *spanning ratio* of H. The graph G is the *underlying* graph of the t-spanner H. In this setting the underlying graph is the complete geometric graph.

An online routing algorithm A is an algorithm that forwards a message from its current vertex s to a destination vertex t where the forwarding decision is made as a function of s, t, $I(s)$ and M where s is the coordinates of the current vertex, t is the coordinates of the destination vertex, $I(s)$ is information stored with vertex s in G and M is some bits of information stored in A. Typically, $I(s)$ is the set of vertices adjacent to s in G but sometimes some additional information is stored. The routing ratio of an algorithm A is the maximum spanning ratio among all s, t pairs of the path followed by A from s to t in G. Thus the routing ratio is an upper bound on the spanning ratio.

The following are some questions regarding spanners where G is the complete geometric graph and H is a planar spanning subgraph of G.

- The spanning ratio of the standard Delaunay triangulation on a set of points is at most 1.998 and in some cases it is at least 1.59. Can we improve the upper or lower bound?
- For points in convex position, the upper bound is 1.83 and the lower bound remains 1.59. Can we improve either bound?
- Is there a plane spanner with maximum degree 3 and constant spanning ratio? The lowest known degree bound is 4.
- Can we find a bounded degree plane spanner with spanning ratio better than 2.91. Currently, the best known bound is degree 14 and spanning ratio 2.91. The best known upper bound on the spanning ratio for spanners with maximum degree less than 10 is a degree bound of 8 and spanning ratio 4.41.
- For standard Delaunay graphs, the routing ratio is at most 3.5 and at least 1.7. Can these bounds be improved?
- Tight spanning ratios of 2.6 and 2 are known for Delaunay graphs where the empty circle is replaced by a square and hexagon, respectively. However, the routing ratio on these graphs is $\sqrt{10}$ and 6.4 respectively which leaves a large gap with the spanning ratio.

5.6   Patterns in Unweighted Planar Graphs

Da Wei Zheng (University of Illinois – Urbana-Champaign, US)

Let $G = (V, E)$ be an undirected unweighted planar graph with n vertices, and let $S = \{s_1, \ldots, s_k\}$ be the (ordered clockwise) vertices of one face of G. Let $d(\cdot, \cdot)$ denote the shortest path metric of G The pattern p_v of a vertex $v \in V$ is the vector

$$p_v = \langle d(v, s_2) - d(v, s_1), d(v, s_3) - d(v, s_2), \ldots, d(v, s_k) - d(v, s_{k-1}) \rangle.$$

Note that as the graph is unweighted and the vertices $s_1, \ldots, s_k$ are adjacent on one face of G the entries of p_v are in $\{1, 0, -1\}$. Since there are n vertices in G there are n patterns. However, Li and Parter [3] showed (using VC dimension type arguments) that there are only at most $O(k^3)$ many distinct patterns:

Theorem.   The number of distinct patterns over all vertices of a graph G is $O(k^3)$

An alternative proof is given in [1], along with a $\Omega(k^2)$ lower bound (a grid graph can also give the same lower bound) and the following conjecture:

Conjecture. The number of distinct patterns over all vertices of a graph G is $O(k^2)$.

What is the true answer? I suspect the conjecture is true and the answer should be $\Theta(k^2)$

Motivation and Consequences. This set system has been generalized to directed minor-free graphs [2, 4] and geometric intersection graphs [5]. Resolving the conjecture either way give use interesting new tools to work with in these settings.

If the conjecture is true, it would imply a simple algorithm (using separators) to compute the diameter of an unweighted planar graph in $O(n^{5/3})$ time (matching the best known algorithm using Voronoi diagrams).

References

1 Shay Mozes, Nathan Wallheimer, and Oren Weimann. Improved compression of the Okamura-Seymour metric. In *ISAAC*, volume 248 of *LIPIcs*, pages 27:1–27:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

2 Adam Karczmarz and Da Wei Zheng. Subquadratic algorithms in minor-free digraphs: (weighted) distance oracles, decrementai reachability, and more. In *SODA*, pages 4338–4351. SIAM, 2025.

3 Jason Li and Merav Parter. Planar diameter via metric compression. In *STOC*, pages 152–163. ACM, 2019.

4 Hung Le and Christian Wulff-Nilsen. VC set systems in minor-free (di)graphs and applications. In *SODA*, pages 5332–5360. SIAM, 2024.

5 Hsien-Chih Chang, Jie Gao, and Hung Le. Computing diameter+2 in truly-subquadratic time for unit-disk graphs. In *SoCG*, volume 293 of *LIPIcs*, pages 38:1–38:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

5.7 Sparse $(1 + \epsilon)$-emulator for Euclidean Point Sets

Hung Le (University of Massachusetts Amherst, US)

Given an Euclidean point set P say in $\mathbb{R}^2$ an emulator for P with stretch $1 + \epsilon$ is a graph $G = (V, E)$ where $P \subseteq V$ such that $\|x, y\|_2 \leq d_G(x, y) \leq (1 + \epsilon) \cdot \|x, y\|_2$ for every pair of points $x, y \in P$ How many edges does G have to have?

Note that vertices in $V \setminus P$ do not have to be points in $\mathbb{R}^2$.

If $V \setminus P$ are restricted be points in $\mathbb{R}^2$ then we know the answer: $O(n/\sqrt{\epsilon})$ edges are both necessary and sufficient. Nothing is known when $V \setminus P$ are not restricted to be points in $\mathbb{R}^2$.

Here is an embarrassingly simple case: points in P are on two opposite sides of a unit square, and for each side, there are $n = O(1/\sqrt{\epsilon})$ points, where the distance between any two consecutive points is $\sqrt{\epsilon}$. The complete bipartite graph, which only use points in P and has stretch 1 has $O(1/\epsilon) = O(n/\sqrt{\epsilon})$ edges. Could we beat this bound, using points that are not in P (and also not in $\mathbb{R}^2$)?

5.8　Tree Cover for Unit Ball Graphs

Jie Gao (Rutgers University – Piscataway, US)

A tree cover for a metric space (X, d) is a collection of trees $\mathcal{T}$ such that for any two points $p, q \in X$ the following two conditions are true:

- for every tree $T \in \mathcal{T}$ $d(p, q) \leq d_T(p, q)$;
- there is a tree $T \in \mathcal{T}$ such that $d_T(p, q) \leq \alpha \cdot d(p, q)$.

The tree cover is called to have distortion α.

Tree covers are known to exist for many settings:

- d-dimensional Euclidean space: $(1 + \epsilon)$-tree cover with $O(\epsilon^{\frac{(-d+1)}{2}} \log(\frac{1}{\epsilon}))$ trees. [1]
- Doubling metrics: $(1 + \epsilon)$-tree cover with $O(\epsilon^{-d} \log(\frac{1}{\epsilon}))$ trees. [2]
- Planar graphs: $(1 + \epsilon)$-tree cover with $O(1)$ trees. [3]

For unit disk graphs, notice that by using the tree cover of planar graphs, one can immediately construct a $O(1)$-tree cover with $O(1)$ trees, by using a constant planar spanner and then results on planar graphs (e.g., [3]). Naturally one can ask, can we find a $(1 + \epsilon)$-tree cover for unweighted unit disk graphs? Through personal communication during the seminar, there seems to be already ongoing work towards resolving the aforementioned question. What is still open is whether such a tree cover can be found for unit ball graphs.

A second question to ask is, what is the best stretch factor one can get to approximate a unit disk graph (or a general geometric intersection graph) by a planar graph?

For unit disk graphs, if we want to take a planar spanner graph (using only edges from the input unit disk graph), the known constructions give a stretch factor such as 449 [4] and 341 [5]. Recently we could get a stretch factor of 286 if we allow Steiner vertices. But these stretch factors are still fairly large. Can we do better? What is a lower bound?

References

1　Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Optimal Euclidean tree covers. In *SoCG*, volume 293 of *LIPIcs*, pages 37:1–37:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

2　Yair Bartal, Ora Nova Fandina, and Ofer Neiman. Covering metric spaces by few trees. *J. Comput. Syst. Sci.*, 130:26–42, 2022.

3　Hsien-Chih Chang, Jonathan Conroy, Hung Le, Lazar Milenkovic, Shay Solomon, and Cuong Than. Covering planar metrics (and beyond): O(1) trees suffice. In *FOCS*, pages 2231–2261. IEEE, 2023.

4　Nicolas Catusse, Victor Chepoi, and Yann Vaxès. Planar hop spanners for unit disk graphs. In *ALGOSENSORS*, volume 6451 of *Lecture Notes in Computer Science*, pages 16–30. Springer, 2010.

5　Ahmad Biniaz. Plane hop spanners for unit disk graphs: Simpler and better. *Comput. Geom.*, 89:101622, 2020.

5.9   Similarity of Curves (Fréchet Distance)

Omrit Filtser (The Open University of Israel – Ra'anana, IL)

A polygonal curve $A : [0, n] \to \mathbb{R}^d$ consists of n line segments $\overline{a_i a_{i+1}}$ for $i = \{0, 1, \ldots, n-1\}$ where $a_i = A(i)$.

The Fréchet distance between two polygonal curves $A : [0, m] \to \mathbb{R}^d$ and $B : [0, n] \to \mathbb{R}^d$ is often described (informally) by the following analogy. Consider a person and a dog connected by a leash, each walking along a curve from its starting point to its end point. Both can control their speed but they are not allowed to backtrack. The Fréchet distance between the two curves is the minimum length of a leash that is sufficient for traversing both curves in this manner (for a formal definition, see below).

The discrete Fréchet distance (DFD for short) is a popular variant in which the two curves are replaced by two sequences of points, and the person and dog are replaced by two frogs, hopping from one point to the next along their respective sequences. The discrete Fréchet distance is defined as the smallest maximum distance between the frogs that can be achieved in such a joint sequence of hops.

Some background.   The Fréchet distance can be computed in $\tilde{O}(n^2)$ time, and the discrete version in $O(n^2)$ time, via DP approach. For both variants there is a lower bound of $O(n^{2-\varepsilon})$ (conditioned on SETH), even in 1D, and for an approximation factor of up to 3. The best known approximation algorithm is a very recent breakthrough result [1]: a $(7 + \varepsilon)$-approximation (w.h.p) in $\tilde{O}(n^{1.99})$ time. Prior to that, the best approximation algorithms presented trade-offs between approximation quality and running time (see e.g. [2, 3]).

Proximity queries.   In short: Construct an efficient NNS data structure or a distance oracle, under the (discrete) Fréchet distance.

Nearest Neighbor Search.   Construct a compact data structure over a set C of n input curves, each of length at most m such that given a query curve Q of length k one can efficiently find the curve from C (approximately) closest to Q.

Distance oracle.   Preprocess a curve of length m into a data structure such that given a query curve of length $k < m$ returns its (approximated) distance to the input curve in $\tilde{O}(k)$ time.

For both problems, under DFD, a $(1 + \varepsilon)$ approximation and (near) linear query time can be achieved, but with space exponential in k $n \cdot O(\frac{1}{\varepsilon})^{kd}$ for NNS [4], $O(\frac{1}{\varepsilon})^{kd}$ for distance oracle [5]).

Open questions.   Can we avoid the exponential dependence on k in the space bound, by allowing a constant approximation factor (instead of $1 + \varepsilon$) and a slightly larger query time? Or, by allowing $O(\log \log n)$ approximation factor? Is the $O(\frac{1}{\varepsilon})^{kd}$ factor in the space bound necessary if we wish to have a small query time and an approximation factor of $1 + \varepsilon$?

For the (continuous) Fréchet distance the situation is much more complicated. Recently there has been some development in 1D case (see [6]).

Exat distance: For $k \in \{1, 2, 3\}$ there are exact distance oracle with near linear size and polylogarithmic query time [7]. For $k = 4$ the query time becomes $O(\sqrt{n})$. Can it be improved? (See also [8] for exact NNS under L_∞ for $k = 2$.)

References

1 Siu-Wing Cheng, Haoqiang Huang, and Shuo Zhang. Constant approximation of Fréchet distance in strongly subquadratic time. In *STOC*, pages 2329–2340. ACM, 2025.

2 Thijs van der Horst and Tim Ophelders. Faster Fréchet distance approximation through truncated smoothing. In *SoCG*, volume 293 of *LIPIcs*, pages 63:1–63:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

3 Timothy M. Chan and Zahed Rahmati. An improved approximation algorithm for the discrete Fréchet distance. *Inf. Process. Lett.*, 138:72–74, 2018.

4 Arnold Filtser, Omrit Filtser, and Matthew J. Katz. Approximate nearest neighbor for curves: Simple, efficient, and deterministic. *Algorithmica*, 85(5):1490–1519, 2023.

5 Arnold Filtser and Omrit Filtser. Static and streaming data structures for Fréchet distance queries. *ACM Trans. Algorithms*, 19(4):39:1–39:36, 2023.

6 Karl Bringmann, Anne Driemel, André Nusser, and Ioannis Psarros. Tight bounds for approximate near neighbor searching for time series under the Fréchet distance. In *SODA*, pages 517–550. SIAM, 2022.

7 Boris Aronov, Tsuri Farhana, Matthew J. Katz, and Indu Ramesh. Discrete Fréchet distance oracles. In *SoCG*, volume 293 of *LIPIcs*, pages 10:1–10:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

8 Boris Aronov, Omrit Filtser, Michael Horton, Matthew J. Katz, and Khadijeh Sheikhan. Efficient nearest-neighbor query and clustering of planar curves. In *WADS*, volume 11646 of *Lecture Notes in Computer Science*, pages 28–42. Springer, 2019.

5.10 Dynamic Subgraph Connectivity

Liam Roditty (Bar-Ilan University – Ramat Gan, IL)

Let $G = (V, E)$ be a fixed undirected graph. In the dynamic subgraph connectivity problem each vertex of G is either marked as on or off. An update either turns off a vertex that is on or turns on a vertex that is off. The data structure has to support connectivity queries on the graph induced by the on vertices while vertices are being updated. The dynamic subgraph connectivity problem was introduced in [1]. In [2] the first data structure for general graphs with $O(m^{0.94})$ amortized update time (using FMM) and $\tilde{O}(m^{1/3})$ query time was presented. In [3] the update time of [2] was improved to amortized update time of $\tilde{O}(m^{2/3})$ (without FMM) while keeping the query time $\tilde{O}(m^{1/3})$.

A data structure with $\tilde{O}(m^{4/5})$ worst case update time and $\tilde{O}(m^{1/5})$ query time was presented in [4]. A randomized data structure with $\tilde{O}(m^{3/4})$ worst case update time and $\tilde{O}(m^{1/4})$ query time was presented in [5]. Notice that the product of the update time and the query time is $\tilde{O}(m)$ in all these results.

This is not a coincidence since there is CLB presented by [6] which states that there is no algorithm with polynomial preprocessing time, $O(m^{\gamma - \Omega(1)})$ amortized update time, and $O(m^{1-\gamma-\Omega(1)})$ amortized query time, for any constants $0 \leq \gamma \leq 1$ as long as $m \leq \min(n^{1/\gamma}, n^{1/(1-\gamma)})$.

Open problem. Does there exist an algorithm for dynamic subgraph connectivity with worst-case update time of $\tilde{O}(m^{2/3})$ and worst-case query time $\tilde{O}(m^{1/3})$?

References
1   Daniele Frigioni and Giuseppe F. Italiano. Dynamically switching vertices in planar graphs. *Algorithmica*, 28(1):76–103, 2000.
2   Timothy M. Chan. Dynamic subgraph connectivity with geometric applications. In *STOC*, pages 7–13. ACM, 2002.
3   Timothy M. Chan, Mihai Pătraşcu, and Liam Roditty. Dynamic connectivity: Connecting to networks and geometry. *SIAM J. Comput.*, 40(2):333–349, 2011.
4   Ran Duan. New data structures for subgraph connectivity. In *ICALP (1)*, volume 6198 of *Lecture Notes in Computer Science*, pages 201–212. Springer, 2010.
5   Ran Duan and Le Zhang. Faster randomized worst-case update time for dynamic subgraph connectivity. In *WADS*, volume 10389 of *Lecture Notes in Computer Science*, pages 337–348. Springer, 2017.
6   Monika Henzinger, Sebastian Krinninger, Danupon Nanongkai, and Thatchaphol Saranurak. Unifying and strengthening hardness for dynamic problems via the online matrix-vector multiplication conjecture. In *STOC*, pages 21–30. ACM, 2015.

5.11   Sparse and Light (k-)Banyans

Sándor Kisfaludi-Bak (Aalto University, FI)

A banyan of a point set $P \subset \mathbb{R}^d$ is a geometric graph G where for any terminal set $K \subset P$ we have that G contains a subtree covering K whose weight is at most $1 + \epsilon$ times the weight of the minimum Euclidean Steiner tree of K. (The minimum Euclidean Steiner tree of K is the connected geometric graph of minimum weight that contains K in its vertex set.)

A k-banyan is as above, but we only require the property for sets K of size at most k. With this definition, 2-banyans are simply (Steiner) spanners.

- What is the best trade-off between the banyan stretch and the number of Steiner vertices (or the number of edges) of a $(k-)$banyan for a given dimension d What about 3-banyans?
- What metrics allow light $(k-)$banyans? What is the optimal lightness in $\mathbb{R}^d$?

There is plenty of work on Steiner spanners, so the case $k = 2$ is well-studied and we have good answers for the optimal sparsity and lightness for fixed d in this case. Light banyans have been constructed in a handful of papers cited below, but with the goal of using them for Steiner tree/forest algorithms; I suspect that these constructions are far from optimal in terms of stretch/edge count and stretch/lightness trade-offs.

Starting points are: [1, 2, 3]. The appendix of [4] may also be relevant.

References
1   Yair Bartal and Lee-Ad Gottlieb. Near-linear time approximation schemes for Steiner tree and forest in low-dimensional spaces. In *STOC*, pages 1028–1041. ACM, 2021.
2   David Eisenstat, Philip N. Klein, and Claire Mathieu. An efficient polynomial-time approximation scheme for Steiner forest in planar graphs. In *SODA*, pages 626–638. ACM-SIAM, 2012.
3   Satish Rao and Warren D. Smith. Approximating geometrical graphs via "spanners" and "banyans". In *STOC*, pages 540–550. ACM, 1998.
4   Sándor Kisfaludi-Bak, Jesper Nederlof, and Karol Wegrzycki. A gap-ETH-tight approximation scheme for Euclidean TSP. In *FOCS*, pages 351–362. IEEE, 2021.

5.12   Online Spanners in Metric Spaces

Sujoy Bhore (Indian Institute of Technology Bombay – Mumbai, IN)

We are given a sequence of points $(s_1, \ldots, s_n)$ in a metric space, where the points are presented one-by-one, i.e., point s_i is revealed at step i and $S_i = \{s_1, \ldots, s_i\}$ for $i \in \{1, \ldots, n\}$. The objective of an online spanner algorithm is to maintain a $t$spanner G_i for S_i for all i. The algorithm is allowed to add edges to the spanner when a new point arrives, however, it is not allowed to remove any edge from the spanner. Moreover, the algorithm does not know the total number of points in advance. The performance of an online $t$spanner algorithm is measured by comparing it with the offline optimum using the standard notion of competitive ratio, which is defined as $\sup_\sigma \frac{ALG(\sigma)}{OPT(\sigma)}$, where the supremum is taken over all input sequences σ, $OPT(\sigma)$ is the minimum weight of a $t$spanner for the (unordered) set of points in σ, and $ALG(\sigma)$ denotes the weight of the $t$spanner produced by the algorithm. Note that, to measure the competitive ratio, it is important that σ is a finite sequence of points.

Problem.   Determine the best possible bounds for the competitive ratios for the weight/-lightness, sparsity, and stretch of online t-spanners, for $t > 1$.

References: [1, 2, 3, 4].

References

1   Sujoy Bhore and Csaba D. Tóth. Online Euclidean spanners. *ACM Trans. Algorithms*, 21(1):5:1–5:22, 2025.
2   Sujoy Bhore, Arnold Filtser, Hadi Khodabandeh, and Csaba D. Tóth. Online spanners in metric spaces. *SIAM J. Discret. Math.*, 38(1):1030–1056, 2024.
3   Noga Alon and Yossi Azar. On-line Steiner trees in the Euclidean plane. In *SCG*, pages 337–343. ACM, 1992.
4   Makoto Imase and Bernard M. Waxman. Dynamic Steiner tree problem. *SIAM J. Discret. Math.*, 4(3):369–384, 1991.

5.13   Steiner $(1 + \varepsilon)$-spanners in $\mathbb{R}^d$

Csaba D. Tóth (California State University Northridge – Los Angeles, US)

For any set S of n points in Euclidean plane and $\varepsilon > 0$ one can construct a Steiner $(1 + \varepsilon)$-spanner of lightness $O(\varepsilon^{-1})$ where the lightness is the ratio between the weight of the spanner and the weight of an MST for the same point set. This bound is the best possible in the plane. In dimensions $d \geq 3$, however, the current best upper and lower bounds do not match: $\Omega(\varepsilon^{-d/2})$ and $\tilde{O}(\varepsilon^{(1-d)/2})$.

Problem.   Close the gap between the upper and lower bounds for $d \geq 3$ (or, at least for $d = 3$).

References: [1, 2].

References

1   Sujoy Bhore and Csaba D. Tóth. Euclidean Steiner spanners: Light and sparse. *SIAM J. Discret. Math.*, 36(3):2411–2444, 2022.
2   Hung Le and Shay Solomon. A unified framework for light spanners. In *STOC*, pages 295–308. ACM, 2023.

6   Working Groups

- Jonathan Conroy, Liam Roditty, and Hsien-Chih Chang: *DAG Covers for Directed Graph Classes*
- Sujoy Bhore, Sándor Kisfaludi-Bak, Lazar Milenković, Csaba D. Tóth, Karol Wegrzycki, and Sampson Wong: *Plane $(1 + \varepsilon)$-spanners in $\mathbb{R}^2$*
- Sergio Cabello, Jie Gao, Paloma Lima, and Nicole Wein: *Vantage Point Selection*
- Édouard Bonnet, Hung Le, and Michał Pilipczuk: *Fat Minors*
- Prosenjit Bose, Omrit Filtser, and Linda Kleist: *Planar Spanners*
- Arnold Filtser, Eunjin Oh, Marcin Pilipczuk, and Da Wei Zheng: *Patterns in Unweighted Planar Graphs*

Participants

- Sujoy Bhore
  Indian Institute of Technology
  Bombay – Mumbai, IN
- Édouard Bonnet
  ENS – Lyon, FR
- Prosenjit Bose
  Carleton University –
  Ottawa, CA
- Sergio Cabello
  University of Ljubljana, SI
- Hsien-Chih Chang
  Dartmouth College –
  Hanover, US
- Jonathan Conroy
  Dartmouth College –
  Hanover, US
- Arnold Filtser
  Bar-Ilan University –
  Ramat Gan, IL
- Omrit Filtser
  The Open University of Israel –
  Ra'anana, IL

- Jie Gao
  Rutgers University –
  Piscataway, US
- Sándor Kisfaludi-Bak
  Aalto University, FI
- Linda Kleist
  Universität Potsdam, DE
- Hung Le
  University of Massachusetts
  Amherst, US
- Paloma Lima
  IT University of
  Copenhagen, DK
- Lazar Milenković
  Tel Aviv University, IL
- Wolfgang Mulzer
  FU Berlin, DE
- Eunjin Oh
  POSTECH – Pohang, KR
- Marcin Pilipczuk
  University of Warsaw, PL

- Michał Pilipczuk
  University of Warsaw, PL
- Liam Roditty
  Bar-Ilan University –
  Ramat Gan, IL
- Csaba D. Tóth
  California State University
  Northridge – Los Angeles, US
- Karol Wegrzycki
  MPI für Informatik –
  Saarbrücken, DE
- Nicole Wein
  University of Michigan –
  Ann Arbor, US
- Sampson Wong
  University of Copenhagen, DK
- Da Wei Zheng
  University of Illinois –
  Urbana-Champaign, US