Some Open Problems in Computational Geometry

Sergio Cabello

University of Ljubljana, Slovenia
Institute of Mathematics, Physics and Mechanics, Ljubljana, Slovenia
https://www.fmf.uni-lj.si/~cabello/
sergio.cabello@fmf.uni-lj.si

Abstract

In this paper we shall encounter three open problems in Computational Geometry that are, in my opinion, interesting for a general audience interested in algorithms.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases barrier resilience, maximum matching, geometric graphs, fixed-parameter tractability, stochastic computational geometry

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.2

Category Invited Talk

Funding Supported by the Slovenian Research Agency, program P1-0297 and projects J1-1693, J1-9109.

1 Introduction

Computational Geometry deals with algorithmic problems where the input are geometric objects. Here is a *tiny* sample of problems and results in the area, just to provide an intuition:

- Given a set P of n points in the Euclidean plane, find the closest pair. This means, find a pair $p_0, q_0 \in P$ such that $|p_0 - q_0| = \min\{|p-q| \mid p,q \in P, p \neq q\}$. The problem can be solved in $O(n \log n)$ time [16] or in randomized linear time, assuming that the floor function is available [17].
- Given a set of n segments in the plane, find the pairs of segments that intersect. The problem can be solved in $O(k + n\log n)$ time, where k is the size of the output, that is, the number of pairs of input segments that intersect [7].
- (Hopcroft's problem.) Given a set of n lines and a set of n points in the plane, decide whether there is any point-line incidence. The problem can be solved in roughly $O(n^{4/3})$ time [14].
- (Klee's measure problem.) Given a set of n axis-parallel boxes in $\mathbb{R}^3$, compute the volume of the union. The problem can be solved in $O(n^{3/2})$ time [6].

Here I would like to explain three problems in Computational Geometry that I like, but I do not know how to solve. Working with colleagues in the last few years, we have obtained partial results in these problems or in related problems, but the main problem or objective has remained open. It is time that others start thinking about the problems, and hopefully this presentation will help that purpose. I have chosen problems that, I believe, are interesting to a general audience interested in algorithms. The tone of the exposition is rather informal, resembling a talk.

2 Stochastic Bounding Box

Let P be a set of points in $\mathbb{R}^d$. A *box* in $\mathbb{R}^d$ is the Cartesian product of closed intervals. The *bounding box* of P, defined as the smallest box that contains P, can be computed trivially in $O(dn)$ time. We just have to compute in each dimension the largest and the smallest coordinate value. The volume of such bounding box can also be computed trivially.

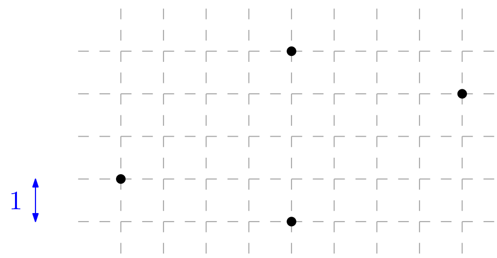

Figure 1 Example of data for the problem. If all points have probability of existence $\pi(\cdot) = \frac{1}{2}$, the expected area of the bounding box is $\frac{1}{16} \cdot 32 + \frac{3}{16} \cdot 16 + \frac{2}{16} \cdot 24 + \frac{2}{16} \cdot 4 + \frac{2}{16} \cdot 12 + \frac{6}{16} \cdot 0$.

Assume now that each point p of P has a number $\pi(p) \in (0, 1]$ associated to it, that determines the probability that the point p exists. The numbers $\pi(p)$ for $p \in P$ are input data. We refer to such an object as a *stochastic point set*. We construct a random subset R of P where we include each point p of P with probability $\pi(p)$ and the decision for each point is made independently. Thus, for each $P' \subseteq P$, we have $\Pr[R = P'] = \prod_{p\in P'} \pi(p) \cdot \prod_{p\in P\setminus P'}(1-\pi(p))$. Can we efficiently compute the expected volume of the bounding box of R? For simplicity, we assume that arithmetic operations take constant time, independently of the size of the numbers. This is relevant to avoid keeping track of the bit-length of the numbers involved in the computations. See Figure 1 for a simple example.

In the plane, when $d = 2$, the problem can be solved in $O(n \log n)$ time [19]. Using the 2-dimensional case as base case, one can solve the problem in $O(n^{d-1} \log n)$ time for each constant $d \geq 3$. What is the complexity of the problem when we take d as part of the input?

I asked the problem at the Dagstuhl Seminar New Horizons in Parameterized Complexity (2019). Together with **Radu Curticapean** and **Mark Jerrum**, we noted that the problem is #P-hard when the dimension d is unbounded. The result is unpublished, so let me reproduce here the main argument.

For any graph G, we construct a point set $P = P(G)$ in $\mathbb{R}^{E(G)}$, as follows. For each edge $e \in E(G)$ and each point $p \in \mathbb{R}^{E(G)}$, we use $x_e(p)$ to denote the coordinate of p in the dimension indexed by e. Thus, $p \in \mathbb{R}^{E(G)}$ has coordinates $(x_e(p))_{e\in E(G)}$. Each vertex $v \in V$ gives a point $p_v \in P$ such that, for each $e \in E(G)$, the coordinate $x_e(p_v)$ is 1 if $v \in e$ and 0 otherwise. Let o denote the origin.

Recall that $U \subseteq V(G)$ is a vertex cover of G if and only if each edge $uv \in E(G)$ has $\{u, v\} \cap U \neq \emptyset$. We have the following observation relating G and P.

▶ **Lemma 1.** *For each subset of vertices $U \subseteq V(G)$, the volume of the bounding box of $P_U = \{p_u \mid u \in U\} \cup \{o\}$ is 1 if U is a vertex cover of G, and 0 otherwise.*

Proof. In each dimension, the smallest coordinate of P_U is 0, because of the origin o, and the largest coordinate is either 0 or 1. Thus, the volume of the bounding box of P_U is either 0 or 1. In a dimension indexed by $e \in E(G)$, the largest coordinate of P_U is 1 if and only if there is some vertex u in U such that $u \in e$. Thus, the volume of the bounding box is 1 if and only if, for each edge $e \in E(G)$, there is some vertex $u \in U$ such that $u \in e$. This is precisely the definition of vertex cover. ◀

Assign probability $\pi(p_v) = 1/2$ for each $v \in V(G)$. Then, each subset of P has the same probability of being in the random sample R, namely $1/2^{|V(G)|}$. We add the origin o to the stochastic point set with probability $\pi(o) = 1$. We then obtain that the expected volume of the bounding box for $R \cup \{o\}$ is exactly the probability that a random subset $U \subset V(G)$ is a

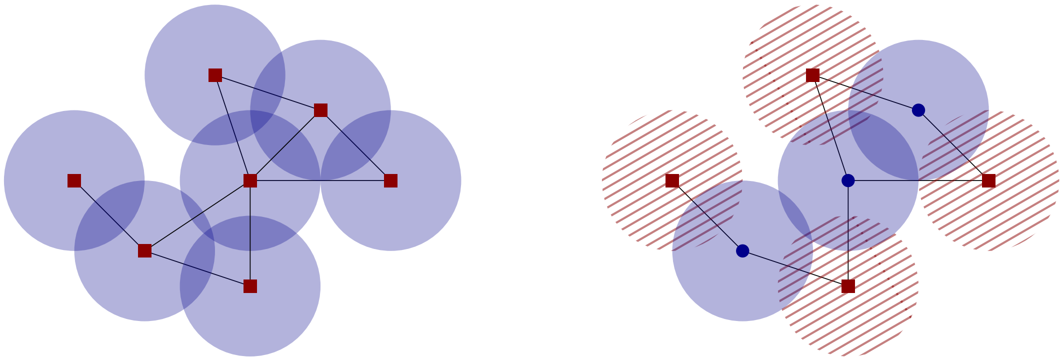

Figure 2 Example of a unit disk graph (left) and a bipartite unit disk graph (right).

vertex cover, which is precisely the number of vertex covers in G divided by $2^{|V(G)|}$. Since computing the number of vertex covers in graphs is #P-hard, even for sparse graphs [20], computing the expected volume is also #P-hard when we have n stochastic points and dimension $d = \Theta(n)$. We conclude the following.

▶ **Theorem 2** (Sergio Cabello, Radu Curticapean, and Mark Jerrum; unpublished)**.** *Computing the expected volume of the bounding box of n stochastic points is $\mathbb{R}^{\Theta(n)}$ is #P-hard.*

The main problem I would like to understand is the dependency on the dimension d. In particular, is the (decision) problem W[1]-hard or FPT when parameterized by the dimension d? Efficient approximation schemes with a small dependency on d would also be interesting.

The concept of stochastic input for geometric problems is relatively recent. See for example [10, 11]. I like this problem because it touches on FPT, and the problem is trivial for non-stochastic data.

3 Maximum Matching in Unit Disk/Square Graphs

Let $\mathcal{U}$ be a set of unit disks or a set of n unit squares (axis-parallel) in the plane. The *intersection graph* $G_{\mathcal{U}}$ of $\mathcal{U}$ has vertex set $\mathcal{U}$ and an edge UV, for distinct $U, V \in \mathcal{U}$, if and only if U and V intersect. See Figure 2, left, for an example. Here it comes the open problem: for given $\mathcal{U}$, can we compute in near-linear time a maximum matching in the intersection graph $G_{\mathcal{U}}$?

Together with **Édouard Bonnet** and **Wolfgang Mulzer** [2] we have shown that the maximum matching can be computed in $O(n^{\omega/2})$ time with high probability, where $\omega > 2$ is a constant such that $n \times n$ matrices can be multiplied in $O(n^{\omega})$ time. Since previous algorithms were using roughly $O(n^{3/2})$ time, this is a substantial improvement.

The algorithm has two main parts, which I describe next. I describe it for unit disks, but the very same ideas work for unit squares.

In the first part, we place a regular grid such that each unit disk contains some grid point and at most $O(1)$ grid points. We cluster the unit disks $\mathcal{U}$ into groups, depending on which grid points it contains. Each cluster is a clique, which intuitively helps when computing a maximum matching because there is much flexibility to arrange a maximum matching within any subset of the cluster. We then show that in each cluster it suffices to keep $O(1)$ unit disks, as the rest can be matched among themselves trivially. We also show that this step can be carried out in $O(n \operatorname{polylog} n)$ time using appropriate geometric data structures. We refer to this step as *sparsification*, since we reduce the instance $\mathcal{U}$ to another instance $\mathcal{U}' \subset \mathcal{U}$ where each point of the plane is covered by $O(1)$ unit disks from $\mathcal{U}'$.

For the second part, we adapt the algorithm of Mucha and Sankowski [15] to compute a maximum matching in a planar graph with n vertices in $O(n^{\omega/2})$ time. The main insight (with several formidable details) is the use of $O(n^{1/2})$-separators in planar graphs to carry out Gaussian elimination and identifying how a maximum matching interacts with the vertices in the separator. This relevance of separators was also exploited by Yuster and Zwick [21] for minor-free graphs. Separators in geometric settings also exist, and when each point of the plane is covered by a few disks, the bounds very much resemble the bounds for planar graphs. The details are a bit tedious and I refer our work [2].

Let me finish mentioning another variant of the problem that is very interesting and useful, namely the bipartite version. Assume that we have a family $\mathcal{U}_R$ of red unit disks and a family $\mathcal{U}_B$ of blue unit disks in the plane. Consider the bipartite intersection graph $G_{\mathcal{U}_R,\mathcal{U}_B}$ with vertex set $\mathcal{U}_R \cup \mathcal{U}_B$ and edges U_rU_b if and only if $U_r \in \mathcal{U}_R$, $U_b \in \mathcal{U}_B$ and $U_r \cap U_b \neq \emptyset$. Thus, we only care about the intersections (and edges) between disks of different colors. See Figure 2, right, for an example.

Efrat, Itai and Katz [9] provided an algorithm to compute a maximum matching in the bipartite graph $G_{\mathcal{U}_R,\mathcal{U}_B}$ defined by n red and blue unit disks (or squares) in roughly $O(n^{3/2})$ time. This remains the best running time. Roughly, it makes $O(n^{1/2})$ rounds, where in each round blocking augmenting paths are computed in near-linear time. Can we find a maximum matching in the bipartite case defined by bichromatic unit disks or squares in near-linear time?

The sparsification step in [2] does not apply in the bipartite case, and the time bound by Efrat, Itai and Katz [9] keeps being the best, in the worst case. The problem is potentially relevant in the context of Computational Topology, where closely-related graphs are considered when computing the distance between persistence diagrams [8]. (In this application the graph is slightly different.)

In both cases, bipartite or not bipartite, recognizing whether the (bipartite) intersection graph has a maximum matching, possibly without explicitly constructing it, seems an interesting challenge.

4 Barrier Resilience

Let me start explaining the barrier resilience problem, introduced by Kumar, Lai and Arora [13]. We have a family $\mathcal{D}$ of unit disks in the plane and two points s and t not covered by any of the disks in $\mathcal{D}$. We want to find an s-t curve in the plane that touches as few disks of $\mathcal{D}$ as possible; it is important that we count the disks *without multiplicity*. Equivalently, we want to remove as few disks as possible from $\mathcal{D}$ so that there is an s-t curve in the plane that does not touch any of the remaining disks.

The problem, as described, is the version in the so-called *annular domain*. In the *rectangular domain*, the point s is above all the disks of $\mathcal{D}$, the point t is below all the disks of $\mathcal{D}$, and we only consider curves contained in a given vertical slab that contains s and t. See Figure 3 for examples of each version.

The problem was considered in the context of sensors. It describes the minimum number of sensors that can fail so that an agent can move from s to t undetected. Kumar, Lai and Arora [13] showed that the problem can be solved in polynomial time in the rectangular domain. The idea is very neat and general. Let L_ℓ and L_r be the boundaries of the slabs and consider the intersection graph G of $\mathcal{D} \cup \{L_\ell, L_r\}$. Using Menger's theorem, one can see that the maximum number of vertex-disjoint L_ℓ-L_r-paths in G is the size of the optimum. Together with **Wolfgang Mulzer** [5] we used geometric data structures to solve the problem in roughly $O(n^{3/2})$ time.

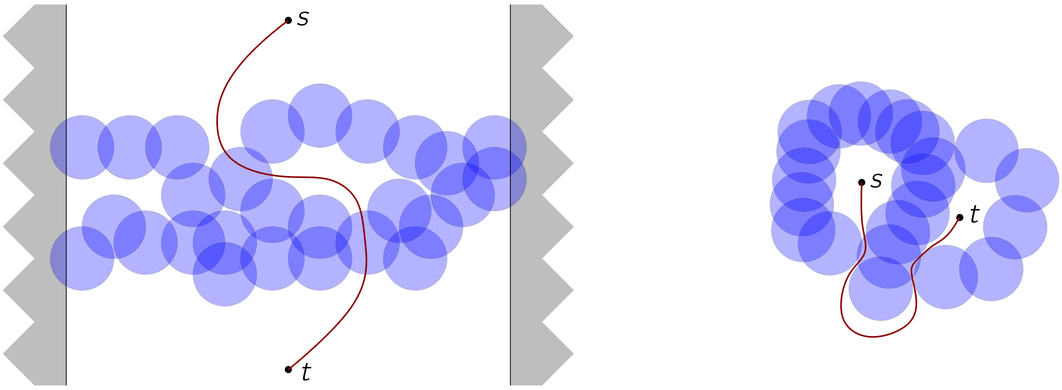

Figure 3 Barrier resilience problem in the rectangular domain (left) and the annular domain (right).

In contrast, we do not know whether the problem can be solved in polynomial time in the annular domain. We do know that the problem is NP-hard in the annular case when the disks are replaced by (unit) segments or rectangles of two different sizes that can cross each other [1, 12, 18]. The computational complexity when we have disks or unit disks, remains elusive.

The problem models complete failure of the sensors. However, in several scenarios, the probability of being undetected is larger when you pass near the boundary of the sensing region, and it is much higher when you pass near the center of the region. Motivated by this, in joint work with **Kshitij Jain**, **Anna Lubiw** and **Debajyoti Mondal** [4] we considered the optimization problem where each disk may be shrunk by a different amount, and we want to minimize the sum of the shrinking over the disks to allow for an s-t path. We showed that the problem has a FPTAS in the rectangular domain, but we do not know whether the problem is NP-hard in the rectangular domain. In a follow up work with **Éric Colin de Verdière** [3] we showed that the problem is weakly NP-hard in the annular domain. The table in Figure 4 provides a summary of the results and the open problems regarding the computational complexities.

	rectangular domain	annular domain
barrier problem total failure	polynomial Menger's theorem max flow	unknown complexity FPT and $(1+\varepsilon)$-approx in some cases
shrinking barrier min $\sum$ shrinking	unknown complexity $(1+\varepsilon)$-approx in $O(n^5/\varepsilon^{2.5})$ time	(weakly!!) NP-hard

Figure 4 Summary of knowledge for barrier resilience problem with unit disks.

References

1 Helmut Alt, Sergio Cabello, Panos Giannopoulos, and Christian Knauer. Minimum cell connection in line segment arrangements. *Int. J. Comput. Geom. Appl.*, 27(3):159–176, 2017. doi:10.1142/S0218195917500017.

2. Édouard Bonnet, Sergio Cabello, and Wolfgang Mulzer. Maximum matchings in geometric intersection graphs. In *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020*, pages 31:1–31:17, 2020. `doi:10.4230/LIPIcs.STACS.2020.31`.
3. Sergio Cabello and Éric Colin de Verdière. Hardness of minimum barrier shrinkage and minimum installation path. *Theoretical Computer Science*, in print. `doi:10.1016/j.tcs.2020.06.016`.
4. Sergio Cabello, Kshitij Jain, Anna Lubiw, and Debajyoti Mondal. Minimum shared-power edge cut. *Networks*, 75(3):321–333, 2020. `doi:10.1002/net.21928`.
5. Sergio Cabello and Wolfgang Mulzer. Minimum cuts in geometric intersection graphs. *CoRR*, abs/2005.00858, 2020. URL: `https://arxiv.org/abs/2005.00858`.
6. Timothy M. Chan. Klee's measure problem made easy. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013*, pages 410–419, 2013. `doi:10.1109/FOCS.2013.51`.
7. Bernard Chazelle and Herbert Edelsbrunner. An optimal algorithm for intersecting line segments in the plane. *J. ACM*, 39(1):1–54, 1992. `doi:10.1145/147508.147511`.
8. Herbert Edelsbrunner and John Harer. *Computational Topology - an Introduction*. American Mathematical Society, 2010. URL: `http://www.ams.org/bookstore-getitem/item=MBK-69`.
9. Alon Efrat, Alon Itai, and Matthew J. Katz. Geometry helps in bottleneck matching and related problems. *Algorithmica*, 31(1):1–28, 2001. `doi:10.1007/s00453-001-0016-8`.
10. Pegah Kamousi, Timothy M. Chan, and Subhash Suri. Stochastic minimum spanning trees in euclidean spaces. In *Proceedings of the 27th ACM Symposium on Computational Geometry, (SoCG 2011)*, pages 65–74, 2011. `doi:10.1145/1998196.1998206`.
11. Pegah Kamousi, Timothy M. Chan, and Subhash Suri. Closest pair and the post office problem for stochastic points. *Comput. Geom.*, 47(2):214–223, 2014. `doi:10.1016/j.comgeo.2012.10.010`.
12. Matias Korman, Maarten Löffler, Rodrigo I. Silveira, and Darren Strash. On the complexity of barrier resilience for fat regions and bounded ply. *Comput. Geom.*, 72:34–51, 2018. `doi:10.1016/j.comgeo.2018.02.006`.
13. Santosh Kumar, Ten-Hwang Lai, and Anish Arora. Barrier coverage with wireless sensors. *Wirel. Networks*, 13(6):817–834, 2007. `doi:10.1007/s11276-006-9856-0`.
14. Jiří Matoušek. Range searching with efficient hiearchical cutting. *Discret. Comput. Geom.*, 10:157–182, 1993. `doi:10.1007/BF02573972`.
15. Marcin Mucha and Piotr Sankowski. Maximum matchings in planar graphs via gaussian elimination. *Algorithmica*, 45(1):3–20, 2006. `doi:10.1007/s00453-005-1187-5`.
16. Franco P. Preparata and Michael I. Shamos. *Computational Geometry - An Introduction*. Texts and Monographs in Computer Science. Springer, 1985. `doi:10.1007/978-1-4612-1098-6`.
17. Michael O. Rabin. Probabilistic algorithms. In *Algorithms and Complexity: New Directions and Recent Results*. Academic Press, 1976.
18. Kuan-Chieh Robert Tseng and David G. Kirkpatrick. On barrier resilience of sensor networks. In *Algorithms for Sensor Systems - 7th International Symposium on Algorithms for Sensor Systems, Wireless Ad Hoc Networks and Autonomous Mobile Entities, ALGOSENSORS 2011*, volume 7111 of *Lecture Notes in Computer Science*, pages 130–144. Springer, 2011. `doi:10.1007/978-3-642-28209-6_11`.
19. Constantinos Tsirogiannis, Frank Staals, and Vincent Pellissier. Computing the expected value and variance of geometric measures. *ACM J. Exp. Algorithmics*, 23, 2018. `doi:10.1145/3228331`.
20. Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM J. Comput.*, 31(2):398–427, 2001. `doi:10.1137/S0097539797321602`.
21. Raphael Yuster and Uri Zwick. Maximum matching in graphs with an excluded minor. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*, pages 108–117, 2007. URL: `http://dl.acm.org/citation.cfm?id=1283383.1283396`.