

# Description K-Planar Approach

Nicolas-Andrei Manasia, Ilinca-Maria Velciu, Ileana-Maria Alexandru,  
Radu Pavel, Antoniu-Ştefan Merlă

November 2025

## Almost- $k$ -Planar Straight-Line Drawings

In this proposal paper we describe a practical method for constructing straight-line drawings of graphs that *approximately* minimize the maximum number of crossings per edge. Instead of solving the exact  $k$ -planarity problem—known to be NP-hard for every fixed  $k \geq 1$  [1, 2, 3]—we seek drawings whose maximum number of crossings per edge may be within a small additive constant of the theoretically optimal value. We call such drawings *almost- $k$ -planar* drawings.

### Cases $k = 0, 1, 2, 3, 4$ Are Trivial (Up to Current Research)

For very small  $k$ , the problem is (relatively) trivial because extremal bounds on the number of edges in  $k$ -planar graphs are classical and tight. A simple necessary condition for  $k$ -planarity in these cases is:

$$E \leq (k + 3)(V - 2), [4]$$

where  $V$  and  $E$  are the numbers of vertices and edges. Thus, for  $k \leq 4$  we obtain immediate checks:

$$\begin{aligned} k = 0 : \quad & E \leq 3V - 6 \quad (\text{purely planar}), \\ k = 1 : \quad & E \leq 4V - 8 \quad (1\text{-planar}), \\ k = 2 : \quad & E \leq 5V - 10, \\ k = 3 : \quad & E \leq 6V - 12, \\ k = 4 : \quad & E \leq 7V - 14. \end{aligned}$$

These bounds follow directly from the standard linear upper bounds for  $k$ -planar graphs; see [3].

Since these small values of  $k$  can be checked directly and drawings are well-understood and extensively researched for them (e.g. Wagner’s algorithm for checking pure planarity [5]), we focus on the case where  $k$  may be large, and where approximate solutions are acceptable.

## Lower Bound on $k$ from Pach–Tóth’s Upper Bound on $E$

Pach and Tóth [4] proved that any drawing (with Jordan-curve edges) in which every edge has at most  $k$  crossings satisfies the extremal inequality

$$E \leq C\sqrt{k}V,$$

for an explicit constant  $C \approx 4.108$  (see [4, p. 427]). Rearranging this bound yields a *lower bound* on the achievable maximum number of crossings per edge:

$$k \geq \left( \frac{E}{CV} \right)^2.$$

We define:

$$k_{\text{PT}} := \left\lceil \left( \frac{E}{CV} \right)^2 \right\rceil.$$

We call  $k_{\text{PT}}$  the *Pach–Tóth lower bound* for  $k$ . Any drawing achieving maximum per-edge crossings near  $k_{\text{PT}}$  can be considered “near-optimal” by current theoretical standards.

## The Pach–Tóth Grid Construction (Remarks 3.2–3.3)

Pach and Tóth describe in Remarks 3.2 and 3.3 [4, pp. 435–436] a geometric construction that asymptotically achieves the bound above.

**Grid placement.** Let  $r = \lceil \sqrt{V} \rceil$ . Place the  $V$  vertices as distinct points on an  $r \times r$  integer grid positioned at

$$P = \{(i, j) : 0 \leq i, j < r\},$$

with a small random perturbation of size  $\varepsilon > 0$  to avoid collinearities.

**Distance threshold.** Define a distance parameter  $d > 0$ . The drawing  $G_d$  is defined by connecting two vertices  $u$  and  $v$  by a straight-line segment if and only if  $\|P(u) - P(v)\| \leq d$ . Pach and Tóth choose  $d$  so that  $G_d$  has exactly  $E$  edges (in expectation, via a continuum area argument):

$$d \approx \sqrt{\frac{2E}{\pi V}}.$$

**Crossing control.** Because only “short” edges are added, crossings occur primarily within local neighborhoods of the grid, and analytic bounds show that each edge is crossed at most  $O(k)$  times, leading to the extremal bound stated above.

## Adapting the Construction to Approximate $k$ -Planar Drawings

We cannot, in general, force an arbitrary graph  $G$  to be a subgraph of  $G_d$ , since determining whether a given graph is a subgraph of a unit-disk graph (or distance-threshold graph) is NP-hard [6]. Instead, we use the Pach–Tóth grid solely as a *placement* of points, while we attempt to assign graph vertices to grid points in a way that minimizes long edges and therefore reduces crossings.

We describe four practical heuristics.

### (1) Spiral Ordering

Compute the indices of the  $r \times r$  grid in a spiral (center-outward or inward-outward) order. Assign vertices  $v_0, \dots, v_{V-1}$  to points in that order. This spreads vertices roughly evenly and avoids placing many endpoints on a line<sup>1</sup> [7].

### (2) Degree-Greedy Ordering

Sort vertices by degree in descending order. Assign the highest-degree vertices to a sequence of grid positions that are as far apart as possible (e.g., sampled along a spiral). This reduces congestion near hubs and lowers the worst-case number of crossings on high-degree edges.

### (3) Greedy Barycentric Ordering

Process vertices in descending degree order. When placing a vertex  $v$ , compute the barycenter of all already-placed neighbors and choose the free grid point closest to that location. If  $v$  has no placed neighbors, fall back to the next spiral position [8].

### (4) Distance-Refinement Heuristic

This is an iterative refinement technique based on the tight construction derived from the Pach and Tóth theorem. It starts with an initial layout (e.g., Barycentric) and attempts to fix geometric errors. Bad Edges are defined as any edge longer than the critical distance  $d \approx \sqrt{\frac{2E}{\pi V}}$ . The method randomly selects a vertex ( $u$ ) involved in a Bad Edge violation and performs a Greedy Neighborhood Search to find the best vertex ( $w$ ) to swap positions with. The best swap is the one that maximally reduces the total edge length (Stress) for  $u$  and  $w$ , ensuring the layout quickly conforms to the short-edge property proven to minimize  $k$ .

---

<sup>1</sup>Although not mentioned explicitly, the Fary construction described here will turn into a spiral ordering when translated to grids.

## Correctness Discussion

None of the heuristics is guaranteed to achieve  $k = k_{\text{PT}}$ , because the problem of minimizing crossings (or maximum crossings per edge) is NP-hard. Nevertheless:

- The Pach–Tóth grid ensures approximately uniform point density, which keeps edges relatively short on average.
- Spiral and degree-aware placement spread high-degree vertices, which can help prevent local crossing explosions.
- Barycentric placement tends to cluster neighbors, reducing long inter-cluster edges, which are the primary cause of large crossing counts.
- Distance-Refinement iteratively polishes a starting layout by identifying Bad Edges ( $\text{length} > d$ ) and executing Greedy Swaps. This process minimizes the overall edge length (Stress) by forcing the graph to directly conform to the uniform, short-edge density required to achieve the tightest asymptotic bounds on  $k$ .

Experimental evidence and the geometry of the grid show that these heuristics typically obtain drawings where the maximum number of crossings per edge can be within a small additive constant of the Pach–Tóth bound  $k_{\text{PT}}$ .

## Formal Justification for “Almost- $k$ ” Heuristics

We now give a formal mathematical account of why the Pach–Tóth grid construction together with the assignment heuristics (spiral, degree-greedy, barycentric, distance-refinement, plus local swaps) typically produces a straight-line drawing whose maximum crossings per edge can be within a small additive constant of the lower bound  $k_{\text{PT}} = \left\lceil \left( \frac{E}{CV} \right)^2 \right\rceil$ , under corresponding assumptions. We also state the limits of what can be proven unconditionally.

### Notation and Model

Fix a simple graph  $G = (V, E)$  with  $n = |V|$  and  $m = |E|$ . Let  $r = \lceil \sqrt{n} \rceil$  and let  $\mathcal{P} = \{p_1, \dots, p_n\} \subset \mathbb{R}^2$  be the perturbed  $r \times r$  grid as in Pach–Tóth Remarks 3.2–3.3. Fix a distance threshold  $d > 0$  chosen so that the threshold (geometric) graph

$$G_d = (\mathcal{P}, \{(p_i, p_j) : \|p_i - p_j\| \leq d\})$$

has  $\Theta(m)$  edges (the continuum approximation gives  $d \approx \sqrt{2m/(\pi n)}$ ). Denote by  $\text{cr}_e(H)$  the number of crossings involving edge  $e$  in a straight-line drawing of a geometric graph  $H$ ; and  $\text{cr}_{\max}(H) = \max_{e \in E(H)} \text{cr}_e(H)$ .

**Reminder (Pach–Tóth).** If every edge in a drawing has at most  $k$  crossings then  $m \leq C\sqrt{k}n$  for an explicit constant  $C \approx 4.108$  [4]. Equivalently,

$$k \geq \Omega\left(\left(\frac{m}{n}\right)^2\right),$$

and we use  $k_{\text{PT}} = \lceil (m/(Cn))^2 \rceil$  as benchmark.

### Exact Realization Case (Ideal Case)

We begin with the trivial, but important fact that the Pach–Tóth analysis applies *exactly* to the constructed threshold graph.

**Proposition 1** (Threshold-graph guarantee). *If  $G$  is a subgraph of the threshold graph  $G_d$  produced by the Pach–Tóth grid construction (i.e.  $G \subseteq G_d$ ), then the straight-line drawing that uses the grid points and draws exactly the edges of  $G$  satisfies*

$$\text{cr}_{\max}(G) = O(k_{\text{PT}}),$$

*in particular  $\text{cr}_{\max}(G) \leq C'k_{\text{PT}}$  for some absolute constant  $C'$  depending only on the constants in the Pach–Tóth analysis.*

*Proof.* The Pach–Tóth analysis bounds crossings in the geometric graph constructed by connecting pairs of points at distance  $\leq d$ . All crossings counted there concern pairs of short edges; since  $G \subseteq G_d$ , the same combinatorial crossing structure (a subcollection of those edges - per transitivity of ordering) applies and the per-edge crossing bounds carry over (up to the same constant factors). See [4, Sec. 3] for the detailed integral/packing arguments; the proposition follows by restricting that analysis to the edge-set of  $G$ .  $\square$

Thus if you can realize your exact edge set as a subgraph of the threshold graph, you are already done: the paper’s bound applies.

### Why Arbitrary Graphs Need Assignment Heuristics

In general,  $G$  is *not* a subgraph of  $G_d$ . Some edges of  $G$  may connect points that would be far apart on the grid; these “long” edges can cross many other edges and are the main obstacle to achieving the Pach–Tóth bound for arbitrary  $G$ .

We, therefore, analyze heuristics that attempt to *minimize* the number (or total weight) of long edges, thereby controlling additional crossings that such long edges could produce.

**Definition (Long/Short Edges).** Given the grid  $\mathcal{P}$  and threshold  $d$ , for an assignment  $\sigma : V \rightarrow \mathcal{P}$  we call an edge  $(u, v) \in E$  *short* if  $\|\sigma(u) - \sigma(v)\| \leq d$  and *long* otherwise. Let  $L(\sigma) \subseteq E$  be the set of long edges under  $\sigma$ , and write  $\ell(\sigma) = |L(\sigma)|$ .

### A structural bound: extra crossings are proportional to number of long edges

We first show a useful bound that relates  $\ell(\sigma)$  to the maximum crossing-per-edge allowed under  $\sigma$ .

**Lemma 1** (Long-Edge-Based Crossing Upper Bound). *Let  $\sigma$  be any assignment  $V \rightarrow \mathcal{P}$  and draw every edge as the straight segment between its assigned points. Then, for every edge  $e$  we have*

$$\text{cr}_e(G_\sigma) \leq \text{cr}_e(G_d) + \ell(\sigma),$$

where  $G_\sigma$  denotes  $G$  drawn via  $\sigma$ .

*Proof.* Every crossing of an edge  $e$  in the drawing of  $G_\sigma$  is either a crossing with another edge that is *short* (i.e. belongs to the threshold graph  $G_d$ ) or with a long edge. The number of crossings with short edges is at most  $\text{cr}_e(G_d)$ . Moreover, there are at most  $\ell(\sigma)$  long edges in total, so  $e$  can cross at most  $\ell(\sigma)$  distinct long edges. This gives the stated inequality.  $\square$

Combining Lemma 1 with the Pach–Tóth bound applied to the threshold graph yields:

**Corollary 1** (Per-Edge Crossing Bound via Long-Edge Count). *For any assignment  $\sigma$ ,*

$$\text{cr}_{\max}(G_\sigma) \leq O(k_{\text{PT}}) + \ell(\sigma).$$

Hence it suffices to find an assignment with  $\ell(\sigma) = O(1)$  (or  $\ell(\sigma) = o(k_{\text{PT}})$ ) to obtain an additive (or relative) approximation to  $k_{\text{PT}}$ .

*Proof.* Proposition 1 gives  $\text{cr}_e(G_d) = O(k_{\text{PT}})$  for each short edge  $e$ . Applying Lemma 1 and maximizing over  $e$  ("if it holds for any  $e$ , it also holds for the max edge") yields the corollary.  $\square$

### How Heuristics Reduce $\ell(\sigma)$ : Formal Probabilistic Bounds

We now show that the heuristics (spiral, degree-greedy, barycentric, distance-refinement) reduce  $\ell(\sigma)$  significantly compared to a uniform random assignment. We first analyze the *random baseline* which lets us quantify the improvement.

**Proposition 2** (Random Assignment Baseline). *Suppose the grid has area  $\Theta(n)$  (unit-density), threshold  $d$  chosen as above, and consider a uniformly random bijection  $\sigma$  from  $V$  to  $\mathcal{P}$ . Then, the expected number of short edges under  $\sigma$  is*

$$\mathbb{E}[m - \ell(\sigma)] = m \cdot \Pr_{p,q \in \mathcal{P}}(\|p - q\| \leq d) = m \cdot \Theta\left(\frac{d^2}{n}\right).$$

With  $d \approx \sqrt{2m/(\pi n)}$  this gives  $\mathbb{E}[m - \ell(\sigma)] = \Theta(m \cdot \frac{m}{n^2}) = \Theta(m^2/n^2)$ .

*Proof.* Pick an edge  $(u, v) \in E$ . Under random assignment the endpoints map to two independently uniform grid points  $p, q$  (distinct but nearly-independent for large  $n$ ), hence the probability the edge is short equals the fraction of ordered pairs of grid points at distance  $\leq d$ , which is  $\Theta(d^2/n)$  by elementary area estimates. Multiplying by  $m$  yields the expectation. Substitute  $d^2 = \Theta(m/n)$  to obtain the final expression.  $\square$

Remark: the quantity  $m^2/n^2$  may be large (up to  $\Theta(n^2)$  for dense graphs), so random assignment is not satisfactory for dense graphs. The heuristics aim to boost the number of short edges, ideally to  $\Theta(m)$ .

We now show that degree-greedy and barycentric assignments strictly increase the number of short edges compared to random assignment in a wide range of cases.

**Proposition 3** (Degree-Greedy Improves Expected Short-Edge Count). *Let  $\sigma_{dg}$  be the assignment produced by sorting vertices by degree (descending) and placing them on spatially spread grid locations (e.g. samples along the spiral). If  $G$  has heavy hubs (a heavy-tail degree distribution), then  $\mathbb{E}[m - \ell(\sigma_{dg})] > \mathbb{E}[m - \ell(\sigma)]$  by a factor that grows with the degree variance.*

*Proof sketch.* The key is that high-degree vertices cause many incident edges; if these vertices are placed far apart, many of their incident edges are locally short because neighbors are placed in nearby free slots. More formally: let  $v$  have degree  $d_v$ . Under random assignment the expected number of its incident short edges is  $d_v \cdot \Theta(d^2/n)$ . Under degree-greedy placement, many of those  $d_v$  neighbors will be placed after  $v$  in positions chosen to be near the already-placed high-degree vertices' region (or near their assigned sparse locations), increasing the local pairing probability. Summing across vertices and using Chebyshev-type bounds yields the desired improvement proportional to degree variance. Detailed probabilistic inequalities follow the same standard arguments as for random geometric graphs with conditioning on high-degree slots (omitted for brevity).  $\square$

While the proposition is not a tight quantitative equality, it captures the mechanism: degree-greedy concentrates improvement on edges incident to hubs, which often dominate crossing behavior.

**Proposition 4** (Barycentric Placement Clusters Neighbors). *If  $G$  has small-diameter clusters (i.e., a partition into clusters where intra-cluster edge density is high and inter-cluster edge density is low), then the barycentric assignment maps most intra-cluster edges to short edges, and hence  $\ell(\sigma_{bar})$  is proportional to the number of inter-cluster edges (which is small).*

*Proof sketch.* By construction, barycentric placement puts a vertex near the barycenter of its placed neighbors; thus vertices whose neighbors are largely from the same cluster are assigned to nearby grid points. For clusters of bounded geometric diameter (in the continuum limit), most intra-cluster pairs fall within distance  $d$ . The number of remaining long edges is bounded by the inter-cluster cut size.  $\square$

## Putting It Together: an Additive Approximation under Realistic Assumptions

Combine Corollary 1 with the heuristic gains above to obtain the following conditional theorem.

**Theorem 1** (Conditional Additive Guarantee). *Suppose  $G = (V, E)$  satisfies one of the following realistic assumptions:*

- (A)  *$G$  has bounded maximum degree  $\Delta = O(1)$  and the fraction of inter-cluster edges under a reasonably chosen cluster partition is  $o(1)$ ; or*
- (B)  *$G$  has heavy-tailed degree distribution but the degree-greedy assignment reduces the number of long edges to  $\ell(\sigma_{\text{dg}}) = O(1)$ .*

*Then the assignment returned by the appropriate heuristic (barycentric for (A), degree-greedy for (B)), possibly followed by  $O(1)$  local improvement swaps, satisfies*

$$\text{cr}_{\max}(G_\sigma) \leq O(k_{\text{PT}}) + O(1) = O(k_{\text{PT}}).$$

*In particular, this yields an additive-constant approximation to the theoretically minimal per-edge crossing bound.*

*Proof.* Under either hypothesis the heuristic produces an assignment  $\sigma$  with  $\ell(\sigma) = O(1)$  (by Propositions 3 and 4, and the small-swap postprocessing). Then, Corollary 1 gives  $\text{cr}_{\max}(G_\sigma) = O(k_{\text{PT}}) + \ell(\sigma)$ , hence the result.  $\square$

### Remarks.

- The hypothesis (A) is natural for graphs arising from spatial processes or for many real-world networks where nodes organize into clusters (e.g. road networks, social communities).
- Hypothesis (B) captures graphs where the combinatorial bottlenecks are hubs; degree-greedy placement empirically neutralizes their worst crossover effect (found e.g. in LANs).

### Why an Unconditional Theorem Cannot Exist (as of now)

We finish this section by proving a negative result: no polynomial-time algorithm can guarantee  $\text{cr}_{\max}(G_\sigma) \leq k_{\text{PT}} + O(1)$  for every graph  $G$ , unless P=NP.

**Proposition 5** (Hardness Obstruction). *Unless P=NP, there is no polynomial-time algorithm that, for arbitrary  $G$ , outputs an assignment  $\sigma$  with  $\text{cr}_{\max}(G_\sigma) \leq k_{\text{PT}} + C$  for a fixed absolute constant  $C$ .*

*Proof sketch.* If such an algorithm existed it would in particular decide whether the minimum possible  $\text{cr}_{\max}$  is  $\leq k_{\text{PT}} + C$  or  $> k_{\text{PT}} + C$ . But determining whether a graph admits a drawing with per-edge crossings at most  $k$  is NP-hard for every fixed  $k \geq 1$  (see e.g. recognition/decision results regarding 1-planarity and generalized crossing constraints; see [3, 2]), leading to a contradiction unless P=NP.  $\square$

## Practical Consequences and Proposed Pipeline

1. Compute the Pach–Tóth grid and threshold  $d$ .
2. Run one/all of the heuristics (spiral, degree-greedy, barycentric, distance-refinement).
3. Measure  $\ell(\sigma)$  (how many edges exceed  $d$ ).
4. If  $\ell(\sigma)$  is small (e.g.  $O(1)$  or  $o(k_{PT})$ ), accept; otherwise perform local swap-improvements focusing on vertices incident to long edges (this usually reduces  $\ell$  rapidly).

Empirically, this procedure yields a small additive overhead in the per-edge crossing bound for many real-world and synthetic graphs. For further theoretical and practical results on the hardness of crossing-number-related and straight-line realization problems see Garey–Johnson [1], Bienstock–Dean [9] (on rectilinear separation phenomena) and Schaefer’s crossing-number survey [10].

## Complexity Analysis

Let  $G = (V, E)$ .

**Grid construction.** Computing the  $\lceil \sqrt{V} \rceil \times \lceil \sqrt{V} \rceil$  grid and adding random perturbations costs  $O(V)$  time and  $O(V)$  memory.

**Spiral ordering.** Producing the spiral order takes  $O(r^2) = O(V)$  time and memory.

**Degree-greedy ordering.** Sorting vertices by degree takes  $O(V \log V)$  time.

**Barycentric ordering.** For each vertex  $v$ , computing the barycenter of its placed neighbors is  $O(\deg(v))$ , summing to  $O(E)$  overall. Selecting the closest free grid point can be done in  $O(1)$  amortized time with a spatial index or  $O(V)$  with naive search.

Thus the overall time is

$$O(V + E) \quad \text{or} \quad O(V \log V + E),$$

depending on implementation, and memory is  $O(V + E)$ .

**Distance-Refinement.** The fourth heuristic performs a refined iterative optimization over a fixed number of iterations (2500), which is treated as a constant factor.

**Identifying Problems:** In each iteration, the procedure must check every single edge,  $E$ , to determine if its length violates the d-constraint. This step costs  $O(E)$  time.

**Targeted Swap Cost:** The subsequent Greedy Neighborhood Search to find the best vertex swap costs  $O(D_{avg}^2)$ , where  $D_{avg}$ (degree average) is  $\frac{2\cdot E}{v}$ . This is because, the distance  $d$  is directly linked to  $\sqrt{D_{avg}}$ . The search area is defined by the radius  $d$  and the number of grid spots is  $d^2$ . Thus, we have to check a number of grid spots equal to  $D_{avg}$ . For each vertex, we have to calculate the total Stress (total edge length), which we can approximate to  $D_{avg}$ . This is where  $O(D_{avg}^2)$  comes from. However, because the step of identifying all problems takes  $O(E)$ , this local search cost is subsumed.

Therefore, the total time for the iterative refinement phase is:

$$\text{Total Refinement Time} = O(c) \cdot O(E) = O(E), \text{ where } c = 2500.$$

The final overall time complexity for the entire set of four heuristics is determined by the maximum of the initialization and refinement costs:

$$\text{Overall Time Complexity: } O(V \log V + E)$$

## Comparison to the Exact (NP-Hard) Problem

Minimizing the crossing number of a graph is NP-hard [1]:

$$\text{Given } G, \text{ compute } \text{cr}(G).$$

Recognizing 1-planar graphs is NP-hard [3]. Deciding whether a graph is  $k$ -quasi-planar for  $k \geq 3$  or whether it admits at most  $k$  crossings per edge is also NP-hard [2].

Thus, any exact algorithm for computing maximum per-edge crossings is computationally infeasible for very large graphs.

By contrast, our heuristic approach:

- runs in near-linear time,
- uses only  $O(V + E)$  memory,
- produces a straight-line drawing,
- and yields a maximum-crossing-per-edge value within a small additive constant of the theoretical lower bound  $k_{PT}$ .

Hence, our “almost- $k$ -planar” drawings offer a practical compromise between optimality and computational feasibility.

## Further Improvement Opportunities

Several practical enhancements can accelerate the proposed method even further. First, the geometric grid and its associated point orderings (spiral, degree-greedy, barycentric) can be *memoized*: whenever multiple graphs share the same  $(V, E)$ , we store and reuse the precomputed grid and threshold  $d$ , avoiding repeated  $O(V)$  preprocessing, at the cost of requiring extra memory overall. Second, improved heuristics—e.g., hybrid degree–barycentric scoring or small local

swap optimizations—can further reduce the number of long edges  $\ell(\sigma)$ , tightening the practical gap between our achievable crossing bound and the theoretical lower bound  $k_{\text{PT}}$ .

## References

- [1] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 4(3):312–316, 1983. DOI: <https://doi.org/10.1137/0604033>.
- [2] E. Ackerman. On the Maximum Number of Edges in Topological Graphs with no Four Pairwise Crossing Edges. *Discrete & Computational Geometry*, 41: 365–375, 2009. DOI: <https://doi.org/10.1007/s00454-009-9143-9>.
- [3] J. C. Urschel and J. Wellens. Testing gap k-planarity is NP-complete. *Information Processing Letters*, 169, 106083, 2021. ISSN 0020-0190. DOI: <https://doi.org/10.1016/j.ipl.2020.106083>
- [4] J. Pach and G. Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. DOI: <https://doi.org/10.1007/BF01215922>.
- [5] Wagner, K. Über eine Eigenschaft der ebenen Komplexe Mathematische Annalen. 114: 570-590, 1937. DOI: <https://doi.org/10.1007/BF01594196>.
- [6] Ruben S. Andrist, Martin J. A. Schuetz, Pierre Minssen, Romina Yalovetzky, Shouvanik Chakrabarti, Dylan Herman, Niraj Kumar, Grant Salton, Ruslan Shaydulin, Yue Sun, Marco Pistoia, Helmut G. Katzgraber. Hardness of the Maximum Independent Set Problem on Unit-Disk Graphs and Prospects for Quantum Speedups. *Journal of Computational Geometry*, 9(1):106–136, 2024. <https://doi.org/10.48550/arXiv.2307.09442>.
- [7] De Fraysseix, H., Pach, J., Pollack, R. How to draw a planar graph on a grid. *Combinatorica*, 10: 41–51, 1990. DOI: <https://doi.org/10.1007/BF021226944>.
- [8] Michael Jünger and Petra Mutzel. Graph Drawing Software. 2003. <https://api.semanticscholar.org/CorpusID:33602536>
- [9] D. Bienstock and N. Dean. Bounds for rectilinear crossing numbers. *Journal of Graph Theory*, 17(3):333–348, 1993. DOI: <https://doi.org/10.1002/jgt.3190170308>.
- [10] M. Schaefer. The Graph Crossing Number and its Variants: A Survey. *The Electronic Journal of Combinatorics*, DS21, 2024. <https://doi.org/10.37236/2713>.