

Drawing Arrangement Graphs in Small Grids, or How to Play Planarity

David Eppstein*

Department of Computer Science, University of California, Irvine, USA

Abstract. We describe a linear-time algorithm that finds a planar drawing of every graph of a simple line or pseudoline arrangement within a grid of area $O(n^{7/6})$. No known input causes our algorithm to use area $\Omega(n^{1+\epsilon})$ for any $\epsilon > 0$; finding such an input would represent significant progress on the famous k -set problem from discrete geometry. Drawing line arrangement graphs is the main task in the *Planarity* puzzle.

1 Introduction

Planarity (<http://planarity.net/>) is a puzzle developed by John Tantaló and Mary Radcliffe in which the user moves the vertices of a planar graph, starting from a tangled circular layout (Figure 1), into a position where its edges (drawn as line segments) do not cross. The game is played in a sequence of levels of increasing difficulty. To construct the graph for the i th level, the game applet chooses $\ell = i + 3$ random lines in general position in the plane. It creates a vertex for each of the $\ell(\ell - 1)/2$ crossings of two lines, and an edge for each of the $\ell(\ell - 2)$ consecutive pairs of crossings on the same line.

One strategy for solving *Planarity* would be to reconstruct a set of lines forming the given graph (Figure 2, left). However, this is tedious to do by hand, and has high computational complexity: testing whether an arrangement of curves is combinatorially equivalent to a line arrangement is NP-hard [1], from which it follows that recognizing line arrangement graphs is also NP-hard [2]. More precisely, both problems are complete for the existential theory of the reals [3]. And although drawings constructed in this way accurately convey the underlying construction of the graph, they have low angular resolution (at most π/ℓ) and close vertex spacing, making them hard to read and hard to place by hand. In practice these puzzles may be solved more easily by an incremental strategy

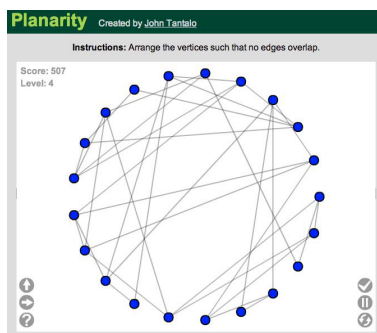


Fig. 1. Initial state of *Planarity*

* This research was supported in part by NSF grants 0830403 and 1217322 and by the Office of Naval Research under grant N00014-08-1-1015.

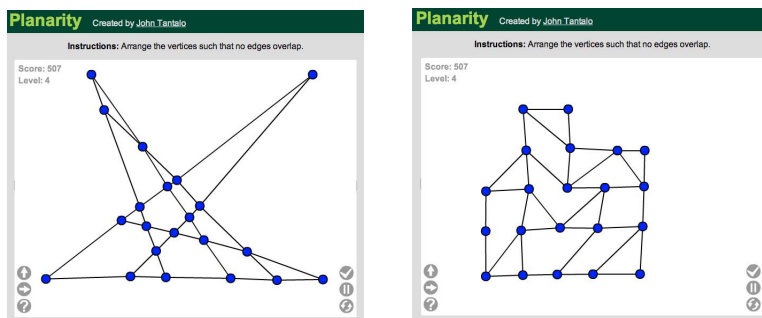


Fig. 2. Two manually constructed solutions to the puzzle from Figure 1. Left: a set of lines with this graph as its arrangement. Right: an (approximate) grid layout.

that maintains a planar embedding of a subgraph of the input, starting from a single short cycle (such as a triangle or quadrilateral), and that at each step extends the embedding by a single face, bounded by a short path connecting two vertices on the boundary of the previous embedding. When using this strategy to solve a Planarity puzzle, the embedding may be kept tidy by placing each vertex into an approximate grid (Figure 2, right). Curiously, the grid drawings found by this incremental grid-placement heuristic appear to have near-linear area; in contrast, there exist planar graphs such as the *nested triangles graph* that cannot be drawn planarly in a grid of less than $\Theta(n^2)$ area [4, 5].

In this paper we explain this empirical finding of small grid area by developing an efficient algorithm for constructing compact grid drawings of the arrangement graphs arising in Planarity. Because recognizing line arrangement graphs is NP-hard, we identify a larger family of planar graphs (the graphs of simple pseudoline arrangements) that may be recognized and decomposed into pseudolines in linear time. We show that every n -vertex simple pseudoline arrangement graph may be drawn in linear time in a grid of size $\kappa_{\max}(O(\sqrt{n})) \times O(\sqrt{n})$; here $\kappa_{\max}(\ell)$ is the maximum complexity of a k -level of a pseudoline arrangement with ℓ pseudolines [6–8], a topological variant of the famous k -set problem from discrete geometry (see Section 3 for a formal definition). The best proven upper bounds of $O(\ell^{4/3})$ on the complexity of k -levels [7–9] imply that the grid in which our algorithm draws these graphs has size $O(n^{2/3}) \times O(\sqrt{n})$ and area $O(n^{7/6})$. However, all known lower bounds on k -level complexity are of the form $\Omega(\ell^{1+o(1)})$ [6, 10], suggesting that our algorithm is likely to perform even better in practice than our worst-case bound. If we could find a constant $\epsilon > 0$ and a family of inputs that would cause our algorithm to use area $\Omega(\ell^{1+\epsilon})$, such a result would represent significant progress on the k -set problem.

We also investigate the construction of *universal point sets* for arrangement graphs, sets of points that can be used as the vertices for a straight-line planar drawing of every n -vertex arrangement graph. Our construction directly provides a universal point set consisting of $O(n^{7/6})$ grid points; we show how to sparsify this structure, leading to the construction of a universal set of $O(n \log n)$ points within a grid whose dimensions are again $O(n^{2/3}) \times O(\sqrt{n})$.

Finally, we formalize and justify an algorithm for manual solution of these puzzles that greedily finds short cycles and adds them as faces to a partial planar embedding. Although this algorithm may fail for general planar graphs, we show that for arrangement graphs it always finds a planar embedding that is combinatorially equivalent to the original arrangement.

2 Preliminaries

Following Shor [1], we define a *pseudoline* to be the image of a line under a homeomorphism of the Euclidean plane. Pseudolines include lines, non-self-crossing polygonal chains starting and ending in infinite rays, and the graphs of continuous real functions. Two pseudolines *cross* at a point x if a neighborhood of x is homeomorphic to a neighborhood of the crossing point of two lines, with the homeomorphism taking the pseudolines to the lines. An *arrangement* of pseudolines is a finite set of pseudolines, the intersection of every two of which is a single crossing point. An arrangement is *simple* if all pairs of pseudolines have distinct crossing points. A *pseudoline arrangement graph* is a planar graph whose vertices are the crossings in a simple pseudoline arrangement, and whose edges connect consecutive crossings on a pseudoline.

Most of the ideas in the following result are from Bose et al. [2], but we elaborate on that paper to show that linear time recognition of arrangement graphs is possible. (See [12] for a more complicated linear time algorithm that recognizes the dual graphs of a wider class of arrangement graphs, the graphs of *weak* pseudoline arrangements in which pairs of pseudolines need not cross)

Lemma 1. *If we are given as input a graph G , then in linear time we can determine whether it is a pseudoline arrangement graph, determine its (unique) embedding as an arrangement graph, and find a pseudoline arrangement for which it is the arrangement graph.*

Proof. Let G^* be formed from a pseudoline arrangement graph G by adding a new vertex v_∞ adjacent to all vertices in G of degree less than four. As Bose et al. [2] show, G^* is 3-connected and planar, and its unique planar embedding is compatible with the embedding of G as an arrangement graph. For convenience we include two edges in G^* from v_∞ to each degree two vertex in G , so that, in G^* , all vertices except v_∞ have degree four. With this modification, the pseudolines of the arrangement for G are represented in G^* by paths starting and ending at v_∞ that, at each other vertex, connect two opposite edges in the embedding.

For any given graph G of maximum degree four we may, in linear time, add a new vertex v_∞ , test planarity of the augmented graph G^* , and embed G^*

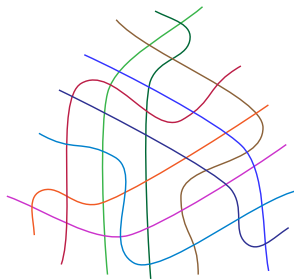


Fig. 3. A simple pseudoline arrangement that cannot be transformed into a line arrangement. Redrawn from Figure 5.3.2 of [11], who attribute this arrangement to Ringel.

in the plane. The edge partition of G^* into paths through opposite edges at each degree four vertex may be found in linear time by connected component analysis. By labeling each edge with the identity of its path, we may verify that this partition does not include cycles disjoint from v_∞ and that no path crosses itself. We additionally check that G has $\ell(\ell-1)/2$ vertices, where ℓ is the number of paths. Finally, by listing the pairs of paths passing through each vertex and bucket sorting this list, we may verify in linear time that no two paths cross more than once. If G passes all of these checks, its decomposition into paths gives a valid pseudoline arrangement, which may be constructed by viewing the embedding of G^* as being on a sphere, puncturing the sphere at point v_∞ , and homeomorphically mapping the punctured sphere to the plane. \square

3 Small Grids

To describe our grid drawing algorithm for pseudoline arrangement graphs, we need to introduce the concept of a *wiring diagram*. A wiring diagram is a particular kind of pseudoline arrangement, in which the ℓ pseudolines largely lie on ℓ horizontal lines (with coordinates $y = 1, y = 2, \dots, y = \ell$). The pseudolines on two adjacent tracks may cross each other, swapping which track they lie on, near points with coordinates $x = 1, x = 2, \dots, x = \ell(\ell - 1)/2$; each crossing is formed by removing two short segments of track and replacing them by two crossing line segments between the tracks. It is convenient to require different crossings to have different x coordinates, following Goodman [13], although some later sources omit this requirement. Figure 4 depicts an example. Wiring diagrams already provide reasonably nice grid drawings of arrangement graphs [14], but are unsuitable for our purposes, for two reasons: they draw the edges connecting pairs of adjacent crossings as polygonal chains with two bends, and for some arrangements, even allowing crossings to share x -coordinates, drawing the wiring diagram of ℓ lines in a grid may require width $\Omega(\ell^2)$ (Figure 5), much larger than our bounds. Instead, we will use these diagrams as a tool for constructing a different and more compact straight-line drawing.

For an arrangement of non-vertical lines in general position, an equivalent wiring diagram may be constructed by a *plane sweep* algorithm [15], which simulates the left-to-right motion of a vertical line across the arrangement. At most

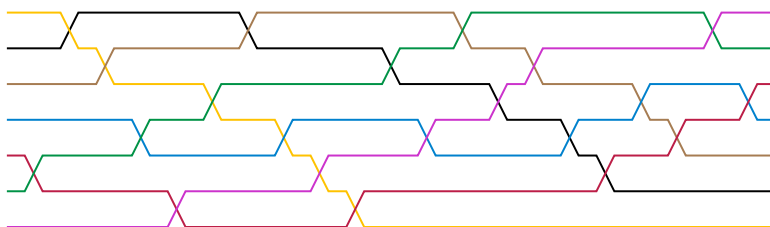


Fig. 4. A wiring diagram formed by a plane sweep of the arrangement from Figure 2

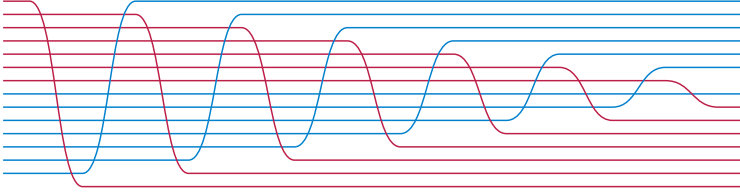


Fig. 5. Cocktail shaker sort corresponds to an arrangement of ℓ pseudolines for which drawing the wiring diagram in a grid requires width $\Omega(\ell^2)$

points in the sweep, the intersection points of the arrangement lines with the sweep line maintain a fixed top-to-bottom order with each other, with their positions in this order reflected in the assignment of the corresponding pseudolines to tracks. When the sweep line crosses a vertex of the arrangement, two intersection points swap positions in the top-to-bottom order, corresponding to a crossing in the wiring diagram. The left-to-right order of crossings in the wiring diagram is thus exactly the sorted order of the crossing points of the arrangement, as sorted by their x coordinates. The wiring diagram in Figure 4 was constructed in this way from the approximate line arrangement depicted in Figure 1.

Every simple pseudoline arrangement, also, has an equivalent wiring diagram, that may be constructed in time linear in its number of crossings. The proof of this fact uses *topological sweeping*, a variant of plane sweeping originally developed to speed up sweeping of straight line arrangements by relaxing the strict left-to-right ordering of the crossing points [16], that can also be extended to apply to pseudoline arrangements [17]. The steps of the topological sweeping algorithm require only determining the relative ordering of crossings along each of the input pseudolines, something that may easily be determined from our path decomposition of a pseudoline arrangement graph by precomputing the position of each crossing on each of the two pseudolines it belongs to.

We define the i th level $L_{\mathcal{D}}(i)$ in a wiring diagram \mathcal{D} to be the set of crossings that occur between tracks i and $i + 1$. A crossing belongs to $L_{\mathcal{D}}(i)$ if and only if $i - 1$ lines pass between it and the bottom face of the arrangement (the face below all of the tracks in the wiring diagram); therefore, once this bottom face is determined, the levels are fixed by this choice regardless of how the crossings are ordered to form a wiring diagram. If we define the size $|\mathcal{D}|$ of a diagram to be its number of pseudolines, and the level complexity $\kappa(\mathcal{D})$ to be $\max_i |L_{\mathcal{D}}(i)|$, then it is a longstanding open problem in discrete geometry (a variant of the k -set problem) to determine the maximum level complexity of an arrangement of ℓ pseudolines, $\kappa_{\max}(\ell) = \max_{|\mathcal{D}|=\ell} \kappa(\mathcal{D})$. (Often this problem is stated in terms of the middle level of an arrangement, rather than the maximum-complexity level, but this variation makes no difference to the asymptotic behavior of the level complexity.) The known bounds on this quantity are $\kappa_{\max}(\ell) = O(\ell^{4/3})$ [7–9], and $\kappa_{\max}(\ell) = \Omega(\ell c^{\sqrt{\log \ell}})$ for some constant $c > 1$ [6, 10], where the last bound is $O(n^{1+\epsilon})$ for all constants $\epsilon > 0$.

Theorem 1. *Let G be a pseudoline arrangement graph with n vertices, determined by $\ell = \Theta(\sqrt{n})$ pseudolines. Then in time $O(n)$ we may construct a planar straight-line drawing of G , in a grid of size $(\ell - 1) \times \kappa_{\max}(\ell) = O(n^{1/2}) \times O(n^{2/3})$.*

Proof. We find a decomposition of G into pseudoline paths, by the algorithm of Lemma 1, and use topological sweeping to convert this decomposition into a wiring diagram. We place each vertex v of G at the coordinates (i, j) , where i is the position of v within its level of the wiring diagram and j is the number of tracks below its level of the wiring diagram.

With this layout, every edge of G either connects consecutive vertices within the same level as each other, or it connects vertices on two consecutive levels. In the latter case, each edge between two consecutive levels corresponds to a horizontal segment of the wiring diagram that lies on the track between the two levels; the left-to-right ordering of these horizontal segments is the same as the left-to-right ordering of both the lower endpoints and the upper endpoints of these edges. Because of this consistent ordering of endpoints, no two edges between the same two consecutive levels can cross. There can also not be any crossings between edges that do not both lie in the same level or connect the same two consecutive levels. Therefore, the drawing we have constructed is planar. By construction, it has the dimensions given in the theorem. \square

Figure 6 depicts the output of our algorithm, using the wiring diagram of Figure 4, for the graph of Figure 1. The arrangement has six levels, with at most five vertices per level, giving a 6×5 grid. Although not as compact as the manually-found 5×5 grid of Figure 2, it is much smaller than standard grid drawings that do not take advantage of the arrangement structure of this graph. A more careful placement of vertices within each row would improve the angular resolution and edge length of the drawing but we have omitted this step in order to make the construction more clear.

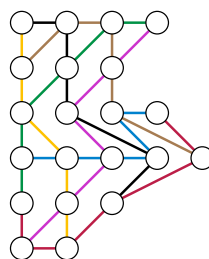


Fig. 6. Output of the drawing algorithm of Theorem 1, based on the wiring diagram of Figure 4

4 Universal Point Sets

A *universal point set* for the n -vertex graphs in a class \mathcal{C} of graphs is a set U_n of points in the plane such that every n -vertex graph in \mathcal{C} can be drawn with its vertices in U_n and with its edges drawn as non-crossing straight line segments [18]. Grids of $O(n) \times O(n)$ points form universal sets of quadratic size for the planar graphs [19, 20], and despite very recent improvements the best upper bound known remains quadratic [21]. A rectangular grid that is universal must have $\Omega(n^2)$ points [4, 5]; the best known lower bounds for universal point sets that are not required to be grids are only linear [18].

Subquadratic bounds are known on the size of universal point sets for subclasses of the planar graphs including the outerplanar graphs [22], simply-nested

planar graphs [21,23], planar 3-trees [24], and graphs of bounded pathwidth [21]; however, these results do not apply to arrangement graphs. The grid drawing technique of Theorem 1 immediately provides a universal point set for arrangement graphs of size $O(n^{7/6})$; in this section we significantly improve this bound, while only increasing the area of our drawings by a constant factor.

Following Bannister et al. [21], define a sequence of positive integers ξ_i for $i = 1, 2, 3, \dots$ by the equation $\xi_i = i \oplus (i - 1)$ where \oplus denotes the bitwise binary exclusive or operation. The sequence of these values begins

$$1, 3, 1, 7, 1, 3, 1, 15, 1, 3, 1, 7, 1, 3, 1, \dots$$

Lemma 2 (Bannister et al. [21]). *Let the finite sequence $\alpha_1, \alpha_2, \dots, \alpha_k$ have sum s . Then there is a subsequence $\beta_1, \beta_2, \dots, \beta_k$ of the first s terms of ξ such that, for all i , $\alpha_i \leq \beta_i$. The sum of the first s terms of ξ is between $s \log_2 s - 2s$ and $s \log_2 s + s$.*

Recall that the grid drawing technique of Theorem 1 produces a drawing in which the vertices are organized into $\ell - 1$ rows of at most $\kappa_{\max}(\ell) = O(\ell^{4/3})$ vertices per row, where $\ell = O(\sqrt{n})$ is the number of lines in the underlying n -vertex arrangement. In this drawing, suppose that there are n_i vertices on the i th row of the drawing, and define a sequence $\alpha_i = \lceil n_i/\ell \rceil$.

Lemma 3. $\sum \alpha_i \leq 3(\ell - 1)/2$.

Proof. We may partition the n_i vertices in the i th row n_i into $\lfloor n_i/\ell \rfloor$ groups of exactly ℓ vertices, together with at most one smaller group; then α_i is the number of groups. The contribution to $\sum \alpha_i$ from the groups of exactly ℓ vertices is at most $n/\ell = (\ell - 1)/2$. There is at most one smaller group per row so the contribution from the smaller groups is at most $\ell - 1$. Thus the total value of the sum is at most $3(\ell - 1)/2$. □

Theorem 2. *There is a universal point set of $O(n \log n)$ points for the n -vertex arrangement graphs, forming a subset of a grid of dimensions $O(\ell) \times \kappa_{\max}(\ell)$.*

Proof. Let $s = 3(\ell - 1)/2$. We form our universal point set as a subset of an $s \times \kappa_{\max}(\ell)$ grid; the area of the grid from which the points are drawn is exactly $3/2$ times the area of the $(\ell - 1)$ -row grid drawing technique of Theorem 1. In the i th row of this grid, we include in our universal point set $\min(\ell \xi_i, \kappa_{\max}(\ell))$ of the grid vertices in that row. It does not matter for our construction exactly which points of the row are chosen to make this number of points.

By Lemma 2, there is a subsequence β_i of the first s rows of sequence ξ , such that the β is termwise greater than or equal to α . This subsequence corresponds to a subsequence $(r_1, r_2, \dots, r_{\ell-1})$ of the rows of our universal point set, such that row r_i has at least $\min(\ell \beta_i, \kappa_{\max}(\ell)) \geq n_i$ points in it. Mapping the i th row of the drawing of Theorem 1 to row r_i of this point set will not create any crossings, because the mapping is monotonic within each row and because all edges of the drawing connect pairs of vertices that are either in the same row or in rows that are consecutive in the selected subsequence.

The number of points in the point set is $O(\ell s \log s)$ where $s = O(\ell)$. Therefore, this number of points is $O(\ell^2 \log \ell) = O(n \log n)$. □

5 Greedy Embedding Algorithm

The algorithm of Lemma 1 uses as a subroutine a linear-time planarity testing algorithm. Although such algorithms may be efficiently implemented on computers, they are not really suitable for hand solution of Planarity puzzles. Instead, it is more effective in practice to build up a planar embedding one face at a time, by repeatedly finding a short cycle in the input graph and attaching it to the previously constructed partial embedding. Here “short” means as short as can be found; it is not possible to limit attention to cycles of length three, four, or any fixed bound. For instance in Figure 3 the central triangle is separated from the rest of the graph by faces with five sides, and by modifying this example it is possible to separate part of an arrangement graph from the rest of the graph by faces with arbitrarily many sides. Thus, this hand-solution heuristic may be formalized by the following steps.

1. Choose an arbitrary starting vertex v .
2. Find a cycle C_1 of minimum possible length containing v .
3. Embed C_1 as a simple cycle in the plane.
4. While some of the edges of the input graph have not yet been embedded:
 - (a) Let C_i be the cycle bounding the current partial embedding. Define an *attachment vertex* of C_i to be a vertex that is incident with edges not already part of the current embedding.
 - (b) Choose two attachment vertices u and v , and a path P_i in C_i from u to v , such that there are no attachment vertices interior to P_i .
 - (c) Find a shortest path S_i from u to v , using only edges that are not already part of the current partial embedding.
 - (d) If necessary, adjust the positions of the embedded vertices (without changing the combinatorial structure of the embedding) so that S_i may be drawn with straight line edges.
 - (e) Add S_i to the embedding, outside C_i , so that the new face between P_i and S_i does not contain C_i . After this change, the new bounding cycle C_{i+1} of the partial embedding is formed from C_i by replacing P_i by S_i .

When it is successful, this algorithm decomposes the input graph into the cycle C_1 and a sequence of edge-disjoint paths S_1, S_2 , etc. Such a decomposition is known as an *open ear decomposition* [25].

This greedy ear decomposition algorithm does not always work for arbitrary planar graphs: even the initial cycle that is found by the algorithm may not be a face of an embedding of the given graph, causing the algorithm to make incorrect assumptions about the structure of the embedding. However (ignoring the possible difficulty of performing step d) the algorithm does always correctly embed the arrangement graphs used by Planarity. These graphs may have multiple embeddings; to distinguish among them, define the *canonical embedding* of an arrangement graph to be the one given by the arrangement from which it was constructed. By Lemma 1, the canonical embedding is unique. As we prove below, the cycles of an arrangement graph that the algorithm assumes to be faces really are faces of the canonical embedding.

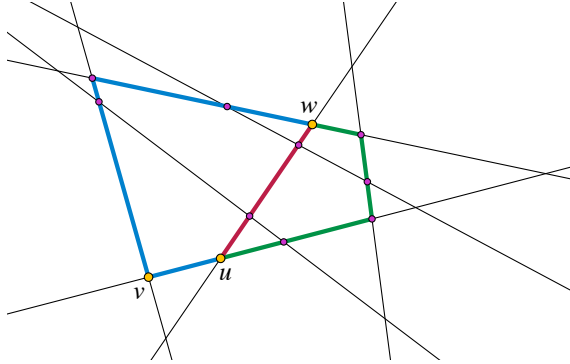


Fig. 7. Illustration for the proof of Lemma 4. Every non-facial cycle C through vertex v (blue and green edges) is crossed by at least one line $\ell = uw$ (red edges), forming a theta-graph. All the vertices on the red path of the theta are matched by an equal number of vertices on each of the other two paths, caused by crossings with the same lines, and the other two paths have additional vertices at their bends, so the red-blue cycle is shorter than the blue-green cycle.

Lemma 4. *Let v be an arbitrary vertex of arrangement graph G , and C be a shortest cycle containing v . Then C is a face of the canonical embedding of G .*

Proof. Let C be an arbitrary simple cycle through v . Then if C is not a face of the arrangement forming G , there is a line ℓ that crosses it; let u and w be two vertices on the boundary of C connected through the interior of C by ℓ (Figure 7). Then C together with the path along ℓ from u to w form a theta-graph, a graph with two degree three vertices (u and w) connected by three paths. Every vertex of ℓ between u and w is caused by a crossing of ℓ with another line that also must cross the other two paths of the theta-graph; in addition, each of these two paths must bend at least once at a vertex that does not correspond to a line that crosses ℓ . Therefore, the path through ℓ is strictly shorter than the other two paths in the theta-graph. Replacing one of the two paths of C from u to w by the path through ℓ produces a shorter cycle that still contains v . Since an arbitrary cycle C that is not a face can be replaced by a shorter cycle through v , it follows that every shortest cycle through v is a face. \square

Lemma 5. *Let D be a drawing of a subset of the faces of the canonical embedding of an arrangement graph G whose union is a topological disk, let u and v be two attachment vertices on the boundary of D with no attachment vertices interior to the boundary path P from u to v , and let S be a shortest path from u to v using only edges not already part of D . Then the cycle formed by the union of P and S is a face of the canonical embedding of G .*

Proof. Assume for a contradiction that $P \cup S$ is not a face; then as in the proof of Lemma 4, this cycle must be crossed by a line ℓ , a path L of which forms a

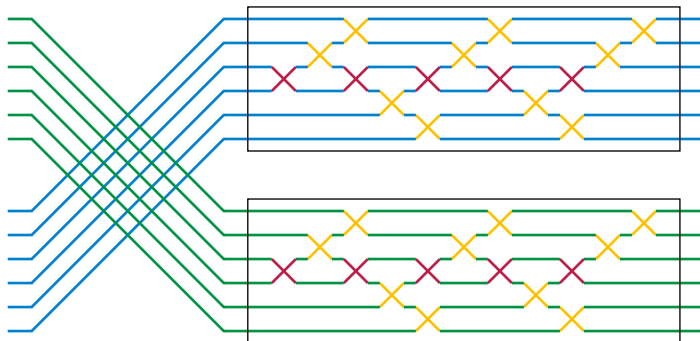


Fig. 8. Two stacked arrangements of $\ell/2$ pseudolines, each with high level complexity, cause our algorithm to create wide drawings no matter how it chooses a wiring diagram

theta-graph together with $P \cup S$. Additionally, because P is assumed to be part of a drawing of a subset of the faces of G , it cannot be crossed by ℓ , for any crossing would cause it to have an attachment vertex between u and v . Therefore, the two degree-three vertices of the theta-graph both belong to S . By the same reasoning as in the proof of Lemma 4, L must be shorter than the other two paths of the theta-graph, so replacing the path that is entirely within S by L would produce a shorter path from u to v , contradicting the construction of S as a shortest path. This contradiction shows that $P \cup S$ must be a face. \square

Theorem 3. *When the greedy ear decomposition embedding algorithm described above is applied to an arrangement graph G , it correctly constructs the canonical embedding of G .*

Proof. We prove by induction on the number of steps of the algorithm that after each step the partial embedding consists of faces of the canonical embedding whose union is a disk. Lemma 4 shows as a base case that the induction hypothesis is true after the first step. In each subsequent step, the ability to find two attachment vertices follows from the fact that arrangement graphs are 2-vertex-connected, which in turn follows from the fact that they can be augmented by a single vertex to be 3-vertex-connected [2]. Lemma 5 shows that, if the induction hypothesis is true after i steps then it remains true after $i + 1$ steps. \square

6 Conclusions

We have found a grid drawing algorithm for pseudoline arrangement graphs that uses area within a small factor of linear, much smaller than the known quadratic grid area lower bounds for arbitrary planar graphs. We have also shown that these graphs have near-linear universal point sets within a constant factor of the same area, and that a simple greedy embedding heuristic suitable for hand solution of Planarity puzzles is guaranteed to find a correct embedding.

The precise area used by our grid drawing algorithm depends on the worst-case behavior of the function $\kappa(\mathcal{D})$ counting the number of crossings in a k -level of an arrangement; closing the gap between the upper and lower bounds for this function remains an important and difficult open problem in combinatorial geometry. However, closing this gap is not the only possible method for improving our drawing algorithm.

A tempting avenue for improvement is to observe that a single pseudoline arrangement may be represented by many different wiring diagrams; therefore, we can select the wiring diagram \mathcal{D} that represent the same pseudoline arrangement and that minimizes $\kappa(\mathcal{D})$. However, this would not improve our worst case width by more than a constant factor. For, if the input forms a pseudoline arrangement constructed by stacking two arrangements of $\ell/2$ lines with maximal k -level complexity, one above the other (Figure 8), then one of these two instances will survive intact in any wiring diagram for the arrangement, forcing our algorithm to produce a drawing with width at least $\kappa_{\max}(\ell/2)$. Further improvements in our algorithm will likely come by finding an alternative layout that avoids the complexity of k -levels, by proving that k -levels are small in the average case if not the worst case, or by reducing the known combinatorial bounds on k -levels.

It is also tempting to consider other drawing styles for arrangement graphs, such as orthogonal drawing (in which each edge is an axis-aligned polyline). Because arrangement graphs contain triangles, such a drawing may be forced to have at least one bend per edge. However, the need either to align neighboring vertices on adjacent rows of the drawing, or to provide space between rows for parallel edge tracks, may cause these drawings to be significantly larger than the straight-line drawings we study, so we have not found an area bound for orthogonal drawing that is as tight as our bound for straight-line drawing.

References

- [1] Shor, P.W.: Stretchability of pseudolines is NP-hard. In: Gritzmann, P., Sturmfels, B. (eds.) *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 4, pp. 531–554. Amer. Math. Soc., Providence (1991)
- [2] Bose, P., Everett, H., Wismath, S.: Properties of arrangement graphs. *International Journal of Computational Geometry & Applications* 13(6), 447–462 (2003)
- [3] Schaefer, M.: Complexity of some geometric and topological problems. In: Eppstein, D., Gansner, E.R. (eds.) *GD 2009*. LNCS, vol. 5849, pp. 334–344. Springer, Heidelberg (2010)
- [4] Dolev, D., Leighton, T., Trickey, H.: Planar embedding of planar graphs. *Advances in Computing Research* 2, 147–161 (1984)
- [5] Valiant, L.G.: Universality considerations in VLSI circuits. *IEEE Transactions on Computers* C-30(2), 135–140 (1981)
- [6] Klawe, M., Paterson, M., Pippenger, N.: Inversions with $n^{1+\Omega(1/\sqrt{\log n})}$ transpositions at the median (September 21, 1982) (unpublished manuscript)
- [7] Tamaki, H., Tokuyama, T.: A characterization of planar graphs by pseudo-line arrangements. *Algorithmica* 35(3), 269–285 (2003)

- [8] Sharir, M., Smorodinsky, S.: Extremal configurations and levels in pseudoline arrangements. In: Dehne, F., Sack, J.-R., Smid, M. (eds.) WADS 2003. LNCS, vol. 2748, pp. 127–139. Springer, Heidelberg (2003)
- [9] Dey, T.L.: Improved bounds for planar k -sets and related problems. *Discrete and Computational Geometry* 19(3), 373–382 (1998)
- [10] Tóth, G.: Point sets with many k -sets. *Discrete and Computational Geometry* 26(2), 187–194 (2001)
- [11] Goodman, J.E.: Pseudoline arrangements. In: Goodman, J.E., O’Rourke, J. (eds.) *Handbook of Discrete and Computational Geometry*, pp. 83–109. CRC Press (1997)
- [12] Eppstein, D.: Algorithms for drawing media. In: Pach, J. (ed.) GD 2004. LNCS, vol. 3383, pp. 173–183. Springer, Heidelberg (2005)
- [13] Goodman, J.E.: Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Mathematics* 32(1), 27–35 (1980)
- [14] Muthukrishnan, S., Sahinalp, S.C., Paterson, M.S.: Grid layout of switching and sorting networks. US Patent 6185220 (2001)
- [15] Bentley, J.L., Ottmann, T.A.: Algorithms for reporting and counting geometric intersections. *IEEE Transactions on Computers* C-28(9), 643–647 (1979)
- [16] Edelsbrunner, H., Guibas, L.J.: Topologically sweeping an arrangement. *Journal of Computer and System Sciences* 38(1), 165–194 (1989); Corrigendum, *JCSS* 42(2), 249–251 (1991)
- [17] Snoeyink, J., Hershberger, J.: Sweeping arrangements of curves. In: Proc. 5th ACM Symp. on Computational Geometry (SCG 1989), pp. 354–363 (1989)
- [18] Chrobak, M., Karloff, H.: A lower bound on the size of universal sets for planar graphs. *SIGACT News* 20, 83–86 (1989)
- [19] de Fraysseix, H., Pach, J., Pollack, R.: Small sets supporting Fáry embeddings of planar graphs. In: Proc. 20th ACM Symp. Theory of Computing (STOC 1988), pp. 426–433 (1988)
- [20] Schnyder, W.: Embedding planar graphs on the grid. In: Proc. 1st ACM/SIAM Symp. Discrete Algorithms (SODA 1990), pp. 138–148 (1990)
- [21] Bannister, M.J., Cheng, Z., Devanny, W.E., Eppstein, D.: Superpatterns and universal point sets. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 208–219. Springer, Heidelberg (2013)
- [22] Gritzmann, P., Mohar, B., Pach, J., Pollack, R.: Embedding a planar triangulation with vertices at specified positions. *Amer. Math. Monthly* 98(2), 165–166 (1991)
- [23] Angelini, P., Di Battista, G., Kaufmann, M., Mchedlidze, T., Roselli, V., Squarcella, C.: Small point sets for simply-nested planar graphs. In: van Kreveld, M., Speckmann, B. (eds.) GD 2011. LNCS, vol. 7034, pp. 75–85. Springer, Heidelberg (2011)
- [24] Fulek, R., Tóth, C.D.: Universal point sets for planar three-trees. In: Dehne, F., Solis-Oba, R., Sack, J.-R. (eds.) WADS 2013. LNCS, vol. 8037, pp. 341–352. Springer, Heidelberg (2013)
- [25] Khuller, S.: Ear decompositions. *SIGACT News* 20(1), 128 (1989)