Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 18, no. 2, pp. 211–231 (2014)
DOI: 10.7155/jgaa.00319

Drawing Arrangement Graphs In Small Grids, Or How To Play Planarity

David Eppstein

Department of Computer Science, University of California, Irvine

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.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
December 2013	February 2014	February 2014	March 2014	April 2014
		Published: May 2014		

Article type:	Communicated by:
Regular paper	S. Wismath and A. Wolff

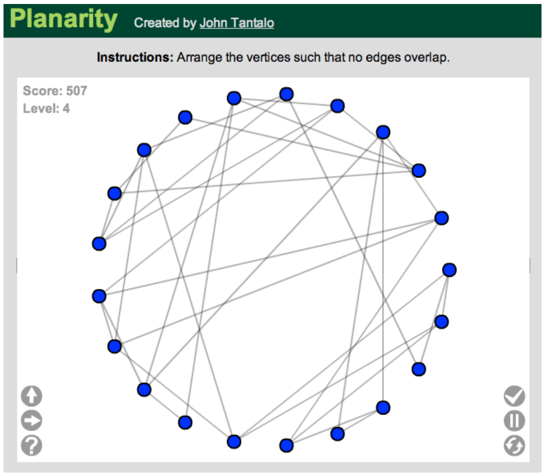
This material was presented in a preliminary form at Graph Drawing 2013 and was supported in part by NSF grants 0830403 and 1217322 and by the Office of Naval Research under grant N00014-08-1-1015.

E-mail address: eppstein@ics.uci.edu (David Eppstein)

1 Introduction

Planarity (`http://planarity.net/`) is a puzzle developed by John Tantalo 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 ith 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.

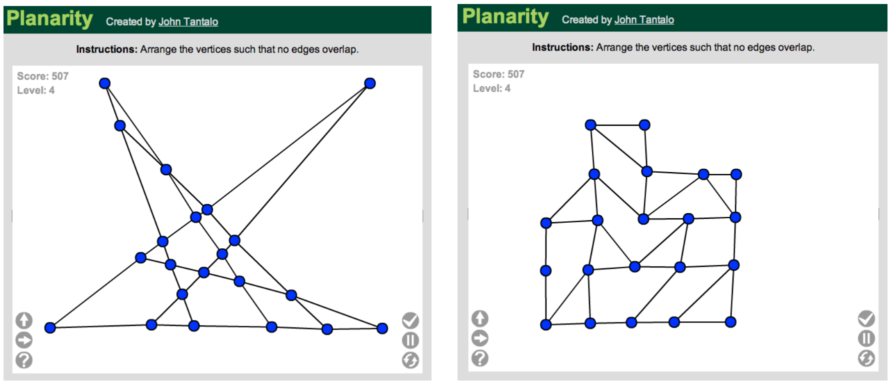
One strategy for solving Planarity would be to search for a line arrangement whose graph matches the input, and to place the vertices on the crossing points of this arrangement (Figure 2, left). From the graph visualization point of view, this method would have the advantage of accurately conveying the underlying construction of the graph. However, placing vertices in this way is tedious to do by hand, and finding the appropriate arrangement has high computational complexity: testing whether an arrangement of curves is combinatorially equivalent to a line arrangement is NP-hard [34], from which it follows that recognizing line arrangement graphs, and finding arrangements that match a given input graph, are both also NP-hard [6]. More precisely, these problems are complete for the existential theory of the reals [31]. An additional problem with drawings constructed in this way is that they necessarily have low angular resolution and high area. *Angular resolution* is a standard quality metric for straight-line graph drawings, equal to the sharpest angle formed by any two edges that meet at a vertex [15,18,27]. The pigeonhole principle shows that, in every arrangement of ℓ lines, some two lines form an angle of π/ℓ or less, and the same angle is formed by two of the edges at the crossing vertex. In addition, there exist arrangements



Instructions: Arrange the vertices such that no edges overlap.

Score: 507
Level: 4

Figure 1: Initial state of Planarity.



Instructions: Arrange the vertices such that no edges overlap.

Instructions: Arrange the vertices such that no edges overlap.

Figure 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.

of lines in which some two crossings must be extremely close together (doubly exponentially close relative to the diameter of the set of crossings) [26], forcing any drawing of this arrangement with unit spacing between its vertices to have double exponential area. Thus, drawing an arrangement graph in a way that makes its arrangement structure visible is difficult and results in a drawing that is hard to read.

Instead, in practice these puzzles may be solved more easily by an incremental strategy 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. (We provide a more formal description of this strategy in Section 5.) 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 other planar graphs such as the *nested triangles graph* that cannot be drawn planarly in a grid of less than $\Theta(n^2)$ area [11, 38].

1.1 New results

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 in Section 2 a larger family of planar graphs (the graphs of simple pseudoline arrangements) that may be recognized and decomposed into their constituent paths in linear time. In Section 3, we show that every n-vertex simple pseudoline arrangement graph may be drawn in linear time in a grid of size $\kappa_{\max}(O(\sqrt{n}))\times O(\sqrt{n})$. In this formula, $\kappa_{\max}(\ell)$ is the maximum complexity of a k-level of a pseudoline arrangement with ℓ pseudolines [25,33,36], a topological variant of the famous k-set problem from discrete geometry (see Definition 6 for a formal definition of $\kappa_{\max}$). The best proven upper bounds of $O(\ell^{4/3})$ on the complexity of k-levels [9, 33, 36] 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 $\Omega(\ell^{1+o(1)})$ [25, 37], suggesting that our algorithm is likely to perform even better in practice than this $O(n^{7/6})$ area bound would suggest. 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 in Section 4 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, in Section 5, 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.

1.2 Related work

Past work on visualizing arrangements has typically focused on the lines or curves of the arrangement, somewhat different from our emphasis on drawing the vertices and edges of arrangement graphs without respect to these curves. A standard tool in the visualization of arrangements (that we also use) is the *wiring diagram* (Figure 5) [20], which replaces the lines of an arrangement with curves that lie on parallel horizontal lines except at their crossings. Eppstein et al. [14] considered the visualization of *weak pseudoline arrangements* using smooth piecewise-circular curves. They showed that arrangements in which all crossings belong to an infinite face can always be drawn with one circular arc per pseudoline, but that in general an arrangement may require a linear number of arcs per pseudoline. Dobkin and Tal [10] study another visualization problem on line arrangements for which the geometry of the lines is already known. They describe a method for approximating any such arrangement by a set of fewer lines that is visually similar to the original arrangement.

Several groups of researchers, motivated in part by the Planarity puzzle, have studied the problem of maximizing the number of points that can be left in their original positions in a solution to the puzzle [5, 8, 19, 30]. Another problem related to Planarity is the choice of an initial placement of the vertices of a graph that maximizes its number of crossings. As Verbitsky [39] shows, the method used in Planarity of randomly permuting the vertices in a circular layout creates a drawing whose number of crossings is within a constant factor of the largest possible number, in expectation.

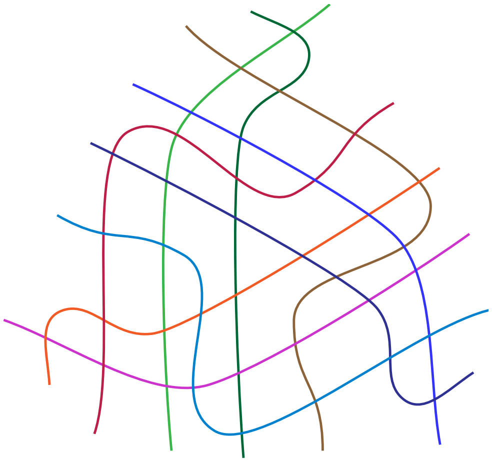


Figure 3: A simple pseudoline arrangement that cannot be transformed into a line arrangement. Redrawn from Figure 5.3.2 of [21], who attribute this arrangement to Ringel.

2 Pseudoline arrangements and their graphs

Definition 1 *A* pseudoline *is the image of a line under a homeomorphism of the Euclidean plane. 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 each of its crossing points is the crossing of only two pseudolines. A* pseudoline arrangement graph *is a graph whose vertices represent the crossings in a simple pseudoline arrangement, and whose edges connect pairs of crossings that are consecutive on the same pseudoline.*

Our definition of pseudolines follows Shor [34], and is somewhat less restrictive than a commonly used alternative definition, that a pseudoline is a non-contractible simple closed curve in the projective plane. Pseudolines as defined by Definition 1 include lines, non-self-crossing polygonal chains starting and ending in infinite rays, and the graphs of continuous real functions. Pseudoline arrangement graphs are necessarily planar, with a planar embedding coming from the arrangement. Every arrangement of lines is a pseudoline arrangement, but there exist *unstretchable* pseudoline arrangements (and more strongly unstretchable simple pseudoline arrangements) that are not combinatorially equivalent to line arrangements. One such example is depicted in Figure 3.

The advantage, for us, of using pseudolines in place of lines is that their arrangement graphs may be recognized more efficiently, as we elaborate below. Most of the ideas in the following result are from Bose et al. [6], but we expand on

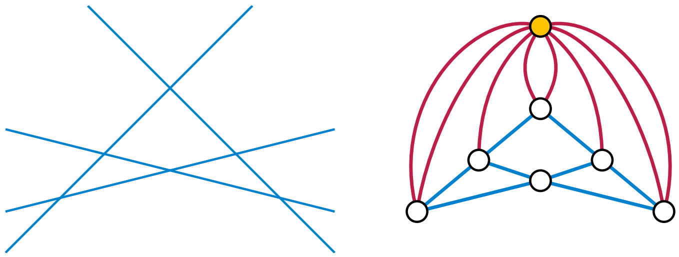


Figure 4: The multigraph G^* formed from an arrangement graph G (white vertices and blue edges) by adding a new vertex v_∞ (yellow) and, for each arrangement vertex u with degree $d < 4$, adding $4 - d$ edges (red) from u to v_∞.

the methods from that paper to show that linear time recognition of arrangement graphs is possible. Relatedly, in [13] we described 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 not every pair of pseudolines is required to cross and in which crossings may involve more than two pseudolines. However, in this work it is the primal graphs and not their duals that we need to recognize.

Lemma 2 *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: First, let G be a pseudoline arrangement graph G, and form a graph G^* from it by adding a new vertex v_∞ adjacent to all vertices in G of degree less than four. G^* is planar (it can be embedded by adding one vertex to the outside face of the embedding of G) and as Bose et al. [6] show, G^* is also 3-connected. Because G^* is a 3-connected and planar graph, it has only one planar embedding (up to the choice of the outer face), which must be the embedding derived from its representation as an arrangement graph.

For convenience we make G^* be a multigraph by including two edges in G^* between v_∞ and each degree two vertex in G, as shown in Figure 4. Duplicating edges in this way cannot decrease the connectivity or change the planarity of G^* but it ensures that all vertices except v_∞ have degree exactly four. In the embedding of G as a pseudoline arrangement, each pseudoline passes directly across each crossing vertex, connecting two opposite edges. Correspondingly, in G^* each pseudoline can be represented in a purely combinatorial way, as a path that starts at v_∞, continues through two opposite edges at each vertex other than v_∞ (using the unique planar embedding to determine which two edges are opposite), and ends again at v_∞. Any two distinct pseudolines are represented in this way by paths that have disjoint sets of edges.

Now, let G be an arbitrary given graph G of maximum degree four, not known to be a pseudoline arrangement graph. Then as above we may, in linear time, add a new vertex v_∞ to create an augmented graph G^* in which all vertices except v_∞ have degree four, test planarity of G^*, and embed G^* in the plane. If G^* is planar, any of its embeddings has a unique decomposition into an arrangement of simply-crossing curves, generalizing the way that we decomposed the graph coming from a pseudoline arrangement: join two edges of G^* into a path or cycle whenever they are opposite at a vertex other than v_∞, and draw a curve in the plane that follows the embedding of each of these paths or cycles. It is straightforward to find this decomposition algorithmically: construct an auxiliary graph that has a vertex for each edge of G^*, and an edge between two vertices whenever they correspond to two opposite edges at some vertex; then the sets of edges in the paths and cycles of the decomposition are given by the connected components of the auxiliary graph. Thus, decomposing G^* into paths and cycles takes time linear in its number of vertices and edges.

If G is indeed a pseudoline arrangement graph then this decomposition will consist only of non-self-crossing paths (not cycles), and any two paths must cross each other exactly once. We now describe how to check these properties. Once we have decomposed G^* into paths, we label each edge with the identity of its path. By comparing the set of labels used in this labeling to the set of labels appearing at v_∞, we may verify that the decomposition contains no cycles. By comparing the four labels present at each vertex other than v_∞, we may verify that no path crosses itself. These two verification steps take linear time. We additionally check that G has $\ell(\ell-1)/2$ vertices, where ℓ is the number of paths. Finally, we must verify that no two paths cross more than once. To do so, we make a list of the pairs of paths crossing at each vertex. The number of pairs is no more than the number of vertices, so we may sort this list in linear time using bucket sorting, and then check that no bucket contains a repeated pair.

If G passes all of these checks, its decomposition into paths gives a valid pseudoline arrangement. One way to show that these paths can be represented geometrically as a pseudoline arrangement is to view the embedding of G^* as being on a sphere. Puncture the sphere at the point v_∞, resulting in a space topologically equivalent to the plane, and homeomorphically map the punctured sphere to the plane. The images of the paths under this map necessarily form a pseudoline arrangement with G as its graph. (In the next section, we instead use wiring diagrams to construct more explicit geometric representations of these arrangements.) □

3 Small Grids

To describe our grid drawing algorithm for pseudoline arrangement graphs, we need to introduce the concept of a *wiring diagram.*

Definition 3 *A wiring diagram is an arrangement of ℓ polygonal pseudolines formed from the ℓ horizontal lines with coordinates $y = 1$, $y = 2$, …, $y = \ell$ by removing $\binom{\ell}{2}$ pairs of short line segments from the same horizontal positions in pairs of lines with adjacent coordinates, and replacing each removed pair of line segments by two crossing line segments.*

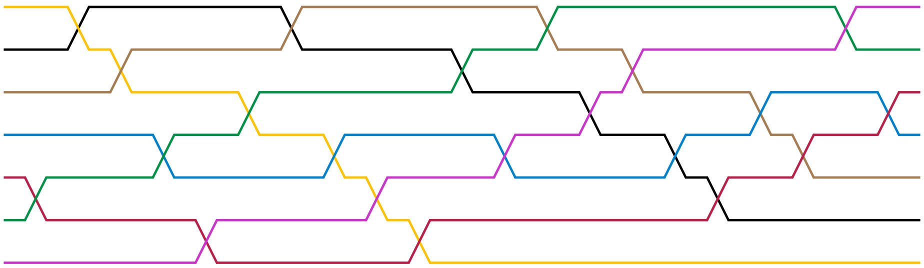


Figure 5: A wiring diagram formed by a plane sweep of the arrangement from Figure 2

Figure 5 depicts an example, a wiring diagram with the same combinatorial structure as the line arrangement whose graph was given as a Planarity puzzle in Figure 1. We will call the horizontal lines from which the wiring diagram is formed *tracks*; each crossing causes the two pseudolines that cross to swap which track they lie on. It may be convenient to require different crossings to have different x coordinates, as depicted in Figure 5. This requirement was part of the original definition of wiring diagrams by Goodman [20], but some later sources allow crossings with equal x-coordinates, a relaxation that leads to narrower diagrams.

Wiring diagrams already provide reasonably nice grid drawings of arrangement graphs [29], but are unsuitable for our purposes, for two reasons. First, they draw the edges connecting pairs of adjacent crossings as polygonal chains with two bends, while the Planarity puzzle requires that edges be drawn as straight line segments. And second, even when we allow the more compact form of wiring diagrams in which crossings may share x-coordinates, some arrangements have wiring diagrams that, when drawn in an integer grid, require width $\Omega(\ell^2)$, much larger than our bounds. Figure 6 depicts an example of this phenomenon, for an arrangement derived from the cocktail shaker sorting algorithm. In this example, there is an x-monotone path in the arrangement that passes through all of the crossings, so they must all be drawn with distinct x coordinates. Since there are $\binom{\ell}{2} = \Omega(\ell^2)$ crossings, the width of any wiring diagram with integer coordinates for this arrangement must be $\Omega(\ell^2)$.

Although wiring diagrams do not directly solve our problem, we will use these diagrams as a tool for constructing a different and more compact form of straight-line drawing. Thus, it is important for us to be able to construct them efficiently. For an arrangement of non-vertical lines in general position, an equivalent wiring diagram may be constructed by a *plane sweep* algorithm [3], which simulates the left-to-right motion of a vertical line across the arrangement. At most 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, and their positions in this order give the y-coordinate of the horizontal line that corresponds to each arrangement line. When the sweep line crosses a vertex of

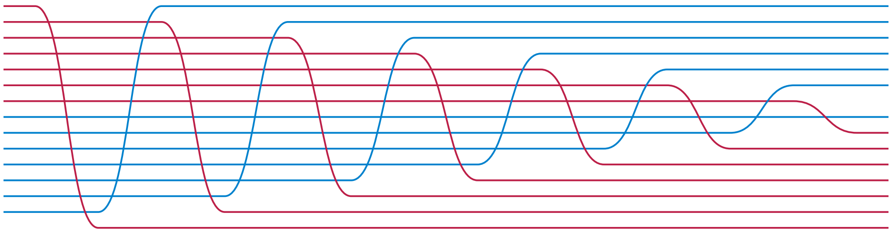


Figure 6: Cocktail shaker sort corresponds to an arrangement of ℓ pseudolines for which drawing the wiring diagram with integer coordinates requires width $\Omega(\ell^2)$

the arrangement, two intersection points swap positions in the top-to-bottom order on the sweep line, and this swap may be represented by introducing a crossing between the corresponding tracks of the wiring diagram. The left-to-right order of crossings in the wiring diagram that results from this sweeping process is thus exactly the sorted order of the crossing points of the original line arrangement, as sorted by their x coordinates. The wiring diagram in Figure 5 was constructed by this plane sweep method from the approximate line arrangement depicted in Figure 2.

Every simple pseudoline arrangement also has an equivalent wiring diagram, that may be constructed in time linear in its number of crossings. One proof of this fact uses *topological sweeping*. Topological sweeping is a variant of plane sweeping, an algorithm for listing all crossing points of an arrangement in sorted order by their x-coordinates. In topological sweeping, this algorithm is sped up by instead listing the points in a topological ordering of the directed acyclic graph formed by orienting each edge of the arrangement graph from left to right [12]. The same method has also been extended to apply to pseudoline arrangements [35], requiring only the availability of a subroutine that determines the relative ordering of two crossings that both belong to the same input pseudoline.

Lemma 4 *A wiring diagram can be constructed from a pseudoline arrangement graph in time linear in the size of the graph.*

Proof: Use the recognition algorithm described in Lemma 2 to partition the input graph into paths that correspond to the pseudolines of an arrangement. Preprocess each path by storing, for each of its vertices, the position of that vertex in the sequence of crossings of the path (storing two position numbers for each vertex, one for each of the two paths it belongs to). Then apply the topological sweeping algorithm for pseudoline arrangements [35] to determine the order in which to place the crossings of a wiring diagram. To implement the subroutine that compares two crossings on the same line, simply compare the precomputed numbers for the two given crossings. □

Definition 5 *We define the ith* 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.

Definition 6 *Define the size* $|\mathcal{D}|$ *of a diagram to be its number of pseudolines, and the level complexity* $\kappa(\mathcal{D})$ *to be* $\max_i |L_{\mathcal{D}}(i)|$*. Let* $\kappa_{\max}(\ell) = \max\{\kappa(\mathcal{D}) : |\mathcal{D}| = \ell\}$ *denote the maximum level complexity of an arrangement of* ℓ *pseudolines,*

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. (Often this problem is stated in terms of the middle level of an arrangement, rather than as here in terms of 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 $\kappa_{\max}(\ell) = O(\ell^{4/3})$ [9, 33, 36], and $\kappa_{\max}(\ell) = \Omega(\ell\, c^{\sqrt{\log \ell}})$ for some constant $c > 1$ [25, 37], where the last bound is $O(\ell^{1+\epsilon})$ 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 2, 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. □

Figure 7 depicts the output of our algorithm, using the wiring diagram of Figure 5, 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

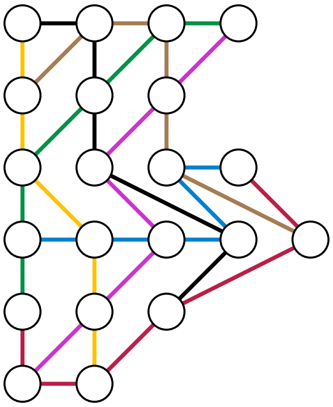


Figure 7: Output of the drawing algorithm of Theorem 1, based on the wiring diagram of Figure 5

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 clarify the construction.

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 [7]. Grids of $O(n) \times O(n)$ points form universal sets of quadratic size for the planar graphs [16, 32], and despite recent improvements the best upper bound known remains quadratic [2]. A rectangular grid that is universal must have $\Omega(n^2)$ points [11, 38]; the best known lower bounds for universal point sets that are not required to be grids are only linear [7].

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 [1, 2], planar 3-trees [17], and graphs of bounded pathwidth [2]. However, the arrangement graphs considered here are not outerplanar (see [14] for alternative methods for drawing weak pseudoline arrangements when their arrangement graphs are outerplanar) and have high treewidth and pathwidth, so these results do not apply to them. Arrangement graphs may be augmented to simply nested graphs by connecting each level of the arrangement into a cycle, but drawing these graphs using methods for simply nested graphs results in an unnecessarily high area. 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.

Definition 7 *Following Bannister et al. [2], define a sequence of positive integers ξ_i for $i = 1, 2, 3, \ldots$ by the equation $\xi_i = i \oplus (i-1)$ where $\oplus$ denotes the bitwise binary exclusive or operation.*

The sequence of these values begins

$$1, 3, 1, 7, 1, 3, 1, 15, 1, 3, 1, 7, 1, 3, 1, \ldots.$$

Lemma 8 (Bannister et al. [2]) *Let the finite sequence $\alpha_1, \alpha_2, \ldots, \alpha_k$ have sum s. Then there is a subsequence $\beta_1, \beta_2, \ldots, \beta_k$ of the first s terms of ξ such that, for all i, $\alpha_i \le \beta_i$. 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 ith row of the drawing, and define a sequence $\alpha_i = \lceil n_i/\ell \rceil$.

Lemma 9 $\displaystyle\sum_{i=1}^{\ell-1} \alpha_i \le 3(\ell-1)/2.$

Proof: We may partition the n_i vertices in the ith 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. Recall that $n = \binom{\ell}{2} = \ell(\ell-1)/2$; therefore, 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 ith 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 8, there is a subsequence β_i of the first s terms of sequence ξ, such that the β is termwise greater than or equal to α. This subsequence corresponds to a subsequence $(r_1, r_2, \ldots r_{\ell-1})$ of the rows of our universal point set, such that row r_i has at least $\min(\ell\beta_i, \kappa_{\max}(\ell)) \ge n_i$ points in it. Mapping the ith 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 $O(\ell^2 \log \ell) = O(n \log n)$. □

As with our other results, Theorem 2 applies both to the graphs of line arrangements and pseudoline arrangements.

5 Greedy embedding algorithm

The algorithm of Lemma 2 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* [24], and the subgraphs of the decomposition are known as *ears*. A given graph has an open ear decomposition if and only if it is 2-vertex-connected [40], and in this case an open ear decomposition may be constructed greedily, at each step arbitrarily choosing an ear to extend the previous partial decomposition. Because the ear decomposition we use is determined by adding as short an ear as possible at each step, we call it a *greedy ear decomposition.*

Since all arrangement graphs are 2-vertex-connected, they will automatically have an ear decomposition. Therefore, one possible failure mode of the algorithm can be ruled out: it will always be possible for the algorithm to find another ear, until it has decomposed the whole graph. However, although arbitrary 2-vertex-connected planar graphs also have ear decompositions, this greedy ear embedding algorithm does not always succeed for all such planar graphs. Even the initial cycle that is found by the algorithm might not be a face of any embedding of the given graph. In this case, the algorithm's incorrect assumption that this cycle is a face will cause it to be unable to find a valid embedding. However, as we will show (modulo 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, we make the following definition.

Definition 10 *The* canonical embedding *of an arrangement graph is the one given by the arrangement from which it was constructed.*

By Lemma 2, 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.

Lemma 11 *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 8). 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.[1] 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. □

[1] The long-standard graph-theoretic nomenclature of theta graphs [4, 28] should not be confused with a newer and unrelated meaning concerning geometric graphs defined by near-neighbors within wedges of fixed angles [23].

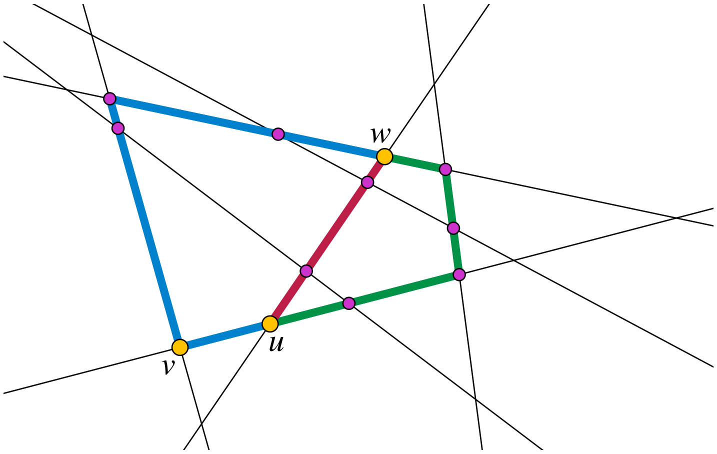


Figure 8: Illustration for the proof of Lemma 11. Every non-facial cycle C through vertex v (such as the one shown by the thick outer blue and green quadrilateral) is crossed by at least one line $\ell = uw$ (the thick inner red line segment), forming a theta-graph. All the vertices on the middle path 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 outer two paths have additional vertices at their bends. Therefore, the outer cycle is longer than either of the two cycles through the inner segment.

Lemma 12 *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 11, this cycle must be crossed by a line ℓ, a path L of which forms a 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 11, 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. □

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 11 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 [6]. Lemma 12 shows that, if the induction hypothesis is true after i steps then it remains true after $i+1$ steps. □

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, suppose that 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 9). A wiring diagram for this arrangement is determined (up to the left-right ordering of independent crossings) by the choice of which one of its 2ℓ unbounded faces is to be the top face. In the figure, the two lines that go to infinity in the top face belong to different copies of the two smaller arrangements, and both of these arrangements are drawn disjointly in the figure, each with high width. If, instead, we chose a top face in which the two lines going to infinity belong to a single copy of the smaller arrangement, then that copy would be drawn differently, but the other copy would be drawn unchanged, again with high width. Thus, no matter which top face is chosen, our algorithm would produce a drawing with width at least $\kappa_{\max}(\ell/2)$. Instead, 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.

An open problem raised by this research concerns the edge length of grid

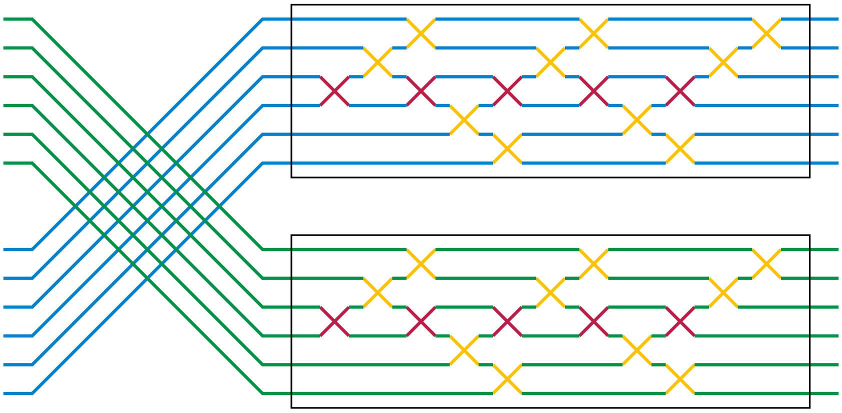


Figure 9: Two stacked arrangements of $\ell/2$ pseudolines, each with high level complexity, cause our algorithm to create wide drawings no matter how it chooses a wiring diagram.

drawings of arrangement graphs. If an arrangement graph G is drawn in a grid in such a way as to minimize the maximum edge length, what edge length is needed (as a function of n, for a worst-case arrangement)? And how can a drawing that approximately minimizes this length be found efficiently?

It is also tempting to consider other drawing styles for arrangement graphs, such as orthogonal drawings in which each edge is represented by an axis-aligned polyline. Because arrangement graphs contain triangles, some edges in an orthogonal drawing may be forced to bend. Additionally, in a layout analogous to ours in which the y-coordinate of each vertex is taken from a wiring diagram, 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. Because of these difficulties, we have not found an area bound for orthogonal drawing that is as tight as our bound for straight-line drawing.

References

[1] P. Angelini, G. Di Battista, M. Kaufmann, T. Mchedlidze, V. Roselli, and C. Squarcella. Small point sets for simply-nested planar graphs. *Proc. 19th Int. Symp. Graph Drawing (GD 2011)*, pp. 75–85. Springer, LNCS 7034, 2012, doi:10.1007/978-3-642-25878-7_8.

[2] M. J. Bannister, Z. Cheng, W. E. Devanny, and D. Eppstein. Superpatterns and universal point sets. *Proc. 21st Int. Symp. Graph Drawing (GD 2013)*, pp. 208–219. Springer, LNCS 8242, 2013, doi:10.1007/978-3-319-03841-4_19, arXiv:1308.0403.

[3] J. L. Bentley and T. A. Ottmann. Algorithms for reporting and counting geometric intersections. *IEEE Trans. Comput.* C-28(9):643–647, 1979, doi:10.1109/TC.1979.1675432.

[4] J. A. Bondy. The "graph theory" of the Greek alphabet. *Graph theory and applications (Proc. Conf., Western Michigan Univ., Kalamazoo, Mich., 1972; dedicated to the memory of J. W. T. Youngs)*, pp. 43–54. Springer, Lecture Notes in Mathematics 303, 1972, doi:10.1007/BFb0067356.

[5] P. Bose, V. Dujmovic, F. Hurtado, S. Langerman, P. Morin, and D. R. Wood. A polynomial bound for untangling geometric planar graphs. *Discrete and Computational Geometry* 42(4):570–585, 2008, doi:10.1007/s00454-008-9125-3.

[6] P. Bose, H. Everett, and S. Wismath. Properties of arrangement graphs. *International Journal of Computational Geometry & Applications* 13(6):447–462, 2003, doi:10.1142/S0218195903001281.

[7] M. Chrobak and H. Karloff. A lower bound on the size of universal sets for planar graphs. *SIGACT News* 20:83–86, 1989, doi:10.1145/74074.74088.

[8] J. Cibulka. Untangling Polygons and Graphs. *Discrete and Computational Geometry* 43(2):402–411, 2009, doi:10.1007/s00454-009-9150-x.

[9] T. L. Dey. Improved bounds for planar k-sets and related problems. *Discrete and Computational Geometry* 19(3):373–382, 1998, doi:10.1007/PL00009354.

[10] D. P. Dobkin and A. Tal. Efficient and small representation of line arrangements with applications. *Proc. 17th ACM Symp. Computational Geometry (SCG 2001)*, pp. 293–301, 2001, doi:10.1145/378583.378707.

[11] D. Dolev, T. Leighton, and H. Trickey. Planar embedding of planar graphs. *Advances in Computing Research* 2:147–161, 1984.

[12] H. Edelsbrunner and L. J. Guibas. Topologically sweeping an arrangement. *Journal of Computer and System Sciences* 38(1):165–194, 1989, doi:10.1016/0022-0000(89)90038-X. Corrigendum, *JCSS* 42 (2): 249–251, 1991.

[13] D. Eppstein. Algorithms for drawing media. *Proc. 12th Int. Symp. Graph Drawing (GD 2004)*, pp. 173–183. Springer, LNCS 3383, 2005, doi:10.1007/978-3-540-31843-9_19, arXiv:cs.DS/0406020.

[14] D. Eppstein, M. Van Garderen, B. Speckmann, and T. Ueckerdt. Convex-arc drawings of pseudolines. *Proc. 21st Int. Symp. Graph Drawing (GD 2013)*, pp. 522–523. Springer, LNCS 8242, 2013.

[15] M. Formann, T. Hagerup, J. Haralambides, M. Kaufmann, F. T. Leighton, A. Symvonis, E. Welzl, and G. Woeginger. Drawing graphs in the plane with high resolution. *SIAM J. Comput.* 22(5):1035–1052, 1993, doi:10.1137/0222063.

[16] H. de Fraysseix, J. Pach, and R. Pollack. Small sets supporting Fáry embeddings of planar graphs. *Proc. 20th ACM Symp. Theory of Computing (STOC 1988)*, pp. 426–433, 1988, doi:10.1145/62212.62254.

[17] R. Fulek and C. Toth. Universal point sets for planar three-trees. *Proc. Algorithms and Data Structures Symposium (WADS 2013)*, pp. 341–352. Springer, LNCS 8037, 2013, doi:10.1007/978-3-642-40104-6_30, arXiv:1212.6148.

[18] A. Garg and R. Tamassia. Planar drawings and angular resolution: Algorithms and bounds. *Proc. 2nd Eur. Symp. Algorithms (ESA 1994)*, pp. 12–23. Springer, LNCS 855, 1994, doi:10.1007/BFb0049393.

[19] X. Goaoc, J. Kratochvíl, Y. Okamoto, C.-S. Shin, A. Spillner, and A. Wolff. Untangling a Planar Graph. *Discrete and Computational Geometry* 42(4):542–569, 2009, doi:10.1007/s00454-008-9130-6.

[20] J. E. Goodman. Proof of a conjecture of Burr, Grünbaum, and Sloane. *Discrete Mathematics* 32(1):27–35, 1980, doi:10.1016/0012-365X(80)90096-5.

[21] J. E. Goodman. Pseudoline arrangements. *Handbook of Discrete and Computational Geometry*, 2nd edition. CRC Press, 2004, doi:10.1201/9781420035315.ch5.

[22] P. Gritzmann, B. Mohar, J. Pach, and R. Pollack. Embedding a planar triangulation with vertices at specified positions. *Amer. Math. Monthly* 98(2):165–166, 1991, `http://www.jstor.org/stable/2323956`.

[23] J. M. Keil. Approximating the complete Euclidean graph. *Proc. 1st Scand. Worksh. Algorithms (SWAT 1988)*, pp. 208–213. Springer, LNCS 318, 1988, doi:10.1007/3-540-19487-8_23.

[24] S. Khuller. Ear decompositions. *SIGACT News* 20(1):128, 1989.

[25] M. Klawe, M. Paterson, and N. Pippenger. Inversions with $n^{1+\Omega(1/\sqrt{\log n})}$ transpositions at the median. Unpublished manuscript, September 21 1982.

[26] J. Kratochvíl and J. Matoušek. Intersection graphs of segments. *Journal of Combinatorial Theory. Series B* 62(2):289–315, 1994, doi:10.1006/jctb.1994.1071.

[27] S. Malitz and A. Papakostas. On the angular resolution of planar graphs. *SIAM J. Discrete Math.* 7(2):172–183, 1994, doi:10.1137/S0895480193242931.

[28] J. Mitchem. Hypo-properties in graphs. *The Many Facets of Graph Theory (Proc. Conf., Western Mich. Univ., Kalamazoo, Mich., 1968)*, pp. 223–230. Springer, 1969, doi:10.1007/BFb0060121.

[29] S. Muthukrishnan, S. C. Sahinalp, and M. S. Paterson. Grid layout of switching and sorting networks. US Patent 6185220, 2001.

[30] J. Pach and G. Tardos. Untangling a polygon. *Discrete and Computational Geometry* 28(4):585–592, 2002, doi:10.1007/s00454-002-2889-y.

[31] M. Schaefer. Complexity of some geometric and topological problems. *Proc. 17th Int. Symp. Graph Drawing (GD 2009)*, pp. 334–344. Springer, LNCS 5849, 2010, doi:10.1007/978-3-642-11805-0_32.

[32] W. Schnyder. Embedding planar graphs on the grid. *Proc. 1st ACM/SIAM Symp. Discrete Algorithms (SODA 1990)*, pp. 138–148, 1990.

[33] M. Sharir and S. Smorodinsky. Extremal configurations and levels in pseudoline arrangements. *Proc. 8th Int. Worksh. Algorithms and Data Structures (WADS 2003)*, pp. 127–139. Springer, LNCS 2748, 2003, doi:10.1007/978-3-540-45078-8_12.

[34] P. W. Shor. Stretchability of pseudolines is NP-hard. *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, pp. 531–554. Amer. Math. Soc., DIMACS Series in Discrete Mathematics and Theoretical Computer Science 4, 1991.

[35] J. Snoeyink and J. Hershberger. Sweeping arrangements of curves. *Proc. 5th ACM Symp. on Computational Geometry (SCG '89)*, pp. 354–363, 1989, doi:10.1145/73833.73872.

[36] H. Tamaki and T. Tokuyama. A characterization of planar graphs by pseudo-line arrangements. *Algorithmica* 35(3):269–285, 2003, doi:10.1007/s00453-002-0999-9.

[37] G. Tóth. Point sets with many k-sets. *Discrete and Computational Geometry* 26(2):187–194, 2001, doi:10.1007/s004540010022.

[38] L. G. Valiant. Universality considerations in VLSI circuits. *IEEE Trans. Comput.* C-30(2):135–140, 1981, doi:10.1109/TC.1981.6312176.

[39] O. Verbitsky. On the obfuscation complexity of planar graphs. *Theoretical Computer Science* 396(1-3):294–300, 2008, doi:10.1016/j.tcs.2008.02.032, arXiv:0705.3748.

[40] H. Whitney. Non-separable and planar graphs. *Trans. AMS* 34:339–362, 1932, doi:10.1090/S0002-9947-1932-1501641-2.