Track Layout Is Hard

Michael J. Bannister[1], William E. Devanny[2(✉)], Vida Dujmović[3], David Eppstein[2], and David R. Wood[4]

[1] Department of Mathematics and Computer Science, Santa Clara University, Santa Clara, CA, USA
mbannister@fastmail.fm

[2] Department of Computer Science, University of California, Irvine, Irvine, CA, USA
{wdevanny,eppstein}@uci.edu

[3] School of Computer Science and Electrical Engineering, University of Ottawa, Ottawa, Canada
vida.dujmovic@uottawa.ca

[4] School of Mathematical Sciences, Monash University, Melbourne, Australia
david.wood@monash.edu

Abstract. We show that testing whether a given graph has a 3-track layout is hard, by characterizing the bipartite 3-track graphs in terms of leveled planarity. Additionally, we investigate the parameterized complexity of track layouts, showing that past methods used for book layouts do not work to parameterize the problem by treewidth or almost-tree number but that the problem is (non-uniformly) fixed-parameter tractable for tree-depth. We also provide several natural classes of bipartite planar graphs, including the bipartite outerplanar graphs, squaregraphs, and dual graphs of arrangements of monotone curves, that always have 3-track layouts.

1 Introduction

A *k-track layout* of a graph is a partition of the vertices into k ordered independent sets called *tracks*, and a partition of the edges into non-crossing subsets that connect pairs of tracks. The *track-number* of a graph is the minimum k for which it has a k-track layout. Track layouts are connected with the existence of low-volume three-dimensional graph drawings: a graph has a three-dimensional drawing in an $O(1) \times O(1) \times O(n)$ grid if and only if it has track-number $O(1)$ [1,2].

Already in 2004, Dujmović et al. [3] asked whether it is computationally feasible to construct optimal track layouts. A graph has track-number 2 if and only if it is a forest of caterpillars [3]. So we can efficiently recognize and construct optimal track layouts for track-number 2 graphs. In this paper we show that the

Michael J. Bannister and David Eppstein were supported in part by NSF grant CCF-1228639. William E. Devanny was supported by an NSF Graduate Research Fellowship under grant DGE-1321846. Vida Dujmović was supported by NSERC. David Wood was supported by the Australian Research Council.

Y. Hu and M. Nöllenburg (Eds.): GD 2016, LNCS 9801, pp. 499–510, 2016.
DOI: 10.1007/978-3-319-50106-2_38

answer to the general question is negative: even recognizing the graphs with 3-track layouts is NP-complete. Our proof is based on the known NP-completeness of level planarity [4], and uses a new characterization of the bipartite graphs with 3-track layouts as being exactly the leveled planar graphs, undirected graphs that can be given a Sugiyama-style layered graph drawing with no crossings and no dummy vertices.

Additionally, we show that known methods of obtaining fixed-parameter tractable algorithms for other types of planar embedding, based on Courcelle's theorem for treewidth [5], or on kernelization of the 2-core for k-almost-trees [6], do not generalize to track number. However, for any fixed bound on the tree-depth of an input graph, the track number can be obtained in linear time.

We also provide several natural classes of bipartite planar graphs, including the bipartite outerplanar graphs, squaregraphs, and dual graphs of arrangements of monotone curves, that always have 3-track layouts.

2 Definitions

A *track layout* of a graph is a partition of its vertices into sequences, called *tracks*, such that the vertices in each sequence form an independent set and the edges between each pair of tracks form a non-crossing set. This means that there do not exist edges uv and $u'v'$ such that u is before u' in one track, but v is after v' in another track; such a pair of edges is said to form a *crossing*. (This ordering constraint on endpoints of pairs of edges connecting two tracks is the same as the constraint on the left-to-right ordering within levels on the endpoints of two edges connecting the same two levels of a layered drawing.)

The *track-number* of a graph G is the minimum number of tracks in a track layout of G; this is finite, since the layout in which each vertex forms its own track is always non-crossing. The set of edges between two tracks form a forest of caterpillars (a forest in which the non-leaf vertices of each component induce a path); in particular, the graphs with track-number 1 are the independent sets, and the graphs with track-number 2 are the forests of caterpillars [7].

A *tree-decomposition* of a graph G is given by a tree T whose nodes index a collection $(B_x \subseteq V(G) : x \in V(T))$ of sets of vertices in G called *bags*, such that:

- For every edge vw of G, some bag B_x contains both v and w, and
- For every vertex v of G, the set $\{x \in V(T) : v \in B_x\}$ induces a non-empty (connected) subtree of T.

The *width* of a tree-decomposition is $\max_x |B_x| - 1$, and the *treewidth* of a graph G is the minimum width of any tree decomposition of G. Treewidth was introduced (with a different but equivalent definition) by Halin [8] and tree decompositions were introduced by Robertson and Seymour [9].

A *layering* of a graph is a partition of the vertices into a sequence of disjoint subsets (called *layers*) such that each edge connects vertices in the same layer or consecutive layers. One way, but not the only way, to obtain a layering is the *breadth first layering* in which we partition the vertices by their distances from a fixed starting vertex, using breadth-first search [10,11].

The class of *leveled planar graphs* was introduced in 1992 by Heath and Rosenberg [4] in their study of queue layouts of graphs. A leveled planar drawing of a graph is a planar drawing in which the vertices are placed on a collection of parallel lines, and each edge must connect vertices in two consecutive parallel lines. Another equivalent way to state this is that this kind of drawing is a Sugiyama-style layered drawing [12] that achieves perfect quality according to two of the most important quality measures for the drawing, the number of edge crossings [13] and the number of dummy vertices [14].

3 Track Layouts and Leveled Planarity

We begin by demonstrating an equivalence between leveled planarity and bipartite 3-track layout.

Lemma 1 (implicit in [15]). *Every leveled planar graph has a 3-track layout.*

Proof. Assign the vertices of the graph to tracks according to the number of their level in the layered drawing, modulo 3, as shown in Fig. 1. Within each track, order the vertices within each level contiguously, and order the levels by their positions in the layered drawing. Two edges that connect the same pair of levels cannot cross because of the chosen vertex ordering within the levels, and two edges that connect different pairs of levels but are mapped to the same pair of tracks cannot cross because of the ordering of the levels within the tracks. □

Lemma 1 can be interpreted as 'wrapping' a layered drawing on to 3 tracks; see [3] for a more general wrapping lemma. As Fig. 1 shows, a 3-track layout can also be interpreted geometrically, as a planar drawing in which the tracks are represented as three rays from the origin; it follows from this interpretation that 3-track graphs have universal point sets of size $O(n)$, consisting of n points on each ray. However, for more than three tracks, a similar embedding of the tracks as rays in the plane would not lead to a planar drawing, because

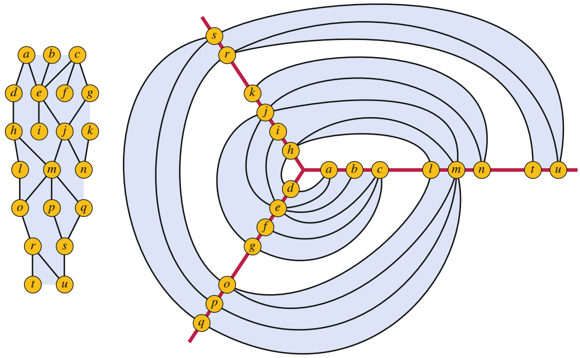

Fig. 1. Converting a layered drawing to a 3-track layout

there is no requirement that edges of the graph connect only consecutive rays. Indeed, all graphs (for example, arbitrarily large complete graphs) have 4-track subdivisions [16].

Define an *arc* of an undirected graph G to be a directed edge formed by orienting one of the edges of G. For a graph G with a 3-track layout, define a function δ from arcs to ± 1 as follows: if an arc uv goes from track i to track $i+1 \pmod 3$ (that is, if it is oriented clockwise in the planar embedding described above), let $\delta(uv) = +1$; otherwise (if it is oriented counterclockwise), let $\delta(uv) = -1$. For an oriented cycle C, we define (by abuse of notation) $\delta(C) = \sum_{uv \in C} \delta(uv)$.

Lemma 2. *Let C be a cycle embedded in a 3-track layout. Cyclically orient the edges of C. If C is even then $\delta(C) = 0$. If C is odd then $|\delta(C)| = 3$. (Here $|x|$ is the absolute value of x.)*

Proof. We proceed by induction on $|C| := |V(C)|$. If $|C| = 3$, then C has one vertex on each track and $\delta(C) \in \{3, -3\}$. If $|C| = 4$, then C has two edges with $\delta = +1$ and two edges with $\delta = -1$, implying $\delta(C) = 0$. Now assume that $|C| \geq 5$. Use the 3-track layout to embed C in the plane as described above, but with straight edges instead of the curved edges shown in the figure. As a planar polygon, C has at least two ears, triangles formed by two of its edges that are empty of other vertices of C (which may be found as the leaf edges in the tree formed as the dual graph of a triangulation of C). If one ear has the same sign of δ for both of the edges that form it, these edges must connect pairs of vertices that are the innermost on their tracks. Therefore, two such ears with same-sign edges could only exist if C is a triangle. For any longer cycle, let uvw be an ear for which $\delta(uv) = -\delta(vw)$; thus edges uv and vw both connect the same two tracks, and (by the assumption that triangle uvw is empty) u and w are consecutive in their track. By deleting v and merging uw into a single vertex, we construct a cycle C' with $|C'| = |C| - 2$, and a 3-track layout of C' with $\delta(C') = \delta(C)$. The result follows by induction. □

The previous lemma can be restated in terms of winding number. The *winding number* of a closed curve C in the plane around a given point x is the number of times that C travels counterclockwise around x. Lemma 2 then says that for an oriented cycle C around the origin in a 3-track representation of C with three rays (as in Fig. 1), if C is even then the winding number is 0, and if C is odd then the winding number is 1.

While Lemma 1 shows that a leveled planar drawing can be wrapped on to three tracks, we now use Lemma 2 to show that a bipartite 3-track layout can be unwrapped to produce a leveled planar drawing.

Theorem 1. *A graph G has a leveled planar drawing if and only if G is bipartite and has a 3-track layout.*

Proof. In one direction, if G has a leveled planar drawing, then it is bipartite (with a coloring determined by the parity of the level numbers of the drawing) and has a 3-track layout by Lemma 1.

In the other direction, suppose that G is bipartite and has a 3-track layout. We may assume without loss of generality that G is connected, for otherwise we can draw each connected component of G separately; let T be a spanning tree of G, and let v be an arbitrary vertex of G. Assign v to level zero of a layered drawing, and assign each other vertex w to the level given by the sum of the numbers $\delta(xy)$ for the edges xy of the oriented path from v to w in T. (Some of these level numbers may be negative.) By construction, the endpoints of each edge of T are assigned to consecutive levels, and by applying Lemma 2 to the oriented cycle formed by a non-tree edge together with the tree path connecting its endpoints, the same can be shown to be true of each edge of $G - E(T)$.

Within each level of the drawing, the vertices all come from the same track, determined by the value of the level modulo 3. Assign the vertices to positions in left-to-right order on this level according to their ordering within this track. Then no two consecutive levels of the drawing can have crossing edges, because such a crossing would also be a crossing in the track layout. Therefore, this assignment of vertices to levels and to positions within these levels gives a leveled planar drawing of G. □

Theorem 2. *Testing whether a given graph has a k-track layout for any constant $k \geq 3$ is NP-complete.*

Proof. For $k = 3$ this follows from Theorem 1 and from the known NP-completeness of level planarity, proven by Heath and Rosenberg [4]. For $k > 3$ this follows by adding $k - 3$ additional vertices, adjacent to all other vertices, to a hard instance of the 3-track layout problem. □

4 Parameterized Complexity

A fixed-parameter tractable problem is also *strongly uniform fixed-parameter tractable*. A problem is *uniformly fixed-parameter tractable* if there is an algorithm that solves it in polynomial time for any value of the parameter, but we cannot compute the dependence on the parameter. Lastly a problem is *non-uniformly fixed-parameter tractable* if there is a collection of algorithms such that for each possible value of the parameter one of the algorithms solves the problem in polynomial time.

4.1 Treewidth

We sketch an argument as to why it is not possible to use Courcelle's Theorem (or any automata methods based on tree decompositions) to produce a fixed-parameter tractable algorithm for leveled planarity with respect to treewidth. Consider the family of graphs depicted in Fig. 2. These graphs have bounded treewidth (in fact pathwidth at most 12) and are leveled planar precisely when $p = q$. However, since p and q are unbounded it is necessary to carry more than a finite amount of state between bags in a treewidth decomposition when parsing the decomposition. Thus, the decompositions corresponding to leveled planar graphs cannot be recognized by automata or methods using automata such as

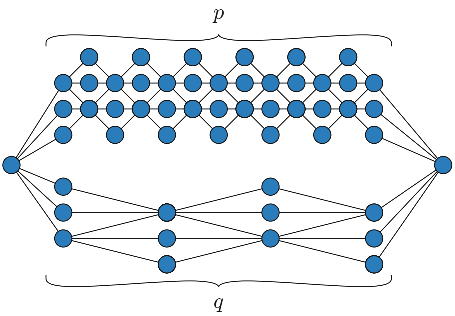

Fig. 2. A family of graphs with bounded treewidth demonstrating that the family of leveled planar graphs is not finite state.

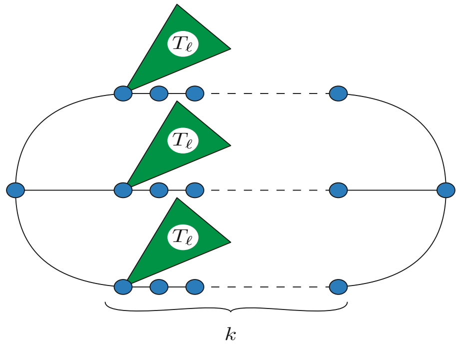

Fig. 3. A family of 2-almost trees for which the standard kernelization cannot decide leveled planarity. The subgraphs T_ℓ are complete binary trees of depth ℓ.

Courcelle's Theorem. This intuitive argument is made formal below using the Myhill-Nerode Theorem for tree automata below.

Following Downey and Fellows [17], we define a *t-boundaried graph* to be a graph G with t designated *boundary* vertices labeled $1, 2, \dots, t$. Given two t-boundaried graphs G_1 and G_2 we define their *gluing* $G_1 \oplus G_2$ by identifying each boundary vertex of G_1 with the boundary vertex of G_2 having the same label.

An n-ary *t-boundaried operator* $\otimes$ consists of a t-boundaried graph $G_\otimes = (V_\otimes, E_\otimes)$ and injections $f_i : \{1, \dots, t\} \to V_\otimes$ for $1 \le i \le n$. Then for t-boundaried graphs $G_1, \dots, G_n$ we define the t-boundaried graph $G_1 \otimes \cdots \otimes G_n$ by gluing each G_i to $G_\otimes$ after applying f_i to the boundary labels of $G_\otimes$. After the gluing the labels of G_i are forgotten.

It can be shown that there exists a *standard set* of t-boundaried operators on t-boundaried graphs that can be used to generate the set of all graphs of treewidth t. Furthermore, it is possible to convert (in linear time) a tree decomposition of width t into a parse tree that uses these standard operators; see Theorem 12.7.1 in [17]. Define $\mathcal{U}_t^{\text{small}}$ to be the *small universe* of t-boundaried graphs obtained by parse trees, using these standard operators. Given a family of graphs F, we define the equivalence relation $\sim_F$ on $\mathcal{U}_t^{\text{small}}$, such that $G_1 \sim_F G_2$ if and only if for all $H \in \mathcal{U}_t^{\text{small}}$, we have $G_1 \oplus H \in F \Leftrightarrow G_2 \oplus H \in F$.

A family of graphs F is said to be *t-finite state* if the family of parse trees for graphs in $F_t = F \cap \mathcal{U}_t^{\text{small}}$ is finite state. Equivalently, such a family of parse trees may be recognized by a finite tree automaton. We can now state the analog of the Myhill–Nerode Theorem (characterizing recognizability of sets of strings by finite state machines) for treewidth t graphs in place of strings and finite tree automata in place of finite state machines.

Lemma 3 (Theorem 12.7.2 of [17]). *Let F be a family of graphs. Then F is t-finite state if and only if $\sim_F$ has finite index over $\mathcal{U}_t^{small}$.*

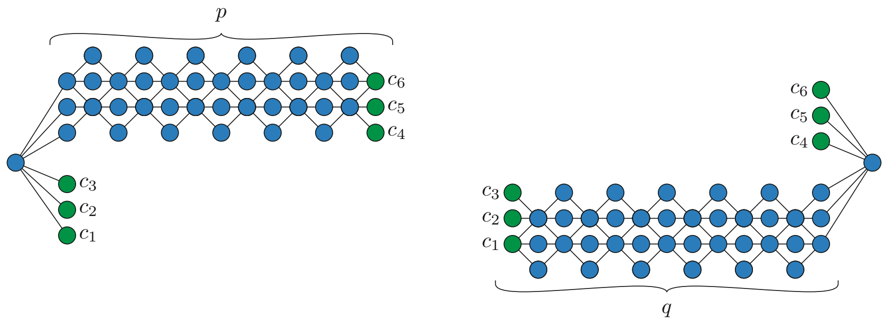

Fig. 4. The 6-boundaried graphs U_p (left) and L_q (right) from the proof of Theorem 3.

As we now show, leveled planarity is not t-finite state when t is sufficiently large.

Theorem 3. *For all $t \geq 6$, the families of leveled planar graphs and of 3-track graphs are not t-finite state.*

Proof. Let F be the family of leveled planar graphs. It suffices to prove the theorem in the case when $t = 6$. Consider the 6-boundaried graphs U_p and L_q shown in Fig. 4, and observe that $U_p \oplus L_q$ is leveled planar if and only if $p = q$. So $U_p \sim_F U_\ell$ if and only if $p = \ell$, which implies that $\sim_{F_6}$ does not have finite index and that in turn F is not 6-finite state by Lemma 3. □

Theorem 3 implies that (when $t \geq 6$) the parse trees of leveled planar graphs with treewidth t are not recognizable by tree automata. Therefore automata-based methods such as Courcelle's Theorem cannot be used to show leveled planarity to be fixed-parameter tractable with respect to treewidth. In particular, leveled planarity cannot be expressed using the forms of monadic second-order graph logic to which Courcelle's Theorem applies.

4.2 Almost-Trees

The *cyclomatic number* (also called *circuit rank*) of a graph is defined to be $r = m - n + c$ where m is the number of edges, n is the number of vertices, and c is the number of connected components in the graph. We say that a graph G is a k-*almost tree* if every biconnected component of G has cyclomatic number at most k. The problems of 1-page and 2-page crossing minimization and testing 1-planarity were shown to be fixed-parameter tractable with respect to the k-almost tree parameter, via the kernelization method [6,18].

In these previous papers, the "standard kernelization" used for a k-almost tree G is constructed by first iteratively removing degree one vertices until no more remain, leaving what is called the *2-core* of G. The 2-core consists of vertices of degree greater than two and paths of degree two vertices connecting these high degree vertices. The paths of degree two vertices are then shortened to a maximum length whose value is a function of k, with a precise form that depends on the specific problem.

However, this kernelization cannot be used to produce a fixed-parameter tractable algorithm for deciding leveled planarity. To see this, consider the graph in Fig. 3, constructed by drawing $K_{2,3}$ in the plane, and replacing each of the three vertices with paths of k vertices, and then rooting a complete binary tree of depth ℓ at one of the vertices of each of these paths. We note that, as complete binary trees have unbounded pathwidth, they also require an unbounded number of layers (depending on ℓ) in any leveled planar drawing. Additionally, depending on the planar embedding chosen for this graph, at most two of its three trees can be drawn on the outside face. So this graph is leveled planar precisely when ℓ is small enough for the remaining tree T_ℓ to be drawn within one of the two bounded faces of the drawing, i.e., the leveled planarity of the graph depends on the relationship between k and ℓ. Since this relationship is not preserved in the kernelization it can not be used to produce a fixed-parameter tractable algorithm for leveled planarity.

4.3 Tree-Depth

The *tree-depth* of a graph G is the minimum height of a forest of rooted trees on the same vertex set as G such that edges in G only go from ancestors to descendants in the forest. It is bounded by pathwidth, and therefore by track-number: track-number$(G) \leq$ pathwidth$(G) + 1 \leq$ tree-depth(G); see [1,19].

Theorem 4. *Computing the track-number of a graph G is non-uniformly fixed-parameter linear in the tree-depth of G.*

Proof. Track-number and layered pathwidth are both monotone (cannot increase) under taking induced subgraphs. The graphs with tree-depth bounded by a constant are well-quasi-ordered under taking induced subgraphs and so for any fixed bound on tree-depth and either track-number or layered pathwidth there exist only a finite number of forbidden induced subgraphs [19]. Since the track-number and pathwidth are both bounded by the tree-depth, the same is true for any fixed bound on tree-depth, regardless of track-number or layered pathwidth.

Because induced subgraph testing is linear time for graphs with tree-depth bounded by a fixed number d, we can for each $t \leq d$ test if the graph has any of the forbidden induced subgraphs to track-number t each in linear time [19]. □

However, this argument does not tell us how to find the set of forbidden induced subgraphs, nor what the dependence of the time bound on the tree-depth is. It would be of interest to replace this existence proof with a more constructive algorithm.

5 Special Classes of Graphs

We consider here particular graph families such as the outerplanar graphs, and prove that these families are leveled planar. Our results are based on

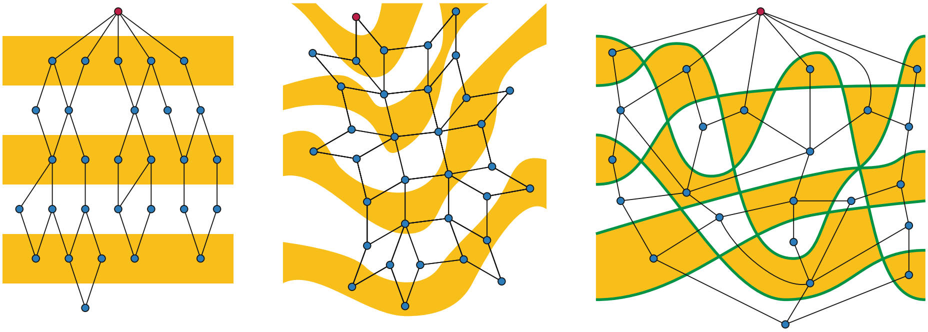

Fig. 5. Examples of graphs with planar breadth-first layerings (start vertex shown in red, and layering in yellow): left, a bipartite outerplanar graph (Theorem 5); center, a squaregraph (Theorem 6); and right, the dual graph of an arrangement of doubly-unbounded monotonic curves (Theorem 7). (Color figure online)

breadth-first layerings; we define a layering of a graph to be *planar* if there exists a non-crossing layered drawing of the graph in which the layers of the drawing are the same as the layers of the layering.

5.1 Bipartite Outerplanar Graphs

Theorem 5 (implicit in [15]). *Every bipartite outerplanar graph is leveled planar and* 3*-track. Every breadth first layering of such a graph G gives a leveled planar drawing.*

Proof. Let v be the starting vertex of a breadth first layering. Then for each face cycle C of the outerplanar embedding of G, there must be a unique nearest neighbor in C to v. For, if v were nearest to distinct vertices u and w in C, then by bipartiteness these two vertices must be non-adjacent in C. In this case, the graph formed by C together with the shortest paths from v to u and w would contain a subdivision of $K_{2,3}$ (with u and w as the degree three vertices, two paths between them in C, and one more path between them through the shortest path tree rooted at v), an impossibility for an outerplanar graph. For the same reason, the distances in v from this nearest neighbor or pair of nearest neighbors must increase monotonically in both directions around C until reaching a unique farthest neighbor, because in the same way any non-monotonicity could be used to construct a subdivision of $K_{2,3}$.

Thus, each face cycle of G has a planar breadth first layering. The result follows from the fact that in a plane graph with an assignment of levels to the vertices, there is a planar drawing consistent with this level assignment and with the given embedding of the graph, if and only if every face cycle of the given graph has a planar drawing consistent with the level assignment [20]. □

5.2 Squaregraphs

A *squaregraph* is defined to be a graph that has a planar embedding in which each bounded face is a 4-cycle and each vertex either belongs to the unbounded face or has four or more incident edges. These graphs may also be characterized in various other ways, for instance as the dual graphs of hyperbolic line arrangements with no three mutually-intersecting lines [21].

Theorem 6. *Every squaregraph G is leveled planar, and 3-track, with a leveled planar drawing coming from a breadth first layering.*

Proof. Because all their bounded faces are even-sided, squaregraphs are necessarily bipartite, so every choice of a starting vertex gives a valid breadth first layering. Bandelt et al. [21, Lemma 12.2] prove that, for every choice of a starting vertex, we can add extra edges to the squaregraph to form a plane multigraph in which the added edges link each layer into a cycle, and in which these cycles are all nested within each other.

Now, choose the starting vertex v to be a vertex of the outer face. Then each cycle added in this augmentation of G contains an edge that separates v from the unbounded face of the augmented graph. If we remove each such edge from the augmented graph, we break each cycle into a path in a consistent way, such that the path ordering within each layer matches the given planar embedding of G. □

5.3 Dual Graphs of Monotone Curves

Theorem 7. *Let A be a collection of finitely many x-monotone curves in the plane, each of whose projection onto the x-axis covers the entire axis, such that any two curves intersect at finitely many crossing points. Then the dual graph of the arrangement of the curves in A is leveled planar and 3-track.*

Proof. Each vertex of the dual graph corresponds to a connected component of the complement of $\bigcup A$; we call this the *region* of the vertex. We may assign each vertex to a layer according to the number of curves in A that pass above it; this is a breadth first layering starting from the vertex corresponding to the topmost (unbounded upward) connected component. Because a single curve separates adjacent regions, vertices in adjacent regions will be assigned to consecutive regions. No two vertices in the same layer have regions that project to overlapping subsets of the x-axis, so we may order the vertices within each layer according to the left-to-right ordering of these projections. This ordering is compatible with the planar embedding of the dual graph given by placing a representative point within each region and connecting each two adjacent regions by a curve crossing their shared boundary. □

See Fig. 5 for examples of the graphs shown to have planar layerings by these theorems. Figure 6 gives another example, demonstrating that Theorem 7 cannot be generalized to monotone curves whose projections do not cover the

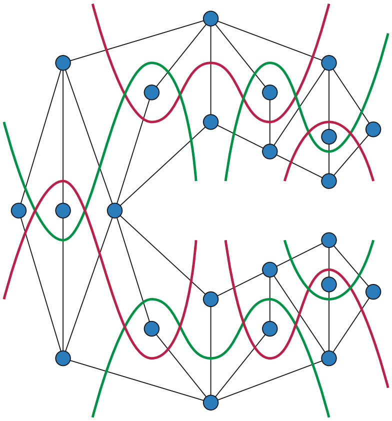

Fig. 6. An arrangement of monotone curves whose dual graph has no planar layering

entire axis: it gives a family of monotone curves, all ending within the outer face of their arrangement, such that the dual graph of the arrangement is not leveled planar. The dual graph is made of multiple $K_{2,3}$ subgraphs, each of which must have the 2-vertex side of its bipartition drawn on two layers with the 3-vertex side of its bipartition in a single layer between them; thus, up to top-bottom reflection, there is only a single layering for this graph that could possibly be planar. However, this layering forced by the planarity of the individual $K_{2,3}$ subgraphs is not planar globally, because it forces one of the two arms of the graph (upper and lower right) to collide with the "armpit" where the other arm meets the body of the graph (left). The graph is drawn without crossings in the figure, but in a way that does not respect any layering of the graph. The dual used in Fig. 6 is non-standard; there is a vertex in the outer face for each pair of consecutive curve endpoints on the outer face. The dual shown can be made as a subgraph of the more standard dual with one vertex on the outer face by adding additional curves. This example is also a series-parallel graph, and shows that Theorem 5 cannot be generalized to series-parallel, treewidth-2, or 2-outerplanar graphs: none of these classes of graphs is leveled planar.

References

1. Dujmović, V., Morin, P., Wood, D.R.: Layout of graphs with bounded tree-width. SIAM J. Comput. **34**, 553–579 (2005)
2. Dujmović, V., Wood, D.R.: Three-dimensional grid drawings with sub-quadratic volume. In: Pach, J., ed.: Towards a Theory of Geometric Graphs, vol. 342, pp. 55–66. Contemporary Mathematics - American Mathematical Society (2004)
3. Dujmović, V., Pór, A., Wood, D.R.: Track layouts of graphs. Discrete Math. Theor. Comput. Sci. **6**, 497–521 (2004)
4. Heath, L.S., Rosenberg, A.L.: Laying out graphs using queues. SIAM J. Comput. **21**, 927–958 (1992)

5. Bannister, M.J., Eppstein, D.: Crossing minimization for 1-page and 2-page drawings of graphs with bounded treewidth. In: Duncan, C., Symvonis, A. (eds.) GD 2014. LNCS, vol. 8871, pp. 210–221. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45803-7_18
6. Bannister, M.J., Eppstein, D., Simons, J.A.: Fixed parameter tractability of crossing minimization of almost-trees. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 340–351. Springer, Heidelberg (2013). doi:10.1007/978-3-319-03841-4_30
7. Harary, F., Schwenk, A.: A new crossing number for bipartite graphs. Utilitas Math. **1**, 203–209 (1972)
8. Halin, R.: S-functions for graphs. J. Geom. **8**, 171–186 (1976)
9. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. J. Algorithms **7**, 309–322 (1986)
10. Dujmović, V., Morin, P., Wood, D.R.: Layered separators for queue layouts, 3D graph drawing and nonrepetitive coloring. In: Proceedings of 54th Symposium on Foundations of Computer Science (FOCS 2013), pp. 280–289. IEEE Computer Society (2013)
11. Dujmović, V., Morin, P., Wood, D.R.: Layered separators in minor-closed graph classes with applications. Electronic preprint arXiv:1306.1595 (2013)
12. Sugiyama, K., Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical system structures. IEEE Trans. Syst. Man Cybern. **SMC–11**, 109–125 (1981)
13. Eades, P., Wormald, N.C.: Edge crossings in drawings of bipartite graphs. Algorithmica **11**, 379–403 (1994)
14. Healy, P., Nikolov, N.S.: How to layer a directed acyclic graph. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 16–30. Springer, Heidelberg (2002). doi:10.1007/3-540-45848-4_2
15. Felsner, S., Liotta, G., Wismath, S.K.: Straight-line drawings on restricted integer grids in two and three dimensions. J. Graph Algorithms Appl. **7**, 363–398 (2003)
16. Dujmović, V., Wood, D.R.: Stacks, queues and tracks: layouts of graph subdivisions. Discrete Math. Theor. Comput. Sci. **7**, 155–201 (2005)
17. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, Berlin (2013)
18. Bannister, M.J., Cabello, S., Eppstein, D.: Parameterized complexity of 1-planarity. In: Dehne, F., Solis-Oba, R., Sack, J.-R. (eds.) WADS 2013. LNCS, vol. 8037, pp. 97–108. Springer, Heidelberg (2013). doi:10.1007/978-3-642-40104-6_9
19. Nešetřil, J., de Mendez, P.O.: Sparsity (Graphs, Structures, and Algorithms). Algorithms and Combinatorics, vol. 28. Springer, Heidelberg (2012)
20. Abel, Z., Demaine, E.D., Demaine, M.L., Eppstein, D., Lubiw, A., Uehara, R.: Flat foldings of plane graphs with prescribed angles and edge lengths. In: Duncan, C., Symvonis, A. (eds.) GD 2014. LNCS, vol. 8871, pp. 272–283. Springer, Heidelberg (2014). doi:10.1007/978-3-662-45803-7_23
21. Bandelt, H.J., Chepoi, V., Eppstein, D.: Combinatorics and geometry of finite and infinite squaregraphs. SIAM J. Discrete Math. **24**, 1399–1440 (2010)