On Area-Optimal Planar Graph Drawings*

Therese Biedl

David R. Cheriton School of Computer Science, University of Waterloo,
Waterloo, Ontario N2L 1A2, Canada
biedl@uwaterloo.ca

Abstract. One of the first algorithmic results in graph drawing was how to find a planar straight-line drawing such that vertices are at grid-points with polynomial coordinates. But not until 2007 was it proved that finding such a grid-drawing with optimal area is NP-hard, and the result was only for disconnected graphs.

In this paper, we show that for graphs with bounded treewidth, we can find area-optimal planar straight-line drawings in one of the following two scenarios: (1) when faces have bounded degree and the planar embedding is fixed, or (2) when we want all faces to be drawn convex. We also give NP-hardness results to show that none of these restrictions can be dropped. In particular, finding area-minimal drawings is NP-hard for triangulated graphs minus one edge.

1 Introduction

A planar graph is a graph that can be drawn without crossing in the plane. Naturally one wonders whether such a drawing must use curves, or whether there exists a *planar straight-line drawing*, i.e., a drawing such that vertices are at points, edges are straight-line segments between their endpoints, no edge overlaps a non-incident vertex, and no two edges cross. It was proved multiple times independently that every planar graph has such a drawing [4][12][14].

To increase readability of such a drawing, vertices should be not too close to each other, but the drawing should fit on a small paper or screen. The first objective can be achieved by demanding a *grid drawing*, where all vertices are placed at points with integer coordinates. The second objective can be achieved by minimizing the area of the smallest enclosing box of such a grid-drawing. In 1990, it was shown independently by de Fraysseix, Pach and Pollack [6] and Schnyder [11] that every planar graph has a grid-drawing of area $O(n^2)$.

Numerous papers have since worked on improving the constant factor in this $O(n^2)$-bound; see e.g. [2] and the references therein. However, very few attempts have been made to find drawings with the *optimal* area. Indeed, it was not even shown until 2007 that finding a grid-drawing with optimal area is NP-hard [8]. One of the very few algorithms that achieves optimal area is by Mondal et al. [9]

* Supported by NSERC and the Ross and Muriel Cheriton Fellowship. Research initiated while participating at Dagstuhl seminar 13421. The author would like to thank Martin Vatshelle; many ideas in this paper were strongly inspired by [1].

J. Esparza et al. (Eds.): ICALP 2014, Part I, LNCS 8572, pp. 198–210, 2014.

for triangulated graphs of treewidth 3. The current paper arose out of an attempt to generalize this results to graphs of larger (but bounded) treewidth. We show:

> Minimizing the area of planar grid-drawings is polynomial-time solvable for any planar graph of bounded treewidth and bounded face degrees for which the planar embedding is fixed.

We also prove NP-hardness as soon as any of the three conditions (bounded treewidth, bounded face degree, fixed planar embedding) is dropped. We then turn to *convex drawings*, where we additionally demand that every face (including the outer-face) is drawn as a convex polygon, and show:

> Minimizing the area of convex planar grid-drawings is polynomial-time solvable for any planar graph of bounded treewidth.

We also prove NP-hardness of finding area-optimal convex drawings for graphs where the treewidth is not constant. Due to space restrictions, many details (especially for the NP-hardness reductions) have been omitted.

2 Background

We assume that $G = (V, E)$ is a planar graph, i.e., it can be drawn in the plane without crossing. Any planar drawing of G defines a *rotation system*, i.e., a clockwise order of edges around each vertex. This defines *facial circuits*, which are boundary cycles of the maximal connected regions (*faces*) of the drawing. The *outer-face* is the unbounded face. The drawing also assigns each *angle* (set of two consecutive edges at a vertex) to a face. A *planar embedding* is a rotation system, an angle-face assignment, and one angle fixed as being on the outer-face; this determines a drawing of G up to deformations of the plane.

A graph is *connected* if there exists a path between any two vertices. For a connected graph every face is bounded by one circuit and hence the angle-face-assignment is unique. A graph is *k-connected* if it remains connected after removing any $k-1$ vertices. Any 3-connected planar graph has a unique rotation system (up to reversal of all orders). A planar graph is called *triangulated* if all faces, including the outer-face, are triangles. Such a graph is always 3-connected.

A *planar straight-line grid-drawing* is a mapping Γ of the vertices of G to distinct points with integer coordinates such that if we draw every edge as a straight-line segment, then no edge overlaps a non-incident vertex, and no two edges cross. The drawing is *convex* if all faces, including the outer-face, are drawn as convex polygons. Angles of 180° are allowed (though the results of the paper can easily be generalized to strictly convex drawings). The *width*, *height* and *area* of Γ is the corresponding measure of the minimum axis-aligned box that encloses all points that Γ maps to.

Definition 1. AreaMinimization *is the following problem: Given a graph G and an integer A, does G have a planar straight-line grid-drawing of area $\leq A$?*

PointSetEmbeddability *is the following problem: Given a graph G and a set S of points in $\mathbb{R}^2$, is there a planar straight-line drawing of G where all vertices are on points of S?*

For both problems, we also consider the CONVEX variant, where we require the planar drawing to be convex.

Treewidth and Sc-decompositions: We will not define the treewidth tw(G) of a graph G, since we do not use it directly. The *pathwidth* pw(G) of a graph G is the smallest k such that G has a vertex order $v_1, \ldots, v_n$ where, for any $1 \leq i < n$, at most k vertices in $v_1, \ldots, v_i$ have a neighbour in $v_{i+1}, \ldots, v_n$. Since $\text{tw}(G) \leq \text{pw}(G)$, it suffices to prove NP-hardness for bounded pathwidth graphs. A *branch decomposition* of a graph G is a rooted binary tree T with edges of G in 1-1 correspondence with the leaves of T. It will be convenient to assume that the root has only one child. For any arc a of T, the *subgraph* G_a *rooted at* a is the graph formed by all edges at leaves that are below a in T. The *separator* σ_a *at* a is the set of vertices with incident edges in both G_a and $G - G_a$. The *width* of a branch decomposition is $\max_{a \in T} |\sigma_a|$, and the *branchwidth* bw(G) of G is the smallest width of a branch decomposition of G. Since $\text{tw}(G) \in \Theta(\text{bw}(G))$ [10], it suffices to give algorithms for bounded branchwidth graphs.

Let G be a planar graph with a fixed rotation system. A *noose* is a sequence $v_0, f_0, v_1, \ldots, v_{k-1}, f_{k-1}$ such that for any $0 \leq i < k$, vertices v_i and v_{i+1} both belong to face f_i (addition modulo k), and no vertex repeats. (Faces may repeat.) An *sc-decomposition* is a branch decomposition of G such that for any arc a of the tree T, there exists a noose N_a whose vertices are exactly the separator σ_a. We can picture N_a as a simple closed curve that intersects a planar drawing of G only at vertices in σ_a, contains G_a on one side and $G - G_a$ on the other side. See Fig. 1. Any 2-connected planar graph G with a fixed planar embedding has an sc-decomposition of width bw(G), and it can be found in polynomial time [3].

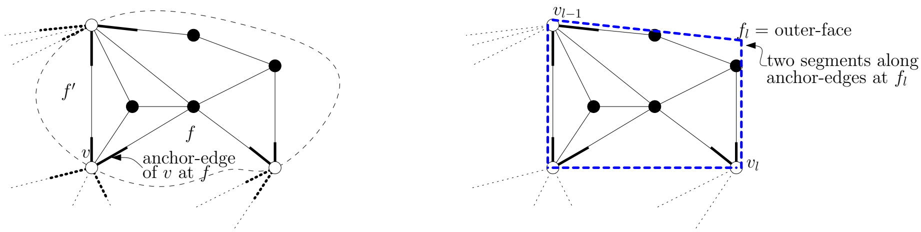

Fig. 1. A noose (dashed) meets the separator vertices (white). Illustrating some concepts for Section 3: Anchor-edges are thick, pole-edges are thick dotted, the noose-polygon is thick dashed.

3 Area-Optimal Drawings

AREAMINIMIZATION is NP-hard even for graphs with a fixed planar embedding and constant treewidth [8]. But it becomes polynomial if additionally face-degrees are bounded:

Theorem 1. *Let G be a planar graph with a fixed planar embedding. If G has bounded treewidth and bounded face-degrees, then* AreaMinimization *can be solved in polynomial time.*

Proof. It was shown in [1] that PointSetEmbeddability is polynomial-time solvable for these graphs. For $W = 1, \ldots, \lceil\sqrt{A}\rceil$ and $H = \lfloor A/W \rfloor$, solve PointSetEmbeddability for G, using the points of a $W \times H$-grid as S. There exists a drawing of area at most A if and only if we succeed for some S. Clearly this yields a polynomial-time algorithm since $A \leq n^2$.[1]

For convex drawings, the results for ConvexPointSetEmbeddability in [1] are more restrictive than we need them to be. We show in the rest of this section that ConvexPointSetEmbeddability is polynomial-time solvable for a planar graph G of bounded treewidth. This was previously only known for graphs with bounded treewidth, bounded vertex degrees, and where the planar embedding is fixed [1]. So assume that we are given a set S of at least n points in the plane. Fix an arbitrary planar embedding of G (we will later explore all possible ones). The main idea is to do dynamic programming in an sc-decomposition of width $\mathrm{bw}(G)$, where the dynamic programming function fixes the position of separator-vertices, as well as their neighbours at the noose.

Formally, let a be an arc of the sc-decomposition tree T, and let v be a vertex of σ_a. By assumption the noose N_a contains f, v, f' as subsequence, for some faces f, f'. By definition of separator, v has incident edges in both G_a and $G - G_a$. Since N_a contains v only once, the incident edges of v hence form two intervals (in the clockwise order around v): one with edges in G_a and one with edges in $G - G_a$. We call the first and last edge of v in G_a the *anchor-edges of* v, and the first and last edge of v in $G - G_a$ the *pole-edges of* v. (Both terms are "with respect to arc a", but arc a will be clear from the context.) Each anchor-edge and pole-edge belongs to either f or f'. See also Fig. 1. An *anchor/pole* of arc a is a vertex x such that (x, v) is an anchor-edge/pole-edge at some vertex v in σ_a. Let A_a be the set of anchors and poles of a, and let G_a^+ be the graph obtained from G_a by adding to it all pole-edges at vertices in σ_a. The dynamic programming function searches for a drawing of G_a^+, subject to a fixed mapping Γ_a of the vertices of $\sigma_a \cup A_a$ to points in S.

From the locations of $\sigma_a \cup A_a$, we can read a polygon that serves as curve for the noose; we call this the *noose-polygon* $\mathcal{P}_a$. Namely, consider the polygon $\Gamma_a(v_0), \ldots, \Gamma_a(v_{k-1})$, where $v_0, f_0, v_1, \ldots, v_{k-1}, f_{k-1}$ are the vertices and faces of the noose N_a. If N_a does not include the outer-face, then $\mathcal{P}_a$ is this polygon. If N_a does include the outer-face, say at f_l, then replace line segment $\overline{\Gamma_a(v_l), \Gamma_a(v_{l+1})}$ by two segments along the supporting lines of the anchor-edge of v_l and v_{l+1} at f_l. See Fig. 1(right). In any convex drawing of G, the noose-polygon $\mathcal{P}_a$ does not cross itself, since every edge of it either resides in an interior face (a convex polygon) or is drawn along two supporting lines of edges on the

[1] The precise run-time is $O(n^3 \Delta(t+1) \log(\Delta(t+1)) \cdot |S|^{1.5\Delta(t+1)})$, where Δ is the maximum degree of a face and t is the treewidth.

outer-face (a convex polygon). If all vertices of σ_a are collinear (e.g. if $|\sigma_a| = 2$), then $\mathcal{P}_a$ may overlap itself; we do not consider this a crossing.

One final notation. In what follows, we consider a drawing Γ of a subgraph G_a^+ of G. We say that an angle is *drawn properly* in Γ if either this angle is not a facial angle of G, or if it belongs to an interior face of G and is drawn with at most 180°, or it belongs to the outer-face of G and is drawn with at least 180°. The function to be computed via dynamic programming is now as follows:

Definition 2. *Let a be an arc of the rooted sc-decomposition. Let Γ_a be a mapping from $\sigma_a \cup A_a$ to S. Define $M(a, \Gamma_a)$ to be true if and only if:*

1. *The noose-polygon $\mathcal{P}_a$ defined by Γ_a has no crossings.*
2. *For any anchor α and any pole ρ of a, any curve from $\Gamma_a(\alpha)$ to $\Gamma_a(\rho)$ contains points of $\mathcal{P}_a$. Put differently, the noose-polygon $\mathcal{P}_a$ forms a boundary between the anchors and the poles.*
3. *There exists a planar drawing Γ of G_a^+ on S such that all angles are drawn properly and Γ coincides with Γ_a on $\sigma_a \cup A_a$.*

Observe that computing $M(a, \Gamma_a)$ for all values of a and Γ_a is sufficient to solve ConvexPointSetEmbeddability for a graph with a fixed planar embedding. For $G_r = G$ at the unique arc r below the root, and so $\bigvee_{\Gamma_r} M(r, \Gamma_r)$ is true if and only if G has a drawing on S for which all angles are drawn properly, i.e., G has a convex drawing on S. We explain how to compute $M(a, \Gamma_a)$ by going bottom-up in the tree T of the sc-decomposition.

$M(a, \Gamma_a)$ at a leaf-arc: Assume first that a is an arc incident to a leaf, say the leaf stores edge (v, w). Then G_a^+ consists of (v, w) as well as up to four pole-edges. (The anchor-edges all coincide with (v, w).) Hence all vertices of G_a^+ belong to $\sigma_a \cup A_a$, so the mapping Γ_a determines the drawing of G_a^+. Testing whether $M(a, \Gamma_a)$ is true hence reduces to checking whether the angles are drawn properly or not. Further, to respect the given planar embedding, the clockwise order of pole-edges and (v, w) must be as induced by the rotation scheme. This can all be tested in constant time for one fixed arc a and mapping Γ_a.

$M(a, \Gamma_a)$ at a non-leaf arc: So assume now that arc a is not incident to a leaf. Then the lower end of a is in turn incident to two other arcs a_1 and a_2. We now show how to extract the value for $M(a, \Gamma_a)$ from those of $M(a_1, \Gamma_{a_1})$ and $M(a_2, \Gamma_{a_2})$ for some suitably chosen mappings Γ_{a_1} and Γ_{a_2}.

Recall that Γ_a determines the positions for all points in $\sigma_a \cup A_a$. With this, we can test the first two conditions of Definition 2 directly, and assume from now on that they are satisfied. In particular, the noose-polygon $\mathcal{P}_a$ then has an *anchor-side*, which is the connected component of $\mathbb{R}^2 - \mathcal{P}_a$ that contains the anchors, and the *pole-side*, which is the connected component that contains the poles. Define $\sigma_\times := (\sigma_{a_1} \cup \sigma_{a_2}) - \sigma_a$ and $A_\times := (A_{a_1} \cup A_{a_2}) - A_a$. If we fix any mapping of $\sigma_a \cup \sigma_\times \cup A_a \cup A_\times$ to points in S, then this fixes (for $i = 1, 2$) also a mapping of $\sigma_{a_i} \cup A_{a_i}$ to points in S. One can easily show the following formula:

Lemma 1. *$M(a, \Gamma_a)$ is true if and only if the first two conditions of Definition 2 are true, and there exists a mapping $\Gamma_\times$ from $\sigma_\times \cup A_\times$ to S such that*

– *$M(a_1, \Gamma_{a_1})$ and $M(a_2, \Gamma_{a_2})$ are true, and*
– *the interior of the anchor-side of $\mathcal{P}_{a_1}$ has no points in common with the interior of the anchor-side of $\mathcal{P}_{a_2}$,*

where Γ_{a_i} is the mapping from $\sigma_{a_i} \cup A_{a_i}$ to S induced by Γ_a and $\Gamma_\times$.

The algorithm for computing $M(a, \Gamma_a)$ is now the obvious: For any choice $\Gamma_\times$ of mapping $\sigma_\times \cup A_\times$ to points of S, compute the induced mappings Γ_{a_i} and look up whether $M(a_i, \Gamma_{a_i})$ is true, for $i = 1, 2$. Also compute $\mathcal{P}_{a_i}$, and check whether the anchor-sides are interior-disjoint. Set $M(a, \Gamma_a)$ to be true if and only if we succeed for some choice of $\Gamma_\times$.

Putting it All Together. So to solve ConvexPointSetEmbeddability for a fixed planar embedding, we go bottom-up in the tree T of the sc-decomposition, and at each arc a and each possible mapping Γ_a compute $M(a, \Gamma_a)$ as explained above. It remains to analyze the run-time. Computing $M(a, \Gamma_a)$ for an arc a incident to a leaf and a fixed Γ_a takes constant time. Doing so for all Γ_a takes $O(|S|^5)$ time since there are five vertices to which we must assign points. To compute $M(a, \Gamma_a)$ for an arc a not incident to a leaf and a fixed assignment Γ_a, we must try all possible mappings $\Gamma_\times$ of points to $\sigma_\times \cup A_\times$. For each of them, we must compute the induced assignments Γ_{a_1} and Γ_{a_2}, look up $M(a_1, \Gamma_{a_1})$ and $M(a_2, \Gamma_{a_1})$, and test whether the noose-polygons are interior-disjoint. This can all be done in $O(n \log n)$ time with suitable data structures. Thus the time to compute $M(a, \cdot)$, for all possible choices of Γ_a and $\Gamma_\times$ is

$$O\left(|S|^{|\sigma_a \cup A_a \cup \sigma_\times \cup A_\times|} n \log n\right).$$

Because any separator-vertex contributes at most two anchors and two poles, and any separator-vertex appears in at least two of $\sigma_a, \sigma_{a_1}, \sigma_{a_2}$, one can argue that $|\sigma_a \cup A_a \cup \sigma_\times \cup A_\times| \leq \frac{15}{2}\mathrm{bw}(G)$. Hence the time to compute $M(a, \cdot)$ is $O(|S|^{7.5\mathrm{bw}(G)} n \log n)$, and doing so for all $O(n)$ arcs of the sc-decomposition adds another $O(n)$-factor. So we have:

Theorem 2. *For any planar graph G and any set S of points in $\mathbb{R}^2$, if the planar embedding of G is fixed then we can solve* ConvexPointSetEmbeddability *in $O((|S|^{7.5bw(G)} n^2 \log n)$ time.*

Now we consider the case when the planar embedding is not fixed. If G is 3-connected, then simply try all possible outer-faces for an additional $O(n)$ run-time overhead. So assume from now on that G has cutting pairs.

A graph may have $\Omega(2^n)$ rotation systems that all lead to a convex drawing, so we cannot explore all of them explicitly. Tutte's characterization [13] states that if G has a convex drawing, then any cutting pair must be on the outer-face and have exactly two cut-components (not counting a possible edge between the cutting pair). Therefore rotation systems of convex drawings can differ only by "flipping" (reversing the rotation sub-systems) of a *leaf-component*, i.e., a 3-connected component that is a leaf in the tree of 3-connected components. We use a special branch-decomposition:

Lemma 2. *For any 2-connected graph G, there exists a branch decomposition T of G of width $bw(G)$ such that*

- *for any leaf-component $\mathcal{C}$ there exists an arc $a_{\mathcal{C}}$ in T with $G_{a_{\mathcal{C}}} = \mathcal{C}$,*
- *for any planar embedding of G and any arc a in T, there exists a noose that contains G_a on one side and $G - G_a$ on the other. Furthermore, the (clockwise or counter-clockwise) order of the vertices of the noose is the same regardless of the planar embedding.*

Previously, anchors and poles were vertices defined by the planar embedding. Now we only know which two vertices v_ℓ and $v_{\ell+1}$ are consecutive in the noose at an arc a, but this is sufficient. Change the definition of Γ_a as follows: let Γ'_a be a mapping that assigns five points to each vertex $v_\ell \in \sigma_a$. These five points belong to v_ℓ, the (unknown) anchor at v_ℓ "towards" $v_{\ell+1}$ (i.e., at the (unknown) face which v_ℓ shares with $v_{\ell+1}$,), the pole at v_ℓ twoards $v_{\ell+1}$, and the anchor and pole at v_ℓ "towards" $v_{\ell-1}$. We allow Γ'_a to repeat points, which avoids having to explore explicitly whether these poles/anchors are distinct vertices. Notice that Γ'_a is sufficient to compute the noose-polygon, as long as we pass along the information which consecutive vertices (if any) of the noose belong to the outer-face. Also, Γ'_a determines a drawing of G^+_a, given one of G_a. Hence define $M'(a, \Gamma'_a)$ to be verbatim the same as $M(a, \Gamma_a)$, except that we use Γ'_a and allow all possible planar embeddings in (3).

The computation of $M'(a, \Gamma'_a)$ is nearly the same as the one of $M(a, \Gamma_a)$, except that in the base case we do not check whether the rotation system is respected, and that at any arc of a leaf component we test both possible choices of which face of the noose is the outer-face. Hence $M'(a, \Gamma'_a)$ explores all possible ways of flipping leaf components, and hence implicitly all planar embeddings that could lead to a convex drawing. Thus the run-time is the same as for the fixed planar embedding, except that we need an additional $O(n)$ factor for trying all possible outer-faces (this is needed only if the graph is 3-connected.) Summarizing, we get:

Theorem 3. *For any planar graph G and any set S of points in $\mathbb{R}^2$, we can solve* ConvexPointSetEmbeddability *in $O((|S|^{7.5bw(G)}n^3 \log n)$ time.*

With the same approach as in Theorem 1 (try all choices of S as a $W \times H$-grid for $W \cdot H \leq A$) we hence have:

Corollary 1. *Let G be a planar graph. If G has bounded treewidth, then* ConvexAreaMinimization *can be solved in polynomial time.*

We can use this to give subexponental exact algorithms for area-optimal convex drawings. We are not familiar with any previous results in this area. The obvious brute-force approach (try for any assignment of grid points to the vertices whether it works) yields an algorithm with run-time $O^*((n^2)^n)$, where the $O^*(\cdot)$ notation hides polynomial terms.

Corollary 2. *There exists an algorithm to find a minimum-area convex grid-drawing of a planar graph in $O^*(2^{O(\sqrt{n}\log n)})$ time.*

Proof. Any planar graph has branchwidth $O(\sqrt{n})$. By solving ConvexAreaMinimization for values $A = 1, 2, \ldots, n^2$ we can hence find the minimum-area convex grid-drawing in time $O^*((n^2)^{O(\sqrt{n})}) = O^*(2^{O(\sqrt{n}\log n)})$ time.

4 NP-Hardness Results

We now give NP-hardness proofs that show that none of the conditions needed for Theorem 1 and Corollary 1 can be dropped. Our reductions borrow many ideas from [8] and [1].

4.1 Small Treewidth, Small Face-Degrees, Flexible Embedding

Recall that AreaMinimization is polynomial-time solvable if the treewidth and the face-degrees are bounded, and the planar embedding is fixed. We now show that if we allow to choose the planar embedding, the problem becomes NP-hard.

The reduction is from the 3-Partition problem defined as follows: Given $3n$ positive integers $a_1, \ldots, a_{3n}$, where $\sum_{i=1}^{3n} a_i = n \cdot B$ and $\frac{1}{4}B < a_i < \frac{1}{2}B$ for all i, is there a partition of $a_1, \ldots, a_{3n}$ into n groups of 3 numbers each such that each group sums to B? It is well-known that 3-Partition is strongly NP-hard [7]. Given an instance of 3-Partition, we first define a frame, shown for $n = 3$ and $B = 6$ in Fig. 2. It consists of a $W \times 4$-grid (for $W \geq n(B+1) + 2$ odd) with n repetitions of a pattern that leaves a face with $B + 3$ points not used by the frame. Above and below this strip are $(W + 1)/2$ stacked cycles (not shown fully in Fig. 2); all except the outer-most one are 4-cycles. Set $A := W(2W + 4)$.

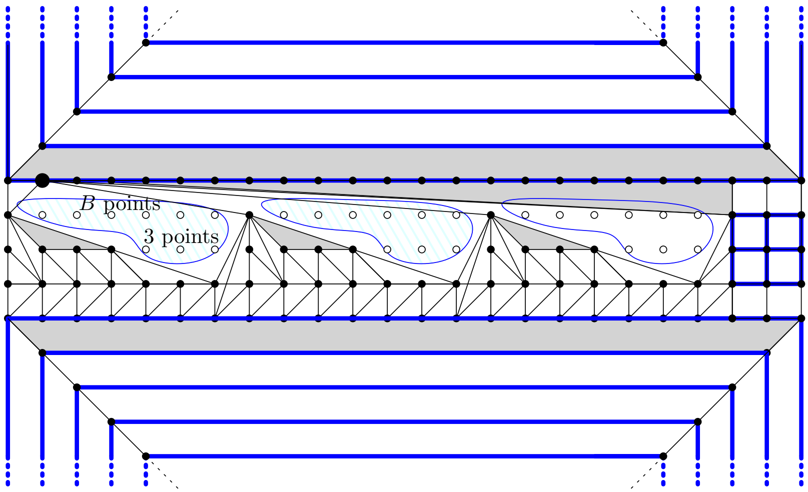

Fig. 2. The frame of the NP-hardness construction. Shaded areas are triangulated; we omit drawing edges in these areas for clarity.

The frame is 3-connected (recall that shaded areas are triangulated) and has a unique rotation system. Any k stacked cycles require a $2k \times 2k$-grid in any planar drawing. Since the frame has two sets of $(W+1)/2$ stacked cycles and a 2×2-grid (shown bold) that are vertex-disjoint, one can argue that any drawing of area A must have the outer-face shown in the figure and the drawing must be (up to rotation) in a $W \times (2W+4)$-grid.

The graph in the middle strip has (once we add the gadgets for the a_i's) *exactly* as many vertices as we can make grid points available to it. So not a single grid point may be "wasted" by not having a vertex on it. One can argue (details are omitted) that this forces the frame to be drawn exactly as shown.

For each $i = 1, \ldots, 3n$, define a path of length a_i. Each vertex on this path is connected to the "fat" vertex of the frame; this is the unique vertex in the graph where the rotation system can be changed. Also add one vertex x_i per index i, which is adjacent to all vertices of the a_i-path. See Fig. 3.

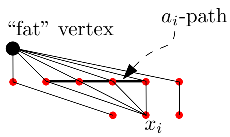

Fig. 3. Encoding $a_1{=}1, a_2{=}4, a_3{=}1$

The frame left n faces with $B+3$ points each. If the 3-partition instance has a solution, then for each group $a_{i_1}, a_{i_2}, a_{i_3}$ that sums to B, we pick one of these faces, place the a_i-paths in the row with B points, and $x_{i_1}, x_{i_2}, x_{i_3}$ in the row below. All edges can be drawn without crossing, and the face that results has degree 17 (12 edges from the three gadgets, and 5 edges from the frame.) Vice versa, if the graph can be drawn in area A, then the frame must be drawn as shown, and so the a_i-gadgets must be split up among the n faces with $B+3$ points each. This gives a partition of the a_i's as desired.

Observe that the middle strip, including the a_i-gadgets, could always be drawn on 5 rows if we allowed an increase in width. It follows that the middle strip has pathwidth at most 5 [5]. Each set of stacked cycles has pathwidth at most 4. By combining their vertex orders, one can hence show that the graph has pathwidth at most 7.

Theorem 4. AreaMinimization *is NP-hard, even for a connected planar graph with pathwidth at most 7, and even if we demand a planar drawing where every face has at most degree 17.*

4.2 3-Connected, and Convex Faces or Small Treewidth

Recall that AreaMinimization is polynomial-time solvable if the treewidth and the face-degrees are bounded, and the rotation system is fixed. We now show NP-hardnes if the condition on treewidth is dropped. The same construction also works for NP-hardness of ConvexAreaMinimization.

Let $a_1, \ldots, a_{3n}$ be an instance of 3-partition. Define $\ell := \max\{12n^2 + 3n, \lceil (B-1)/2 \rceil\}$. We construct a graph that is 3-connected, hence has a unique rotation system. All faces are triangles except one face of degree 4. Fig. 4 shows the *frame* of the graph. The width and height of this drawing is $W := 6\ell + 2n + 6$, and we set $A := W^2$. The frame has three connected components, the *outer frame* as well as two *blobs* that consist of ℓ stacked cycles with one vertex in the middle.

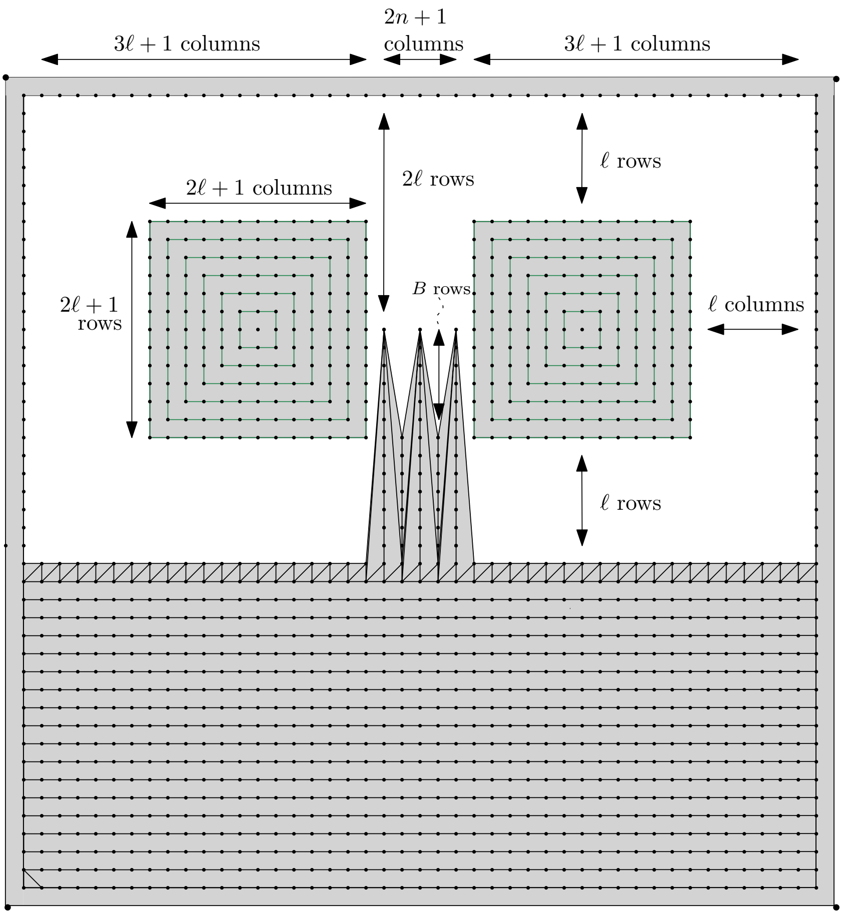

Fig. 4. The NP-hardness reduction for inner triangulated graphs. The frame for $n = 2$ and $B = 6$. The picture uses $\ell = 6$ (while $\ell = 56$ would be correct). Shaded areas are triangulated; we omit showing these edges for clarity.

Any drawing of area $\leq A = W^2$ has at most $(W-1)^2$ grid points that are not on the four extreme grid lines. G will have $(W-1)^2+4$ vertices, so in any drawing with outer-face f it has $(W-1)^2+4-\deg(f)$ vertices that are not on the outer-face, hence cannot be placed on extreme grid lines. It follows that if there exists a drawing of G of area $\leq A$, then it must be on a $W \times W$-grid, and the outer-face of G must be the (unique) face that has degree 4. Furthermore, in such a drawing not a single grid point not on an extreme grid-line may be wasted. Using this, one can argue that the outer frame is drawn exactly as in Fig. 4, up to reflection and rotation. In particular, we have $n+1$ "teeth" that stick into the middle region, each column of a tooth has 2ℓ grid points left (i.e., not used by the outer frame), and each column between two teeth has $2\ell + B$ grid points left. The two blobs are too big to fit into the columns at the teeth, and too big to both fit left of the teeth, so one must be left of the teeth and the other right of the teeth.

Between the outer frame and the blobs we add ℓ stacked cycles called *layers*; the two blobs are inside all these cycles. Each layer must start in the far left (to surround the left blob), go past all teeth to the right (to surround the right blob), and then go back. The length of each layer is set so that it exactly fills the grid points encountered along this path. The ℓ layers hence use up all grid points except for B grid points in each column between two teeth.

There are $\ell \geq 12n^2 + 3n = (3n) \cdot (4n+1)$ layers. For $i = 1, \ldots, 3n$, insert a path of length a_i in the face between layer $i(4n)$ and layer $i(4n)+1$. These paths can only be placed in the columns between teeth, so a drawing of area A gives an assignments of the a_i's into groups that sum to B each, as desired.

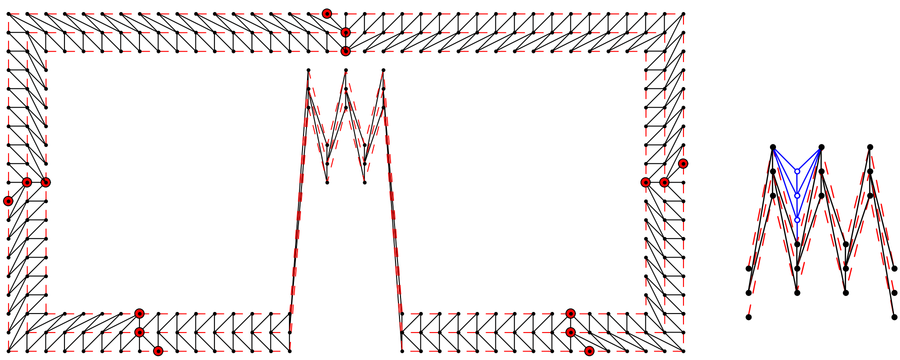

Fig. 5. Triangulating between layers (red, dashed), and how to attach the a_i-path

To make the graph inner triangulated, we connect two consecutive layers with a zig-zag line, except for "collector-points" (thick and red in Fig. 5) that have three or four neighbours on the previous layer, including the previous collector-points. There are two ways to draw these connections; in one way the collector-points are aligned vertically/horizontally, while in the other they are shifted clockwise by one unit. We use this for $2n$ pairs of layers, and then for the next $2n$ pairs of layers use a symmetric construction that allows collector-points to be aligned or to be shifted counter-clockwise by one unit. Finally, when adding the a_i-path, we attach it at the vertex diametrically opposite to the top collector-point, and connect all vertices on the path to the two vertices before/after that attachment-point. Over the course of the $4n$ layers between paths, we can shift collector-points by up to $\pm 2n$ units clockwise or counter-clockwise. With this, we can bring the attachment-point of the a_i-path to any of the n columns between teeth, regardless of where the a_{i-1}-path was. Hence for any solution of 3-partition we can draw G in a $W \times W$-grid, and since faces are triangles or squares, the drawing is convex. We conclude:

Theorem 5. AreaMinimization *and* ConvexAreaMinimization *are NP-hard, even for an internally triangulated 3-connected planar graph for which the outer-face is a 4-cycle.*

By omitting the triangulation edges in shaded areas of Fig. 4, and (roughly speaking) replacing the triangulation between any second pair of layers by three edges only, we can create a variant of this graph that has pathwidth at most 7 and for which any drawing of area A implies a solution to 3-partition.

Theorem 6. AreaMinimization *is NP-hard, even for a 3-connected planar graph of bounded pathwidth.*

5 Conclusion

This paper revisited the problem of drawing planar graphs with optimal area. We showed that finding a convex planar drawing with optimal area is possible in polynomial time for all graphs with bounded treewidth, even if the planar embedding is not fixed. Based on results for point-set embeddability, we also showed that finding an area-optimal planar drawing is polynomial-time solvable if the graph has bounded treewidth, bounded face-degrees, and the planar embedding is fixed.

As for open problems, can one approximate the optimal area, or is the optimization version of AreaMinimization APX-hard? For many other problems, finding a poly-time algorithm for bounded-treewidth graphs was a first step towards approximation algorithms. To do so, it would be helpful to find algorithms that are fixed-parameter tractable in the treewidth, but ours is not. Is Convex-AreaMinimization $W[1]$-hard with respect to the parameter treewidth?

References

1. Biedl, T., Vatshelle, M.: The point-set embeddability problem for plane graphs. International Journal of Computational Geometry and Applications (in press, 2014)
2. Brandenburg, F.: Drawing planar graphs on (8/9)*n^2 area. Electronic Notes in Discrete Mathematics 31, 37–40 (2008)
3. Dorn, F.: Dynamic programming and planarity: Improved tree-decomposition based algorithms. Discrete Applied Mathematics 158(7), 800–808 (2010)
4. Fáry, I.: On straight line representation of planar graphs. Acta Scientiarum Mathematicarum (Szeged) 11(4), 229–233 (1948)
5. Felsner, S., Liotta, G., Wismath, S.: Straight-line drawings on restricted integer grids in two and three dimensions. Journal of Graph Algorithms and Applications 7(4), 335–362 (2003)
6. de Fraysseix, H., Pach, J., Pollack, R.: How to draw a planar graph on a grid. Combinatorica 10, 41–51 (1990)

7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman (1979)
8. Krug, M., Wagner, D.: Minimizing the area for planar straight-line grid drawings. In: Hong, S.-H., Nishizeki, T., Quan, W. (eds.) GD 2007. LNCS, vol. 4875, pp. 207–212. Springer, Heidelberg (2008)
9. Mondal, D., Nishat, R.I.: Md. S. Rahman, and Md. J. Alam. Minimum-area drawings of plane 3-trees. J. Graph Algorithms Appl. 15(2), 177–204 (2011)
10. Robertson, N., Seymour, P.D.: Graph minors. X. Obstructions in tree-decompositions. J. Combin. Theory Ser. B 52, 153–190 (1991)
11. Schnyder, W.: Embedding planar graphs on the grid. In: ACM-SIAM Symposium on Discrete Algorithms (SODA 1990), pp. 138–148 (1990)
12. Stein, S.: Convex maps, vol. 2, pp. 464–466. American Mathematical Society (1951)
13. Tutte, W.T.: Convex representations of graphs. Proc. London Math. Soc. 10(3), 304–320 (1960)
14. Wagner, K.: Bemerkungen zum Vierfarbenproblem. Jahresbericht der Deutschen Mathematiker-Vereinigung 46, 26–32 (1936)