Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Planar drawings of fixed-mobile bigraphs ☆

Michael A. Bekos [a], Felice De Luca [b,g], Walter Didimo [b,*], Tamara Mchedlidze [c], Martin Nöllenburg [d], Antonios Symvonis [e], Ioannis Tollis [f]

[a] *Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Germany*
[b] *Dipartimento di Ingegneria, Università degli Studi di Perugia, Italy*
[c] *Karlsruhe Institute of Technology, Germany*
[d] *TU Wien, Vienna, Austria*
[e] *National Technical University of Athens, Greece*
[f] *University of Crete, Greece*
[g] *University of Arizona, USA*

A R T I C L E I N F O

Article history:
Received 13 April 2018
Received in revised form 22 February 2019
Accepted 15 July 2019
Available online xxxx
Communicated by T. Calamoneri

Keywords:
Graph drawing
Graph planarity
Bipartite graphs
Fixed-mobile bigraphs

A B S T R A C T

A *fixed-mobile bigraph* G is a bipartite graph such that the vertices of one partition set are given with *fixed* positions in the plane and the *mobile* vertices of the other partition, together with the edges, must be added to the drawing without any restriction on their positions. We assume that G is planar and study the problem of finding a planar straight-line drawing of G. We show that deciding whether G admits such a drawing is NP-hard in the general case. Under the assumption that each mobile vertex is placed in the convex hull of its neighbors, we are able to prove that the problem is also in NP. Moreover, if the intersection graph of these convex hulls is a path, a cycle or, more generally, a cactus, the problem is polynomial-time solvable through a dynamic programming approach. Finally, we describe linear-time testing algorithms when the fixed vertices are collinear or when they lie on a finite set of horizontal levels (lines) and no edge can intersect a level except at its fixed vertex.

1. Introduction

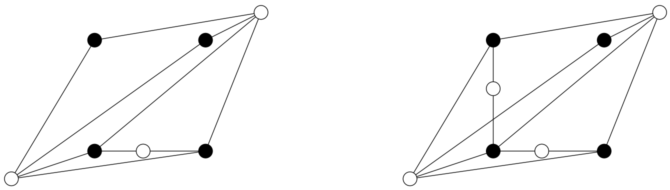
This paper considers the following problem. Let $G = (V_f, V_m, E)$ be a planar bipartite graph such that the vertices in V_f, called *fixed vertices*, have fixed distinct locations (i.e., points) in the plane, while the vertices in V_m, called *mobile vertices*, can be freely placed (here we stress that by mobile vertex we do not mean a vertex that dynamically moves over time, but a vertex whose position can be freely chosen). Does G admit a crossing-free drawing Γ with straight-line edges? We assume that each vertex of G is drawn in Γ as a distinct point in the plane. We refer to G as an *FM-bigraph* and to Γ as a *planar straight-line drawing* of G.

Fig. 1 shows two different instances of our problem, both having the same set of four fixed vertices, colored black (the graph of each instance is planar and bipartite). The FM-bigraph in Fig. 1(a) has three mobile vertices (colored white) and

☆ An extended abstract of this research has been published in the proceedings of the 25th International Symposium on Graph Drawing and Network Visualization, GD 2017.

* Corresponding author.
E-mail addresses: bekos@informatik.uni-tuebingen.de (M.A. Bekos), felicedeluca@email.arizona.edu (F. De Luca), walter.didimo@unipg.it (W. Didimo), mched@iti.uka.de (T. Mchedlidze), noellenburg@ac.tuwien.ac.at (M. Nöllenburg), symvonis@math.ntua.gr (A. Symvonis), tollis@csd.uoc.gr (I. Tollis).

https://doi.org/10.1016/j.tcs.2019.07.025



(a) (b)

Fig. 1. Two instances of our problem with fixed vertices in black and mobile vertices in white: (a) positive instance; (b) negative instance (any placement of the mobile vertices causes at least one edge crossing).

admits a planar straight-line drawing. The FM-bigraph in Fig. 1(b) has one more mobile vertex, which makes it impossible to find a planar straight-line drawing. For example, the placement of the mobile vertices showed in the figure requires at least one edge crossing, and it can be easily proven that this is true for any other placement.

Besides its intrinsic theoretical interest, our problem is motivated by the following practical scenario. Fixed vertices represent geographic locations and each mobile vertex is an attribute of one or more locations. One wants to place each mobile vertex as a label in the plane and connect it to its associated locations, while guaranteeing a "readable" layout. We interpret readability in terms of planarity and straight edges, motivated by several studies that show how edge crossings and edge bends are among the most important parameters that negatively affect the readability of a graph layout (see, e.g., [35,36]). Additional aesthetic criteria that can be studied and that are not considered in this paper are, for example, angular resolution, edge length, and drawing area (see, e.g., [11,37]).

Contribution. The main results of this paper are as follows.

i. We prove that deciding whether an FM-bigraph G admits a planar straight-line drawing is NP-hard. In a broader scenario that allows edge bends, the same hardness technique is used to prove that deciding if G admits a planar drawing with at most $k \geq 0$ bends per edge is at least as hard as deciding if a planar graph admits a planar drawing at fixed vertex locations, with at most $2k+1$ bends per edge.
ii. If all fixed vertices of G are collinear, the existence of a planar straight-line drawing of G can be tested in linear time. More in general, assuming that the fixed vertices are distributed along (i.e., covered by) a finite set of horizontal levels (lines), we can decide in linear time the existence of a planar straight-line drawing of G, in which no edge crosses a level and no mobile vertex lies on a level. Clearly, since there always exists a finite set of horizontal levels that covers all the fixed vertices, this corresponds to finding a planar straight-line drawing of G where no edge intersects a level except at its fixed vertex.
iii. Since it is difficult to discretize the problem in the general case [17,33], we investigate the scenario in which each mobile vertex is restricted to lie in the convex hull of its neighbors. This is reasonable in practice, as the user may expect that each attribute is placed in a sort of "barycentric" position with respect to its associated locations. For this scenario, we prove that testing the existence of a planar straight-line drawing is a problem in NP. With a reduction to a combinatorial problem, which is of its own independent interest but still NP-hard in its general form, we obtain polynomial-time solutions when the intersection graph of the convex hulls is a path, a cycle or, more generally, a cactus.

Notation. We assume familiarity with basic graph theoretic concepts (see, e.g., [19]). For standard definitions on *planar graphs* and *drawings*, we point the reader to [11,23]. We denote an FM-bigraph by a pair $\langle G, \phi \rangle$, where $G = (V_f, V_m, E)$ is a bipartite planar graph and $\phi : V_f \to \mathbb{R}^2$ is a function that maps each vertex $v \in V_f$ to a distinct point $p_v = \phi(v)$. We denote by n_f and n_m the number of fixed and the number of mobile vertices of G, respectively, i.e., $n_f = |V_f|$ and $n_m = |V_m|$. By n we denote the total number of vertices of G, i.e., $n = n_f + n_m$. A *planar straight-line drawing* of $\langle G, \phi \rangle$ is a geometric representation of G such that: (*i*) The vertices of G are mapped to distinct points of the plane and each vertex $v \in V_f$ is mapped to $\phi(v)$; (*ii*) the edges of G are drawn as straight lines; (*iii*) no two edges intersect except at their common end-vertices.

Structure. The remainder of the paper is structured as follows. In Section 2 we discuss the main literature related to our problem, by highlighting similarities and differences with our research. In Section 3 we prove the NP-hardness of testing the existence of planar straight-line drawings of FM-bigraphs and present a linear-time algorithm for fixed vertices on one or more parallel levels. In Section 4 we describe the results for the scenario in which each mobile vertex is constrained to lie in the convex hull of its neighbors. Conclusions and open problems are in Section 5.

2. Related work

Our problem is related to several problems addressed in the literature, but it also has substantial differences from all of them. In the following, we give a short overview of the most relevant ones.

Point labeling. A close connection is with the problem of labeling a given set of points in the plane (see, e.g., [31,39]), because mobile vertices can be regarded as labels for the fixed vertices (points). Similarly to our setting, in the *many-to-one boundary labeling* problem [3,26] each label can have multiple associated vertices and is visually connected to them by edges. However, in the boundary labeling problem, the edges are drawn as chains of horizontal and vertical segments (which may partially overlap) and all the labels are placed outside a rectangular region that encloses all the vertices. Variants of the boundary labeling problem where each fixed vertex is associated with exactly one label have also been studied (see, e.g., [2,4,25]). Note that, in the point labeling problem, labels are geometric shapes of non-empty area, while we model mobile vertices as points.

Partial drawings. Our problem is a special case of the problem of extending a partial drawing of a (not necessarily bipartite) planar graph G to a planar straight-line drawing of G. This problem is NP-hard in general [33] and polynomial-time solvable for restricted cases [9,14,20,27,38].

Point-set embedding. In a point-set embedding problem, a planar graph with n vertices must be planarly mapped onto a given set of n points, with or without a predefined correspondence between the vertices and the points (see, e.g., [1,8,12,13,24,32]). Thus, in all settings of the point-set embedding problem, each vertex can only be mapped to a finite set of points. The results in [1,32] imply that any n-vertex planar FM-bigraph admits a planar drawing in which each edge is a poly-line with $O(n)$ bends. Indeed, [1,32] prove that any n-vertex planar graph can be planarly mapped onto any set of n points, with given correspondences, using a linear number of bends per edge (which is also necessary in some cases, even for bipartite graphs). Hence, for a given FM-bigraph, one can place the mobile vertices anywhere so to realize a planar drawing, with a possibly high number of bends per edges.

Constrained drawings of bipartite graphs. Misue [29] proposed a model and a technique for drawing bipartite graphs such that the vertices of a partition set, called *anchors*, are evenly distributed on a circle. Anchors are similar to fixed vertices in our setting, but the order of the anchors in Misue's model can be freely chosen. Extensions to the 3D space and to semi-bipartite graphs have been subsequently presented [22,30]. Finally, several papers study how to draw a bipartite graph such that the vertices of each partition set are on a line or within a specific plane region (see, e.g., [5,6,16]). In these scenarios, however, the vertices do not have predefined locations.

3. NP-hardness and fixed vertices on parallel levels

In this section we first prove the hardness of deciding whether an FM-bigraph admits a planar straight-line drawing. Then, we describe a linear-time testing algorithm when the fixed vertices lie on one or more parallel levels and we look for a planar straight-line drawing where no edge intersects a level except at its fixed vertex.

NP-hardness. Our NP-hardness proof uses a reduction from the 1-bend point set embeddability with correspondence problem (or 1-BPSEWC, for short), which has been proven to be NP-hard by Goaoc et al. [17]. Problem 1-BPSEWC is defined as follows: *Given a planar graph $G = (V, E)$, a set S of $|V|$ points in the plane, and a one-to-one correspondence ζ between V and S, is there a planar drawing of G with at most one bend per edge and such that each vertex v is mapped to point $\zeta(v)$?*

Theorem 1. *Deciding whether an FM-bigraph admits a planar straight-line drawing is* NP*-hard, even if each mobile vertex has degree at most two.*

Proof. Let $\langle G = (V, E), S, \zeta\rangle$ be an instance of 1-BPSEWC. We construct an instance $\langle G' = (V_f, V_m, E'), \phi\rangle$ of our FM-bigraph problem:

- $V_f = V$, $\phi = \zeta$, and
- for each edge $e = (u, v) \in E$, define a corresponding vertex $w_e \in V_m$ and two edges (w_e, u), (w_e, v) in E'.

Clearly, $\langle G', \phi\rangle$ can be constructed in linear time. Also, G has a drawing Γ with at most one bend per edge that respects ζ if and only if G' has a planar straight-line drawing Γ' that respects ϕ. To see this, observe that the position of a bend along an edge $e = (u, v)$ of Γ corresponds to the positions of the mobile vertex w_e in Γ'; if e has no bend, w_e is drawn anywhere along the segment $\overline{uv}$. □

We observe that the reduction used in Theorem 1 can be used to prove a hardness result for the more general scenario in which we allow up to $k \geq 0$ bends per edge in a planar drawing of an FM-bigraph. Namely, this reduction can be applied without any modifications to prove that, for any $k \geq 0$, deciding whether an FM-bigraph G admits a planar drawing with at most k bends per edge is at least as difficult as the $(2k + 1)$-BPSEWC problem, where up to $(2k + 1)$ bends per edge are allowed. We summarize this observation in the following theorem.

Theorem 2. *For any given integer $k \geq 0$, deciding if an FM-bigraph admits a planar drawing with at most k bends per edge is at least as hard as the $(2k + 1)$-BPSEWC problem.*

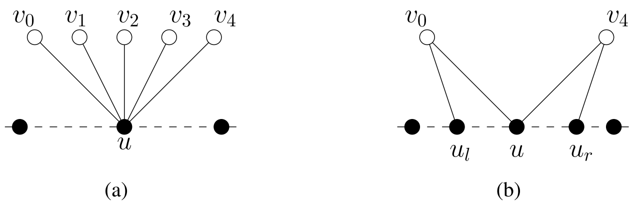


Fig. 2. (a) A star of mobile vertices of degree one attached to the same fixed vertex u. (b) The extremal vertices v_0 and $v_h = v_4$ can be made vertices of degree two by connecting them to two dummy fixed vertices u_l and u_r close to u. The other mobile vertices attached to u are temporarily removed and trivially reinserted in the drawing at the end of the drawing algorithm.

Fixed vertices on parallel lines. We first consider the special case in which all the fixed vertices of an FM-bigraph $\langle G, \phi\rangle$ are collinear. We describe a linear-time algorithm to test whether $\langle G, \phi\rangle$ admits a planar straight-line drawing and to compute such a drawing if the test is positive. The techniques used in this scenario are then extended to the case of fixed vertices on multiple horizontal lines.

Lemma 3. *Let $\langle G = (V_f, V_m, E), \phi\rangle$ be an n-vertex FM-bigraph such that all the vertices of V_f are collinear. There exists an $O(n)$-time algorithm that tests whether $\langle G, \phi\rangle$ admits a planar straight-line drawing and that computes such a drawing if the test is positive.*

Proof. Assume w.l.o.g. that all the vertices of V_f lie on a horizontal line ℓ. In any planar straight-line drawing of $\langle G, \phi\rangle$, we can assume that every vertex $w \in V_m$ is either above or below ℓ. Indeed, if w lies on ℓ then w has degree either one or two, and in the latter case it is incident to two consecutive vertices of V_f along ℓ. In fact, we can always slightly move w above or below ℓ without changing the planar embedding of the drawing. Hence, deciding whether $\langle G, \phi\rangle$ has a planar straight-line drawing is equivalent to deciding whether there exists an assignment of each mobile vertex to one of the two half planes delimited by ℓ that avoids edge crossings.

The latter problem is equivalent to testing the planarity of a graph G' obtained by augmenting G with a cycle C that connects all fixed vertices in the order they appear along the line ℓ. More precisely, a vertex inside C corresponds to a vertex above ℓ; symmetrically, a vertex outside C corresponds to a vertex below ℓ. Since the size of G' is linear in the size of G and since the graph planarity testing problem is linear-time solvable [7,21], the whole test is executed in $O(n)$ time.

If the test is positive, different techniques can be used to compute a straight-line drawing of G where each fixed vertex v is placed at point $\phi(v)$. A quadratic-time drawing technique is given in [16]; a more complex algorithm for general plane graphs with outer vertices at prescribed locations is described in [10]. Here, we give a simple algorithm that works in linear time.

Consider the planar embedding of G' determined by the planarity testing algorithm. We describe how to place the mobile vertices that are inside C, i.e., those above ℓ. The algorithm for placing the vertices that are outside C (those below ℓ) is symmetric. Denote by $G'' \subseteq G'$ the planar embedded subgraph induced by the fixed vertices and by the mobile vertices that are inside C, plus the edges in the path of C going from the leftmost fixed vertex to the rightmost fixed vertex. To ease the description, we claim that is not a loss of generality to assume that each mobile vertex is connected to at least two fixed vertices. Indeed, suppose that u is a fixed vertex and that $v_0, v_1, \ldots, v_k$ are all the mobile vertices of degree one attached to u (if any), in this clockwise order around u (refer to Fig. 2). We can always add a dummy fixed vertex u_l to the left of u such that no other fixed vertices lie between u_l and u. Symmetrically, if $v_h \neq v_0$ we can add a dummy fixed vertex u_r immediately to the right of u. Then, we planarly augment the graph by connecting u_l to v_0 and u_r to v_h. In this way, v_0 and v_h have degree two. The vertices $v_1, \ldots, v_{h-1}$ can be temporarily removed and trivially reinserted in the drawing once all the other vertices of the graph have been drawn. Observe that the number of dummy fixed vertices added in this way is linear in the number of fixed vertices of the graph.

Denote by $V_0 \subseteq V_m$ the mobile vertices on the outer face of G'' and by V_1 the mobile vertices (if any) that belong to the outer face after the removal of the vertices in V_0. More in general, for any $i = 1, \ldots, k$ for which V_i is not empty, let V_{i+1} be the mobile vertices that are on the outer face after the removal of the vertices in $V_0 \cup V_1 \cup \cdots \cup V_i$. With this procedure, k is the largest integer for which V_k is not empty, and the subsets $V_0, \ldots, V_k$ form a partition of the mobile vertices of G', which can be easily computed in $O(n)$ time by an iterative procedure. The drawing algorithm first adds to the drawing all the mobile vertices in V_0 and their incident edges, then it adds the mobile vertices in V_1 and their incident edges, and so on, until all the vertices in V_k and their incident edges have been drawn. To this aim, it first executes a plane augmentation of the embedded graph by adding a linear number of dummy edges, which will be removed from the final drawing at the end of the algorithm. The augmentation is as follows; refer to Figs. 3(a) and 3(b) for an illustration. Let v be a vertex in V_i and let u_l and u_r be the leftmost and the rightmost fixed vertices connected to v. Add a suitable set of dummy edges (dashed in the figure) that connect v to fixed vertices between u_l and u_r in such a way that the boundary of each internal face that contains v also contains at most one vertex of V_{i+1}. Clearly, the total number of these dummy edges is linear in the number of vertices of the graph. Also, these edges can be easily added in linear time by walking on the boundary of each internal face of the graph. Then, we add another type of dummy edges (dotted in the figure) in the augmentation phase. Namely, for every vertex $v \in V_i$ and for every internal face f incident to v, consider the vertex $v' \in V_{i+1}$ along the

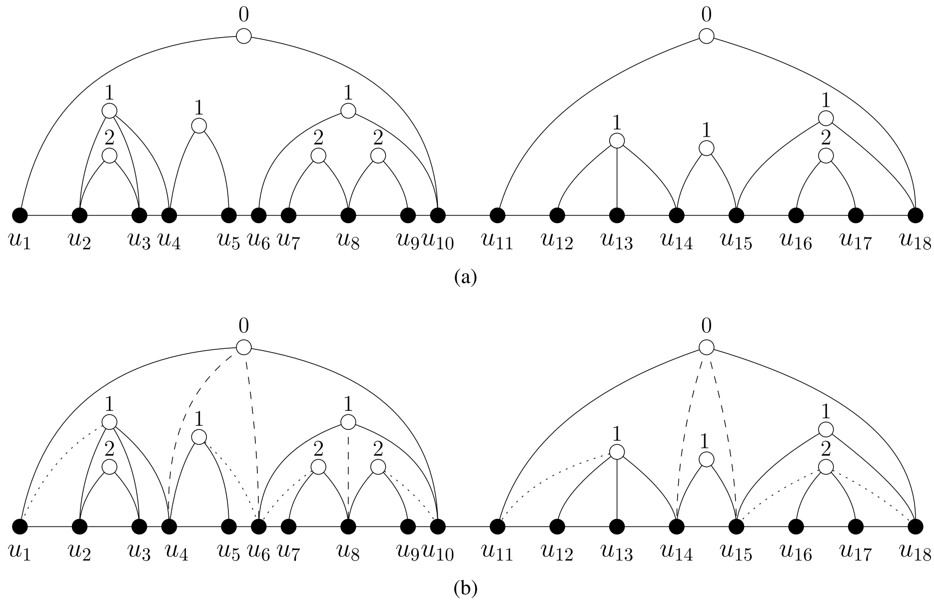


Fig. 3. Illustration plane augmentation phase in the proof of Lemma 3. (a) The planar embedded graph G'', with the fixed vertices in black and the mobile vertices in white; the label i of each mobile vertex indicates that it belongs to V_i. (b) The plane augmentation of G'' with dashed edges and the dotted edges.

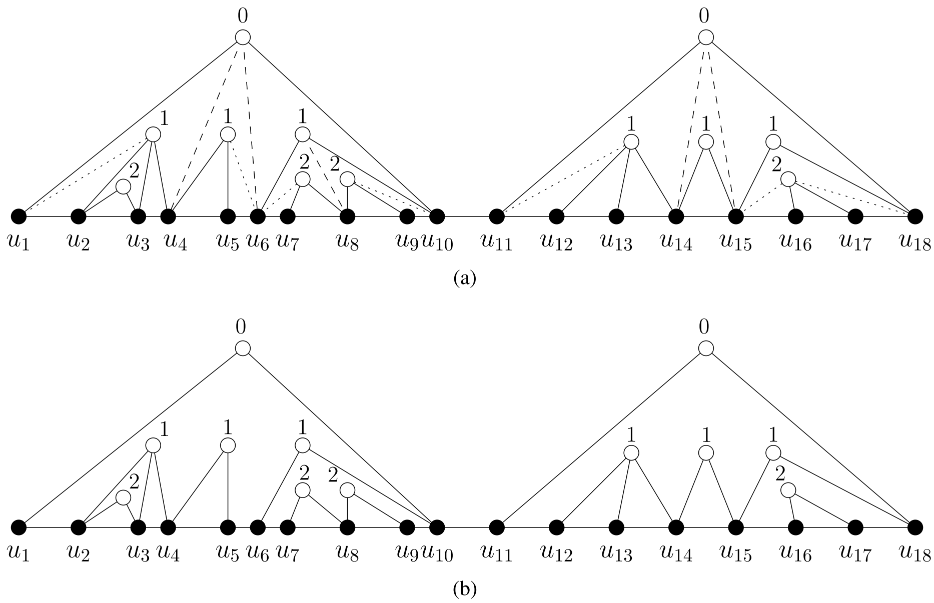


Fig. 4. Illustration of the drawing phase in the proof of Lemma 3. (a) Drawing of the plane augmented graph. (b) Drawing of G''.

boundary of f, if any (thanks to the previous augmentation, there cannot be two vertices of V_{i+1} along the boundary of f). Denote by u_l and u'_l the leftmost adjacent vertices of v and of v', respectively. Analogously, let u_r and u'_r be the rightmost adjacent vertices of v and of v', respectively. If $u_l \neq u'_l$, we add the edge (u_l, v'). Symmetrically, if $u_r \neq u'_r$, we add the edge (u_r, v'). This completes the augmentation phase.

We now describe how to construct the drawing; refer to Fig. 4(a). First notice that if v and w are any two distinct vertices in V_0, the open geometric interval determined by the leftmost and the rightmost fixed vertices connected to v is always disjoint from the open geometric interval determined by the leftmost and the rightmost fixed vertices connected to w. This induces a left-to-right order of the vertices in V_0, which corresponds to the left-to-right order of the intervals determined by their extremal adjacent fixed vertices. The vertices in V_0 can be added to the drawing at increasing arbitrary x-coordinates, according to their left-to-right order. For example, a natural (yet not restrictive) option is to add every vertex of V_0 at the

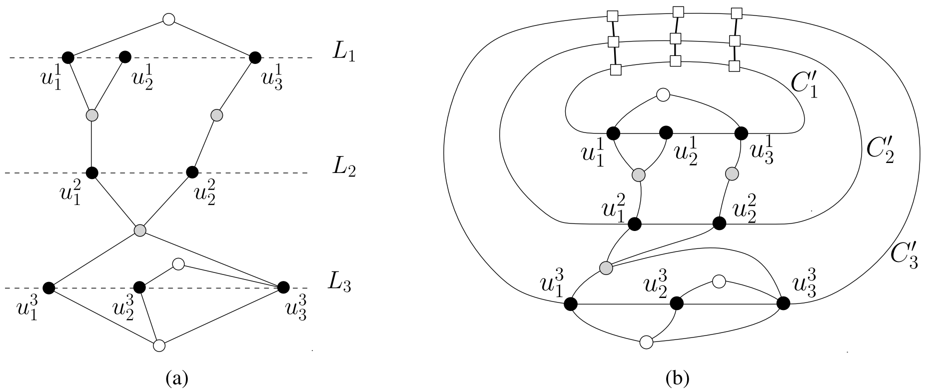


Fig. 5. Illustration for the proof of Theorem 4: (a) A leveled planar straight-line drawing. (b) The corresponding planar embedding of the graph G'.

x-coordinate of the center of its corresponding interval. Also, all the vertices in V_0 can be placed above the fixed vertices at any desired y-coordinate (all of them have the same y-coordinate). For $i = 1, \ldots, k$, the vertices of V_i are added to the drawing in any order. Let $v \in V_i$ be the next vertex to be added and let u_l and u_r be its leftmost and rightmost adjacent fixed vertices, respectively. Thanks to the plane augmentation described above, v belongs to a quadrangular face f. The vertices on the boundary f other than v are the vertices u_l and u_r, and a vertex $w \in V_{i-1}$ (whose leftmost and rightmost adjacent fixed vertices coincide with u_l and u_r, respectively). Vertex v can be added anywhere inside the triangular region formed by the vertices w, u_l, u_r. This guarantees that the edges incident to v will not cross any other edge in the drawing constructed so far. The final drawing of G'' is obtained by removing all the dummy edges added in the augmentation phase. It is clear that this drawing algorithm can be easily implemented to run in $O(n)$ time. □

Suppose now that the fixed vertices of $\langle G, \phi \rangle$ lie on $h \geq 1$ horizontal lines, which we call *levels*. We want to test whether $\langle G, \phi \rangle$ admits a planar straight-line drawing such that no edge crosses a level and no mobile vertex lies on a level. As already observed, this corresponds to finding a planar straight-line drawing of $\langle G, \phi \rangle$ where no edge intersects a level except at its fixed vertex. We call such a drawing a *leveled planar straight-line drawing* of $\langle G, \phi \rangle$. We prove the following result.

Theorem 4. *Let $\langle G = (V_f, V_m, E), \phi \rangle$ be an FM-bigraph. There exists an $O(n)$-time algorithm that test whether $\langle G, \phi \rangle$ admits a leveled planar straight-line drawing and that computes such a drawing if the test is positive.*

Proof. Denote by $\mathcal{L} = \{L_1, \ldots, L_h\}$ ($h \geq 1$) the set of levels, numbered from top to bottom, that covers all the fixed vertices of G. If $h = 1$, all the fixed vertices are collinear and the statement follows from Lemma 3. Hence, assume that $h \geq 2$. Call *white* a mobile vertex with all neighbors in the same level and *gray* a mobile vertex that is connected to two distinct levels. A gray vertex with neighbors in two consecutive levels must lie between them, while a white vertex can lie either above or below the level of its neighbors. If a gray vertex has neighbors that are not in two consecutive levels, the instance is immediately rejected.

For each level $L_i \in \mathcal{L}$, let $V_f^i = \{u_1^i, \ldots, u_{r_i}^i\}$ be the left-to-right sequence of fixed vertices on L_i. Testing whether a leveled planar straight-line drawing of $\langle G, \phi \rangle$ exists generalizes the testing algorithm for collinear fixed vertices. We reduce the problem to testing planarity for a graph G' suitably defined by augmenting G. Namely, for each L_i, add a cycle C_i connecting all the vertices of V_f^i in their left-to-right order; then, subdivide edge $(u_1^i, u_{r_i}^i)$ of C_i with three dummy vertices v_1^i, v_2^i, v_3^i, in this order from u_1^i to $u_{r_i}^i$, and call C_i' the subdivision of C_i; finally, for each $i = 1, \ldots, h-1$ and $j = 1, 2, 3$, connect v_j^i to v_j^{i+1}. See Fig. 5 for an illustration; in the figure, the dummy vertices v_j^i are represented by small squares. In a planar embedding of G', a vertex inside C_i' will be drawn above L_i in the leveled planar straight-line drawing, while a vertex outside C_i' will be drawn below L_i ($1 \leq i \leq h$). Since there are $3h$ dummy vertices and $h = O(n)$, the test is executed in $O(n)$ time.

If the test is positive, a leveled planar straight-line drawing of $\langle G, \phi \rangle$ can be computed by applying a technique similar to the one described in the proof of Lemma 3, level by level from top to bottom. More precisely, we execute h steps: At step $i = 1, \ldots, h$ we place all the mobile vertices connected to level L_i that have not been placed yet. For $1 \leq i \leq h$, denote by ${G_i'}^A$ (resp. ${G_i'}^B$) the plane graph induced by the mobile vertices connected to i that are above (resp. below) L_i, plus the path connecting all the fixed vertices of L_i. Note that, by definition, the gray vertices of ${G_i'}^B$ coincide with the gray vertices of ${G_{i+1}'}^A$; all these vertices are on the outer face of both ${G_i'}^B$ and ${G_{i+1}'}^A$. The mobile vertices connected to L_1 are placed as described in Lemma 3, in such a way that those of them that are on the outer face of ${G_1'}^B$ receive an arbitrary y-coordinate between the y-coordinates of L_1 and L_2. For any successive step $i \geq 2$, the mobile vertices connected to L_i are placed with the algorithm given in Lemma 3, in such a way that the (outer) mobile vertices of ${G_i'}^A \cap {G_{i-1}'}^B$ keep the positions

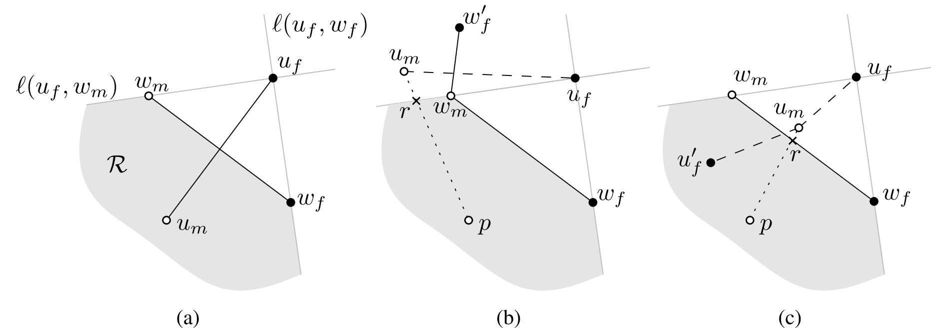


Fig. 6. Illustration for the proof of Lemma 5. In figures (b) and (c) part of the line T between r and r' is drawn dotted.

already assigned in the previous step (the remaining vertices of ${G'_i}^A$ are placed at the same y-coordinate and at arbitrary x-coordinates that respect the left-to-right order defined in Lemma 3) and, if $i < h$, the outer mobile vertices of ${G'_i}^B$ receive an arbitrary y-coordinate between the y-coordinates of L_i and L_{i+1}. Since for every step $i = 1, \ldots, h$ the algorithm takes a time that is linear in the number of vertices of ${G'_i}^A \cup {G'_i}^B$, the whole time complexity is $O(n)$. □

4. Mobile vertices at internal positions

In this section, we focus on drawings, in which the fixed vertices are in general position and each mobile vertex lies in the convex hull of its (fixed) neighbors. For the sake of simplicity, we refer to such drawings as *convex-hull drawings*.

Before we proceed with the description of our approach, we first introduce some necessary definitions. Let $u_m \in V_m$ be a mobile vertex. With slight abuse of notation, we denote by $CH(u_m)$ the convex hull of the neighbors of u_m. Let $\mathcal{A} = \mathcal{A}(V_f)$ be the arrangement of lines defined by all pairs of fixed points; see Fig. 7(a). It is not difficult to see that $\mathcal{A}$ has $O(n_f^2)$ lines, which define $O(n_f^4)$ cells (see also [18]). The following lemma allows us to discretize the set of possible positions for the mobile vertices. In particular, it implies that all positions of a mobile vertex u_m in the same cell of $\mathcal{A}$ within $CH(u_m)$ are equivalent for a planar straight-line drawing of $\langle G, \phi \rangle$.

Lemma 5. *Let $\langle G = (V_f, V_m, E), \phi \rangle$ be an FM-bigraph. Let $u_m \in V_m$ be a mobile vertex, and C a cell of arrangement $\mathcal{A} = A(V_f)$ inside $CH(u_m)$. For a pair of points p and p' in cell C, let Γ be a planar straight-line drawing of $\langle G, \phi \rangle$ where u_m is at point p and let Γ' be a planar straight-line drawing of $\langle G, \phi \rangle$ obtained from Γ by only moving u_m from point p to point p'. Then, Γ' is planar if and only if Γ is planar.*

Proof. For a proof by contradiction assume that drawing Γ' is planar, while drawing Γ is non-planar; the proof for the other direction is symmetric. Since Γ' is planar, while Γ is non-planar, it follows that moving u_m along the straight line T from p' to p, at some point we get a crossing along one of the edges incident to u_m. Since cell C is convex, clearly T lies in C. Let r be the point of T closest to p', such that placing u_m at r causes such a crossing. By definition of point r, it follows that placing u_m on any point between r and p' implies no crossing.

Let (u_m, u_f) be an edge crossed by some other edge (w_m, w_f), when u_m is placed at r. Assume w.l.o.g. that w_m is a mobile vertex, w_f is a fixed vertex, and that u_m lies to the right of the oriented edge (w_m, w_f); see Fig. 6(a). Denote by $\ell(u_f, w_f)$ the line through u_f and w_f and by $\ell(u_f, w_m)$ the line through u_f and w_m. Let $\mathcal{R}$ be the region that contains u_m and is delimited by lines $\ell(u_f, w_f)$, $\ell(u_f, w_m)$ and the edge (w_m, w_f). Let r' be a point of T lying between r and p'. Notice that r' has to lie outside of $\mathcal{R}$. Thus, T crosses the boundary of region $\mathcal{R}$. Let us additionally assume that r' lies arbitrarily close to the boundary of $\mathcal{R}$. We distinguish three cases, based on whether T crosses $\ell(u_f, w_f)$, $\ell(u_f, w_m)$, or the edge (w_m, w_f).

Case 1. *T crosses $\ell(u_f, w_f)$.* Since line $\ell(u_f, w_f)$ is part of $\mathcal{A}$, r' lies outside C. However, this is a contradiction to the assumption that T lies in C.

Case 2. *T crosses $\ell(u_f, w_m)$*; see Fig. 6(b) for an illustration. Since w_m is in the convex hull of its neighbors, there is an edge (w_m, w'_f), such that w_f and w'_f are on different sides of the line $\ell(u_f, w_m)$. Placing u_m at r' yields a crossing with (w_m, w'_f), as r' is arbitrarily close to $\ell(u_f, w_m)$. This is a contradiction to the choice of r.

Case 3. *T crosses (w_m, w_f)*; for an illustration refer to Fig. 6(c). Since u_m lies in the convex hull of its neighbors, there is an edge (u_m, u'_f), such that u'_f and u_f are on different sides of the line through edge (w_f, w_m). Placing u_m at r' would introduce a crossing between (u_m, u'_f) and (w_m, w_f), as r' lies arbitrarily close to (w_m, w_f). This again contradicts the choice of r. □

Please cite this article in press as: M.A. Bekos et al., Planar drawings of fixed-mobile bigraphs, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2019.07.025

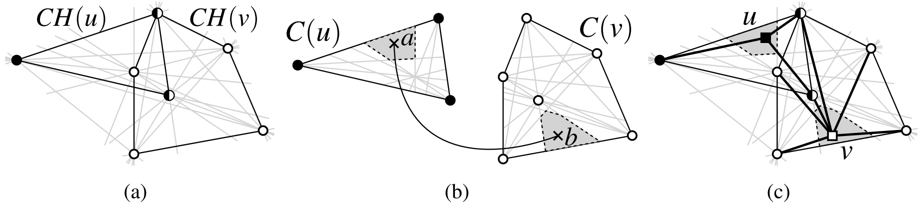


Fig. 7. (a) Line arrangement $\mathcal{A} = \mathcal{A}(V_f)$ with the neighbors $N(u)$ in black and $N(v)$ in white (the vertices in $N(u) \cap N(v)$ are colored black-and-white); $CH(u)$ and $CH(v)$ intersect and thus form an edge in G_x. (b) Two clusters $C(u)$ and $C(v)$ of G_c with one exemplary edge between two cell vertices a and b. (c) Placing u inside cell(a) and v inside cell(b) yields a planar drawing of the FM-bigraph (thick edges); u is colored black and v is colored white, consistently with the colors given to their neighbors.

Lemma 5 directly implies that testing if an FM-bigraph admits a planar straight-line drawing is a problem in NP for convex-hull drawings.[1] More precisely, a non-deterministic algorithm guesses an assignment of the mobile vertices to the $O(n_f^4)$ cells and, since G is planar, checks in $O(n_f^2)$ time whether the corresponding straight-line drawing is planar (note that $n_m = O(n_f)$). We summarize this observation in the following theorem.

Theorem 6. *Testing whether an FM-bigraph admits a planar straight-line drawing is a problem in* NP *if each mobile vertex must lie in the convex hull of its neighbors.*

In the remainder of this section, we prove that the problem is in fact in P for certain input configurations. Central ingredients in our approach are the CH intersection graph G_x of G, the cell graph G_c of G, and the skeleton graph G_s of G_c, which we formally define in the following.

The *CH intersection graph* G_x is defined as the intersection graph [28] of the convex hulls of the neighbors of the mobile vertices. Formally, (i) for each mobile vertex u, graph G_x has a vertex associated with $CH(u)$, and (ii) for any two distinct mobile vertices u and v, graph G_x has an edge between the vertices corresponding to $CH(u)$ and $CH(v)$ if and only if $CH(u)$ and $CH(v)$ overlap, that is, $CH(u) \cap CH(v) \neq \emptyset$.

The *cell graph* G_c is a clustered graph (see, e.g., Eades and Feng [15]), which is defined as follows; see Fig. 7(b) for an illustration. Each mobile vertex u is associated with a cluster $C(u)$; the vertices of $C(u)$, called *cell vertices*, are the cells of $\mathcal{A}$ that intersect with $CH(u)$ (and in fact are contained in $CH(u)$). The vertices of G_c are defined by the disjoint union of the vertices of all clusters,[2] that is,

$$V(G_c) = \uplus_{u \in V_m} C(u).$$

For a cell vertex a of G_c, we denote by cell(a) the cell corresponding to a in $\mathcal{A}$. For a pair of mobile vertices u and v such that $CH(u) \cap CH(v) \neq \emptyset$, a cell vertex $a \in C(u)$ is adjacent to a cell vertex $b \in C(v)$ if and only if placing u in cell(a) and v in cell(b) produces no crossing among the edges incident to u and v; see Fig. 7(c). Note that graph G_c has $O(n_f^4 n_m)$ vertices and $O(n_f^8 n_m^2)$ edges. Also, by definition, any two mobile vertices u and v, such that $CH(u) \cap CH(v) \neq \emptyset$, can be positioned within their convex hulls without creating edge crossings if and only if there exist two adjacent cell vertices $a \in C(u)$ and $b \in C(v)$ in G_c.

Finally, the *skeleton graph* G_s is created by selecting exactly one cell vertex, called a *skeleton vertex*, from each cluster of G_c, such that for every pair of mobile vertices u and v with $CH(u) \cap CH(v) \neq \emptyset$, the skeleton vertices of $C(u)$ and $C(v)$ are adjacent in G_c. Graph G_s is the subgraph of G_c induced by the skeleton vertices. Note that G_s might not exist. If G_s exists, then it is isomorphic to G_x. The following characterization is an immediate consequence of our definitions.

Lemma 7. *An FM-bigraph $\langle G, \phi \rangle$ admits a planar straight-line convex-hull drawing if and only if cell graph G_c has a skeleton.*

Proof. For the forward direction, observe that a planar straight-line drawing immediately defines a skeleton. Conversely, if graph G_c has a skeleton G_s, a planar straight-line drawing Γ of $\langle G, \phi \rangle$ is obtained by placing each mobile vertex u in the cell corresponding to the skeleton vertex of $C(u)$ in G_s. Since crossings may only occur between edges incident to mobile vertices u and v such that $CH(u) \cap CH(v) \neq \emptyset$, the obtained drawing Γ is planar. □

The characterization of Lemma 7 allows us to translate the geometric problem of finding a straight-line convex-hull drawing of an FM-bigraph $\langle G, \phi \rangle$ to a purely combinatorial problem on a support clustered graph G_c constructed from

[1] We remark that in a preliminary version of [33], it is claimed membership in NP for the partial planarity extension problem [34], which would imply membership in NP also for our problem. That claim, however, lacks a proof in [34] and the author was only able to prove the NP-hardness of the problem in [33] (personal communication).

[2] Note that cells in the intersection of two convex hulls correspond to different vertices of G_c.

$\langle G, \phi \rangle$. Unfortunately, however, this combinatorial problem is NP-hard in its general form, as proved in Theorem 8. Note that, the graphs G_x and G_c in the statement of Theorem 8 are algebraic graphs, not associated with a geometry; we call *cells* the vertices of G_c to easily maintain correspondence with the geometric setting previously defined.

Theorem 8. *Let $G_\mathsf{x} = (\mathcal{C}, \mathcal{E})$ be a graph, where $\mathcal{C}$ is a set of disjoint clusters of cells. Also, let $G_\mathsf{c} = (V, E)$ be a graph, where each vertex v of G_c is a cell of a cluster $C(v)$ in $\mathcal{C}$ and an edge (u, v) belongs to G_c only if $C(u)$ and $C(v)$ are adjacent in G_x. It is* NP*-complete to test if there is a subset $V' \subseteq V$ of skeleton vertices, containing exactly one cell from each cluster in $\mathcal{C}$ such that the induced subgraph $G_\mathsf{c}[V']$ is isomorphic to G_x.*

Proof. The problem is clearly in NP. The proof of the hardness is by reduction from the well-known 3SAT problem. Starting from a boolean 3SAT formula ψ, we create a cluster $C(x)$ for each variable x in ψ and a cluster $C(\gamma)$ for each clause γ of ψ. In G_x each clause cluster is adjacent to the three clusters of the variables occurring in the clause. Each variable cluster $C(x)$ consists of two cells in G_c, one for the positive literal x and one for its negation $\neg x$. Also, each clause cluster $C(\gamma)$ contains three cells, one for each literal. Finally, connect each literal cell of a clause λ to the corresponding cell of its variable cluster and to all four cells of the other two variables of γ.

We now show that ψ has a satisfying truth assignment if and only if there exists a subset of skeleton vertices in G_c that induces a subgraph isomorphic to G_x. Assume that we know a satisfying variable assignment of ψ. We select the "*true*" cells of the variable clusters and one satisfied literal of each clause. The induced subgraph of this set of cells is isomorphic to G_x as the satisfied literal cell of each clause γ covers all three edges of $C(\gamma)$ to its three adjacent variable clusters. Conversely, if we have a subset of skeleton vertices of G_c that induce a subgraph isomorphic to G_c, then setting the literals of the set of selected literal cells to *true* satisfies ψ. Otherwise, in an unsatisfied clause, none of the three clause cells would connect to the selected cells of all three adjacent variable clusters, which contradicts the skeleton property. □

Assuming that P $\neq$ NP and given the fact that the graph constructed in the reduction proof of Theorem 8 is bipartite, there can be no polynomial-time algorithm for the case where the intersection graph G_x is a general bipartite graph. Nonetheless, we are able to solve this more general combinatorial problem efficiently when G_x is a cactus (refer to Theorem 10), which includes the special cases in which G_x is a cycle or a tree. It is an interesting open question to investigate the computational complexity of the original problem of deciding whether an FM-bigraph admits a planar straight-line convex-hull drawing, when the underlying *geometry* of the problem is taken into account. The next two lemmas are base cases for Theorem 10.

Lemma 9. *Let $\langle G, \phi \rangle$ be an FM-bigraph, whose CH intersection graph G_x is a path or a cycle. There exists a polynomial-time algorithm that tests whether $\langle G, \phi \rangle$ has a planar straight-line convex-hull drawing.*

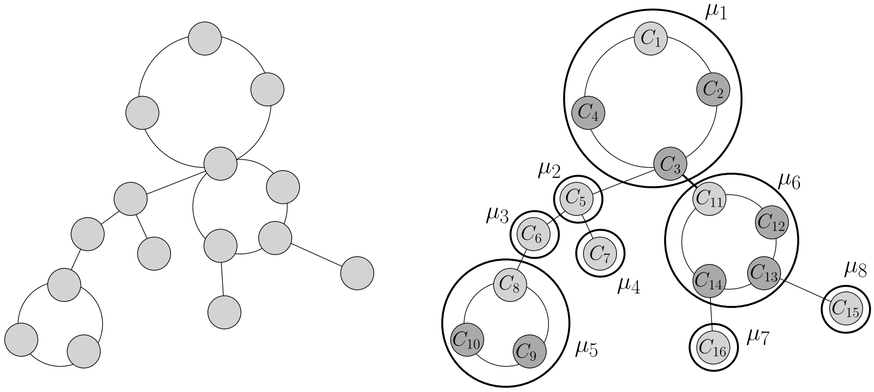
Proof. We first discuss the case of a path G_x and then extend the idea to a cycle. So let G_x be a path. By Lemma 7, it is enough to test whether G_c has a skeleton. Let $u_1, \ldots, u_\lambda$ be the mobile vertices in the order their convex hulls appear along path G_x. Call a cell vertex a of $C(u_i)$ *active* if and only if the subgraph of G_c induced by $C(u_1) \cup \cdots \cup C(u_i)$ has a skeleton containing a, where $1 \leq i \leq \lambda$. Thus, G_c has a skeleton if and only if there is an active cell vertex in $C(u_\lambda)$.

A simple algorithm that tests this condition works as follows. Initially, mark all cell vertices of $C(u_1)$ as active, and then *propagate* this information forward to the cell vertices of $C(u_\lambda)$ as follows. For each $i = 2, \ldots, \lambda$, mark each cell vertex of $C(u_i)$ as active if it has an active neighbor in $C(u_{i-1})$. The time complexity of this simple algorithm is bounded by the number of vertices and edges in G_c, which is polynomial in the number of vertices of G.

Consider now the case where G_x is a cycle. The main difference to the case of a path is that, for the existence of a skeleton, we need to find an active cell vertex in $C(u_\lambda)$ from which we can "close the cycle" to a compatible active cell vertex in $C(u_1)$. In order to do so, when we do the forward propagation for $i = 2, \ldots, \lambda$, we additionally keep track of all possible *originating* cell vertices from $C(u_1)$ for each cell vertex of $C(u_i)$. More precisely, the set of originating vertices of each cell vertex of $C(u_1)$ is by definition a singleton containing the cell vertex itself. For $i = 2, \ldots, \lambda$, the set of originating vertices of each cell vertex v in $C(u_i)$ is the union of the sets of originating vertices of the active neighbors of v in $C(u_{i-1})$. So when we reach $C(u_\lambda)$ we still know from which active cell vertices of $C(u_1)$ a particular cell vertex of $C(u_\lambda)$ can be reached in a skeleton of the path from u_1 to u_λ. In the last step we close the cycle and mark each cell vertex w in $C(u_1)$ as *confirmed*, if w has an active neighbor v in $C(u_\lambda)$ *and* w appears in the set of originating cell vertices of v. Now G_c has a skeleton if and only if we marked at least one vertex in $C(u_1)$ as confirmed. The book keeping of the originating vertices adds at most a factor that is linear in the number of cells of $C(u_1)$ to the running time.

Note that if the test is positive, a skeleton of G_c can be easily constructed as follows. Initially, we add to the skeleton of G_c an active cell vertex α_λ of $C(u_\lambda)$, which has in its set of originating cell vertices at least one confirmed neighbor in $C(u_1)$. Then, for each $i = \lambda - 1, \ldots, 1$, we add to the skeleton of G_c a cell vertex α_i, such that α_i is in the cell of originated vertices of α_{i+1}. By the choice of the initial cell vertex α_λ, it follows that α_1 must be a confirmed cell vertex of $C(u_1)$. Therefore, the reported cell vertices form indeed a skeleton of G_c. □

We now extend the previous technique to the case in which G_x is a cactus, which also covers the case of a tree. We recall that a *cactus* is a connected graph in which any two simple cycles share at most one vertex. A cactus is an outerplanar



(a) G_{x} (b) $\mathcal{T}$

Fig. 8. (a) An intersection graph G_{x} that is a cactus. (b) The decomposition tree $\mathcal{T}$ of G_{x}. Clusters C_{11} of μ_6 and C_3 of μ_1 correspond to the same vertex of G_{x}.

graph and can always be decomposed into a tree where each node corresponds to either a single vertex or a simple cycle (refer to Fig. 8(a) for an illustration).

Theorem 10. *Let $\langle G, \phi \rangle$ be an FM-bigraph, whose CH intersection graph G_{x} is a cactus. There exists a polynomial-time algorithm that tests whether $\langle G, \phi \rangle$ has a planar straight-line convex-hull drawing.*

Proof. By Lemma 9, the statement holds when G_{x} is a path or a cycle. In the general case, our testing algorithm decomposes G_{x} into its tree $\mathcal{T}$ (as in Fig. 8(b)), roots $\mathcal{T}$ at any node, and visits $\mathcal{T}$ bottom-up. More precisely, each vertex of G_{x} corresponds to a convex hull $CH(u)$ of a mobile vertex u, and it has a one-to-one correspondence with a cluster $C(u)$ of G_{c}. Thus, each node μ of $\mathcal{T}$ corresponds to either a single cluster of G_{c} or to a cycle of clusters of G_{c}. Note that when in G_{x} two cycles share a vertex (cluster of G_{c}), we replicate such a vertex in both nodes of $\mathcal{T}$ that correspond to the two cycles. For example, in Fig. 8(b) cluster C_{11} inside μ_6 and cluster C_3 inside μ_1 correspond to the same vertex of G_{x}. Once the root of $\mathcal{T}$ has been chosen, we define the *anchor* of μ as the cluster that connects μ to its parent node in $\mathcal{T}$ (refer to the light-gray colored clusters in Fig. 8(b)). During the bottom-up visit of $\mathcal{T}$, two cases are possible when a node μ is visited:

- ***μ* is a leaf**. If μ contains a single cluster (i.e., its anchor), then all its cell vertices are marked as confirmed; if μ contains a cycle of clusters, then the confirmed cell vertices of its anchor are computed as for $C(u_1)$ in the proof of Lemma 9.
- ***μ* is an internal node**. Let $\nu_1, \nu_2, \dots, \nu_k$ be the children of μ in $\mathcal{T}$ and denote by C_{q_i} the cluster of μ connected to the anchor of ν_i. Note that C_{q_i} may coincide with some C_{q_j} if $q_i \neq q_j$. Also, the anchor of ν_i and C_{q_i} may correspond to the same vertex of G_{x}.
  - For each $i = 1, \dots, k$, if the anchor of ν_i differs from C_{q_i} in G_{x}, remove from C_{q_i} all cell vertices that are not connected to a confirmed cell vertex of the anchor of ν_i in G_{c}, as they cannot occur in any skeleton of G_{c}. On the other hand, if the anchor of ν_i and C_{q_i} coincide in G_{x}, remove from C_{q_i} all vertices of the anchor of ν_i that are not marked as confirmed.
  - Now, if μ contains a single cluster (i.e., its anchor), then all its remaining cell vertices are marked as confirmed; if μ contains a cycle of clusters, then the confirmed cell vertices of its anchor are computed as in Lemma 9. At this point, if the anchor of μ contains no confirmed vertex, the algorithm stops and the instance is rejected, as a skeleton does not exist.

Once the bottom-up visit of $\mathcal{T}$ ends, the test is positive if and only if the anchor of the root node of $\mathcal{T}$ has a confirmed cell vertex w, and in this case one can reconstruct a skeleton of G_{c} starting from w and visiting $\mathcal{T}$ top-down. In particular, during the top-down visit, for each node μ of $\mathcal{T}$, any confirmed vertex in the anchor of μ can be arbitrarily selected, as it is connected to the parent node of μ by construction. Also, if μ corresponds to a simple cycle of clusters, the construction of a cycle that connects these clusters is done as in Lemma 9.

Concerning the time complexity, the above algorithm takes polynomial time in the size of G_{c}. Indeed, the number of clusters that may occur in multiple nodes of $\mathcal{T}$ (i.e., those that are shared by multiple cycles of clusters) is at most the number of cycles in G_{x}. Therefore, the total number of clusters over all nodes of $\mathcal{T}$ is linear in the number of clusters of G_{c}. This also implies that the total number of cell vertices over all clusters of $\mathcal{T}$ is linear in the number of cell vertices in G_{c}. Finally, each node μ of $\mathcal{T}$ is visited twice (once in the bottom-up visit and once in the top-down visit), and in each visit of μ the algorithm has a running time that is polynomial in the number of cell vertices in the clusters of μ. $\square$

5. Conclusions and open problems

In this paper, we initiated the study of FM-bigraphs, showed that it is NP-hard in the general case to test if they admit a planar straight-line drawing, and gave polynomial-time algorithms in some interesting restricted cases. In the following, we list several interesting open research questions:

Q1. We solved the problem for convex-hull drawings if the CH intersection graph is a cactus and showed that it is NP-complete in a non-geometric setting. Can we solve the problem for larger classes of convex-hull drawings in polynomial time (e.g., when the CH intersection graph is a triconnected planar graph) or extend the NP-completeness to our geometric setting? More generally, for which other layout constraints or sub-families of FM-bigraphs does the problem become tractable?

Q2. Our focus was on proving the existence of polynomial-time algorithms under certain layout constraints, but some of the algorithms have high time complexity. Thus, finding more efficient algorithms is of interest.

Q3. We focused on crossing-free drawings of FM-bigraphs. Relaxing the planarity requirement (e.g., for a given maximum number of permitted crossings per edge) is an interesting variant, as well as, designing heuristics or exact approaches for crossing/bend minimization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Research in this work started at the Bertinoro Workshop on Graph Drawing 2016. We thank all the participants and in particular Seok-Hee Hong for useful discussions. We also thank the anonymous reviewers of this paper for their valuable comments. The research in this paper is partially supported by the NSF grant DMS-1839274 and it is also part of the Italian MIUR Project “Smartour” SCN_166.

References

[1] M. Badent, E. Di Giacomo, G. Liotta, Drawing colored graphs on colored points, Theor. Comput. Sci. 408 (2–3) (2008) 129–142.
[2] L. Barth, A. Gemsa, B. Niedermann, M. Nöllenburg, On the readability of boundary labeling, in: E. Di Giacomo, A. Lubiw (Eds.), Graph Drawing and Network Visualization (GD'15), in: LNCS, vol. 9411, Springer International Publishing, 2015, pp. 515–527.
[3] M.A. Bekos, S. Cornelsen, M. Fink, S. Hong, M. Kaufmann, M. Nöllenburg, I. Rutter, A. Symvonis, Many-to-one boundary labeling with backbones, J. Graph Algorithms Appl. 19 (3) (2015) 779–816.
[4] M.A. Bekos, M. Kaufmann, A. Symvonis, A. Wolff, Boundary labeling: models and efficient algorithms for rectangular maps, Comput. Geom. 36 (3) (2007) 215–236.
[5] T.C. Biedl, Drawing planar partitions I: LL-drawings and LH-drawings, in: R. Janardan (Ed.), Computational Geometry (SoCG'98), ACM, 1998, pp. 287–296.
[6] T.C. Biedl, M. Kaufmann, P. Mutzel, Drawing planar partitions II: HH-drawings, in: J. Hromkovic, O. Sýkora (Eds.), Graph-Theoretic Concepts in Computer Science (WG'98), in: LNCS, vol. 1517, Springer, 1998, pp. 124–136.
[7] K.S. Booth, G.S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms, J. Comput. Syst. Sci. 13 (3) (1976) 335–379.
[8] U. Brandes, C. Erten, A. Estrella-Balderrama, J.J. Fowler, F. Frati, M. Geyer, C. Gutwenger, S. Hong, M. Kaufmann, S.G. Kobourov, G. Liotta, P. Mutzel, A. Symvonis, Colored simultaneous geometric embeddings and universal pointsets, Algorithmica 60 (3) (2011) 569–592.
[9] Erin W. Chambers, David Eppstein, Michael T. Goodrich, Maarten Löffler, Drawing graphs in the plane with a prescribed outer face and polynomial area, in: Graph Drawing, 2010, pp. 129–140.
[10] E.W. Chambers, D. Eppstein, M.T. Goodrich, M. Löffler, Drawing graphs in the plane with a prescribed outer face and polynomial area, J. Graph Algorithms Appl. 16 (2) (2012) 243–259.
[11] G. Di Battista, P. Eades, R. Tamassia, I.G. Tollis, Graph Drawing: Algorithms for the Visualization of Graphs, Prentice-Hall, 1999.
[12] E. Di Giacomo, W. Didimo, G. Liotta, H. Meijer, F. Trotta, S.K. Wismath, k-Colored point-set embeddability of outerplanar graphs, J. Graph Algorithms Appl. 12 (1) (2008) 29–49.
[13] E. Di Giacomo, G. Liotta, F. Trotta, Drawing colored graphs with constrained vertex positions and few bends per edge, Algorithmica 57 (4) (2010) 796–818.
[14] C.A. Duncan, M.T. Goodrich, S.G. Kobourov, Planar drawings of higher-genus graphs, J. Graph Algorithms Appl. 15 (1) (2011) 7–32.
[15] P. Eades, Q.-W. Feng, Multilevel visualization of clustered graphs, in: S. North (Ed.), Graph Drawing (GD'96), in: LNCS, vol. 1190, Springer-Verlag, 1997, pp. 101–112.
[16] U. Fößmeier, M. Kaufmann, Nice drawings for planar bipartite graphs, in: G.C. Bongiovanni, D.P. Bovet, G. Di Battista (Eds.), Algorithms and Complexity (CIAC'97), in: LNCS, vol. 1203, Springer, 1997, pp. 122–134.
[17] X. Goaoc, J. Kratochvíl, Y. Okamoto, C. Shin, A. Spillner, A. Wolff, Untangling a planar graph, Discrete Comput. Geom. 42 (4) (2009) 542–569.
[18] D. Halperin, Arrangements, in: J.E. Goodman, J. O'Rourke (Eds.), Handbook of Discrete and Computational Geometry, CRC Press LLC, Boca Raton, FL, 2004, pp. 529–562 (Chapter 24).
[19] F. Harary, Graph Theory, Addison-Wesley, Reading, MA, 1972.
[20] S.-H. Hong, H. Nagamochi, Convex drawings of graphs with non-convex boundary constraints, Discrete Appl. Math. 156 (12) (2008) 2368–2380.
[21] J.E. Hopcroft, R.E. Tarjan, Efficient planarity testing, J. ACM 21 (4) (1974) 549–568.
[22] T. Ito, K. Misue, J. Tanaka, Sphere anchored map: a visualization technique for bipartite graphs in 3D, in: J.A. Jacko (Ed.), Human-Computer Interaction (HCI'09), in: LNCS, vol. 5611, Springer, 2009, pp. 811–820.

[23] M. Kaufmann, D. Wagner (Eds.), Drawing Graphs, Methods and Models, LNCS, vol. 2025, Springer, 2001.
[24] M. Kaufmann, R. Wiese, Embedding vertices at points: few bends suffice for planar graphs, J. Graph Algorithms Appl. 6 (1) (2002) 115–129.
[25] P. Kindermann, B. Niedermann, I. Rutter, M. Schaefer, A. Schulz, A. Wolff, Multi-sided boundary labeling, Algorithmica 76 (1) (2016) 225–258.
[26] C. Lin, Crossing-free many-to-one boundary labeling with hyperleaders, in: IEEE Pacific Visualization Symposium (PacificVis'10), IEEE Computer Society, 2010, pp. 185–192.
[27] T. Mchedlidze, M. Nöllenburg, I. Rutter, Extending convex partial drawings of graphs, Algorithmica 76 (1) (2016) 47–67.
[28] T.A. McKee, F.R. McMorris, Topics in Intersection Graph Theory, SIAM Monographs on Discrete Mathematics and Applications, 1999.
[29] K. Misue, Anchored map: graph drawing technique to support network mining, IEICE Trans. 91-D (11) (2008) 2599–2606.
[30] K. Misue, Q. Zhou, Drawing semi-bipartite graphs in anchor+matrix style, in: Information Visualisation (IV'11), IEEE Computer Society, 2011, pp. 26–31.
[31] G. Neyer, Map labeling with application to graph drawing, in: M. Kaufmann, D. Wagner (Eds.), Drawing Graphs, Methods and Models, in: LNCS, vol. 2025, Springer, 1999, pp. 247–273.
[32] J. Pach, R. Wenger, Embedding planar graphs at fixed vertex locations, Graphs Comb. 17 (4) (2001) 717–728.
[33] M. Patrignani, On extending a partial straight-line drawing, Int. J. Found. Comput. Sci. 17 (5) (2006) 1061–1070.
[34] M. Patrignani, On extending a partial straight-line drawing, in: P. Healy, N.S. Nikolov (Eds.), Graph Drawing (GD'05), in: LNCS, vol. 3843, Springer, 2006, pp. 380–385.
[35] H.C. Purchase, Metrics for graph drawing aesthetics, J. Vis. Lang. Comput. 13 (5) (2002) 501–516.
[36] H.C. Purchase, D.A. Carrington, J. Allder, Empirical evaluation of aesthetics-based graph layout, Empir. Softw. Eng. 7 (3) (2002) 233–255.
[37] R. Tamassia (Ed.), Handbook on Graph Drawing and Visualization, Chapman and Hall/CRC, 2013.
[38] W.T. Tutte, How to draw a graph, Proc. Lond. Math. Soc. 13 (3) (1963) 743–768.
[39] A. Wolff, T. Strijk, The map-labeling bibliography, http://i11www.ira.uka.de/map-labeling/bibliography, 1996.

Please cite this article in press as: M.A. Bekos et al., Planar drawings of fixed-mobile bigraphs, Theoret. Comput. Sci. (2019), https://doi.org/10.1016/j.tcs.2019.07.025