On the Complexity of Recognizing k^+-Real Face Graphs

Michael A. Bekos
University of Ioannina, Greece

Giuseppe Di Battista
University of Roma Tre, Italy

Emilio Di Giacomo
University of Perugia, Italy

Walter Didimo
University of Perugia, Italy

Michael Kaufmann
University of Tübingen, Germany

Fabrizio Montecchiani
University of Perugia, Italy

Abstract

A nonplanar drawing Γ of a graph G divides the plane into topologically connected regions, called *faces* (or *cells*). The boundary of each face is formed by vertices, crossings, and edge segments. Given a positive integer k, we say that Γ is a k^+*-real face drawing* of G if the boundary of each face of Γ contains at least k vertices of G. The study of k^+-real face drawings started in a paper by Binucci et al. (WG 2023), where edge density bounds and relationships with other beyond-planar graph classes are proved. In this paper, we investigate the complexity of recognizing k^+-real face graphs, i.e., graphs that admit a k^+-real face drawing. We study both the general unconstrained scenario and the 2-layer scenario in which the graph is bipartite, the vertices of the two partition sets lie on two distinct horizontal layers, and the edges are straight-line segments. We give NP-completeness results for the unconstrained scenario and efficient recognition algorithms for the 2-layer setting.

2012 ACM Subject Classification Mathematics of computing → Graph algorithms; Mathematics of computing → Graph theory; Theory of computation → Graph algorithms analysis

Keywords and phrases Beyond planarity, k^+-real face drawings, 2-layer drawings, recognition algorithm, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.GD.2024.32

Funding *Giuseppe Di Battista*: MUR PRIN Proj. 2022TS4Y3N – EXPAND: scalable algorithms for EXPloratory Analyses of heterogeneous and dynamic Networked Data.
Walter Didimo: AIDMIX – Artificial Intelligence for Decision Making: Methods for Interpretability and eXplainability – Ricerca di Base 2021. RASTA: Realtà Aumentata e Story-Telling Automatizzato per la valorizzazione di Beni Culturali ed Itinerari; Italian MUR PON Proj. ARS01_00540.
Fabrizio Montecchiani: MUR PRIN Proj. 2022ME9Z78 – NextGRAAL: Next-generation algorithms for constrained GRAph visuALization.

Acknowledgements Research started at the Dagstuhl Seminar: Beyond-Planar Graphs: Models, Structures and Geometric Representations; seminar number: 24062, February 4-9, 2024.

1 Introduction

Drawing nonplanar graphs in the plane while avoiding forbidden crossing configurations is a prominent line of research in graph drawing. Over the past twenty years, numerous papers have addressed this topic, commonly recognized as *beyond-planar graph drawing*. For

32nd International Symposium on Graph Drawing and Network Visualization (GD 2024).
Editors: Stefan Felsner and Karsten Klein; Article No. 32; pp. 32:1–32:22
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

example, one of the most studied graph hierarchies is this context is the one of *k-planar graphs*, i.e., graphs that admit a drawing in which each edge is crossed at most k times (see, e.g., [24, 26]). Another classical example is the class of *right-angle-crossing graphs* (*RAC graphs*), which admit a straight-line drawing where any two crossing edges form 90° angles at their crossing point (see, e.g., [11, 12]). We refer the reader to surveys, books, and seminar reports for a comprehensive overview on beyond-planar graph drawing [13, 18, 19, 21].

In this paper, we study *k^+-real face graphs*, a beyond-planar graph hierarchy recently introduced in [5, 6] and further studied in [23]. Namely, a nonplanar graph Γ of a graph G divides the plane into topologically connected regions, called *faces* (or *cells*). The boundary of each face is formed by vertices, crossings, and edge segments. Given a positive integer k, the drawing Γ is a *k^+-real face drawing* of G if the boundary of each face of Γ contains at least k vertices of G. In this case, G is a *k^+-real face graph*. The research in [5, 6, 23] concentrates on providing tight bounds on the edge density of k^+-real face graphs, and on establishing relationships between k^+-real face graphs and other prominent beyond-planar graph classes. Conversely, testing which graphs admit a k^+-real face drawing and computing such a drawing when the test is positive are almost unexplored problems. Trivial recognition algorithms exist only for complete graphs and complete bipartite graphs, exploiting results about edge density and crossing numbers for these families [5].

Contribution. Following a consolidated line of research in beyond-planar graph drawing, and addressing a problem mentioned in [5], we investigate the complexity of recognizing k^+-real face graphs. We study both the general unconstrained scenario and the classical 2-layer scenario for bipartite graphs, in which the vertices of the two partition sets are placed on two distinct horizontal lines and the edges are drawn as straight-line segments. We remark that the 2-layer scenario has a long tradition in graph drawing (see, e.g., [1, 15, 16, 22, 27, 28]) and in beyond-planar graph drawing (see, e.g., [3, 4, 7, 9, 10]). Our results are as follows:

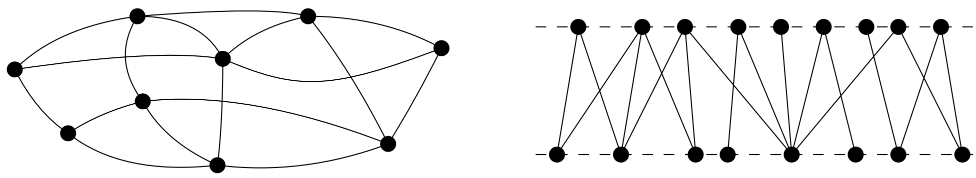
- We prove that, for the set of instances $\langle G, k\rangle$, where G is a graph and k is a positive integer, testing whether G admits a k^+-real face drawing is NP-complete. More specifically, we prove that the problem is already NP-complete for $k \in \{1, 2\}$ and even if G is biconnected (Section 3). This excludes that recognizing k^+-real face graphs is fixed-parameter tractable (FPT) or even slicewise polynomial (XP) when parameterized by k, unless P=NP.
- We provide tight upper bounds on the edge density of 2-layer k^+-real face graphs for any positive integer k (Section 4.1). Then, we describe linear-time algorithms for recognizing 2-layer k^+-real face graphs for any $k \geq 2$, and for recognizing *optimal* 2-layer k^+-real face graphs for any given $k \geq 1$ (Section 4.2). The optimal graphs are those that match the maximum possible edge density. Recognizing optimal graphs for specific beyond-planar graph families is also a classical problem in graph drawing [2, 8, 20].

For space reasons, some proofs are sketched or omitted.

2 Basic Terminology and Tools

We consider connected, *simple* graphs, i.e., without parallel edges and self-loops. Given a graph $G = (V, E)$ and a set $E' \subseteq E$, let $V' \subseteq V$ be the set of the end-vertices of the edges in E'. The graph $G' = (V', E')$ is the subgraph of G *induced by* E'. A *block* B of G is a biconnected component of G. If B is an edge, it is a *trivial block*, otherwise B is *non-trivial.*

In a *drawing* Γ of a graph G the vertices are represented as points of the plane and the edges are simple Jordan arcs. We only consider *simple* drawings, that is: (*i*) adjacent edges do not intersect, except at their common endpoint; (*ii*) two independent (i.e., non-adjacent)



(a) (b)

Figure 1 (a) A 2^+-real face drawing. (b) A 2-layer 2^+-real face drawing.

edges intersect at most in one of their interior points, called a *crossing point*; and (*iii*) no three edges intersect at a common crossing point. A *vertex* of Γ is either a point corresponding to a vertex of G, called a *real-vertex*, or a crossing point, called a *crossing-vertex*. An *edge* of Γ is a curve connecting two vertices of Γ; an edge of Γ whose endpoints are both real-vertices coincides with an edge of G; otherwise, it is just a proper portion of an edge of G. A drawing Γ subdivides the plane into topologically connected regions, called *faces* (or *cells*). The boundary of each face is a circular sequence of vertices and edges of Γ. The face corresponding to the the infinite plane region is the *external face* of Γ; the other faces are the *internal faces*.

For an integer $k \geq 0$, we say that f is a *k-real face* (resp. *k^+-real face*) if it contains exactly (resp. at least) k real vertices. For an integer $k \geq 1$, a *k^+-real face drawing* of a graph G is a drawing such that each face is a k^+-real face (see Figure 1a for an example with $k = 2$). If G admits such a drawing, it is a *k^+-real face graph*. A k^+-real face graph whose number of edges is the maximum possible over all its number of vertices is said to be *optimal*.

2-layer drawings. In a *2-layer drawing* Γ of a bipartite graph $G = (V_1 \cup V_2, E)$, the vertices in V_1 and in V_2 lie on two distinct horizontal lines L_1 and L_2, called *layers*, and the edges are straight-line segments. If Γ is also k^+-real face then it is a *2-layer k^+-real face drawing*, and G is a *2-layer k^+-real face graph* (see Figure 1b for an example with $k = 2$). Again, a 2-layer k^+-real face graph is *optimal* if it matches the maximum possible edge density.

Given a 2-layer drawing Γ of a graph G, we say that there is a *fan crossing* in Γ if two adjacent edges of G are crossed by a third one in Γ; we also say that these three edges *form a fan crossing*. Further, Γ is a *2-layer RAC drawing* if any two crossing edges only cross at right angles. A *2-layer RAC graph* is a graph admitting a 2-layer RAC drawing.

Given a bipartite graph $G = (V_1 \cup V_2, E)$, let π_1 and π_2 be two linear (left-to-right) orderings of the vertices in V_1 and in V_2, respectively. A *2-layer embedding* $\gamma = (\pi_1, \pi_2)$ of G is the equivalence class of 2-layer drawings of G that induce the orderings π_1 and π_2. In other words, γ is an abstraction of a 2-layer drawing where only the vertex orderings on the layers matter, independent of the vertex coordinates. A *drawing of* γ is any 2-layer drawing of G in the class γ. If γ has a 2-layer RAC drawing, it is a *2-layer RAC embedding*. Analogously, if γ has a k^+-real face drawing (for some $k \geq 1$), then γ is a *2-layer k^+-real face embedding*. Note that, in fact, if γ is a 2-layer k^+-real face embedding, *every* drawing of γ is a 2-layer k^+-real face drawing. Indeed, it is not difficult to see that any two distinct drawings of the same 2-layer k^+-real face embedding γ have the same set of faces, which is uniquely determined by the linear orderings of the vertices on the two layers[1] Hence, for a 2-layer k^+-real face embedding γ, we will refer to the faces of γ to indicate the faces of any 2-layer drawing of γ. Similarly, the edge crossings of a graph G in a 2-layer drawing Γ of G only

[1] If a 2-layer drawing Γ has a face without real-vertices, its 2-layer embedding γ does not uniquely determine the set of faces, i.e., another drawing of γ may have a set of faces different from that of Γ.

depend on the 2-layer embedding γ of Γ. A crossing in γ between two edges of G refers to the crossing formed by these edges in any drawing of γ; real- and crossing-vertices of γ refer to the real- and crossing-vertices of any drawing of Γ. Since two real-vertices of the same layer cannot belong to the same internal face of a 2-layer drawing, the next property holds.

▶ **Property 1.** *Let γ be a 2-layer embedding of a connected bipartite graph G. In every drawing of γ, each internal face has at most two real-vertices.*

A *caterpillar* is a tree such that removing all its leaves yields a path. A *ladder* is a bipartite outerplanar graph consisting of two paths of the same length $\langle u_1, \ldots, u_{\frac{n}{2}} \rangle$ and $\langle w_1, \ldots, w_{\frac{n}{2}} \rangle$ plus the edges (u_i, w_i) for $i = 1, \ldots, \frac{n}{2}$; the edges (u_1, w_1) and $(u_{\frac{n}{2}}, w_{\frac{n}{2}})$ are the *extremal edges* of the ladder (see Figure 6). The next result will be used in the following.

▶ **Theorem 2** (Di Giacomo et al. [9]). *A 2-layer embedding γ of a bipartite graph is a 2-layer RAC embedding if and only if there is no fan crossings in γ.*

3 Recognizing Unconstrained k^+-real face graphs

For optimal graphs, the results in [5, 6] imply that for $k \geq 3$, recognizing k^+-real face graphs corresponds to recognizing planar graphs that have an embedding with all faces of degree k. This last problem is tractable for $k \leq 6$ whereas it is NP-complete for odd $k \geq 7$ and for even $k \geq 10$ [25]. Moreover, recognizing optimal 2^+-real face graphs is equivalent to recognizing optimal 1-planar graphs [6], which is linear-time solvable [8]. In this section we prove that recognizing k^+-real face graphs is NP-complete; in particular, the next theorem shows that the problem is NP-complete for $k = 2$ and even for biconnected graphs.

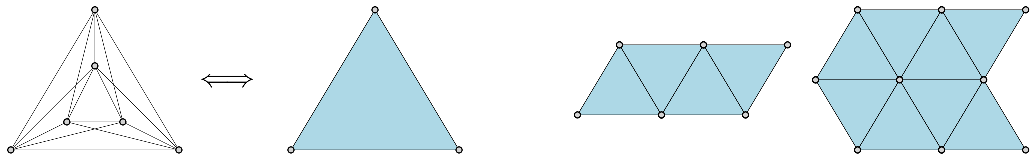
▶ **Theorem 3.** *Deciding if a graph G is 2^+-real face is* NP*-complete, even if G is biconnected.*

Membership of the problem in NP can be easily verified using standard arguments. We reduce from the 3-Partition problem, which is known to be strongly NP-hard [17]. Recall that an instance of 3-Partition consists of a set $A = \{a_1, a_2, \ldots, a_{3m}\}$ of $3m$ integers, each of which is strictly between $B/4$ and $B/2$, where $B = \frac{1}{m}\sum_{i=1}^{3m} a_i$. Then, the problem asks whether A can be partitioned into m subsets $A_1, A_2, \ldots, A_m$, each of cardinality 3, such that the sum of the integers in each subset is B.

Proof overview. The idea is to construct a rigid frame which admits a unique 2^+-real face drawing (up to a homeomorphism of the plane) and contains a large face. Inside this face, we arrange, in a grid-like fashion, $3m$ vertical gadgets and m horizontal paths. The former, called *columns*, encode the integers of an instance A of 3-Partition; see Figure 5a. If a 2^+-real face drawing exists, one can read a solution for A by looking at how the paths intersect the columns; see Figure 5b. A crucial ingredient is an intertwined design of the columns and of the path gadgets such that the latter must cross the former in a controlled manner.

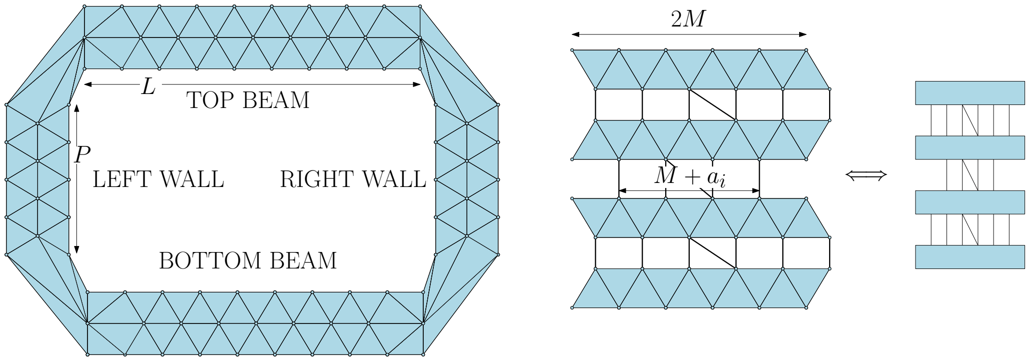
Construction. Let A be an instance of 3-Partition. We will construct in polynomial time a graph G, such that A admits a partition if and only if G admits a 2^+-real face drawing. In our construction, we will leverage K_6 as a building block, since any 2^+-real face drawing of it is 1-planar as we prove in the following lemma; refer to Figure 2a for an illustration.

▶ **Lemma 4.** *Any 2^+-real face drawing of K_6 is 1-planar.*



(a) (b)

Figure 2 (a) A 1-planar and 2^+-real face drawing of K_6, and its schematic representation. (b) Schematic representations of a belt of length three, and of a 2-belt.



(a) (b)

Figure 3 (a) A ring barrier. (b) Three channels of a column, two are dense and one is sparse, along with their schematic representation.

Proof. By Lemma 1 of [6] it follows that, for a 2^+-real face drawing, it holds $\chi \leq n-2$, where χ is the number of crossings in the drawing and n is the number of real vertices. For K_6, this implies $\chi \leq 4$. Suppose by contradiction that K_6 admits a 2^+-real face drawing Γ with an edge crossed at least twice. Thus, Γ contains a face of degree at least four (namely a face with at least two real vertices and two crossing-vertices). Since each other face of Γ has at least degree three and since the sum of the degrees of the faces of Γ equals twice the number μ of its edges, we have $2\mu \geq 3(\phi-1)+4 = 3\phi+1$, where ϕ denotes the number of faces of Γ. Since $\mu = m+2\chi$ and $\phi = m+\chi+2-n$, the last inequality implies that $2(m+2\chi) \geq 3(m+\chi+2-n)+1$ or equivalently $\chi \geq m-3n+7$ holds. For K_6, we know $n=6$ and $m=15$, thus $\chi \geq 10$, which contradicts the bound $\chi \leq 4$. ◀

A *belt of length* k is a chain of k copies of K_6 that are glued such that two consecutive copies share one edge; see also Figure 2b. A b*-belt of length* k is obtained by merging together $b > 1$ belts of length k, as shown in Figure 2b. To construct graph G, we first create a quite rigid structure, called *ring barrier* R, consisting of four components: the *top beam*, the *right wall*, the *bottom beam* and the *left wall*. Each of the top and bottom beams consists of a T-belt of length L, while each of the left and right walls consists of a T-belt of length P, with the following choice of parameters: (i) $M = \lceil B/2 \rceil + 1$; (ii) $X = 2M$; (iii) $L = 3mX$; (iv) $P = 3m+2$; (v) $T = L^2$. Note that $M > a_i$ for each $i = 1, \ldots, 3m$. Also, L and P are chosen to accommodate $3m$ *column gadgets* and m *transversal gadgets*, to be defined later. T makes the ring barrier thick enough; it is formed by gluing in a circular arrangement the endpoints of the top beam, right wall, bottom beam and left wall; see Figure 3a.

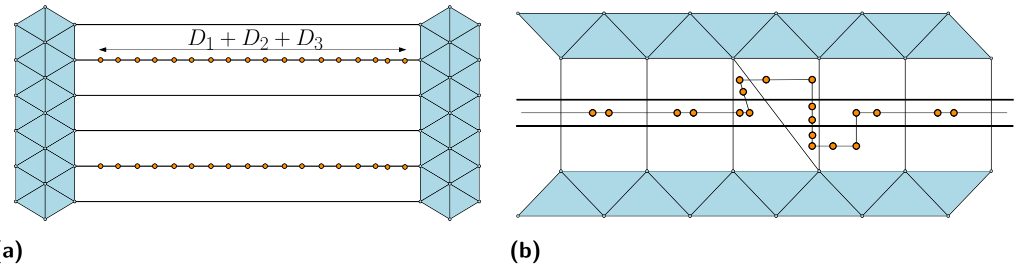


Figure 4 (a) Two transversal gadgets (only a part of the ring barrier is shown and columns are omitted). (b) A transversal gadget routed through a channel.

The top and bottom beams are connected by a set of $3m$ *columns*; see Figure 3b. Each column contains a stack of $2m-2$ copies of the *barrier-gadget*, which is formed by gluing X copies of K_6. Within each column, consecutive copies of the barrier-gadget are connected by an even set of pairwise disjoint edges whose size will be defined later, interleaved by an edge in the middle of the sequence forming two triangles, as shown in Figure 3b. The topmost and the bottommost copy of the barrier-gadget of each column is connected to the top and bottom beam, respectively, also in the same fashion. The edges that realize these connections are called *vertical edges* and form the so-called *channels*. In particular, there are $m-1$ *topmost channels*, one *central channel* and $m-1$ *bottommost channels*. The central channel of the i-th column is *sparse* containing only $M + a_i$ vertical edges (note that, by construction, $X = 2M > M + a_i$); the remaining ones are *dense* containing X edges each.

We conclude the construction of graph G by introducing m pairwise disjoint gadgets, $\pi_1, \pi_2, \dots, \pi_m$, called the *transversal gadgets* of G. Each transversal gadget consists of two edges, called *guide edges*, and one path of $D_1 + D_2 + D_3$ vertices, where the three parameters are specified below. The intuition is that, in order to realize a 2^+-real face drawing, each path must cross all columns through one of their channels, and each path will be able to do so if and only if it can be routed through exactly three sparse channels whose number of vertical edges is equal to B; therefore, the length of a single path is crucial: (i) $D_1 = (3m-3)(2X+8)$, for the path to be able to cross $3m-3$ dense channels; (ii) $D_2 = 2B + 6M + 24$, for the path to be able to cross 3 sparse channels; (iii) $D_3 = 2(3m-1)$, for the path to be able to cross the faces between consecutive columns. These gadgets are attached at independent consecutive vertices along the left and right walls, as shown in Figure 4a (each of the m gadgets takes three vertices on the left wall and on the right wall, which are both made by T-belts of length $3m+2$). Note that G does not contain any cut-vertex.

Proof sketch for Theorem 3. To prove that A admits a partition if and only if G admits a 2^+-real face drawing, we need a few definitions. A *canonical drawing* of G is one such that, if two edges cross, then one of the following cases applies: (i) both edges are part of a K_6, or (ii) one edge is part of a transversal gadget and the other is a vertical edge of a channel, or (iii) one is a guide edge of transversal gadget and the other is an edge of the path of the same transversal gadget. Consider a column C and a channel c of C. We say that a transversal gadget π is *routed through* c, if c is the only channel of C whose edges are crossed by some edges of π, see Figure 4b. If $A_1, \dots, A_m$ is a solution of A, then a 2^+-real face drawing of G can be obtained by routing each transversal gadget π_i through $m-3$ dense channels and 3 sparse channels corresponding to the elements of A_i as shown in Figure 5. Proving the other direction requires a more involved argument. We first prove that a crossing-minimal 2^+-real face drawing of G, if any, is a canonical drawing. Next we

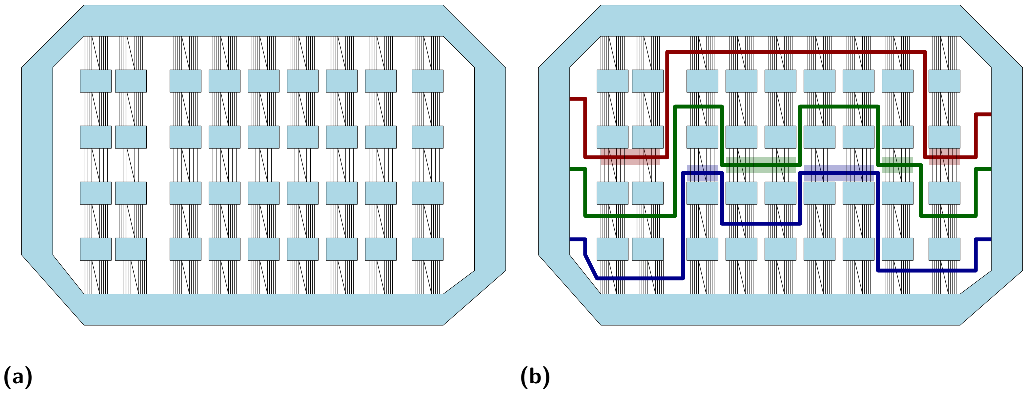


Figure 5 (a) A canonical drawing (schematic representation) where the transversal gadgets are omitted. (b) A complete canonical drawing in which the three transversal gadgets are schematized as thick polylines of different colors and their intersections with sparse cells are highlighted.

can show that each transversal gadget is routed through $m-3$ dense channels and 3 sparse channels such that no two gadgets traverse the same channel. This allows the construction of a solution $A_1, \dots, A_m$ of A as follows: if π_j crosses the κ-th, λ-th and μ-th columns of G through sparse channels, where $1 \leq \kappa, \lambda, \mu \leq 3m$, then $A_j = \{a_\kappa, a_\lambda, a_\mu\}$. ◀

▶ Remark 5. The proof of Theorem 3 can be adapted to show that recognizing 1^+-real face graphs is also NP-complete. At high level, we substitute the copies of K_6 with copies of K_7, slightly modify the channel structure by introducing two edges that cross (to ensure that a single guide edge cannot be routed through a channel without an associated path) and halve the length of the paths. In the interest of space, we defer the details to the journal version.

4 2-Layer k^+ graphs

In this section we focus on 2-layer k^+-real face drawings. We start giving edge-density results for each positive integer k (Section 4.1); they represent a preliminary step for the recognition problem. Then, we describe algorithms to recognize 2-layer k^+-real face graphs for $k \geq 2$, and algorithms to recognize optimal 2-layer k^+-real face graphs for $k \geq 1$ (Section 4.2).

4.1 Edge Density

We give tight upper bounds on the edge density of n-vertex 2-layer k^+-real face graphs, for any $k \in [1, n]$. For $k \in [3, n]$ the next theorem establishes that the n-vertex connected 2-layer k^+-real face graphs are caterpillars, thus they have $n-1$ edges.

▶ **Theorem 6.** *An n-vertex connected graph is 2-layer k^+-real face for any $k \in [3, n]$ if and only if it is a caterpillar.*

Proof. A connected graph has a 2-layer planar embedding if and only if it is a caterpillar [14]. Suppose first that γ is a 2-layer k^+-real face embedding of a connected graph G for a given $k \in [3, n]$. By Property 1, γ cannot have internal faces, i.e., all real-vertices of γ belong to the external face. In particular, there cannot be any two edges of G that cross in γ, otherwise, since G is connected, there would be an internal face in γ. Hence G is 2-layer

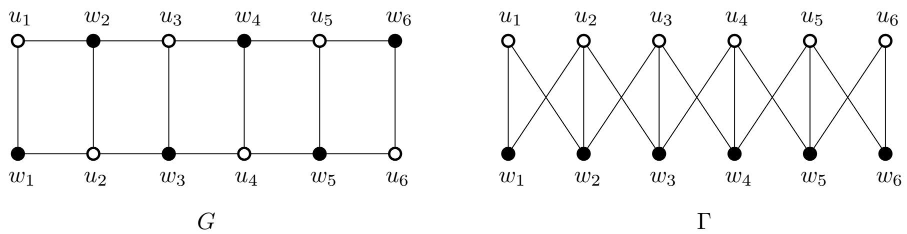


Figure 6 Illustration for Theorem 7. An optimal 2-layer 2^+-real face graph G with $n = 12$ vertices (left) and a 2-layer 2^+-real face drawing Γ of G (right).

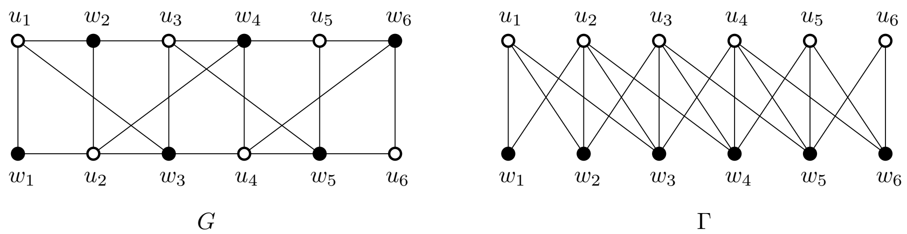


Figure 7 Illustration for Theorem 8. An optimal 2-layer 1^+-real face graph G with $n = 12$ vertices (left) and a 2-layer 1^+-real face drawing Γ of G (right).

planar, which implies that it is a caterpillar. Conversely, if G is a caterpillar it has a 2-layer planar embedding. This embedding has a unique face (the external face), which contains all the vertices of the graph, thus it is a 2-layer k^+-real face embedding for every $k \in [3, n]$. ◀

▶ **Theorem 7.** *Any n-vertex 2-layer 2^+-real face graph has at most $1.5n - 2$ edges, and this bound is tight.*

Proof. Let $G = (V_1 \cup V_2, E)$ be an n-vertex bipartite graph that admits a 2-layer 2^+-real face drawing Γ, and let m be the number of edges of G. Augment Γ (and G) with $n - 2$ (non-crossing) straight-line edges that connect all the vertices in each vertex set V_i ($i = 1, 2$), in their linear ordering along the corresponding layer. The resulting drawing Γ' is an *outer 2^+-real face drawing* of a graph G' with $n' = n$ vertices, i.e., a 2^+-real face drawing with all vertices on the external face. In [6] it is proved that such a graph G' has at most $2.5n - 4$ edges. Since G' has $m' = m + n - 2$ edges, we have $m \leq 1.5n - 2$. About the tightness of the bound, the ladders on n vertices are optimal 2-layer 2^+-real face graphs (Figure 6). ◀

▶ **Theorem 8.** *Any n-vertex 2-layer 1^+-real face graph has at most $2n - 4$ edges, and this bound is tight.*

Sketch. The proof is analogous to that of Theorem 7, but exploits the fact that an outer 1^+-real face drawing has at most $3n - 6$ edges [6]. A 2-layer 1^+-real face graph that matches this bound is a ladder with some extra edges (see Figure 7 when $n = 12$). Namely, let $V_1 = \{u_1, \ldots, u_{\frac{n}{2}}\}$ and $V_2 = \{w_1, \ldots, w_{\frac{n}{2}}\}$. For $i = 1, \ldots, \frac{n}{2}$ there is an edge (u_i, w_i). For $i = 1, \ldots, \frac{n}{2} - 1$ there are the two edges (u_i, w_{i+1}) and (u_{i+1}, w_i). For each $i \in \{1, \ldots, \frac{n}{2} - 2\}$ there is an edge (u_i, w_{i+2}). This graph has in total $m = \frac{n}{2} + n - 2 + \frac{n}{2} - 2 = 2n - 4$ edges. ◀

4.2 Recognition

For $k \in [3, n]$, Theorem 6 implies that testing whether an n-vertex graph G is 2-layer k^+-real face is equivalent to testing whether G is a caterpillar. Hence, the following holds.

▶ **Theorem 9.** *Let G be any connected graph with n vertices and let $k \geq 3$. There exists an $O(n)$-time algorithm that tests whether G is 2-layer k^+-real face and that computes a 2-layer k^+-real face drawing if one exists.*

Theorem 9 extends trivially to the recognition of optimal 2-layer k^+-real face graphs for $k \geq 3$, since by Theorem 6 any connected 2-layer k^+-real face graph with $k \geq 3$ is optimal. Recognizing 2-layer k^+-real face graphs for $k \in \{1, 2\}$ is more involved. The following definition will be used in the next subsections. Let G be a bipartite graph, P be a simple path in G, and γ be a 2-layer embedding of G. We say that P is a *zig-zag path* in γ if the restriction of γ to P is not self-crossing (see, e.g., the path from w to w' in Figure 8). If P is a zig-zag path, there exists a drawing of γ in which P is x-monotone; thus, with a slight abuse of terminology, we also refer to the left-to-right order of the vertices of P in γ.

4.2.1 2-layer 2^+-real face graphs

We start with the following inclusion relationship.

▶ **Lemma 10.** *Any 2-layer 2^+-real face embedding is a 2-layer RAC embedding.*

Proof. Suppose that γ is a 2-layer 2^+-real face embedding of a graph. There cannot be a fan crossing in γ, as otherwise γ would have a triangular 1-real face, contradicting the hypothesis that γ is a 2^+-real face embedding. Hence, by Theorem 2, γ is a 2-layer RAC graph. ◀

Lemma 10 implies that the family of 2-layer 2^+-real face graphs is included in the family of 2-layer RAC graphs. The reverse does not hold, as the next lemma proves.

▶ **Lemma 11.** *There exist infinitely many graphs that are 2-layer RAC but not 2-layer 2^+-real face.*

Sketch. For any even positive integer k, consider a bipartite graph $G = (V_1 \cup V_2, E)$ consisting of: (i) two 4-cycles B and B'; (ii) a path $\overline{P}$ of length k between a vertex $w \in V_2$ of B and a vertex $w' \in V_2$ of C'; (iii) two paths P and P' of length $\frac{k}{2} + 2$, where P is attached to w and P' is attached to w'. See Figure 8 for an illustration where: $k = 4$, $w = w_2$, $w' = w_8$; C, C', and $\overline{P}$ are in bold; P is in blue and P' is in red. In any 2-layer RAC embedding of G, P and P' are zig-zag paths that cross each other, thus forming a 1-real face f (see Figure 8). ◀

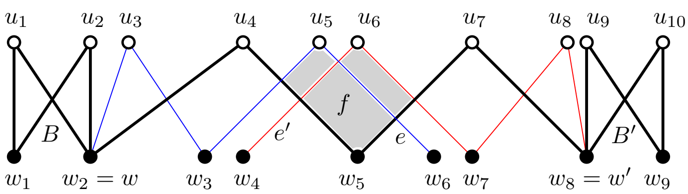


Figure 8 Illustration for Lemma 11. A 2-layer RAC graph that is not 2-layer 2^+-real face.

Note however that the graphs of Lemma 11 are not biconnected. If we restrict to biconnected graphs, we are able to prove that 2-layer RAC graphs and 2-layer 2^+-real face graphs are in fact the same family. More precisely, the following result is known (see [9]).

▶ **Theorem 12** (Di Giacomo et al. [9]). *An n-vertex biconnected graph G is 2-layer RAC if and only if it is a spanning subgraph of a ladder. Also, there exists an $O(n)$-time algorithm that tests whether G is 2-layer RAC, and computes a 2-layer RAC drawing of G if one exists.*

From the considerations above, we derive the following characterization.

▶ **Lemma 13.** *A biconnected graph is 2-layer* 2^+*-real face if and only if it is a spanning subgraph of a ladder.*

Proof. As shown, every ladder is a 2-layer 2^+-real face graph, and hence every spanning subgraph of a ladder is a 2-layer 2^+-real face graph. Vice versa, let G be a biconnected 2-layer 2^+-real face graph and let γ be a 2-layer 2^+-real face embedding of G. By Lemma 10, γ is 2-layer RAC, and hence, G is a spanning subgraph of a ladder by Theorem 12. ◀

The next theorem follows by combining Lemma 13 and Theorem 12.

▶ **Theorem 14.** *An n-vertex biconnected G is 2-layer* 2^+*-real face if and only if it is 2-layer RAC. Also, there exists an $O(n)$-time algorithm that tests whether G is 2-layer* 2^+*-real face and that computes a 2-layer* 2^+*-real face drawing if one exists.*

Moreover, by Lemma 13 and since any n-vertex ladder is optimal 2-layer 2^+-real face (Theorem 7), whereas any non-biconnected n-vertex 2-layer 2^+-real face graph has less than $1.5n - 2$ edges, we get the following result for optimal 2-layer 2^+-real face graphs.

▶ **Corollary 15.** *An n-vertex graph G is optimal 2-layer* 2^+*-real face if and only if it is a ladder. Optimal 2-layer* 2^+*-real face graphs can be recognized in $O(n)$ time.*

We now focus on the recognition of connected 2-layer 2^+-real face graphs that are not biconnected. By Lemma 10, every 2-layer 2^+-real face embedding of a bipartite graph G (if any) must be searched in the space of 2-layer RAC embeddings of G. Hence, we first recall in some details what is the structure of any 2-layer RAC embedding γ of a connected graph G; then we establish an extra property that γ must fulfill to be a 2-layer 2^+-real face embedding.

Structure of 2-layer RAC embeddings. Let G be a 1-connected graph and let γ be a 2-layer RAC embedding of G. As showed in [9], the embedding γ consists of two parts:

- Skeleton. The first part, called *skeleton*, is a (possibly empty) left-to-right sequence of non-trivial blocks; any two consecutive blocks either share a cut-vertex of G or are connected by a zig-zag path, placed between them, which we call an *in-between path.* The first (last) block of the sequence may be preceded (followed) by a *maximal* zig-zag path attached to it, called an *extremal path.* If G is a tree (without non-trivial blocks), the skeleton is just a single zig-zag path; by convention, the extremal paths coincide with such a zig-zag path. We denote by $\mathrm{skel}_\gamma(\mathrm{G})$ the subgraph of G induced by the edges of G in the skeleton of γ.
- Dangling Paths. The second part is a set of zig-zag paths, each path P sharing exactly one vertex w with the skeleton. We call P a *dangling path* and w the *attaching vertex* of P. Two dangling paths are either edge-disjoint or they have exactly one edge in common, which is the one containing the attaching vertex of the paths. We denote by $\mathrm{dang}_\gamma(\mathrm{G})$ the subgraph of G induced by the edges that belong to the dangling paths.

Clearly, the edges of $\mathrm{skel}_\gamma(\mathrm{G})$ and of $\mathrm{dang}_\gamma(\mathrm{G})$ partition the edge set of G. When G is not a tree, we also denote by $\mathrm{skel}^-_\gamma(\mathrm{G})$ the subgraph of $\mathrm{skel}_\gamma(\mathrm{G})$ consisting only of the non-trivial blocks and their in-between paths. Figure 9 shows an example of 2-layer RAC embedding and its parts. Note that, by definition, if G is not a tree, each zig-zag path of $\mathrm{skel}_\gamma(\mathrm{G})$ is attached to two non-trivial blocks (if this path is an in-between path) or to one non-trivial block (if this path is an extremal path). Further, by Lemma 13, each non-trivial

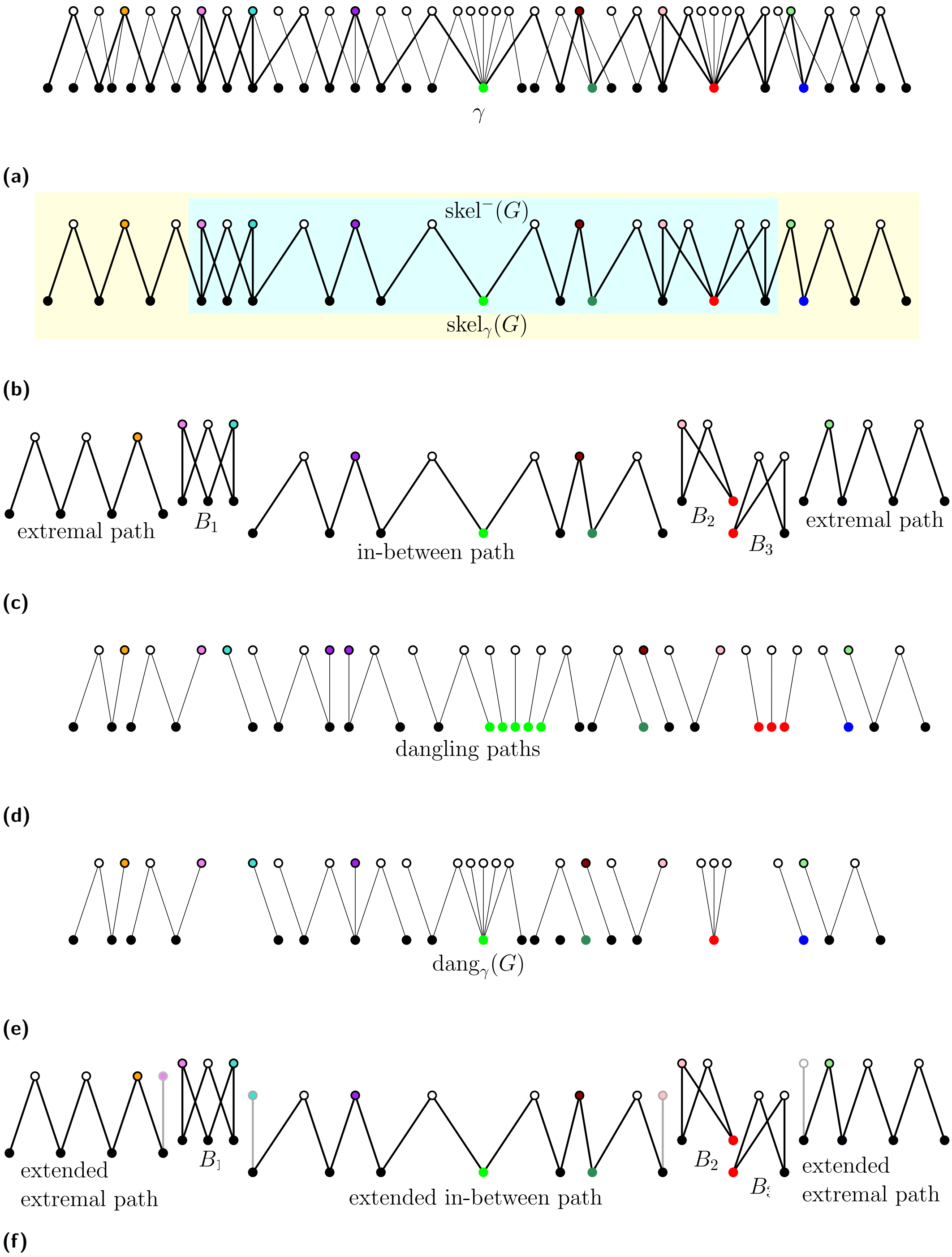


Figure 9 (a) A 2-layer RAC embedding γ of a graph G; the colored vertices are attaching vertices; (b) $\mathrm{skel}_\gamma(\mathrm{G})$ and $\mathrm{skel}^-(\mathrm{G})$; (c) the components of $\mathrm{skel}_\gamma(\mathrm{G})$; (d) the dangling paths of γ; (e) $\mathrm{dang}_\gamma(\mathrm{G})$. (f) The extended zig-zag paths of the 2-layer RAC embedding γ.

block B of G is a spanning subgraph of a ladder, and the leftmost and rightmost edges of B, which we call the *extremal edges of* B in γ, coincide with the extremal edges of such a ladder. A skeleton zig-zag path extended with the extremal edges of the non-trivial blocks to which it is attached will be called an *extended zig-zag path* of $\mathrm{skel}_\gamma(\mathrm{G})$. For example, Figure 9f shows the extended zig-zag paths of the 2-layer RAC embedding of Figure 9a.

The following properties hold for any 2-layer RAC embedding γ, otherwise it is easy to see that γ would contain a fan-crossing (see [9] for details).

▶ **Property 16.** *A dangling path cannot cross any non-trivial block in* γ*.*

▶ **Property 17.** *Let P and P' be two edge-disjoint dangling paths of γ that cross each other, and let w and w' be their attaching vertices. Then:* (*a*) *w and w' are distinct vertices of the same extended zig-zag path of the skeleton;* (*b*) *if (w.l.o.g.) w is to the left of w', then all the vertices of P (P') distinct from w (w') are to the right of w (to the left of w').*

▶ **Property 18.** *Each edge of a dangling path in γ crosses at most one edge of the skeleton. Also, if P and P' are two edge-disjoint dangling paths that cross each other in γ, then each edge of P and of P', with the possible exception of the edges incident to their attaching vertices, crosses exactly one edge of the skeleton.*

▶ **Property 19.** *Let w be any vertex of an extended zig-zag path of the skeleton. Then, there are at most two dangling paths of length larger than one attached to w.*

▶ **Property 20.** *If G is not a tree and if γ' is a 2-layer RAC embedding of G distinct from γ, then* $\mathrm{skel}^-_\gamma(\mathrm{G}) = \mathrm{skel}^-_{\gamma'}(\mathrm{G})$*. Also the restriction of γ to* $\mathrm{skel}^-_\gamma(\mathrm{G})$ *coincides with the restriction of γ' to* $\mathrm{skel}^-_{\gamma'}(\mathrm{G})$ *(up to mirroring).*

By Property 20, we can denote $\mathrm{skel}^-_\gamma(\mathrm{G})$ by $\mathrm{skel}^-(\mathrm{G})$, as it does not depend on the specific embedding γ. We can get $\mathrm{skel}^-(\mathrm{G})$ by recursively removing from G the degree-1 vertices.

Structure of 2-layer 2^+-real face graphs. A 2-layer RAC embedding γ of a graph G is *dangling-crossing free* if it does not have two dangling paths that cross each other. The next lemma is a key ingredient to efficiently recognize 2-layer 2^+-real face graphs.

▶ **Lemma 21.** *Let γ be a 2-layer RAC embedding of a connected graph G. Then, γ is a 2-layer 2^+-real face embedding if and only if it is dangling-crossing free.*

Sketch. Let $n \geq 2$ denote the number of vertices of G. Suppose first that γ is dangling-crossing free. The external face of γ contains all the vertices of G, thus it is an n-real face, and hence a 2^+-real face. Consider now any internal face f of γ. By Property 16, a dangling path cannot cross any edge of a non-trivial block of G in γ. Hence, if f is formed only by skeleton edges of γ then it is a face internal to the embedding of a non-trivial block B. Since by Lemma 13, B is a spanning subgraph of a ladder (and it is drawn RAC), the face f is either a triangle with two real-vertices or a quadrilateral with two real-vertices (see Figure 10a). Finally, assume that f is formed by skeleton edges and by edges of a single dangling path P (f cannot contain edges of two distinct dangling paths, because we are assuming that the dangling paths are pairwise non-crossing). In this case, the skeleton edges that cross P to form f belong to a zig-zag path (either an extremal path or an in-between path of γ). Since by Property 18 each edge of P crosses at most one skeleton edge and since P is also a zig-zag path, f is either a 2-real triangle or a 2-real quadrilateral (see Figure 10b).

Suppose vice versa that γ is a 2-layer 2^+-real face embedding. If there were in γ two dangling paths that cross each other, they would form a 1-real face, like f in Figure 8. ◀

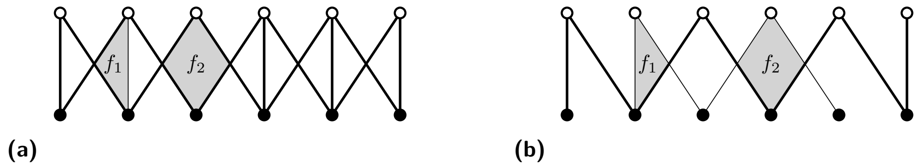


Figure 10 (a) Internal faces created by skeleton edges (bold); (b) Internal faces created by skeleton edges (bold) and dangling path edges (thin).

Before stating the main result of this section, we give an additional auxiliary lemma.

▶ **Lemma 22.** *If G is a 2-layer 2^+-real face graph, then G admits a 2-layer 2^+-real face embedding such that the attaching vertex of each dangling path has degree at least three.*

▶ **Theorem 23.** *Let G be an n-vertex bipartite graph. There exists an $O(n)$-time algorithm that tests whether G is 2-layer 2^+-real face, and that computes a 2-layer 2^+-real face drawing of G if one exists.*

Proof. Based on Lemma 21, we describe an algorithm that attempts to construct a dangling-crossing free 2-layer RAC embedding γ of G, if one exists. We distinguish two main cases: Case 1 – G is not a tree (G contains at least one non-trivial block); Case 2 – G is a tree.

Case 1 – G is not a tree. The algorithm executes the following steps:

Step 1. (See Figure 11.) It tests whether there exists a subgraph of G that is a valid $\text{skel}^-(\text{G})$. To this aim, it recursively removes from G all the vertices of degree one and then applies the algorithm of Theorem 12 to check whether the resulting graph G' admits a 2-layer RAC embedding, and to compute one if any. If such an embedding does not exist, the algorithm stops and rejects the instance. Otherwise, G' coincides with $\text{skel}^-(\text{G})$ and, by Property 20, its 2-layer RAC embedding γ' is unique (up to mirroring); the algorithm goes to the next step.

Step 2. Let $B_1, \dots, B_h$ ($h \geq 1$) be the non-trivial blocks in the left-to-right order defined by γ'. The edges of $G \setminus \text{skel}^-(\text{G})$ form a forest $\mathcal{F}$ of trees, each tree sharing exactly one vertex with $\text{skel}^-(\text{G})$. At most two of these trees share a vertex with the leftmost extremal edge of B_1 in γ'; let $\mathcal{F}^\ell$ be the subset of $\mathcal{F}$ that contains these (at most two) trees. Analogously, at most two trees share a vertex with the rightmost extremal edge of B_h in γ'; let $\mathcal{F}^r$ be the subset of $\mathcal{F}$ that contains these trees. In this step, the algorithm tests if the trees in $\mathcal{F} \setminus \{\mathcal{F}^\ell, \mathcal{F}^r\}$ form a valid set of dangling paths that can be attached to γ' to get a dangling-crossing free 2-layer RAC embedding γ''. This is done by executing the following substeps.

- Step 2.1. (See Figure 12a.) First, the algorithm considers the paths attached to every cut-vertex shared by two non-trivial blocks. Specifically, for each such cut-vertex w, all the paths attached to w can be successfully embedded (between the two non-trivial blocks sharing w) if and only if each of them consists of a single edge; otherwise one of these paths would cross a non-trivial block, thus violating Property 16.
- Step 2.2. (See Figure 12b.) For each in-between path $\overline{P}$, delimited by two blocks B_i and B_{i+1}, the algorithm checks whether there are some paths attached to the vertex u of the rightmost extremal edge of B_i not in $\overline{P}$, or attached to the vertex v of the leftmost extremal edge of B_{i+1} not in $\overline{P}$. In particular, there can be at most one such a path, call it P (resp. P'), attached to u (resp. to v), because two paths attached to u (or to v) cannot be embedded without causing a fan crossing with $\overline{P}$. Further, P and P' must

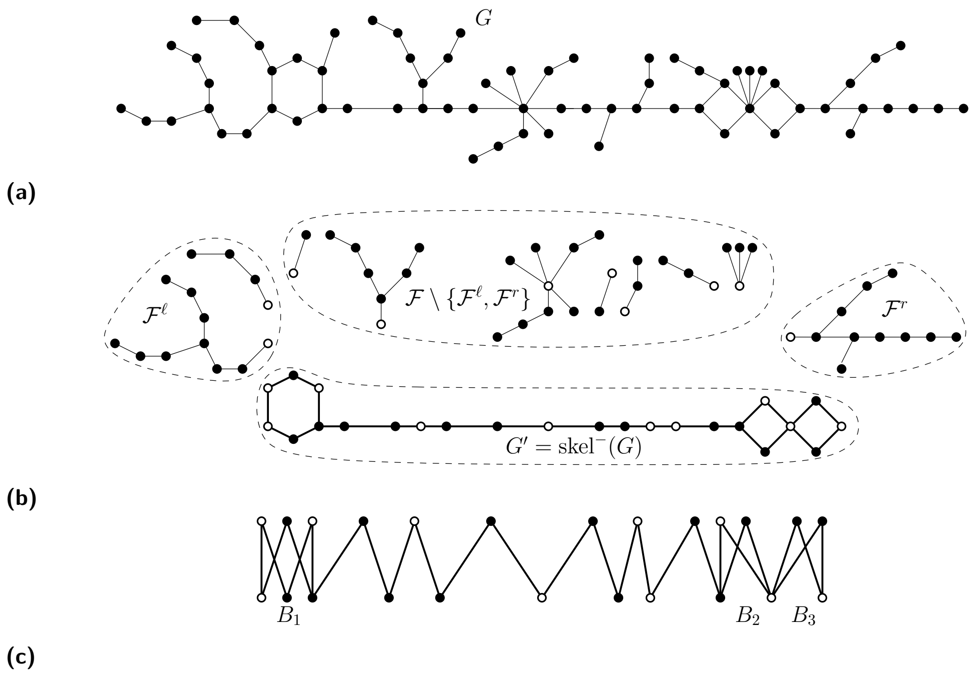


Figure 11 Step 1: (a) an input bipartite graph G; (b) decomposition of G into skel$^-$(G) and a forest $\mathcal{F}$; (c) a 2-layer RAC embedding of skel$^-$(G). The attaching vertices are in white.

necessarily be embedded to the right of u and to the left of v, respectively. Hence, if both P and P' exist in the graph, the algorithm checks that they do not cross each other in their unique embedding. If only one among P and P' exists, say P, the algorithm checks that it does not cross B_{i+1} in its unique embedding. If any of the above checks fails, then the algorithm rejects the instance, otherwise it continues with the next substep.

- Step 2.3. (See Figures 13a and 13b.) For each in-between path $\overline{P}$ of γ', the algorithm considers the paths that in G are attached to $\overline{P}$, and checks if they can be embedded so to be pairwise non-crossing. Formally, let B_i and B_{i+1} be the two non-trivial blocks that delimit $\overline{P}$ $(1 \leq i \leq h-1)$. Also, let $W = \langle w_1, w_2, \ldots, w_p \rangle$ be the left-to-right sequence of attaching vertices of $\overline{P}$, i.e., the vertices to which some trees of $\mathcal{F} \setminus \{\mathcal{F}^\ell, \mathcal{F}^r\}$ are attached. By Property 16, all the paths attached to $\overline{P}$ must be embedded between B_i and B_{i+1}. Hence, we can process the vertices of W in their left-to-right order and test, for each vertex $w_j \in W$, if all paths attached to w_j can be suitably embedded as zig-zag paths so that: (i) they do not cross with any previously embedded dangling paths attached to w_g, with $g < j$, or with B_i, or with the dangling path P attached to B_i embedded in Step 2.2; (ii) they leave the maximum degree of freedom for embedding the dangling paths attached to w_{j+1}, subject to condition (i). Conditions (i) and (ii) together guarantee the correctness of the testing algorithm. However, to satisfy these conditions, we sometimes need to process contemporary all the vertices in specific subsequences of W, as done in [9] for testing 2-layer RAC embeddability (a necessary condition in our case).

  More in details as proved in [9], there are only three possible types of graph structures, called *feasible structures*, that could be attached to the vertices of W without necessarily creating fan crossings (see Figures 14a and 14b):

  - star-tree: it is a subdivision of a star rooted at a vertex $w \in W$. By Property 19, at most two paths attached to w have length larger than one. Call them the *long paths*.
  - y-tree: it is a tree attached to a vertex $w \in W$ and consisting of two paths sharing

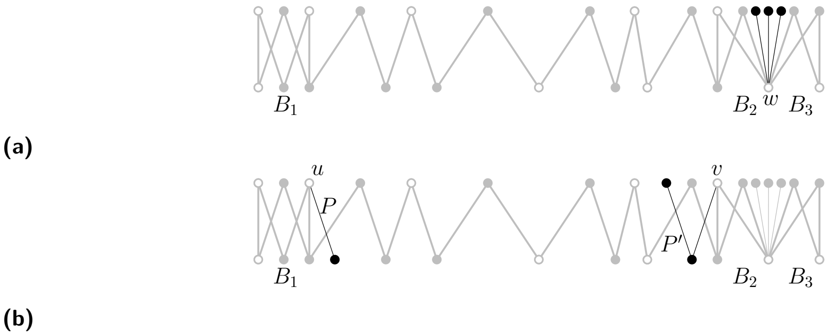


Figure 12 (a) Step 2.1: Addition of paths attached to a cut-vertex w shared by two non-trivial blocks; (b) Step 2.2: Addition of the paths P and P'.

only the edge that contains w.

- k-fence: it is a set of vertex-disjoint paths attached to a maximal subsequence W' of vertices of W that are consecutive in $\overline{P}$. Each $w \in W'$ has exactly one path attached to it. Also, if $k \geq 5$, all the paths attached to the vertices of W', except possibly those attached to the first two and to the last two vertices of W', have length at most two.

Hence, the testing algorithm first checks whether all the vertices in W can be partitioned into maximal subsequences $W_1, W_2, \dots, W_q$ ($q \geq 1$), such that each subsequence W_i contains the attaching vertices of a feasible structure, which we denote by $T(W_j)$ ($|W_j| = 1$ if $T(W_j)$ is a star-tree or a y-tree). If this is not possible, the algorithm rejects the instance. Otherwise, it searches for an embedding of each $T(W_j)$ such that all these embeddings, along with the embeddings of B_i, B_{i+1}, and $\overline{P}$, result in a 2-layer RAC embedding that is dangling-crossing free. Namely, in [9] it is shown that, for the 2-layer RAC embeddability, the number of candidate embeddings for each structure is bounded by a small constant. More precisely (see Figures 14c–14f): (i) For a star-tree it must be decided which of the (at most) two long paths can go to the left and which to the right (the paths of length one can always be embedded without crossing the skeleton); (ii) for a y-tree it must be decided which of the two paths goes to the left and which to the right; (iii) for a k-fence, the set of candidate embeddings is at most 2 if $k = 2$, or at most 3 if $k = 3$, or at most 4 if $k \geq 4$. In particular, in a valid embedding of each k-fence, one of its paths will be embedded to the left of the k-fence, one to the right, and the others in-between the leftmost and the rightmost vertices of the k-fence (see [9] for more details).
However, differently from [9], we can only accept the embeddings of $T(W_1), \dots, T(W_q)$ where no two dangling paths cross each other. To this aim, our testing algorithm processes all W_j from left to right in a greedy fashion. Each time a subsequence W_j is considered, the algorithm checks if the feasible structure $T(W_j)$ has some candidate RAC embeddings that do not cause crossings with dangling paths already embedded to the left of W_j. If not, the instance is rejected; otherwise, among the candidate embeddings, the algorithm selects one for which the dangling path of $T(W_j)$ that goes to the right is as short as possible, which maximizes the degrees of freedom for dangling paths that will be processed in the future. This guarantees that the test is positive if and only if a 2-layer 2^+-real face embedding exists. At the end, the algorithm checks whether the embedding of $T(W_q)$ causes a crossing of a dangling path of $T(W_q)$ with B_{i+1} (or with a dangling path attached to B_{i+1} in Step 2.2). If so, it rejects the instance, otherwise goes to the next step.

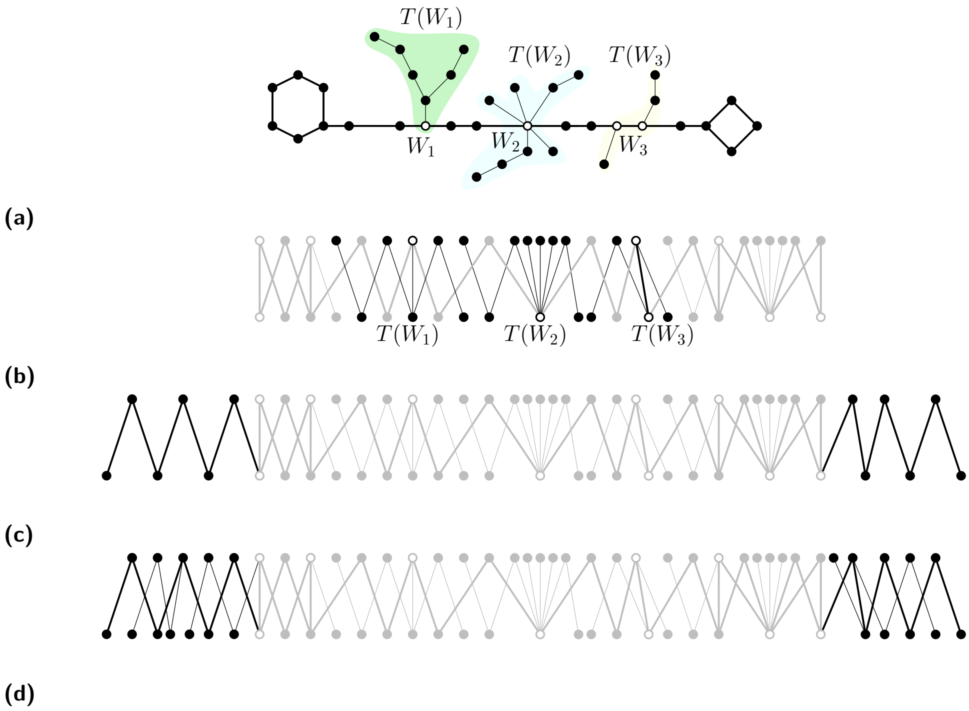


Figure 13 Step 2.3: (a) feasible structures attached to the in-between path of the graph of Figure 11a; (b) addition of $T(W_1)$, $T(W_2)$, and $T(W_3)$. Step 3: (c) addition of the extremal paths; (d) completion of the 2-layer RAC embedding.

Step 3.(See Figures 13c and 13d.) The algorithm tests if $\mathcal{F}^\ell$ and $\mathcal{F}^r$ can be added to the embedding γ'' of Step 2, to form the final dangling-crossing free 2-layer RAC embedding γ. If so, $\mathcal{F}^\ell$ and $\mathcal{F}^r$ coincide with the graph formed by the two extremal paths of $\text{skel}_\gamma(\text{G})$ and by the dangling paths attached to (the extended version of) these extremal paths. To perform this test for $\mathcal{F}^r$, the algorithm has to determine the path corresponding to the extremal path $\overline{P}$ in $\mathcal{F}^r$. Then, it will apply the same procedures as in Steps 2.2 and 2.3 to test whether the remaining part of $\mathcal{F}^r$ consists of dangling paths attached to (the extended version of) $\overline{P}$, and whether they can be embedded without crossing each other and without creating fan crossings. The test for $\mathcal{F}^\ell$ is the same, but the procedures of Steps 2.2 and 2.3 are applied going from right to left. We now explain how the algorithm can test for the existence of a valid $\overline{P}$ in $\mathcal{F}^r$ (for $\mathcal{F}^\ell$ the algorithm is symmetric).

Let $e = (u, v)$ be the rightmost extremal edge of the last non-trivial block B_h, and let T_u and T_v be the two trees of $\mathcal{F}^r$ attached to u and v, respectively. If both T_u and T_v are non-empty and contain vertices of degree larger than two (other than u and v), then $\mathcal{F}^r$ cannot be the union of an extended extremal path with dangling paths attached to it. Hence, the algorithm can reject the instance in this case. On the other hand, if both T_u and T_v are paths, the algorithm can arbitrarily choose one of them as the desired extremal path and the other as a dangling path; these two paths can always be embedded as zig-zag paths going from left to right, without creating fan crossings. Finally, assume without loss of generality, that T_u is the only tree (among T_u and T_v) that contains vertices of degree larger than two (in addition to u). Denote by W the set of vertices of degree at least three in T_u (including u). The desired extremal path (if any) must contain all vertices in W. If this is not the case, the algorithm rejects the instance. Otherwise, let $\overline{P'}$ be the path in T_u starting from u and

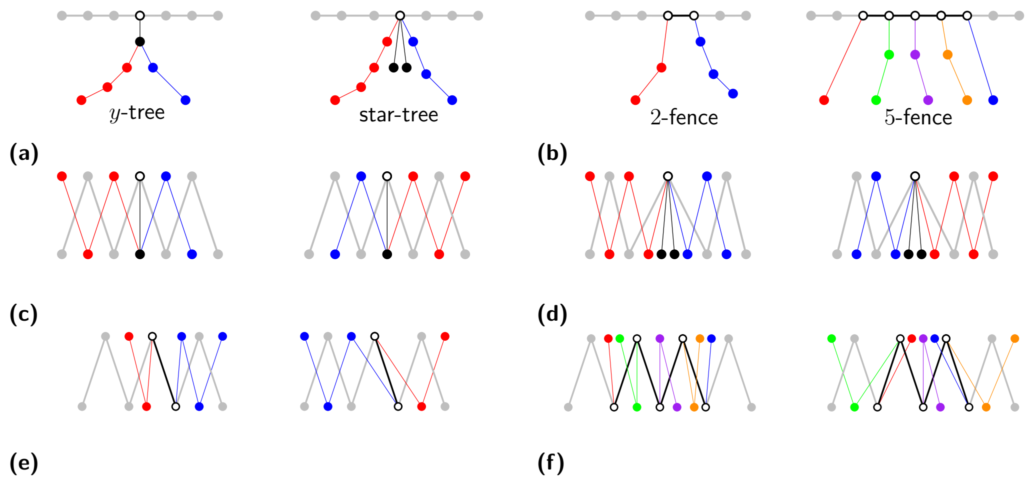


Figure 14 (a) Examples of a y-tree and a star-tree; (b) Examples of a 2-fence and of a 5-fence; (b)-(e) Two possible 2-layer RAC embeddings for each of the structures in (a) and (b).

containing all the vertices of W; also, let w be the last vertex of W encountered along $\overline{P'}$. The desired extremal path $\overline{P}$ must be obtained by extending $\overline{P'}$ with a path attached to w. Let $\mathcal{P}$ denote the set of paths attached to w and extending $\overline{P'}$. Note that, there can be at most three paths in $\mathcal{P}$ with more than one edge, because, by Property 19, there can be at most two dangling paths of length larger than one attached to w. We can get $\overline{P}$ by extending $\overline{P'}$ with an arbitrarily chosen path $P \in \mathcal{P}$ of maximum length. Indeed, suppose that there exists a 2-layer 2^+-real face embedding γ where $\overline{P}$ is extended with a path $P' \in \mathcal{P}$ shorter than P. We can always obtain from γ a new 2-layer embedding by exchanging P' with P, possibly after a horizontal flip of these two paths. Since P' is shorter than P, such a flip can only reduce the number of crossings in γ and does not create two dangling paths that cross each other; thus the new embedding remains a 2^+-real face embedding.

Case 2 – G is a tree. In this case, if γ is a 2-layer 2^+-real face embedding of G, the skeleton $\text{skel}_\gamma(G)$ of γ is simply a zig-zag path, and the graph structures formed by the dangling paths attached to $\text{skel}_\gamma(G)$ must be star-trees, or y-trees, or k-fences. Since, by Lemma 10, γ is also a 2-layer RAC embedding, the algorithm can just apply the procedure of Step 2.3 on every path of G that is a candidate skeleton for a 2-layer RAC embedding of G. In [9], it is proved that the number of such candidate skeletons is bounded by a small constant (precisely, there are at most 49 candidate skeletons).

Time Complexity. About the time complexity of the described algorithm, consider first the case in which G is not a tree. The recursive removal of degree-1 vertices in Step 1 is executed in $O(n)$ time, and testing if the resulting graph admits a 2-layer RAC embedding is done in $O(n)$ time by the algorithm in [9]. Hence, Step 1 takes overall $O(n)$ time.

About Step 2, we have that: (i) Steps 2.1 and 2.2 can be easily executed in $O(n)$ time by visiting the subgraphs in the set $\mathcal{F} \setminus \{\mathcal{F}^r, \mathcal{F}^\ell\}$. (ii) In Step 2.3, we apply the $O(n)$-time algorithm in [9] to test whether all subgraphs attached to an in-between path are feasible structures and to partition W into maximal subsequences $W_1, W_2, \ldots, W_q$. Then, the subsequent greedy procedure that processes $W_1, W_2, \ldots, W_q$ from left to right can be executed in linear time, because for each W_j the algorithm evaluates a constant number of candidate embeddings. Hence, Steps 2 takes $O(n)$ time.

About Step 3, the procedure described to find a candidate extremal path $\overline{P}$ is executed in linear time by simply visiting $\mathcal{F}^r$ and $\mathcal{F}^\ell$. The remaining part of this step uses the same strategy as Steps 2.2 and 2.3, thus it takes $O(n)$ time.

Finally, if G is a tree, the algorithm computes all candidate skeletons, which are bounded by a constant number. This is done through the $O(n)$-time procedure in [9]. For each candidate skeleton, the algorithm uses the strategy of Steps 2.2, which takes $O(n)$ time. ◀

4.2.2 Optimal 2-layer 1^+-real face graphs

To derive a linear-time recognition algorithm for optimal 2-layer 1^+-real face graphs, we prove several structural properties of these graphs. We denote by G an optimal 2-layer 1^+-real face graph and by Γ a 2-layer 1^+-real face drawing of G. Since the removal of vertices or edges from Γ yields a 2-layer 1^+-real face drawing, any subgraph of G is also 2-layer 1^+-real face.

▶ **Property 24.** *The minimum vertex-degree of G is* 2.

Proof. Assume for a contradiction that G has a vertex u with $\deg(u) = 1$. Let H be the graph obtained from G by removing vertex u, that is, $H = G \setminus \{u\}$. Since G has n vertices and $2n - 4$ edges, graph H has $n - 1$ vertices and $2n - 3$ edges. By Theorem 8, this is a contradiction, since graph H is a 2-layer 1^+-real face graph (as a subgraph of G). ◀

▶ **Property 25.** *The maximum vertex-degree of G is* 4.

Proof. Assume to the contrary that G has a vertex u with $\deg(u) \geq 5$. To derive a contradiction, we first consider the case in which $\deg(u) = 5$. Assume without loss of generality that u belongs to layer L_1 of Γ and let $u_1, \ldots, u_5$ be the neighbors of u in L_2 in this left-to-right order. We observe that vertex u_3 cannot have two neighbors, say v and w, that are both either to the left or to the right of u in L_1, as otherwise the edges (u_3, v) and (u_3, w) together with the edges (u, u_1) and (u, u_2) in the former case or with the edges (u, u_4) and (u, u_5) in the latter case would form a face in Γ that does not contain a real-vertex on its boundary (see the gray colored face in Figure 15a); a contradiction. This implies that $\deg(u_3) \leq 3$, namely, u_3 can be adjacent to u, to a vertex v to the left of u, and to a vertex w to the right of u in L_1; see Figure 15b. In particular, if (u_3, v) belongs to G, then v cannot be connected to a vertex to the right of u_3 in L_2, as otherwise this connection together with the edges (u_3, v), (u, u_1), and (u, u_2) would form a face in Γ without real-vertices on its boundary (see the gray colored face in Figure 15b); a contradiction. Symmetrically, if (u_3, w) belongs to G, then w cannot be connected to a vertex to the left of u_3 in L_2. Let H be the graph obtained by (i) removing vertex u_3 from G, (ii) adding the edge (u_4, v), if the edge (u_3, v) belongs to G, and (iii) adding the edge (u_2, w), if the edge (u_3, w) belongs to G. Since graph G is a 2-layer 1^+-real face graph, graph H is a 2-layer 1^+-real face graph, as well. Since G has n vertices and $2n - 4$ edges, graph H has $n - 1$ vertices and $2n - 3$ edges. By Theorem 8, this is a contradiction, since graph H is a 2-layer 1^+-real face graph.

To complete the proof, consider the case in which $\deg(u) \geq 6$. Let $u_1, \ldots, u_{\deg(u)}$ be the neighbors of u in L_2 in this left-to-right order. Vertices u_3 and u_4 can have neither a common neighbor nor two distinct neighbors that are both to the left or to the right of u in L_1, as otherwise this would imply a face in Γ with no real-vertex on its boundary; see Figures 15d and 15e, respectively. Further, as in the case in which $\deg(u) = 5$, we can prove that neither u_3 nor u_4 can have two neighbors that are both to the left or both to the right of u in L_1; see Figure 15a. Hence, by Property 24, each of u_3 and u_4 has degree exactly 2 in G. In particular, there exist two vertices v and w on opposite sides of u along L_1, such that u_3 is adjacent to v and u_3 is adjacent to w. Let H be the graph obtained by removing vertex

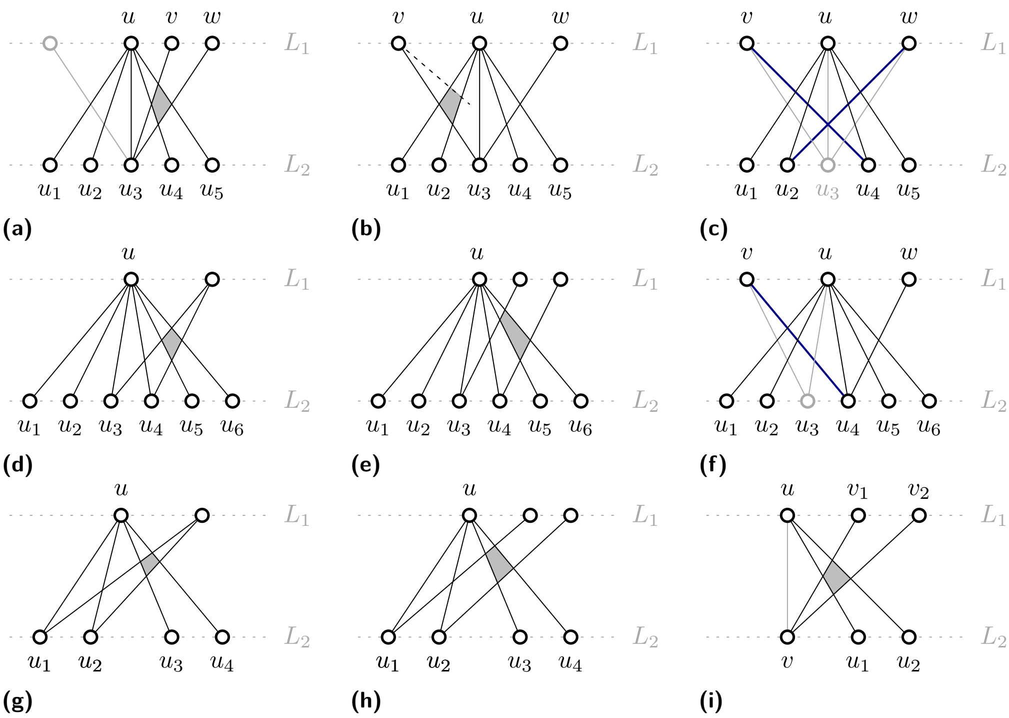


Figure 15 Illustrations for the proof of Theorem 31.

u_3 from G and by adding the edge (u_4, v); see Figure 15f. Since G is a 2-layer 1^+-real face graph, graph H is a 2-layer 1^+-real face graph, as well. Since G has n vertices and $2n-4$ edges, graph H has $n-1$ vertices and $2n-3$ edges, which again contradicts Theorem 8. ◀

▶ **Property 26.** *Neither the leftmost nor the rightmost vertex of each layer in* Γ *has degree* 4.

Proof. Assume to the contrary that the leftmost vertex, say u, of L_1 has degree 4 and let $u_1, \ldots, u_4$ be its neighbors in L_2 in this left-to-right order. By Property 24, $\deg(u_1) \geq 2$ and $\deg(u_2) \geq 2$; hence, u_1 and u_2 either have a common neighbor to the right of u in L_1 or have two distinct neighbors that are both to the right of u in L_1. Both cases, however, cause a face without real-vertices in Γ; see Figures 15g and 15h, respectively; a contradiction. ◀

▶ **Property 27.** *Either the leftmost (rightmost) vertex of* L_1 *or the leftmost (rightmost) vertex of* L_2 *in* Γ *has degree* 2.

Proof. Assume to the contrary that neither the leftmost vertex u in L_1 nor the leftmost vertex v in L_2 has degree 2. By Properties 24–26, $\deg(u) = \deg(v) = 3$. Let u_1 and u_2 be two neighbors of u in L_2, and let v_1 and v_2 be two neighbors of v in L_1, such that $\{u, v\} \cap \{u_1, u_2, v_1, v_2\} = \emptyset$. The edges (u, u_1), (u, u_2), (v, v_1), and (v, v_2) form a face in Γ without real-vertices (see Figure 15h); a contradiction. ◀

▶ **Property 28.** *The leftmost (rightmost) vertex of* L_1 *and the leftmost (rightmost) vertex of* L_2 *in* Γ *are adjacent.*

Proof. If the leftmost (resp. rightmost) vertices of L_1 and L_2 are not adjacent, one can connect them without introducing any crossing, which contradicts that G is optimal. ◀

▶ **Property 29.** *The leftmost (rightmost) vertex of L_1 and the leftmost (rightmost) vertex of L_2 in Γ cannot be both of degree 2.*

Proof. Assume to the contrary that the leftmost vertex u in L_1 and the leftmost vertex v in L_2 are both of degree 2. By Property 28, u and v are adjacent. Let H be the subgraph of G obtained by removing u and v from G. Since G has n vertices and $2n-4$ edges, it follows that H has $n-2$ vertices and $2n-7$ edges, which contradicts Theorem 8. ◀

▶ **Property 30.** *If G has at least seven vertices, the vertex to the right (left) of the leftmost (rightmost) degree-2 vertex in Γ has degree 3.*

Proof. By Properties 26–28, the leftmost two vertices of G are of degree 2 and 3, and the same holds for the rightmost two vertices of G. Since by Property 25 the maximum degree of G is 4 and since G has n vertices and $2n-4$ edges, by the hand-shaking lemma G has two more vertices of degree 3 (besides the two aforementioned extreme ones), while each of the remaining $n-6$ vertices has degree 4. Assume by contradiction that the vertex to the right of the leftmost degree-2 vertex has degree 4. Let H be the graph obtained by removing the leftmost degree-2 vertex from G. The obtained subgraph H has $n-1$ vertices and $2n-6 = 2(n-1)-4$ edges, thus it is still optimal. Hence, it satisfies Properties 26 and 28. In particular, none of its two leftmost vertices can be of degree 4, a contradiction. ◀

▶ **Theorem 31.** *Let G be an n-vertex bipartite graph. There exists an $O(n)$-time algorithm that tests whether G is an optimal 2-layer 1^+-real face graph, and that computes a 2-layer 1^+-real face drawing of G in one exists.*

Proof. Let $G = (V_1 \cup V_2, E)$ be an n-vertex optimal 2-layer 1^+-real face graph and let Γ be a 2-layer 1^+-real face drawing of G. The vertex-degree of G ranges between 2 and 4 (Properties 24 and 25). However, neither the leftmost nor the rightmost vertex of each of the layers L_1 and L_2 of Γ has degree 4 (Property 26). In particular, assuming that the graph has at least seven vertices, the leftmost (rightmost) two vertices of L_1 and L_2 are adjacent, such that one of them has degree 2 and the other has degree 3 (Properties 27, 28, and 30).

From these properties, we can derive our linear-time recognition algorithm. If $n \leq 6$, then we can check whether G is an optimal 2-layer 1^+-real face graph by generating all its 2-layer embeddings and checking whether at least one of them is a 2-layer 1^+-real face. If $n \geq 7$, then we identify one of its vertices of degree 2, say v. If there is no such vertex, then the instance is rejected (by Property 27). Otherwise, we additionally check whether v is neighboring a degree-3 vertex, say w. If no such vertex exists, then the instance is rejected (by Properties 24, 26, and 29). Otherwise, we remove v from G and recursively check whether the obtained instance is a 2-layer 1^+-real face graph starting now from w (which has degree 2). The implementation is straightforward, and the algorithm works in $O(n)$ time. The correctness follows from a direct application of Properties 24–30. ◀

5 Open Problems

A question that directly stems from our research is whether 2-layer 1^+-real face graphs can be recognized efficiently. In the unconstrained scenario, are there subfamilies of k^+-real face graphs that can be recognized efficiently? Also, are there meaningful parameterizations that make the recognition problem tractable?

References

1 Abu Reyan Ahmed, Patrizio Angelini, Michael A. Bekos, Giuseppe Di Battista, Michael Kaufmann, Philipp Kindermann, Stephen G. Kobourov, Martin Nöllenburg, Antonios Symvonis, Anaïs Villedieu, and Markus Wallinger. Splitting vertices in 2-layer graph drawings. *IEEE Computer Graphics and Applications*, 43(3):24–35, 2023. doi:10.1109/MCG.2023.3264244.

2 Michael A. Bekos, Sabine Cornelsen, Luca Grilli, Seok-Hee Hong, and Michael Kaufmann. On the recognition of fan-planar and maximal outer-fan-planar graphs. In Christian A. Duncan and Antonios Symvonis, editors, *Graph Drawing - 22nd International Symposium, GD 2014, Würzburg, Germany, September 24-26, 2014, Revised Selected Papers*, volume 8871 of *Lecture Notes in Computer Science*, pages 198–209. Springer, 2014. doi:10.1007/978-3-662-45803-7_17.

3 Carla Binucci, Markus Chimani, Walter Didimo, Martin Gronemann, Karsten Klein, Jan Kratochvíl, Fabrizio Montecchiani, and Ioannis G. Tollis. 2-layer fan-planarity: From caterpillar to stegosaurus. In *GD*, volume 9411 of *Lecture Notes in Computer Science*, pages 281–294. Springer, 2015. doi:10.1007/978-3-319-27261-0_24.

4 Carla Binucci, Markus Chimani, Walter Didimo, Martin Gronemann, Karsten Klein, Jan Kratochvíl, Fabrizio Montecchiani, and Ioannis G. Tollis. Algorithms and characterizations for 2-layer fan-planarity: From caterpillar to stegosaurus. *J. Graph Algorithms Appl.*, 21(1):81–102, 2017. doi:10.7155/jgaa.00398.

5 Carla Binucci, Giuseppe Di Battista, Walter Didimo, Vida Dujmović, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Graphs drawn with some vertices per face: Density and relationships. *IEEE Access*, 12:68828–68846, 2024. doi:10.1109/ACCESS.2024.3401078.

6 Carla Binucci, Giuseppe Di Battista, Walter Didimo, Seok-Hee Hong, Michael Kaufmann, Giuseppe Liotta, Pat Morin, and Alessandra Tappini. Nonplanar graph drawings with k vertices per face. In Daniël Paulusma and Bernard Ries, editors, *Graph-Theoretic Concepts in Computer Science - 49th International Workshop, WG 2023, Fribourg, Switzerland, June 28-30, 2023, Revised Selected Papers*, volume 14093 of *Lecture Notes in Computer Science*, pages 86–100. Springer, 2023. doi:10.1007/978-3-031-43380-1_7.

7 Carla Binucci, Emilio Di Giacomo, Walter Didimo, Fabrizio Montecchiani, Maurizio Patrignani, Antonios Symvonis, and Ioannis G. Tollis. Fan-planarity: Properties and complexity. *Theor. Comput. Sci.*, 589:76–86, 2015. doi:10.1016/j.tcs.2015.04.020.

8 Franz J. Brandenburg. Recognizing optimal 1-planar graphs in linear time. *Algorithmica*, 80(1):1–28, 2018. doi:10.1007/S00453-016-0226-8.

9 Emilio Di Giacomo, Walter Didimo, Peter Eades, and Giuseppe Liotta. 2-layer right angle crossing drawings. *Algorithmica*, 68(4):954–997, 2014. doi:10.1007/S00453-012-9706-7.

10 Walter Didimo. Density of straight-line 1-planar graph drawings. *Inf. Process. Lett.*, 113(7):236–240, 2013. doi:10.1016/j.ipl.2013.01.013.

11 Walter Didimo. Right angle crossing drawings of graphs. In *Beyond Planar Graphs*, pages 149–169. Springer, 2020. doi:10.1007/978-981-15-6533-5_9.

12 Walter Didimo, Peter Eades, and Giuseppe Liotta. Drawing graphs with right angle crossings. *Theor. Comput. Sci.*, 412(39):5156–5166, 2011. doi:10.1016/j.tcs.2011.05.025.

13 Walter Didimo, Giuseppe Liotta, and Fabrizio Montecchiani. A survey on graph drawing beyond planarity. *ACM Comput. Surv.*, 52(1):4:1–4:37, 2019. doi:10.1145/3301281.

14 Peter Eades, Brendan D. McKay, and Nicholas C. Wormald. On an edge crossing problem. In *9th Australian Computer Science Conference*, pages 327–334, 1986. URL: https://api.semanticscholar.org/CorpusID:116397155.

15 Peter Eades and Sue Whitesides. Drawing graphs in two layers. *Theor. Comput. Sci.*, 131(2):361–374, 1994. doi:10.1016/0304-3975(94)90179-1.

16 Peter Eades and Nicholas C. Wormald. Edge crossings in drawings of bipartite graphs. *Algorithmica*, 11(4):379–403, 1994. doi:10.1007/BF01187020.

17 M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
18 Seok-Hee Hong. Beyond planar graphs: Introduction. In *Beyond Planar Graphs*, pages 1–9. Springer, 2020. doi:10.1007/978-981-15-6533-5_1.
19 Seok-Hee Hong, Michael Kaufmann, Stephen G. Kobourov, and János Pach. Beyond-planar graphs: Algorithmics and combinatorics (Dagstuhl Seminar 16452). *Dagstuhl Reports*, 6(11):35–62, 2016. doi:10.4230/DagRep.6.11.35.
20 Seok-Hee Hong and Hiroshi Nagamochi. A linear-time algorithm for testing full outer-2-planarity. *Discret. Appl. Math.*, 255:234–257, 2019. doi:10.1016/J.DAM.2018.08.018.
21 Seok-Hee Hong and Takeshi Tokuyama, editors. *Beyond Planar Graphs, Communications of NII Shonan Meetings*. Springer, 2020. doi:10.1007/978-981-15-6533-5.
22 Michael Jünger and Petra Mutzel. 2-layer straightline crossing minimization: Performance of exact and heuristic algorithms. *J. Graph Algorithms Appl.*, 1(1):1–25, 1997. doi:10.7155/JGAA.00001.
23 Michael Kaufmann, Boris Klemz, Kristin Knorr, Meghana M. Reddy, Felix Schröder, and Torsten Ueckerdt. The density formula: One lemma to bound them all. *CoRR*, abs/2311.06193, 2023. doi:10.48550/arXiv.2311.06193.
24 Stephen G. Kobourov, Giuseppe Liotta, and Fabrizio Montecchiani. An annotated bibliography on 1-planarity. *Comput. Sci. Rev.*, 25:49–67, 2017. doi:10.1016/j.cosrev.2017.06.002.
25 Giordano Da Lozzo, Vít Jelínek, Jan Kratochvíl, and Ignaz Rutter. Planar embeddings with small and uniform faces. In Hee-Kap Ahn and Chan-Su Shin, editors, *Algorithms and Computation - 25th International Symposium, ISAAC 2014, Jeonju, Korea, December 15-17, 2014, Proceedings*, volume 8889 of *Lecture Notes in Computer Science*, pages 633–645. Springer, 2014. doi:10.1007/978-3-319-13075-0_50.
26 János Pach and Géza Tóth. Graphs drawn with few crossings per edge. *Combinatorica*, 17(3):427–439, 1997. doi:10.1007/BF01215922.
27 Vicente Valls, Rafael Martí, and Pilar Lino. A branch and bound algorithm for minimizing the number of crossing arcs in bipartite graphs. *European Journal of Operational Research*, 90(2):303–319, 1996. doi:10.1016/0377-2217(95)00356-8.
28 David R. Wood. 2-layer graph drawings with bounded pathwidth. *J. Graph Algorithms Appl.*, 27(9):843–851, 2023. doi:10.7155/JGAA.00647.