

Hierarchical Partial Planarity

Patrizio Angelini, Michael A. Bekos

Wilhelm-Schickhard-Institut für Informatik, Universität Tübingen, Germany
{angelini,bekos}@informatik.uni-tuebingen.de

Abstract

In this paper we consider graphs whose edges are associated with a degree of *importance*, which may depend on the type of connections they represent or on how recently they appeared in the scene, in a streaming setting. The goal is to construct layouts of these graphs in which the readability of an edge is proportional to its importance, that is, more important edges have fewer crossings. We formalize this problem and study the case in which there exist three different degrees of importance. We give a polynomial-time testing algorithm when the graph induced by the two most important sets of edges is biconnected. We also discuss interesting relationships with other constrained-planarity problems.

1 Introduction

Describing a graph in terms of a stream of nodes and edges, arriving and leaving at different time instants, is becoming a necessity for application domains where massive amounts of data, too large to be stored, are produced at a very high rate. The problem of visualizing graphs under this streaming model has been introduced only recently.

In particular, the first step in this direction was performed in [7], where the problem of drawing trees whose edges arrive one-by-one and disappear after a certain amount of steps has been studied, from the point of view of the area requirements of straight-line planar drawings. Later on, it was proved [18] that polynomial area could be achieved for trees, tree-maps, and outerplanar graphs if a small number of vertex movements are allowed after each update. The problem has also been studied [13] for general planar graphs, relaxing the requirement that edges have to be straight-line.

In this paper we introduce a problem motivated by this model, and in particular by the fact that the *importance* of vertices and edges in the scene decays with time. In fact, as soon as an edge appears, it is important to let the user clearly visualize it, possibly at the cost of moving “older” edges in the more cluttered part of the layout, which may be unavoidable if the graph is large or dense. The idea is that the user may not need to *see* the connection between two vertices, as she *remembers* it from the previous steps.

Visually, one could associate the decreasing importance of an edge with its fading; theoretically, one could associate it with the fact that it becomes more acceptable to let

it participate in some crossings. As a general framework for this kind of problems, we associate a weight $w(e)$ to every edge $e \in E$ and define a function $f : E \times E \rightarrow \{\text{YES}, \text{NO}\}$ that, given a pair of edges e and e' , determines whether it is allowed to have a crossing between e and e' based on their weights. Of course, if no assumption is made on function $f(\cdot)$, this model allows to encode instances of the NP-complete problem WEAK REALIZABILITY [21], in which the pairs of edges that are allowed to cross are explicitly given as part of the input. On the other hand, already the “natural” assumption that, if an edge e is allowed to cross an edge e' , then it is also allowed to cross any edge e'' such that $w(e'') \leq w(e')$, could potentially make the problem tractable.

As a first step towards a formalization of this general idea, we introduce problem HIERARCHICAL PARTIAL PLANARITY, which takes as input a graph $G = (V, E = E_p \cup E_s \cup E_t)$ whose edges are partitioned into the *primary* edges in E_p , the *secondary* edges in E_s , and the *tertiary* edges in E_t . The goal is to construct a drawing of G in which the primary edges are crossing-free, the secondary edges can only cross tertiary edges, while these latter edges can also cross one another. We say that any crossing that involves a primary edge or two secondary ones is *forbidden*. We remark that this problem can be easily modeled under the general framework we described above. Namely, we can say that all edges in E_p , E_s , and E_t have weights 4, 2, and 1, respectively, and function $f(\cdot)$ is such that $f(e, e') = \text{YES}$ if and only if $w(e) + w(e') \leq 3$.

We observe that our problem is a generalization of the recently introduced PARTIAL PLANARITY problem [1, 22], in which the edges of a certain subgraph of a given graph must not be involved in any crossings. An instance of this problem is in fact an instance of our problem only composed of edges in E_p and E_t .

Our main contribution is an $O(|V|^3 \cdot |E_t|)$ -time algorithm for HIERARCHICAL PARTIAL PLANARITY when the graph induced by the primary and the secondary edges is biconnected (see Section 4). Our result builds upon a formulation of the problem in terms of a *constrained-planarity* problem, which we believe to be interesting in its own. Our algorithm for this constrained-planarity problem is based on the use of SPQR-trees [14, 15]. This formulation also allows us to uncover interesting relationships with other important graph planarity problems, like PARTIALLY EMBEDDED PLANARITY [4, 20] and SIMULTANEOUS EMBEDDING WITH FIXED EDGES [8, 11] (see Section 3).

In Section 2 we give definitions, and in Section 5 we conclude with open problems.

2 Preliminaries

A graph $G = (V, E)$ containing neither loops nor multiple edges is *simple*. We consider simple graphs, if not otherwise specified. A *drawing* Γ of G maps each vertex of G to a point in the plane and each edge of G to a Jordan curve between its two end-points.

A drawing is *planar* if no two edges cross except, possibly, at common endpoints. A planar drawing partitions the plane into connected regions, called *faces*. The unbounded one is called *outer face*. A graph is *planar* if it admits a planar drawing. A *planar embedding* of a planar graph is an equivalence class of planar drawings that define the same set of faces and outer face. Let H be a subgraph of a planar graph G , and let \mathcal{G} be

a planar embedding of G . We call *restriction* of \mathcal{G} to H the planar embedding of H that is obtained by removing the edges of $G \setminus H$ from \mathcal{G} (and potential isolated vertices).

A graph is *connected* if for any pair of vertices there is a path connecting them. A graph is *k-connected* if the removal of $k - 1$ vertices leaves it connected. A 2- or 3-connected graph is also referred to as *biconnected* or *triconnected*, respectively.

The SPQR-tree \mathcal{T} of a biconnected graph G is a labeled tree representing the decomposition of G into its triconnected components [14, 15]. Every triconnected component of G is associated with a node μ in \mathcal{T} . The triconnected component itself is referred to as the *skeleton* of μ , denoted by G_μ^{skel} , whose edges are called *virtual edges*. A node $\mu \in \mathcal{T}$ can be of one of four different types: (i) *S-node*, if G_μ^{skel} is a simple cycle of length at least 3; (ii) *P-node*, if G_μ^{skel} is a bundle of at least three parallel edges; (iii) *Q-node*, if G_μ^{skel} consists of two parallel edges; (iv) *R-node*, if G_μ^{skel} is a simple triconnected graph. The set of leaves of \mathcal{T} coincides with the set of Q-nodes, except for one arbitrary Q-node ρ , which is selected as the root of \mathcal{T} . Also, neither two S-nodes, nor two P-nodes are adjacent in \mathcal{T} . Each virtual edge in G_μ^{skel} corresponds to a node ν that is adjacent to μ in \mathcal{T} , more precisely, to another virtual edge in G_ν^{skel} . In particular, the skeleton of each node μ (except the one of ρ) contains a virtual edge, called *reference edge* and denoted by $ref(\mu)$, that has a counterpart in the skeleton of its parent. The endvertices of $ref(\mu)$ are the *poles* of μ . The subtree \mathcal{T}_μ of \mathcal{T} rooted at μ induces a subgraph G_μ^{pert} of G , called *pertinent*, which is described by \mathcal{T}_μ in the decomposition. The SPQR-tree of G is unique, up to the choice of the root, and can be computed in linear time [19].

3 Problem Formulation & Relationships to Other Problems

In this section we define a problem, called FACIAL-CONSTRAINED CORE PLANARITY, that will serve as a tool to solve HIERARCHICAL PARTIAL PLANARITY and to uncover interesting relationships with other important graph planarity problems. This problem takes as input a graph $G = (V, E_1 \cup E_2)$ and a set $W \subseteq V \times V$ of pairs of vertices. Let H be the subgraph of G induced by the edges in E_1 , which we call *core* of G . The goal is to construct a planar embedding \mathcal{G} of G whose restriction \mathcal{H} to H is such that, for each pair $\langle u, v \rangle \in W$, there exists a face of \mathcal{H} that contains both u and v .

Theorem 1. *Problems FACIAL-CONSTRAINED CORE PLANARITY and HIERARCHICAL PARTIAL PLANARITY are linear-time equivalent.*

Proof. We show how to construct in linear time an instance $\langle G' = (V, E_1 \cup E_2), W \rangle$ of FACIAL-CONSTRAINED CORE PLANARITY starting from an instance $G = (V, E_p \cup E_s \cup E_t)$ of HIERARCHICAL PARTIAL PLANARITY. Graph G' has the same vertex-set V as G . Also, we set $E_1 = E_p$ and $E_2 = E_s$. Finally, for each edge $(u, v) \in E_t$, we add pair $\langle u, v \rangle$ to W . The reduction in the opposite direction is symmetric.

Suppose that $\langle G', W \rangle$ is a positive instance, and let \mathcal{G}' be a corresponding planar embedding of G' . We show how to construct a drawing Γ of G not containing any forbidden crossing. First, initialize Γ to a planar drawing of G' whose embedding is \mathcal{G}' . Note that restricting Γ to the core H' of G' yields a planar drawing Γ' of H' in which,

for each pair $\langle u, v \rangle \in W$, there exists a face $f_{u,v}$ of \mathcal{H}' that contains both u and v . This implies that it is possible to draw edge $(u, v) \in E_t$ in Γ' as a curve from u to v lying completely in the interior of $f_{u,v}$, and hence not crossing any primary edge. Repeating this operation for every pair in W yields a drawing Γ with no forbidden crossings.

Suppose that G is a positive instance, and let Γ be the corresponding drawing of G . We show how to construct an embedding \mathcal{G}' of G' such that for every pair $\langle u, v \rangle \in W$, vertices u and v lie in the same face of \mathcal{H}' . First, note that the drawing $\Gamma_{p,s}$ induced by the edges in E_p and E_s is planar, due to the definition of HIERARCHICAL PARTIAL PLANARITY. Also, note that $\Gamma_{p,s}$ is a planar drawing of G' , since $E_1 = E_p$ and $E_2 = E_s$. Let \mathcal{G}' be the planar embedding of G' corresponding to $\Gamma_{p,s}$. Let \mathcal{H}' be the restriction of \mathcal{G}' to H' . Consider a pair $\langle u, v \rangle \in W$ and let $e = (u, v)$ be the corresponding tertiary edge of G . Since e can be drawn in $\Gamma_{p,s}$ without crossing any primary edge, vertices u and v are incident to the same face of \mathcal{H}' . This concludes the proof. \square

In the following, we describe relationships between HIERARCHICAL PARTIAL PLANARITY and other important graph planarity problems, as PARTIAL PLANARITY [1, 22] and PARTIALLY EMBEDDED PLANARITY [4, 20], and SIMULTANEOUS EMBEDDING WITH FIXED EDGES [8].

In PARTIAL PLANARITY [1], given a non-planar graph $G = (V, E)$ and a subset $F \subseteq E$ of its edges, the goal is to compute a drawing Γ of G , if any, in which the edges of F are not crossed by any edge of G . Positive and negative results are given in [1] if the graph induced by F is a connected spanning subgraph of G . In [22], the corresponding decision problem is shown to be polynomial-time solvable. By setting $E_p = F$, $E_s = \emptyset$, and $E_t = E \setminus F$, we can model any instance of PARTIAL PLANARITY as an instance of HIERARCHICAL PARTIAL PLANARITY. We thus have the following.

Theorem 2. PARTIAL PLANARITY *can be reduced in linear time to HIERARCHICAL PARTIAL PLANARITY.*

In PARTIALLY-EMBEDDED PLANARITY [4], given a planar graph G and a planar embedding \mathcal{H} of a subgraph H of G , the goal is to determine whether \mathcal{H} can be extended to a planar embedding of G , and to compute this embedding, if it exists. The problem is linear-time solvable [4] and characterizable in terms of forbidden subgraphs [20]. We prove that HIERARCHICAL PARTIAL PLANARITY can be used to encode instances of PARTIALLY-EMBEDDED PLANARITY in which H is biconnected. Note that this special case is a central ingredient in the algorithm in [4] for the general case.

Theorem 3. PARTIALLY-EMBEDDED PLANARITY *with biconnected H can be reduced in quadratic time to HIERARCHICAL PARTIAL PLANARITY.*

Proof. Let $\langle G' = (V, E), H, \mathcal{H} \rangle$ be an instance of PARTIALLY-EMBEDDED PLANARITY in which H is biconnected. We construct an instance $\langle G = (V, E_1 \cup E_2), W \rangle$ of FACIAL-CONSTRAINED CORE PLANARITY on the same vertex set V as G' , as follows. Set E_1 contains all the edges of E that are contained in H ; set E_2 contains the other ones, that is, $E_2 = E \setminus E_1$. Finally, for every pair of non-adjacent vertices $\langle u, v \rangle$ that are on the

same face of \mathcal{H} , we add a pair $\langle u, v \rangle$ to W . This last step requires quadratic time and guarantees that in the solution of FACIAL-CONSTRAINED CORE PLANARITY, for each face f of \mathcal{H} , all the vertices of f are incident to the same face f' of the planar embedding of the core of G . These vertices appear in the same order along f and f' , since H is biconnected and thus this order is unique. Hence, $\langle G', H, \mathcal{H} \rangle$ is a positive instance if and only if $\langle G, W \rangle$ is. The statement follows by Theorem 1. \square

A *simultaneous embedding* of two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ embeds each graph in a planar way using the same vertex positions for both embeddings; edges are allowed to cross only if they belong to different graphs (see [8] for a survey). Our problem is related to a well-studied version of this problem, called SIMULTANEOUS EMBEDDING WITH FIXED EDGES (SEFE) [3, 5, 9, 10, 11], in which edges that are *common* to both graphs must be embedded in the same way (and hence, cannot be crossed by other edges). So in our setting, these edges correspond to the primary ones. However, to obtain a solution for SEFE, it does not suffice to assume that the *exclusive* edges of G_1 and G_2 are the secondary and tertiary ones, respectively, as we could not guarantee that the edges of G_2 do not cross each other. So, in some sense, our problem seems to be more related to *nearly-planar simultaneous embeddings*, where the input graphs are allowed to cross, as long as they avoid some local crossing configurations, e.g., by avoiding triples of mutually crossing edges [16]. Note that the SEFE problem has also been studied in several settings [2, 6, 12, 17]. An interpretation of PARTIAL PLANARITY, which also extends to HIERARCHICAL PARTIAL PLANARITY, in terms of a special version of SEFE, called SUNFLOWER SEFE [8], was already observed in [1].

The algorithm we present in Section 4 is inspired by an algorithm to decide in linear time whether a pair of graphs admits a SEFE if the common graph is biconnected [5]. The main part of that algorithm is to find an embedding of the common graph in which every pair of vertices that are joined by an exclusive edge are incident to the same face; so, these edges play the role of the pairs in W . In a second step, it checks for crossings between exclusive edges of the same graph. Since the common graph is biconnected, the existence of these crossings does not depend on the choice of the embedding.

Thus, for instances of our problem in which the core H of G is biconnected, we can employ the main part of the algorithm in [5] to find a planar embedding of H in which every two vertices that either are joined by an edge of E_2 or form a pair of W are incident to the same face of H ; note that in this case it is not even needed to perform the second check for the pairs in W . In this paper we extend this result to the case in which H is not biconnected, but it becomes so when adding the edges of E_2 . The main difficulty here is to “control” the faces of H by operating on the embeddings of the biconnected graph G composed of H and of the edges of E_2 . In Section 4 we discuss the problems arising from this fact and our proposed solution.

4 Biconnected Facial-Constrained Core Planarity

In this section, we provide a polynomial-time algorithm for instances $\langle G = (V, E_1 \cup E_2), W \rangle$ of FACIAL-CONSTRAINED CORE PLANARITY in which G is biconnected. Recall that the goal is to find a planar embedding \mathcal{G} of G so that for each pair $\langle x, y \rangle \in W$ vertices x and y lie in the same face of the restriction \mathcal{H} of \mathcal{G} to the core H of G .

4.1 High-Level Description of the Algorithm

We first give a high-level description of our algorithm. We perform a bottom-up traversal of the SPQR-tree \mathcal{T} of G . At each step of the traversal, we consider a node $\mu \in \mathcal{T}$ and we search for an embedding $\mathcal{G}_\mu^{\text{pert}}$ of G_μ^{pert} satisfying the following requirements.

- R.1 For every pair $\langle x, y \rangle \in W$ such that x and y belong to G_μ^{pert} , vertices x and y lie in the same face of the restriction $\mathcal{H}_\mu^{\text{pert}}$ of $\mathcal{G}_\mu^{\text{pert}}$ to the part of the core H in G_μ^{pert} .
- R.2 For every pair $\langle x, y \rangle \in W$ such that exactly one vertex, say x , belongs to G_μ^{pert} , vertex x lies in the outer face of $\mathcal{H}_\mu^{\text{pert}}$ (note that y belongs to $G \setminus G_\mu^{\text{pert}}$).

In general, there may exist several “candidate” embeddings of G_μ^{pert} satisfying R.1 and R.2. If there exists none, the instance is negative. Otherwise, we would like to select one of them and proceed with the traversal. However, while it would be sufficient to select *any* embedding of G_μ^{pert} satisfying R.1, it is possible that some of the embeddings satisfying R.2 are “good”, in the sense that they can be eventually extended to an embedding of G satisfying both R.1 and R.2, while some others are not. Unfortunately, we cannot determine which ones are good at this stage of the algorithm, as this may depend on the structure of a subgraph that is considered later in the traversal. Thus, we have to maintain succinct information to describe the properties of the embeddings of G_μ^{pert} that satisfy R.1 and R.2, so to group these embeddings into equivalence classes.

We denote by x_1, \dots, x_k the vertices belonging to pairs $\langle x_i, y_i \rangle \in W$ such that $x_i \in G_\mu^{\text{pert}}$ and $y_i \notin G_\mu^{\text{pert}}$. In other words, vertices x_1, \dots, x_k are those that must lie on the outer face of $\mathcal{H}_\mu^{\text{pert}}$ due to R.2. To describe the information to maintain, we need the following definition. We say that μ is *non-traversable* if there is a cycle C_μ composed of edges of E_1 that contains both poles u and v of μ , at least one edge of G_μ^{pert} , and at least one of $G \setminus G_\mu^{\text{pert}}$; see Fig. 1a. Otherwise, μ is *traversable*, i.e., either in G_μ^{pert} or in $G \setminus G_\mu^{\text{pert}}$ every path between u and v contains edges of E_2 ; see Fig. 1b.

Intuitively, when μ is non-traversable, cycle C_μ splits the outer face of $\mathcal{H}_\mu^{\text{pert}}$ into two faces f_μ^l and f_μ^r of \mathcal{H} in any planar embedding of G . Hence, R.2 must be refined to take into account the possible partitions of x_1, \dots, x_k with respect to their incidence to f_μ^l and f_μ^r . For a single vertex $x_i \in \{x_1, \dots, x_k\}$ this is not an issue, as a flip of G_μ^{pert} can transform an embedding of G in which x_i is incident to one of these faces into another one in which it is incident to the other face. However, there may exist dependencies among different vertices of $\{x_1, \dots, x_k\}$, given by the structure of G_μ^{pert} , which enforce the relative positions of these vertices with respect to f_μ^l and f_μ^r . More precisely, let $\langle x, y \rangle, \langle x', y' \rangle \in W$ be two pairs such that $x, x' \in G_\mu^{\text{pert}}$ and $y, y' \notin G_\mu^{\text{pert}}$. Then, vertices

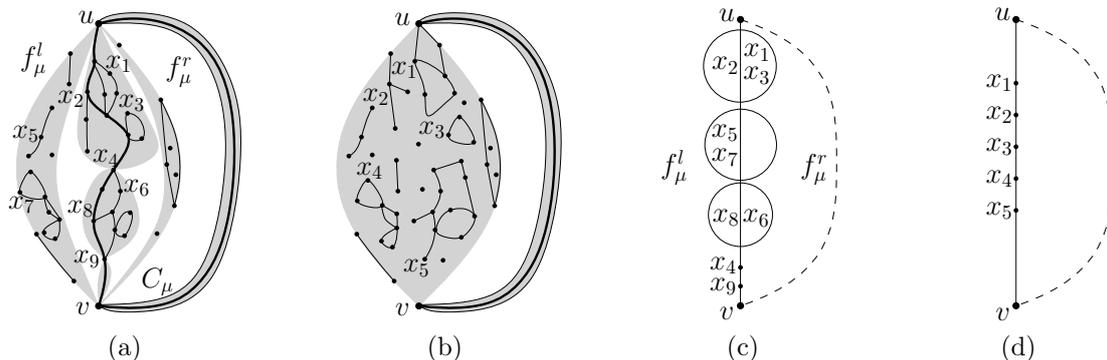


Figure 1: (a–b) Graph G_μ^{pert} when μ is (a) non-traversable and (b) traversable. (c–d) The bags of the nodes in (a) and in (b), respectively. A segment between u and v separates faces f_μ^l and f_μ^r ; each bag B_μ^i is represented by a circle across the segment, with its pockets S_μ^i and T_μ^i on the two sides; the vertices in the special bag \mathcal{B}_μ lie along the segment, as they are incident to both faces.

x and x' may be enforced to be incident to the same face, either f_μ^l or f_μ^r (see x_1 and x_3 in Fig. 1a), they may be enforced to be incident to different faces (see x_2 and x_3 in Fig. 1a), or they may be independent in this respect (see x_1 and x_6 in Fig. 1a).

We encode this information by associating a set of *bags* with μ , which contain vertices x_1, \dots, x_k . Each bag is composed of two *pockets*; all the vertices in a pocket must be incident to the same face of \mathcal{H} in any candidate embedding of G_μ^{pert} , while all the vertices in the other pocket must be incident to the other face. Vertices of different bags are independent of each other. For the vertices of $\{x_1, \dots, x_k\}$ that are incident to both f_μ^l and f_μ^r in any embedding (see x_4 in Fig. 1a), we add a *special* bag, composed of a single set containing all such vertices; note that if a vertex of $\{x_1, \dots, x_k\}$ is a pole of μ , then it belongs to the special bag. See Fig. 1c for the bags of the node in Fig. 1a.

When μ is traversable, instead, the outer face of $\mathcal{H}_\mu^{\text{pert}}$ corresponds to a single face of \mathcal{H} in any planar embedding \mathcal{G} of G . Thus, we do not need to maintain any information about the relative positions of x_1, \dots, x_k , and we can place all of them in the special bag. An illustration of the bags of the node represented in Fig. 1b is given in Fig. 1d.

If the visit of the root ρ of \mathcal{T} at the end of the bottom-up traversal is completed without declaring $\langle G, W \rangle$ as negative, we have that $G_\rho^{\text{pert}} = G$ admits a planar embedding satisfying R.1 and thus $\langle G, W \rangle$ is a positive instance.

As anticipated in Section 3, we discuss two main problems arising when extending the algorithm in [5] for SEFE to solve our problem when H is not biconnected.

First, when H is biconnected it is always possible to decide the flip of every child component for every node that is either an R- or a P-node, but not when it is an S-node. On the other hand, the fact that no two S-nodes can be adjacent to each other in the SPQR-tree ensures that this choice is always fixed in the next step of the algorithm (refer to *visible nodes* in [5]). When H is not biconnected, even the flips of the children of R- and P-nodes (and of S-nodes) may be not uniquely determined. So, there is no guarantee

that a choice for these flips can be done in the next step; in fact, it is sometimes necessary to defer this choice till the end of the algorithm. This comes with another difficulty. In the course of the algorithm, it could be required to make “partial” choices for these flips, in the sense that constraints imposed by the structure of the graph could enforce two or more components to be flipped in the same way (without enforcing, however, a specific flip for them). To encode the possible flips of the components that are enforced by the constraints considered till a certain point of the algorithm, we introduced the bags, which represent the main technical contribution of this work.

Second, the order of the vertices along the faces of H is not unique if H is not biconnected. For FACIAL-CONSTRAINED CORE PLANARITY, this is not an issue, as it is enough that the vertices belonging to the pairs in W share a face, but we do not impose any requirement on their order along these faces. On the other hand, if we were able to also control these orders, we could provide an algorithm for instances of SEFE in which one of the two graphs is biconnected, which would be a significant step ahead in the state of the art for this problem. We recall that an efficient algorithm for this case (even with the additional restriction that the common graph is connected) would imply an efficient algorithm for all the instances in which the common graph is connected (and no restriction on the two input graphs), as observed in [3].

4.2 Detailed Description of the Algorithm

We give the details of the algorithm. Let \mathcal{T} be the SPQR-tree of G , rooted at a Q-node ρ . First, we compute for each node $\mu \in \mathcal{T}$, whether μ is traversable or not, that is, whether there exist two paths composed of edges of H between the poles of μ , one in G_μ^{pert} and one in $G \setminus G_\mu^{pert}$. A naïve approach would be to perform a BFS-visit restricted to the edges of H in each of the two graphs in linear time per node, and thus in total quadratic time. For a linear-time algorithm, we proceed as follows; see also [4]. We traverse \mathcal{T} bottom-up to compute for each node μ whether there exists the desired path in G_μ^{pert} , using the same information computed for its children. Then, with a top-down traversal, we search for the path in $G \setminus G_\mu^{pert}$, using the information computed in the first traversal.

The main part of our algorithm consists of a bottom-up traversal of \mathcal{T} . For a node $\mu \in \mathcal{T}$, let $\langle x_1, y_1 \rangle, \dots, \langle x_k, y_k \rangle$ be all pairs of W such that $x_i \in G_\mu^{pert}$ and $y_i \notin G_\mu^{pert}$. We denote by B_μ^1, \dots, B_μ^q the bags of μ and by \mathcal{B}_μ its special bag; these bags determine a partition of the vertices x_1, \dots, x_k that are required to be on the outer face of \mathcal{H}_μ^{pert} due to R.2. The vertices of each bag $B_\mu^i = \langle S_\mu^i, T_\mu^i \rangle$ are partitioned into its two pockets S_μ^i and T_μ^i ; all vertices of S_μ^i must lie in the same face of \mathcal{H} , either f_μ^l or f_μ^r , while all vertices of T_μ^i must lie on the other face.

We first describe an operation, called MERGE-BAGS, to modify the bags of a node μ in order to satisfy the constraints that may be imposed by R.1 when there exists a pair $\langle x, y \rangle \in W$ such that $x, y \in G_\mu^{pert}$. Refer to Figs. 2a– 2b. In particular, if at least one of x and y belongs to the special bag \mathcal{B}_μ (see $\langle x_4, x_6 \rangle$ in the figure), or if x and y belong to the same pocket of a bag B_μ^i , with $1 \leq i \leq q$, then we do not modify any bag. If $x \in S_\mu^i$ and $y \in T_\mu^i$, for some $1 \leq i \leq q$, or vice versa, then we declare the instance

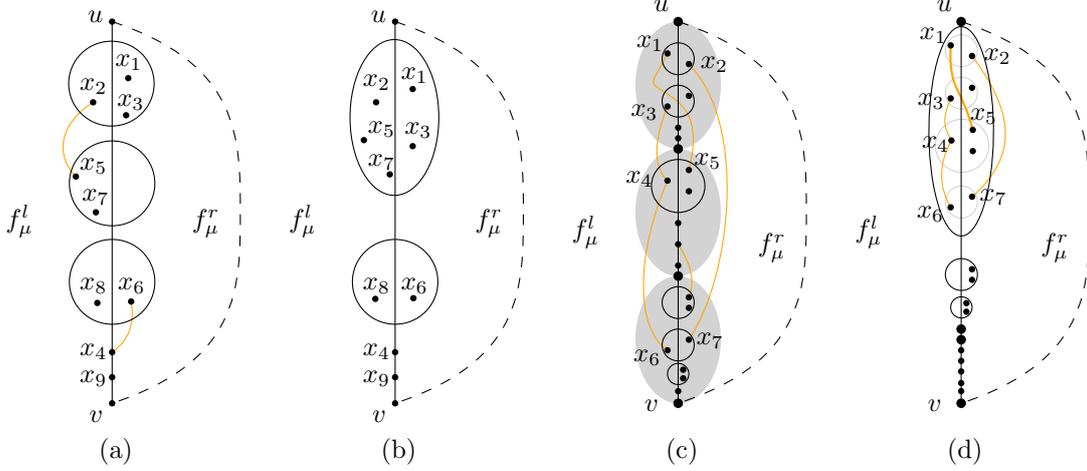


Figure 2: (a) The bags of a node μ and two pairs $\langle x_2, x_5 \rangle, \langle x_4, x_6 \rangle \in W$ (orange curves). (b) The bags of μ after operation MERGE-BAGS. Pair $\langle x_2, x_5 \rangle$ merged two bags, while $\langle x_4, x_6 \rangle$ did not modify any bag, since $x_4 \in \mathcal{B}_\mu$. (c) Initialization of the bags of an S-node μ . (d) The bags of μ after MERGE-BAGS. The instance is negative, as pair $\langle x_1, x_5 \rangle$ is such that $x_1 \in S_\mu^1$ and $x_5 \in T_\mu^1$.

negative. Otherwise, we have $x \in B_\mu^i$ and $y \in B_\mu^j$, for some $1 \leq i \neq j \leq q$, and we merge B_μ^i and B_μ^j into a single bag $B_\mu = \langle S_\mu, T_\mu \rangle$, i.e., we merge into S_μ the pockets of B_μ^i and B_μ^j containing x and y , respectively, and we merge into T_μ the other two pockets of B_μ^i and B_μ^j ; see $\langle x_2, x_5 \rangle$ in the figure. We finally remove $\langle x, y \rangle$ from W and, if there is no other pair in W containing x (resp., y), we remove it from the bag it belongs to.

At each step of the traversal of \mathcal{T} , we consider a node μ , with poles u and v , and children ν_1, \dots, ν_h in \mathcal{T} . We denote by e_i , for $i = 1, \dots, h$, the virtual edge of G_μ^{skel} corresponding to ν_i .

Suppose that μ is a Q-node. If any of the two poles of μ belongs to $\{x_1, \dots, x_k\}$, then we add it to \mathcal{B}_μ , independently of whether μ is traversable or not.

Suppose that μ is an S-node. We initialize special bag \mathcal{B}_μ to the union of the special bags of ν_1, \dots, ν_h . Note that if μ is traversable, then all of its children are traversable. So, in this case, we already have that all vertices x_1, \dots, x_k are in \mathcal{B}_μ . Further, if μ is non-traversable, we add to the set of bags of μ all the non-special bags of its children; see Fig. 2c. Finally, as long as there exists a pair $\langle x, y \rangle \in W$ such that both x and y belong to G_μ^{pert} , we apply operation MERGE-BAGS to $\langle x, y \rangle$. This may result in uncovering a negative instance, but only when μ is non-traversable. See Fig. 2d.

Suppose that μ is an R-node. See Fig. 3a. Let H_μ^{skel} be the graph composed of the vertices of G_μ^{skel} and of the virtual edges corresponding to non-traversable children of μ , plus $ref(\mu)$ if μ is non-traversable; see Fig. 3b. Let \mathcal{H}_μ^{skel} be the restriction of the unique planar embedding of the triconnected graph G_μ^{skel} to H_μ^{skel} . Note that, for each traversable child ν_i of μ , virtual edge e_i is contained in one face f_{ν_i} of \mathcal{H}_μ^{skel} ; in Fig. 3b,

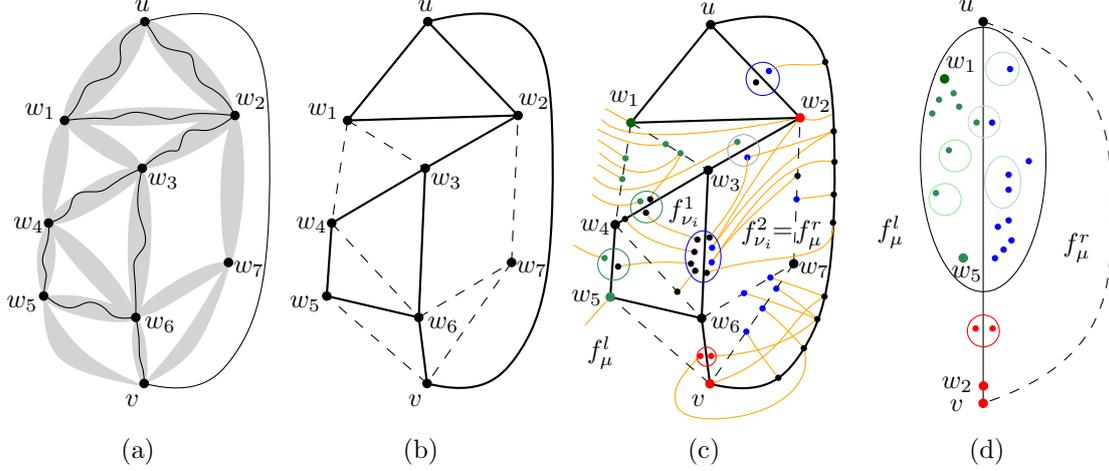


Figure 3: (a) Graph G_μ^{pert} when μ is a non-traversable R-node. (b) Graph H_μ^{skel} (solid) and the traversable children (dashed) of μ . Children corresponding to virtual edges (v, w_6) and (w_2, w_3) are 2-sided. (c) Association of pockets with faces. Blue (green) pockets are associated with f_μ^r (f_μ^l , resp.). Red pockets are not associated. Gray pockets belong to 2-sided children, but they are associated with f_μ^r and f_μ^l . (d) The bags of μ .

(w_4, w_6) is contained in face $\{w_3, w_4, w_5, w_6\}$. For a non-traversable child ν_i , denote by $f_{\nu_i}^1$ and $f_{\nu_i}^2$ the two faces of \mathcal{H}_μ^{skel} virtual edge e_i is incident to. For a vertex $x \in V$ that does not belong to G_μ^{skel} , we denote by $e_\mu(x)$ either the virtual edge e_i , if $x \in G_{\nu_i}^{pert}$, or the virtual edge $ref(\mu)$ representing the parent of μ , if $x \in G \setminus G_\mu^{pert}$.

Suppose that μ is non-traversable; see Fig. 3c. Recall that in this case $ref(\mu) \in H_\mu^{skel}$; let f_μ^l and f_μ^r be the two faces of \mathcal{H}_μ^{skel} incident to $ref(\mu)$. Any other virtual edge e_i of H_μ^{skel} such that $\{f_{\nu_i}^1, f_{\nu_i}^2\} = \{f_\mu^l, f_\mu^r\}$ is called *2-sided*; see $(w_2, w_3), (v, w_6)$ in Fig. 3c.

We consider each pair $\langle x, y \rangle \in W$ such that $x \in G_{\nu_i}^{pert}$, with $1 \leq i \leq h$, and $y \notin G_{\nu_i}^{pert}$. Let $e^x = e_\mu(x)$ and $e^y = e_\mu(y)$. A necessary condition for R.1 and R.2 is that e^y is either contained in or incident to face f_{ν_i} (if ν_i is traversable) or one of $f_{\nu_i}^1$ and $f_{\nu_i}^2$ (if ν_i is non-traversable). If this is not the case, we declare the instance negative.

Another constraint imposed by this pair is the following. Suppose that x belongs to a pocket, say S_{ν_i} , of a bag B_{ν_i} of ν_i (this can only happen if ν_i is non-traversable). If e^x and e^y share exactly one face, say $f_{\nu_i}^1$, then all pairs $\langle x', y' \rangle \in W$ with $x' \in S_{\nu_i}$ must be such that $e_\mu(y')$ is either contained in or incident to f_{ν_i} ; also, all the pairs $\langle x'', y'' \rangle \in W$ with $x'' \in T_{\nu_i}$ must be such that $e_\mu(y'')$ is either contained in or incident to f_{ν_i} . This is due to the fact all the vertices in the same pocket must be incident to the same face of \mathcal{H}_μ^{skel} . So, if this is not the case, we declare the instance negative. Otherwise, we *associate* S_{ν_i} with $f_{\nu_i}^1$ and T_{ν_i} with $f_{\nu_i}^2$. If e^x and e^y share both faces $f_{\nu_i}^1$ and $f_{\nu_i}^2$, instead, we have to postpone the association of S_{ν_i} and T_{ν_i} , as at this point we cannot make a unique choice. Note that an association for these pockets may be performed later, due to another pair of W . Suppose now that x belongs to the special bag \mathcal{B}_{ν_i} of ν_i . Then, we associate

\mathcal{B}_{ν_i} to either f_{ν_i} , if ν_i is traversable, or to both $f_{\nu_i}^1$ and $f_{\nu_i}^2$, if it is non-traversable. This completes the process of pair $\langle x, y \rangle$.

Once all children ν_1, \dots, ν_h of μ have been considered, there may still exist pockets that are not associated. Let S_{ν_i} be one of such pockets, and consider each pair $\langle x, y \rangle \in W$ such that $x \in S_{\nu_i}$. Note that $e_\mu(x)$ shares both faces $f_{\nu_i}^1$ and $f_{\nu_i}^2$ with $e_\mu(y)$. If y belongs to a pocket, say T_{ν_j} , that is associated with one of $f_{\nu_i}^1$ and $f_{\nu_i}^2$, say $f_{\nu_i}^1$, then we associate S_{ν_i} with $f_{\nu_i}^1$ and T_{ν_j} with $f_{\nu_i}^2$. In fact, the association of T_{ν_j} with $f_{\nu_i}^1$ implies that y will be incident to $f_{\nu_i}^1$ in any embedding of G that is a solution for $\langle G, W \rangle$. If two pairs determine different associations for S_{ν_i} and T_{ν_j} , we declare the instance negative.

We repeat the above process as long as there exist pockets that can be associated by means of this procedure. Note that this does not necessarily result in an association for all pockets; however, we can say that all the mandatory choices for G_μ^{pert} have been performed. Consider any of the remaining pockets S_{ν_i} . If ν_i is not 2-sided, then we associate S_{ν_i} with $f_{\nu_i}^1$ and T_{ν_i} with $f_{\nu_i}^2$. This association can be done arbitrarily since its effect is limited to G_μ^{pert} and not to $G \setminus G_\mu^{pert}$, as ν_i is not 2-sided. Then, we propagate this association to other pockets by performing the procedure described above. We repeat this process until the only pockets that are not associated, if any, belong to bags of 2-sided children of μ . Note that the previous arbitrary association cannot be propagated to pockets of 2-sided children, since their virtual edges are only incident to f_μ^l and f_μ^r .

Based on the association of ν_1, \dots, ν_h with the faces of \mathcal{H}_μ^{skel} , we determine the bags of μ ; see Fig. 3d. The special bag \mathcal{B}_μ of μ contains the poles of μ , if they belong to $\{x_1, \dots, x_k\}$, and the union of the special bags of the 2-sided children of μ . Next, we create a bag $B_\mu = \langle S_\mu, T_\mu \rangle$, such that S_μ and T_μ contain all the vertices of the pockets associated with f_μ^l and f_μ^r , respectively. Finally, we add to the set of bags of μ the non-special bags of the 2-sided children of μ whose pockets have not been associated with any face of \mathcal{H}_μ^{skel} (this allows us to postpone their association). Then, we apply operation MERGE-BAGS to all pairs $\langle x, y \rangle \in W$ such that both x and y belong to G_μ^{pert} in order to merge the bags of different 2-sided children of μ (again this may result in uncovering a negative instance). This completes the case in which μ is non-traversable.

It remains to consider the simpler case in which μ is traversable. In this case virtual edge $ref(\mu)$ does not belong to H_μ^{skel} , hence faces f_μ^l and f_μ^r do not exist, and none of the children of μ is 2-sided. This implies that performing all the operations described above results in an association of each pocket and of each special bag of the children of μ with some face of \mathcal{H}_μ^{skel} . Recall that, since μ is traversable, μ has only its special bag \mathcal{B}_μ . We add to \mathcal{B}_μ all the vertices of the pockets and of the special bags that have been associated with the outer face of \mathcal{H}_μ^{skel} . This concludes the R-node case.

Suppose that μ is a P-node. Refer to Fig. 4. We distinguish three cases, based on whether μ has (i) zero, (ii) one, or (iii) more than one non-traversable child.

In Case (i), we have that μ is traversable. So, it has only its special bag \mathcal{B}_μ , in which we add all the vertices of the special bags of its children. Note that all virtual edges in G_μ^{skel} are incident to the same face of \mathcal{H}_μ^{pert} and hence R.1 and R.2 are trivially satisfied.

Next, we consider Case (ii), in which μ has exactly one non-traversable child, say ν_1 ; see Fig. 4a. In this case, μ is non-traversable, since the path of $G \setminus G_{\nu_1}^{pert}$ composed of

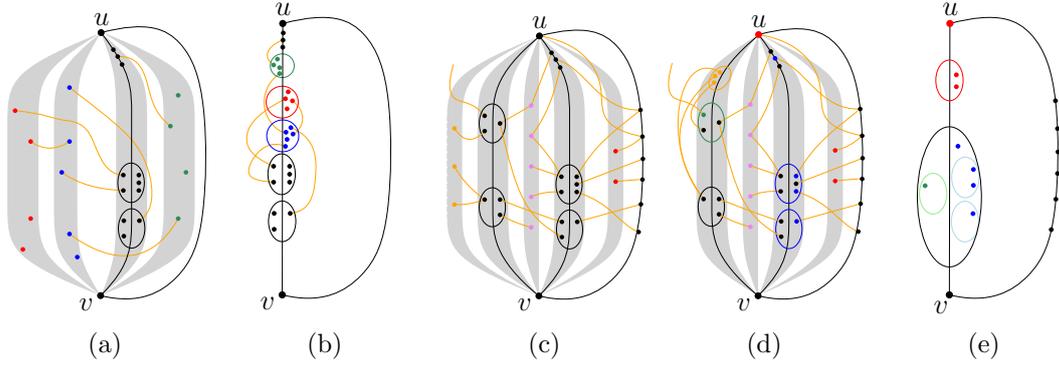


Figure 4: Illustration for the case in which μ is a P-node with: (a-b) one, and (c-e) more than one non-traversable children. The color-scheme of this figure follows the one of Fig. 3

edges of H also belongs to $G \setminus G_\mu^{pert}$. We initialize the set of bags of μ to the set of bags of ν_1 . For each traversable child ν_i , with $i = 2, \dots, h$, we add to μ a new bag B_μ^i , where S_μ^i contains all the vertices in the special bag \mathcal{B}_{ν_i} of ν_i , while T_μ^i is empty; see Fig. 4b. This represents the fact that all the vertices in $G_{\nu_i}^{pert}$ must lie on the same side of the cycle passing through $G_{\nu_1}^{pert}$ and $G \setminus G_\mu^{pert}$ to satisfy R.2. Finally, we apply operation MERGE-BAGS to all pairs $\langle x, y \rangle \in W$ such that both x and y belong to G_μ^{pert} .

Finally, we consider Case (iii), in which μ has more than one non-traversable child; see Figs. 4c-4e. We construct an auxiliary graph G_{aux} with a vertex v_i for each child ν_i of μ , which is colored *black* if ν_i is non-traversable and *white* otherwise. Graph G_{aux} also has a vertex v corresponding to $ref(\mu)$, which is colored black if μ is non-traversable and white otherwise. Then, we consider every pair $\langle x, y \rangle \in W$ such that $x \in G_{\nu_i}^{pert}$, for some child ν_i of μ . If $y \in G_{\nu_j}^{pert}$, for some $j \neq i$, then we add edge (v_i, v_j) to G_{aux} , while if $y \in G \setminus G_\mu^{pert}$, then we add edge (v_i, v) to G_{aux} . If G_{aux} has multiple copies of an edge, we keep only one of them. We assume w.l.o.g. that no two white vertices are adjacent in G_{aux} , as otherwise we could contract them to a new white vertex. In fact, the virtual edges representing traversable children of μ corresponding to adjacent white vertices must be contained in the same face of \mathcal{H}_μ^{pert} , due to R.1.

Consider each white vertex w of G_{aux} . If w has more than two black neighbors, we declare the instance negative, as the virtual edge of the traversable child of μ corresponding to w should share a face in \mathcal{H}_μ^{pert} with more than two virtual edges representing non-traversable children of μ , which is not possible. If w has at most one black neighbor, we remove w from G_{aux} . Finally, if w has exactly two black neighbors b and b' , then we remove w from G_{aux} and we add edge (b, b') to G_{aux} (if it is not present). Once we have considered all white vertices, the resulting graph \bar{G}_{aux} has only black vertices.

We check whether \bar{G}_{aux} is either a cycle through all its vertices or a set of paths (some of which may consist of single vertices). The necessity of this condition can be proved similar to [5]. The only difference is in the edges between black vertices that are introduced due to degree-2 white vertices. Let (b, b') be one of such edges and let

w be the white vertex that was adjacent to b and b' . Also, let e_b , $e_{b'}$, and e_w be the virtual edges representing the children of μ (or virtual edge $ref(\mu)$, if μ is non-traversable) corresponding to b , b' , and w , respectively. Then, e_b and $e_{b'}$ must share a face in \mathcal{H}_μ^{pert} , and this face must contain e_w , due to R.1 and R.2. If the above condition on \overline{G}_{aux} is not satisfied, then we declare the instance negative; otherwise, we fix an order of the black vertices of \overline{G}_{aux} based either on the cycle or on an arbitrary order of the paths.

We now construct graph H_μ^{skel} in the same way as for the R-node. Note that, also in this case, the embedding \mathcal{H}_μ^{skel} of H_μ^{skel} is fixed, since the order of the black vertices of \overline{G}_{aux} induces an order of the virtual edges of H_μ^{skel} . We will again use \mathcal{H}_μ^{skel} to either determine whether the instance is negative or to construct the bags of μ .

The case in which μ is traversable is identical to the R-node case. When μ is non-traversable, we have $ref(\mu) \in H_\mu^{skel}$, and thus there exist the two faces f_μ^l and f_μ^r incident to $ref(\mu)$. However, since μ has at least two non-traversable children, every two virtual edges of H_μ^{skel} share at most one face in \mathcal{H}_μ^{skel} , and thus μ has no 2-sided children.

We now consider each traversable child ν_i of μ . Contrary to the R-node case, the face of \mathcal{H}_μ^{skel} in which ν_i is contained is not necessarily defined in this case by the rigid structure underneath, as the embedding of G_μ^{skel} is not unique. Recall that ν_i corresponds to a white vertex v_i of G_{aux} . If v_i has exactly two black neighbors, then they must be connected by an edge in \overline{G}_{aux} after the removal of v_i . So, they are consecutive in the order of the black vertices that we used to construct \mathcal{H}_μ^{skel} . Thus, the two virtual edges of H_μ^{skel} corresponding to them share a face in \mathcal{H}_μ^{skel} . We say in this case that e_i is *contained in* this particular face. If v_i has exactly one black neighbor in G_{aux} , then e_i may be contained in any of the two faces of \mathcal{H}_μ^{skel} incident to the virtual edge e corresponding to this black vertex. However, we cannot make a choice at this stage, as this may depend on other pairs whose vertices belong to the subgraph of G represented by e (that is, $G_{\nu_j}^{pert}$, if $e = e_j$, for some $1 \leq j \leq h$, and $G \setminus G_\mu^{pert}$, if $e = ref(\mu)$). If $e = e_j$, then we add a new bag B_{ν_j} to the child ν_j of μ , so that S_{ν_j} contains all the vertices of the special bag of ν_i , while T_{ν_j} is empty. The association of S_{ν_j} with one of the two faces incident to e_j , to be performed later, will determine the face in which e_i is contained. In the case in which $e = ref(\mu)$, virtual edge e_i should be contained either in f_μ^l or in f_μ^r , but again we cannot determine which of the two. Furthermore, we cannot even delegate this choice to the association of the pockets, since $ref(\mu)$ does not correspond to a child of μ . Thus, we do not associate it to any face, but we will use it to create the bags of μ . Finally, when v_i has no black neighbors, its special bag is empty.

Once all traversable children have been considered, we associate the special bags and the pockets of the non-special bags with the faces of \mathcal{H}_μ^{skel} , as in the R-node case. Then, we construct the bags of μ . We add the poles of μ to its special bag, if they belong to $\{x_1, \dots, x_k\}$. As in the R-node case, we add to μ a bag B_μ , whose pockets S_μ and T_μ have all the vertices of the special bags and of the pockets associated with f_μ^l and f_μ^r , respectively. Finally, for each traversable child ν_i of μ that has not been associated, we add a new bag B_μ^i so that S_μ^i contains all the vertices of the special bag of ν_i , while T_μ^i is empty. Finally, we apply operation MERGE-BAGS to all pairs $\langle x, y \rangle \in W$ such that both x and y belong to G_μ^{pert} . Hence, R.1 and R.2 are satisfied by any embedding \mathcal{G}_μ^{pert} of G_μ^{pert} .

that is described by the bags of μ . This concludes the P-node case.

At the end of the traversal, if root ρ has been visited without declaring the instance negative, the fact that $G_\rho^{pert} = G$ admits a planar embedding satisfying R.1 implies that $\langle G, W \rangle$ is a positive instance. The proof of the following theorem is in the appendix.

Theorem 4. *Let $G = (V, E_p \cup E_s \cup E_t)$ be an instance of HIERARCHICAL PARTIAL PLANARITY such that the graph induced by the edges in $E_p \cup E_s$ is biconnected. We can test in $O(|V|^3 \cdot |E_t|)$ time whether G has a drawing with no forbidden crossing.*

Proof. By Theorem 1, it suffices to prove that the algorithm described in Section 4.2 decides in $O(|V|^3 \cdot |W|)$ whether an instance $\langle G = (V, E_1 \cup E_2), W \rangle$ of FACIAL-CONSTRAINED CORE PLANARITY in which the graph H induced by the edges of E_1 is biconnected is positive.

The correctness of the algorithm follows from the fact that, as already discussed during the description of the algorithm, for each node $\mu \in \mathcal{T}$, requirements R.1 and R.2 are satisfied by any embedding \mathcal{G}_μ^{pert} of G_μ^{pert} that is described by the bags of μ (if any). In particular, this holds also for the root ρ of \mathcal{T} .

Regarding the time complexity, we observe that the construction of the SPQR-tree \mathcal{T} and of the auxiliary graphs G_{aux} and H_μ^{skel} can be done in $O(|V| + |W|)$ time. Operation MERGE-BAGS needs constant time, adopting elementary data structures to maintain the references between vertices and bags or pockets. Thus, the complexity of our algorithm is dominated by the association of the bags and of the pockets to the faces of \mathcal{H}_μ^{skel} , in the R- and P-node cases. In this phase of the algorithm, every bag of a child of a node μ could be considered a number of times that is linear in the total number of bags, which is $O(|V|)$. Also, every time one of these bags is considered, we perform $O(|W|)$ checks. Since the number of bags over all the children of μ is $O(|V|)$, we have a total $O(|V|^2 \cdot |W|)$ processing time for μ , which hence results in a total $O(|V|^3 \cdot |W|)$ time for G , and the statement follows. \square

5 Conclusions

In this paper we studied the problem HIERARCHICAL PARTIAL PLANARITY, in which a graph whose edges are of three types (primary, secondary, and tertiary) is given and the goal is to construct a drawing in which crossings are allowed only if they involve a tertiary edge. For this problem, we gave an efficient algorithm when the graph induced by the primary and secondary edges is biconnected.

The main open problem raised by our work is to determine the complexity in the general case, where the biconnectivity restriction is relaxed. It is also of interest to broaden the study towards the case in which there exist more than three levels of importance for the edges. As a first step, one could consider the case in which there are four levels and the first two form a biconnected graph. Finally, the relationship with SEFE should be further investigated to understand whether the techniques used in this paper can be applied to solve some of its open cases.

References

- [1] P. Angelini, C. Binucci, G. Da Lozzo, W. Didimo, L. Grilli, F. Montecchiani, M. Patrignani, and I. G. Tollis. Algorithms and bounds for drawing non-planar graphs with crossing-free subgraphs. *Comput. Geom.*, 50:34–48, 2015. doi:10.1016/j.comgeo.2015.07.002.
- [2] P. Angelini, S. Chaplick, S. Cornelsen, G. Da Lozzo, G. Di Battista, P. Eades, P. Kindermann, J. Kratochvíl, F. Lipp, and I. Rutter. Simultaneous orthogonal planarity. In Y. Hu and M. Nöllenburg, editors, *Graph Drawing*, volume 9801 of *LNCS*, pages 532–545. Springer, 2016. doi:10.1007/978-3-319-50106-2_41.
- [3] P. Angelini, G. Da Lozzo, and D. Neuwirth. Advancements on SEFE and partitioned book embedding problems. *Theor. Comput. Sci.*, 575:71–89, 2015. doi:10.1016/j.tcs.2014.11.016.
- [4] P. Angelini, G. Di Battista, F. Frati, V. Jelínek, J. Kratochvíl, M. Patrignani, and I. Rutter. Testing planarity of partially embedded graphs. *ACM Trans. Algorithms*, 11(4):32, 2015. doi:10.1145/2629341.
- [5] P. Angelini, G. Di Battista, F. Frati, M. Patrignani, and I. Rutter. Testing the simultaneous embeddability of two graphs whose intersection is a biconnected or a connected graph. *J. Discrete Algorithms*, 14:150–172, 2012. doi:10.1016/j.jda.2011.12.015.
- [6] M. A. Bekos, T. C. van Dijk, P. Kindermann, and A. Wolff. Simultaneous drawing of planar graphs with right-angle crossings and few bends. *J. Graph Algorithms Appl.*, 20(1):133–158, 2016. doi:10.7155/jgaa.00388.
- [7] C. Binucci, U. Brandes, G. Di Battista, W. Didimo, M. Gaertler, P. Palladino, M. Patrignani, A. Symvonis, and K. A. Zweig. Drawing trees in a streaming model. *Inf. Process. Lett.*, 112(11):418–422, 2012. doi:10.1016/j.ipl.2012.02.011.
- [8] T. Bläsius, S. G. Kobourov, and I. Rutter. Simultaneous embedding of planar graphs. In R. Tamassia, editor, *Handbook on Graph Drawing and Visualization.*, pages 349–381. Chapman and Hall/CRC, 2013.
- [9] T. Bläsius and I. Rutter. Disconnectivity and relative positions in simultaneous embeddings. *Comput. Geom.*, 48(6):459–478, 2015. doi:10.1016/j.comgeo.2015.02.002.
- [10] T. Bläsius and I. Rutter. Simultaneous pq-ordering with applications to constrained embedding problems. *ACM Trans. Algorithms*, 12(2):16:1–16:46, 2016. doi:10.1145/2738054.
- [11] P. Braß, E. Cenek, C. A. Duncan, A. Efrat, C. Erten, D. Ismailescu, S. G. Kobourov, A. Lubiw, and J. S. B. Mitchell. On simultaneous planar graph embeddings. *Comput. Geom.*, 36(2):117–130, 2007. doi:10.1016/j.comgeo.2006.05.006.

- [12] T. M. Chan, F. Frati, C. Gutwenger, A. Lubiw, P. Mutzel, and M. Schaefer. Drawing partially embedded and simultaneously planar graphs. *J. Graph Algorithms Appl.*, 19(2):681–706, 2015. doi:10.7155/jgaa.00375.
- [13] G. Da Lozzo and I. Rutter. Planarity of streamed graphs. In V. T. Paschos and P. Widmayer, editors, *CIAC*, volume 9079 of *LNCS*, pages 153–166. Springer, 2015. doi:10.1007/978-3-319-18173-8_11.
- [14] G. Di Battista and R. Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15(4):302–318, 1996. doi:10.1007/BF01961541.
- [15] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996. doi:10.1137/S0097539794280736.
- [16] E. Di Giacomo, W. Didimo, G. Liotta, H. Meijer, and S. K. Wismath. Planar and quasi-planar simultaneous geometric embedding. *Comput. J.*, 58(11):3126–3140, 2015. doi:10.1093/comjnl/bxv048.
- [17] C. Erten and S. G. Kobourov. Simultaneous embedding of planar graphs with few bends. *J. Graph Algorithms Appl.*, 9(3):347–364, 2005. doi:10.7155/jgaa.00113.
- [18] M. T. Goodrich and P. Pszona. Streamed graph drawing and the file maintenance problem. In S. K. Wismath and A. Wolff, editors, *Graph Drawing*, volume 8242 of *LNCS*, pages 256–267. Springer, 2013. doi:10.1007/978-3-319-03841-4_23.
- [19] C. Gutwenger and P. Mutzel. A linear time implementation of SPQR-trees. In J. Marks, editor, *Graph Drawing*, volume 1984 of *LNCS*, pages 77–90. Springer, 2000. doi:10.1007/3-540-44541-2_8.
- [20] V. Jelínek, J. Kratochvíl, and I. Rutter. A Kuratowski-type theorem for planarity of partially embedded graphs. *Comput. Geom.*, 46(4):466–492, 2013. doi:10.1016/j.comgeo.2012.07.005.
- [21] J. Kratochvíl. String graphs. I. The number of critical nonstring graphs is infinite. *J. Comb. Theory, Ser. B*, 52(1):53–66, 1991. doi:10.1016/0095-8956(91)90090-7.
- [22] M. Schaefer. Picking planar edges; or, drawing a graph with a planar subgraph. In C. A. Duncan and A. Symvonis, editors, *Graph Drawing*, volume 8871 of *LNCS*, pages 13–24. Springer, 2014. doi:10.1007/978-3-662-45803-7_2.