Planarity of Graphs with Crossable Edges

Seok-Hee Hong
University of Sydney, Australia,
shhong@it.usyd.edu.au

Hiroshi Nagamochi
Kyoto University, Japan,
nag@amp.i.kyoto-u.ac.jp

Abstract

Let $H = (V_H, E_H)$ be an undirected graph with a subset E of the edge set E_H, where we call the edges in E *red edges* and the edges in $E_H - E$ *blue edges*. An embedding of H into the plane is called an E-*planar* embedding if no red edge crosses any other edges whereas two blue edges may cross. In this paper, we give a complete characterization of an instance (H, E) that admits no E-planar embedding via forbidden subgraphs with red/blue edges. Furthermore we design a linear-time algorithm that finds either an E-planar embedding or a forbidden subgraph.

The problem setting can enable us to formulate a problem of finding a planar embedding of a planar graph G with an additional constraint such that, for specified sets $S_1, S_2, \ldots, S_k$ of vertices, all the vertices in each S_i appear along the same facial cycle. To see this, we regard all edges in G as red edges and add a star s_i with blue edges s_it, $t \in S_i$ for each i. For example, this allows us to find a planar embedding of a planar graph G such that the rotation systems of some vertices are predetermined.

1 Introduction

Planar graphs are graphs that can be embedded in the plane without edge crossings, and extensively studied by researchers in Graph Theory and Graph Algorithms, for example, planar graphs with n vertices can have at most $3n - 6$ edges. A graph is planar if and only if it contains no subgraph that is a subdivision of K_5 and $K_{3,3}$ [32]. Testing *planarity* of a graph can be solved in linear time [28, 33], and some methods also produce a planar embedding [9, 10, 12, 17, 18, 34, 36, 37] or forbidden minors [9, 10].

Variations of planarity with additional embedding or desired drawing *constraints* were studied. For example, testing planarity with embedding constraints such as a fixed *rotation system* (i.e., the circular ordering of edges for a vertex) of each vertex [22], and partially-fixed planar embeddings [2] are considered. Testing planarity with additional drawing constraints for extended graph models and digraphs such as clustered (compound) graph planarity (*C-planarity*, in short) [19], *hierarchical* (or *level*) planarity [**?**], and *upward* planarity [20] for digraphs were extensively studied.

Another recent variant of the planarity problem includes the *simultaneous embedding* which given two planar graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ with the same vertex set V, asks the existence of two planar drawings D_1 and D_2 of G_1 and G_2, respectively, such that each vertex $v \in V$ is mapped to the same point in D_1 and D_2. Unfortunately, the problem of testing whether two planar graphs admit a *geometric* simultaneous embedding, where D_1 and D_2 are required to be straight-line drawings, is NP-hard [16].

[1]Technical report 2016-003, July 14, 2016.

Note that simultaneous embedding can be considered as an embedding of two planar graphs, red and blue, where red-red edge crossings and blue-blue edge crossings are not allowed, however red-blue crossings are allowed in a combined drawing consisting of a red graph and a blue graph.

A recent research topic in topological graph theory generalizes the notion of planarity to *beyond planar graphs*, i.e., non-planar graphs with some specific crossings, or with some forbidden crossing patterns. Examples include k*-planar* graphs (i.e., graphs that can be embedded with at most k crossings per edge) [35], k*-quasi-planar* graphs (i.e., graphs that can be embedded without k mutually crossing edges) [1], *RAC* graphs (i.e., graphs that can be embedded with right angle crossings) [14], and *fan-planar* graphs (i.e., graphs that can be embedded with fan-crossings) [30].

Recently, algorithmics and complexity for such graphs have been investigated. Unfortunately, testing 1-planarity of a graph is NP-complete [21, 31], and testing whether a given graph is a RAC graph is NP-hard [3]. Similarly, testing fan-planarity of a graph is NP-hard [8], even if a rotation system of each vertex in a graph is fixed [6].

On the positive side, linear-time algorithms are available for special subclasses of beyond planar graphs. For example, testing *maximal 1-planarity* (i.e., the addition of an edge destroys 1-planarity) of a graph can be solved in linear time, if a rotation system of each vertex is given [15]. Testing *outer-1-planarity* (i.e., 1-planar graphs with each vertex on the outer face) of a graph can be solved in linear time [4, 25]. Testing *maximal outer-fan-planarity* (i.e., fan-planar graphs with each vertex on the outer face and the addition of an edge destroys outer-fan-planarity) [6], and testing *full outer-2-planarity* (i.e., 2-planar graphs with each vertex on the outer face and there is no crossing on the outer face) [26] can be solved in linear time.

As another problem on beyond planarity, this paper studies a problem of drawing a subset of edges as a plane embedding while the other edges are allowed to cross each other. Let $H = (V_H, E_H)$ be an undirected graph with a subset E of the edge set E_H, where we call the edges in E *red edges* and the edges not in E *blue edges*. An embedding of H into the plane is called E*-planar* if no edge in E crosses any other edges whereas two edges in $E_H - E$ may cross. The main problem of this paper can be defined as follows.

Embedding a Graph with Crossable Edges

Input: A graph $H = (V_H, E_H)$ with an edge subset $E \subseteq E_H$.

Output: Test whether H admits an E-planar embedding and construct an E-planar embedding if one exists.

See Fig. 1 for an example of a graph H with an edge subset E and its E-planar embedding. As a main result in this paper, we give a complete characterization of an instance (H, E) that does not admit an E-planar embedding via forbidden subgraphs with red/blue edges. Furthermore we design a linear-time algorithm that finds either an E-planar embedding or a forbidden subgraph.

Our problem setting can enable us to formulate a problem of finding a planar embedding of a planar graph $G = (V, E)$ with an additional constraint such that, for specified sets $S_1, S_2, \ldots, S_k$ of vertices, all the vertices in each S_i appear along the same facial cycle. We call the constraint the *facing constraint*. To see this, we regard all edges in G as red edges and add a star s_i with blue edges $s_i t$, $t \in S_i$ for each i. Then we observe that the augmented graph $H = (V \cup \{s_1, \ldots, s_k\}, E \cup (\cup_{1 \leq i \leq k}\{s_i t \mid t \in S_i\}))$ admits an E-planar embedding if and only if G admits a planar embedding satisfying the facing constraint with $\{S_1, S_2, \ldots, S_k\}$.

For example, this allows us to find a planar embedding of a planar graph G such that the rotation systems of some vertices are predetermined, as studied by Gutwenger et al. [22]. Let $\{u_1, u_2, \ldots, u_d\}$ be the neighbors of a vertex v in G. Then by setting $S_i = \{u_i, u_{i+1}\}$ for all $i = 1, 2, \ldots, d-1$, we see that for any planar embedding G the facing constraint with $\{S_1, S_2, \ldots, S_d\}$, the rotation system of v is either

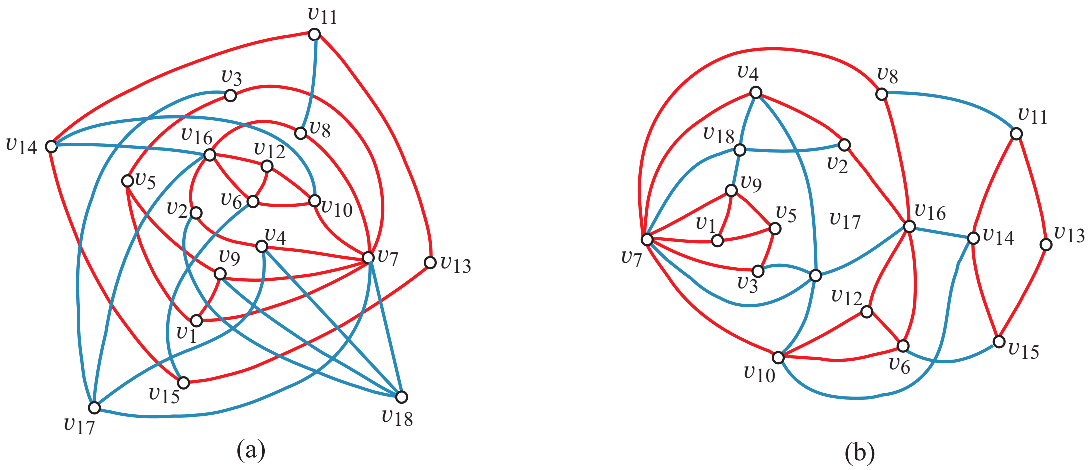


Figure 1: (a) An instance (H,E) with a red edge set $E = \{v_1v_5, v_1v_7, v_1v_9, v_2v_4, v_2v_{16}, v_3v_5, v_3v_7, v_4v_7, v_5v_9, v_6v_{10}, v_6v_{12}, v_6v_{16}, v_7v_8, v_7v_{10}, v_8v_{16}, v_{10}v_{12}, v_{11}v_{13}, v_{11}v_{14}, v_{12}v_{16}, v_{13}v_{15}, v_{14}v_{15}\}$; (b) An E-planar embedding of H.

$(u_1, u_2, \ldots, u_d)$ or its reversal. We can impose the same constraint for some other vertices at the same time, since we allow blue edges to cross each other.

Another way of asking our problem is to find a partition $\{X_1, X_2, \ldots, X_h\}$ of a specified vertex subset X in a graph H such that the graph H' obtained from H by contracting each set X_i into a single vertex x_i becomes planar, where we call such a partition *planarizing*. It is not difficult to see that the problem of finding a planarizing partition can be reduced to our problem by regarding all edges incident to a vertex in X as blue edges.

The paper is organized as follows. Section 2 introduces basic notations and discusses data structure for ordered trees. Section 3 states our main result that the instances that have no E-planar embeddings can be characterized by five types of forbidden subgraphs, and shows how to restrict given instances of the problem to instances with a special structure called "star instances," where the red graph is connected. Section 4 describes how to reduce a star instance with a red connected graph into a star instance with a red biconnected graph in linear time. Section 5 presents an algorithm for testing whether given instance (H,E) with a triconnected red graph is E-planar, where we use a geometric argument based on convex grid drawings to make a naive quadratic time algorithm run in linear time. The last case where the red graph is biconnected is treated by three sections. Section 6 first reviews a method of decomposing a biconnected graph into triconnected components, and observes how the forbidden subgraphs may appear in such triconnected components of the red biconnected graph. Our algorithm for testing whether some forbidden subgraph appears in a triconnected component of the red biconnected graph consists of two major phases. Section 7 shows the first phase which detects some types of forbidden subgraphs appear in a triconnected component of the red graph, whereas Section 8 presents the second phase that mainly detects the last type of forbidden subgraphs in a given instance with a red biconnected graph.

2 Preliminaries

This section introduces basic notations and discusses data structure for ordered trees.

2.1 Terminology

For a graph G, let $V(G)$ and $E(G)$ denote the sets of vertices and edges in G, respectively. Let $G = (V, E)$ be a graph, where n denotes $|V|$ unless stated otherwise. For a vertex $v \in V$, let $\deg(v)$ denote the degree of v, and $N(v)$ denote the set of neighbors u of v. For a subset $E' \subseteq E$ of edges, let $V(E')$ denote the set of end-vertices of all edges in E', and let $G - E'$ denote the graph obtained from G by removing the edges in E'. Let $X \subseteq V$ be subsets of vertices. Let $\delta(X)$ denote the set of edges between X and $V - X$, where we may denote $\delta(\{v\})$ by $\delta(v)$. Let $G - X$ denote the graph obtained from G by removing the vertices in X together with the edges incident to any vertex in X. Let $G[X]$ denote the subgraph induced from G by the vertices in X, i.e., $G[X] = G - (V - X)$. We may indicate the underlying graph G in these notations in such a way that $\delta(X)$, $\deg(v)$ and $N(v)$ are written as $\delta(X; G)$, $\deg(v; G)$ and $N(v; G)$. A vertex of degree d is called a *degree-d vertex*. A *star* is a graph with a center s and some other vertices which are incident to only s (possibly with multiple edges), which we may simply denote by the center s, the edge set $\{st \mid t \in N(s)\}$ or the neighbor set $N(s)$ if it is clear from the context. A biconnected component of a graph is called a *block*.

We say that two paths are *internally disjoint* if no internal vertex of one of the paths is contained by the other. For two vertex subsets $S, T \subseteq V$, a simple path P with end vertices $s \in S$ and $t \in T$ such that $V(P) \cap (S \cup T) = \{s, t\}$ called an *S, T-path*. We may denote $\{s\}, T$-path by s, T-path and $\{s\}, \{t\}$-path by s, t-path. For a path P, let $V_{\text{in}}(P)$ denote the set of internal vertices in P. *Subdividing* an edge $e = uv$ is to replace the edge with a u, v-path $u, w_1, w_2, \ldots, w_k, v$ with k (≥ 1) new degree-2 vertices w_i, $i = 1, 2, \ldots, k$. A graph H is a *subdivision* of G if H is obtained by subdividing some edges in G. A graph H is called *pseudo-triconnected* if it is a subdivision of a triconnected graph G. It is known that a planar embedding of a pseudo-triconnected graph is unique up to reversal or a choice of outer face.

For a tree T and two vertices u and v in T, let $P(u, v; T)$ denote the unique u, v-path in T. A *rooted tree* is a tree with a designated vertex r, called the *root*, which introduces a parent-child order among vertices and defines the *depth* $\text{dt}(v)$ of each vertex v to be the length of the path from r to v.

A *u, v-chain* is a graph obtained from a u, v-path $(u_1 = u, u_2, \ldots, u_p = v)$ by replacing some edges $u_i u_{i+1}$ with two internally disjoint u_i, u_{i+1}-paths P_i^1 and P_i^2, and the *length* of the chain is defined to be $p - 1$, the length of the original u, v-path. Let C_i denote the cycle formed by P_i^1 and P_i^2 (possibly C_i is a cycle of length 2). Hence it is a sequence of edges $u_i u_{i+1}$ or simple cycles C_i for $i = 1, 2, \ldots, p - 1$, where we call each of such cycles C_i a *factor* of the chain and each of P_i^1 and P_i^2 of a factor C_i is a *side* of C_i. We define a *circular chain* to be a u, v-chain with $u = v$.

Let K_5^* denote the graph obtained from the complete graph K_5 with five vertices by splitting a vertex into two degree-3 vertices u_1 and v_1 with a new edge $u_1 v_1$, where we call u_1 and v_1 the *split vertices*.

A *topological graph* or *embedding* γ of a graph H is a representation of a graph (possibly with multiple edges) in the plane, where each vertex is a point and each edge is a Jordan arc between the points representing its endpoints. Two edges *cross* if they have a point in common, other than their endpoints. The point in common is a *crossing*. To avoid pathological cases, standard non-degeneracy conditions apply: (i) two edges intersect at most one point; (ii) an edge does not contain a vertex other than its endpoints; (iii) no edge crosses itself; (iv) edges must not meet tangentially; (v) no three edges share any crossing point; and (vi) no two edges that share an endpoint cross.

2.2 Ordered Trees

Let T be an *ordered tree*, i.e., a rooted tree with a left-right order, a total order over the children of each vertex. For each vertex v in T, let $D(v; T)$ denote the set of vertex v and all descendants of v, and let $T(v)$ denote the ordered subtree $T[D(v; T)]$ induced from T by $D(v; T)$. For two vertices u and v in T, let $\text{lca}(u, v; T)$ be the *least common ancestor* of u and v. In T, we define the *left dfs order* ld, the *right dfs order* rd, the *left post order* lp and the *right post order* rp to be functions from V to the set of nonnegative integers such that

(i) $\mathrm{ld}(u) > \mathrm{ld}(v)$, $\mathrm{rd}(u) > \mathrm{rd}(v)$, $\mathrm{lp}(u) < \mathrm{lp}(v)$ and $\mathrm{rp}(u) < \mathrm{rp}(v)$ for a vertex v with a child u; and

(ii) $\mathrm{ld}(u) < \mathrm{ld}(v)$, $\mathrm{rd}(u) > \mathrm{rd}(v)$, $\mathrm{lp}(u) < \mathrm{lp}(v)$ and $\mathrm{rp}(u) > \mathrm{rp}(v)$ for two siblings u and v such that u is to the left of v.

We assume that the maximum value used in these functions is $O(n)$. See Fig. 2(a) for an illustration of an ordered tree T with vertices indexed by the left post order lp.

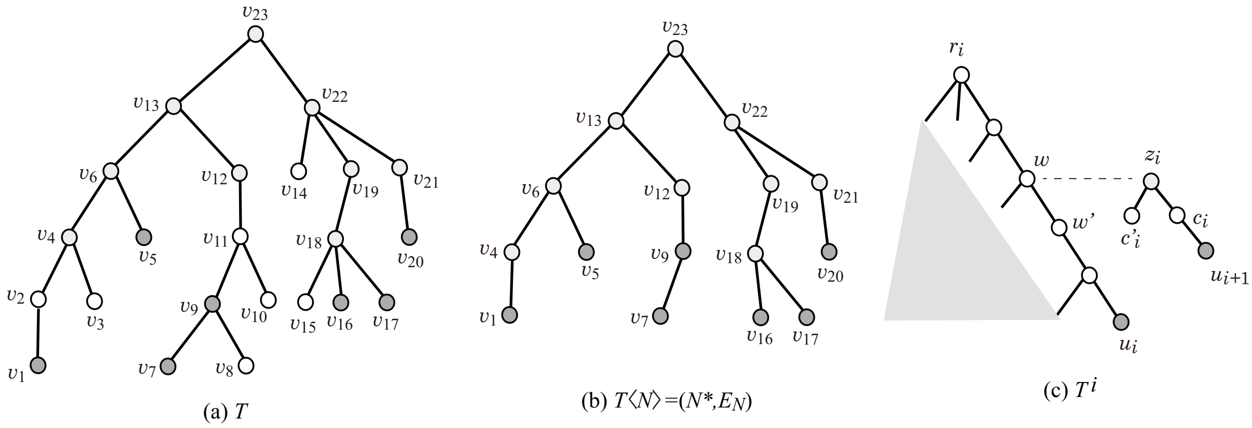


Figure 2: Illustration of an ordered tree and a mimic tree: (a) An instance ordered tree T rooted at vertex v_{23}, where the vertices are indexed by the left post order lp; (b) The mimic tree $T\langle N\rangle = (N^*, E_N)$ for $N = \{v_1, v_5, v_7, v_9, v_{17}, v_{16}, v_{20}\}$, where $N^* - N = \{v_4, v_6, v_{12}, v_{13}, v_{18}, v_{19}, v_{22}, v_{21}, v_{23}\}$; and (c) The mimic tree $T^i = T\langle N^i\rangle$ of the set $N^i = \{u_1, u_2, \ldots, u_i\}$.

Note that $u \notin D(v;T)$ if and only if $\mathrm{ld}(u) < \mathrm{ld}(v)$ or $\mathrm{rd}(u) < \mathrm{rd}(v)$.

For a vertex $v \in T$, we say that a vertex u is *on the left side of* v if $\mathrm{ld}(u) < \mathrm{ld}(v)$ and $\mathrm{lp}(u) < \mathrm{lp}(v)$, which means that $\mathrm{lca}(u, v; T)$ has children w_1 and w_2 such that $u \in D(w_1;T)$, $v \in D(w_2;T)$ and w_1 is to the left of w_2. Symmetrically we say that a vertex u is *on the right side of* v if $\mathrm{rd}(u) < \mathrm{rd}(v)$ and $\mathrm{rp}(u) < \mathrm{rp}(v)$. Observe that $\mathrm{lp}(u) \le \mathrm{lp}(w)$ implies that $u \in D(w;T)$ or $\mathrm{ld}(u) < \mathrm{ld}(w)$.

Lemma 1 *For a subset S of vertices in an ordered tree T and each function $f \in \{\mathrm{dt}, \mathrm{lp}, \mathrm{rp}\}$, let $\mathrm{a}_{f,S}$ denote a vertex $u \in S$ with $f(u) = \min\{f(s) \mid s \in S\}$. Let v be a vertex in T. Then:*

(i) *If $\mathrm{ld}(\mathrm{a}_{\mathrm{lp},S}) < \mathrm{ld}(v)$ and $\mathrm{lp}(\mathrm{a}_{\mathrm{lp},S}) < \mathrm{lp}(v)$, then $\mathrm{a}_{\mathrm{lp},S}$ is a vertex on the left side of v; Conversely if the set $S - D(v;T)$ contains a vertex on the left side of v, then $\mathrm{ld}(\mathrm{a}_{\mathrm{lp},S}) < \mathrm{ld}(v)$ and $\mathrm{lp}(\mathrm{a}_{\mathrm{lp},S}) < \mathrm{lp}(v)$;*

(ii) *If $\mathrm{rd}(\mathrm{a}_{\mathrm{rp},S}) < \mathrm{rd}(v)$ and $\mathrm{rp}(\mathrm{a}_{\mathrm{rp},S}) < \mathrm{rp}(v)$, then $\mathrm{a}_{\mathrm{rp},S}$ is a vertex on the right side of v; Conversely if the set $S - D(v;T)$ contains a vertex on the right side of v, then $\mathrm{rd}(\mathrm{a}_{\mathrm{rp},S}) < \mathrm{rd}(v)$ and $\mathrm{rp}(\mathrm{a}_{\mathrm{rp},S}) < \mathrm{rp}(v)$; and*

(iii) *Assume that $S - D(v;T)$ contains no vertex on the left or right side of v. If $\mathrm{dt}(\mathrm{a}_{\mathrm{dt},S}) < \mathrm{dt}(v)$, then $\mathrm{a}_{\mathrm{dt},S}$ is an ancestor of v; Conversely if the set $S - D(v;T)$ contains an ancestor of v, then $\mathrm{dt}(\mathrm{a}_{\mathrm{dt},S}) < \mathrm{dt}(v)$.*

Proof. (i) Let $u = \mathrm{a}_{\mathrm{lp},S}$. Hence if $\mathrm{ld}(u) < \mathrm{ld}(v)$ and $\mathrm{lp}(u) < \mathrm{lp}(v)$ then u is on the left side of v by definition. Assume that $S - D(v;T)$ contains a vertex w on the left side of v. By definition, $\mathrm{ld}(w) < \mathrm{ld}(v)$ and $\mathrm{lp}(w) < \mathrm{lp}(v)$, from which $\mathrm{lp}(u) \le \mathrm{lp}(w) < \mathrm{lp}(v)$ since $u = \mathrm{a}_{\mathrm{lp},S}$. Recall that $\mathrm{lp}(u) \le \mathrm{lp}(w)$ implies that $u \in D(w;T)$ or $\mathrm{ld}(u) < \mathrm{ld}(w)$, from which we know $\mathrm{ld}(u) < \mathrm{ld}(v)$.

(ii) Symmetrically with (i).

(iii) Assume that $S - D(v;T)$ contains no vertex on the left or right side of v. Then any vertex $w \in S - D(v;T)$ is an ancestor of v, and satisfies $\mathrm{lca}(w, v; T) = w$. Let $u = \mathrm{lca}(\mathrm{a}_{\mathrm{dt},S}, v; T)$. If $\mathrm{dt}(u) < \mathrm{dt}(v)$, then $u \in S - D(v;T)$ is an ancestor of v. If $S - D(v;T)$ contains an ancestor w of v, then clearly $\mathrm{dt}(u) \leq \mathrm{dt}(w) < \mathrm{dt}(v)$. □

2.3 Mimic Trees

Let $T = (V, E)$ be an ordered tree. It is known that we can find $\mathrm{lca}(u, v; T)$ for any query of two vertices u and v in $O(1)$ time after an $O(|V|)$-time preprocessing on the rooted tree T (see [7, 24]).

For two distinct vertices $v_1, v_2 \in V$, define $\mathrm{lca}^*(v_1, v_2; T)$ to be the set of at most three vertices z, z_1 and z_2 such that $z = \mathrm{lca}(v_1, v_2; T)$, and z_i, $i = 1, 2$ is the children of z which is v_i or an ancestor of v_i (no such vertex z_i exists if $v_i = \mathrm{lca}^*(v_1, v_2; T)$); i.e., if $z = \mathrm{lca}^*(v_1, v_2; T) \neq v_1, v_2$ then $\mathrm{lca}^*(v_1, v_2; T) = \{\mathrm{lca}(v_1, v_2; T), c_1, c_2\}$ with $c_i \in \mathrm{Ch}(z; T)$ with $v_i \in D(c_i; T)$; and if $z = \mathrm{lca}^*(v_1, v_2; T) \neq v_i$ for $i = 1$ or 2 then $\mathrm{lca}^*(v_1, v_2; T) = \{\mathrm{lca}(v_1, v_2; T), c_j\}$ with $c_j \in \mathrm{Ch}(z; T)$ with $v_i \in D(c_j; T)$ for $j \neq i$. We observe that given a query of two vertices v_1 and v_2, $\mathrm{lca}^*(v_1, v_2; T)$ can be obtained in $O(1)$ time using the procedure for finding the least common ancestors. For example, if T is a binary tree, then we can easily find the children c_i of $\mathrm{lca}(v_1, v_2; T)$ that is an ancestor of v_i in $O(1)$ time if one exists. When T is not a binary tree, each vertex v in T with $d \geq 3$ children $u_1, u_2, \ldots, u_d$ can be split into $d - 2$ vertices $v_1, \ldots, v_{d-1}$ to obtain a binary ordered tree T_{left} (resp., T_{right}) such that $u_i, v_{i+1} \in \mathrm{Ch}(v_i; T_{\mathrm{left}})$, $i = 1, 2, \ldots, d-2$ and $u_{d-1}, u_d \in \mathrm{Ch}(v_{d-1}; T_{\mathrm{left}})$ (resp., $u_{d-i+1}, v_{i+1} \in \mathrm{Ch}(v_i; T_{\mathrm{right}})$, $i = 1, 2, \ldots, d-2$ and $u_1, u_2 \in \mathrm{Ch}(v_{d-1}; T_{\mathrm{right}})$), where u_i appears before u_{i+1} in the left depth-first order along any of T_{left} and T_{right}. Using the both modified trees T_{left} and T_{right}, we can find the right child c_i of $\mathrm{lca}(v_1, v_2; T)$ in T in $O(1)$ time.

For a subset N of vertices in an ordered tree $T = (V, E)$, let N^* denote the set $N \cup (\bigcup_{u,v \in N} \mathrm{lca}^*(u, v; T))$, and we call an ordered tree $T\langle N \rangle = (N^*, E_N)$ the *mimic tree* induced from T by N if
(i) the edge set E_N contains an edge uv when u is an ancestor of v and no other vertex $w \in N^* - \{u, v\}$ lies along the path between u and v in T; and
(ii) for two siblings u and v in $T\langle N \rangle$, u is to the left of v in $T\langle N \rangle$, when u is on the left side of v.
Note that $|N^*| \leq 3|N|$. See Fig. 2(b) for an illustration of a mimic tree.

Lemma 2 *Given an ordered tree T with n vertices and a family $\{N_1, N_2, \ldots, N_k\}$ of subsets of vertices, the mimic trees $T\langle N_i \rangle$, $i = 1, 2, \ldots, k$ can be constructed in $O(n + \sum_{1 \leq i \leq k} |N_i|)$ time.*

Proof. First for each set $N \in \{N_1, N_2, \ldots, N_k\}$, sort the vertices in N so that $N = \{u_1, u_2, \ldots, u_p\}$ satisfies $\mathrm{ld}(u_1) < \mathrm{ld}(u_2) < \cdots < \mathrm{ld}(u_p)$. This can be done in $O(n + \sum_{1 \leq i \leq k} |N_i|)$ time by visiting each vertex v in the ordered tree T according to the left dfs order and placing the vertex v as the latest one in a new list for each set N_i with $v \in N_i$.

Next we construct the mimic tree $T\langle N \rangle$ for each set $N \in \{N_1, N_2, \ldots, N_k\}$. To prove the lemma, it suffices to show that each $T\langle N \rangle$ can be constructed in $O(|N|)$ time.

Let $N = \{u_1, u_2, \ldots, u_p\}$, where $\mathrm{ld}(u_i) < \mathrm{ld}(u_{i+1})$, $i = 1, 2, \ldots, |N| - 1$. For each $i = 1, 2, \ldots, p - 1$, we compute $\mathrm{lca}^*(u_i, u_{i+1}; T)$ in $O(1)$ time, and denote by z_i $\mathrm{lca}(u_i, u_{i+1}; T)$ and by c_i (resp., c'_i) the child of z_i that is an ancestor of u_i (resp., u_{i+1}) if one exists. Note that N^* is obtained by $N \cup (\bigcup_{i=1,2,\ldots,p-1} \mathrm{lca}^*(u_i, u_{i+1}; T))$.

For each $i = 1, 2, \ldots, p$, let T^i denote the mimic tree $T\langle N^i \rangle$ of the set $N^i = \{u_1, u_2, \ldots, u_i\}$ of the first i vertices. Clearly T^1 is the tree consisting of vertex u_1. Assuming that T^i for some $i < |N|$ is obtained, we show how to construct T^{i+1}. Since $u_1, u_2, \ldots, u_p$ are indexed according to the left dfs order, we observe that the path $P(r_i, u_i; T^i)$ between the current root r_i of T^i is the rightmost path, i.e., we arrive at u_i from r_i by choosing the rightmost child. See Fig. 2(c) for an illustration of T^i. To determine the

right position where vertices z_i, c_i, c'_i and u_{i+1} are inserted or added in T^i to obtain T^{i+1}, we traverse path $P(r_i, u_i; T^i)$ from u_i toward r_i to find the lowest vertex w in the path such that $\mathrm{dt}(w) \geq \mathrm{dt}(z_i)$. We distinguish three cases:
(i) No such vertex w exists in the path: Then let r_i be a child of a new root z_i;
(ii) Such a vertex w exists and $\mathrm{dt}(w) = \mathrm{dt}(z_i)$: Then $w = z_i$ holds; and
(iii) Such a vertex w exists and $\mathrm{dt}(w) > \mathrm{dt}(z_i)$: Then insert z_i between w and its rightmost child w'.

In any of (i)-(iii), (a) if z_i has a child and the right child of z_i is not c'_i in the current tree, then insert c'_i between z_i and the right child; (b) if $z_i \neq u_{i+1} \neq c'_i$ (resp., if $z_i \neq u_{i+1} = c'_i$) then append c'_i as the rightmost child of w and u_{i+1} as a new child of c'_i (resp., append u_{i+1} as the rightmost child of w). Note that $|V(T^p)| = |N^*| \leq 3|N|$. When we traverse the rightmost path $P(r_i, u_i; T^i)$, the edges in the path traversed will not be traversed again later. This implies that the total time for constructing $T^p = T\langle N\rangle$ is $O(|V(T^p)|) = O(|N|)$, as required. □

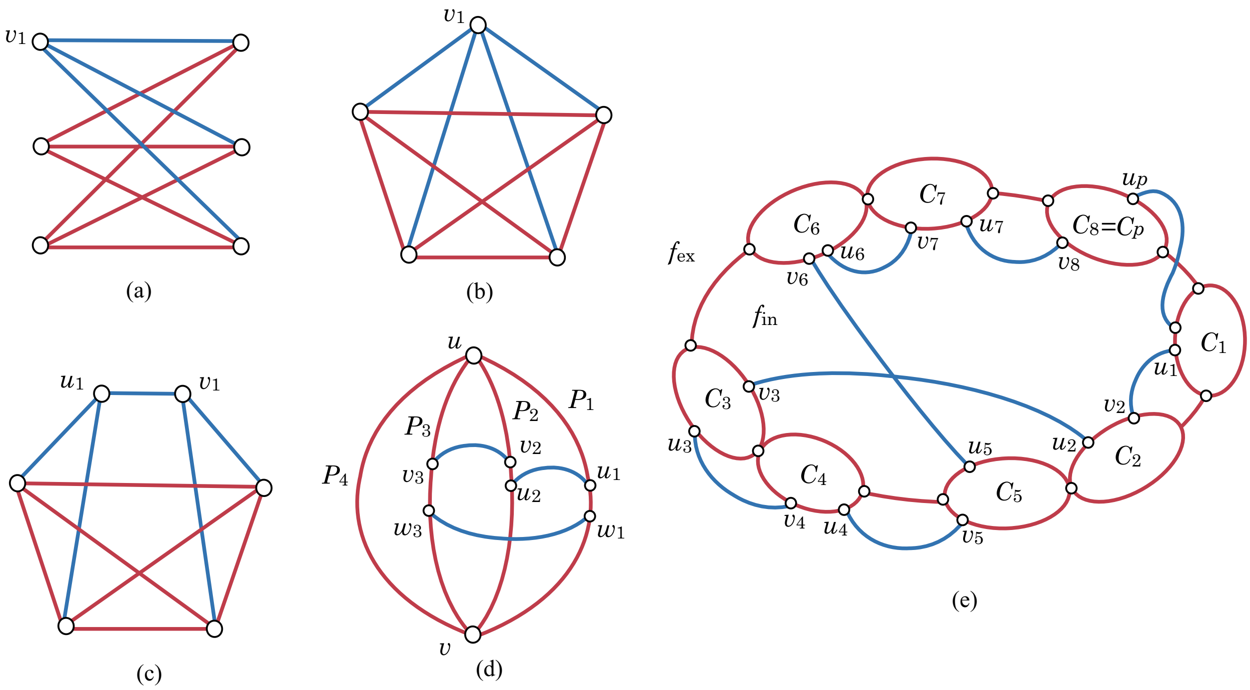


Figure 3: Illustration of primely forbidden subgraphs: (a) $K_{3,3}$ with at most one vertex to which a blue edge is incident; (b) K_5 with at most one vertex to which a blue edge is incident; (c) K_5^* with blue edges incident to only the split vertices; (d) A set of four red internally disjoint u, v-paths P_1, P_2, P_3 and P_4 together with three blue edges u_1u_2, v_2v_3 and w_3w_1 such that $u_1, w_1 \in V(P_1) - \{u, v\}$, $u_2, v_2 \in V(P_2) - \{u, v\}$, $v_3, w_3 \in V(P_3) - \{u, v\}$; and (e) A red circular chain with $p \geq 2$ factors $C_1, C_2, \ldots, C_p$ and p blue edges $u_1v_2, u_2v_3, \ldots, u_{p-1}v_p$ and u_pv_1 such that for each $i = 1, 2, \ldots, p-1$, it holds $u_i \in V_{\mathrm{in}}(P_i^k)$ and $v_{i+1} \in V_{\mathrm{in}}(P_{i+1}^k)$ for the same $k \in \{1, 2\}$, but $u_p \in V_{\mathrm{in}}(P_p^k)$ and $v_{i+1} \in V_{\mathrm{in}}(P_1^\ell)$ for $k \neq \ell$

3 Main Theorem

As our main result, this section states that every instance that has no E-planar embedding can be characterized by five types of forbidden subgraphs, and shows how to restrict given instances of the problem to instances with special structure called "star instances," where the red graph is connected.

Let $(H = (V_H, E_H), E)$ be a given instance. For convenience, we call the edges in E *red edges* and the edges in $E_H - E$ *blue edges*. A graph is called *red* (resp., *blue*) if it contains only red (resp., blue) edges.

A graph is called *gray* if it is allowed to contain both a red edge and a blue edge.

We call a graph H with red and blue edges *primely forbidden* if it satisfies one of the following:

(i) $H = K_{3,3}$ with at most one vertex to which a blue edge is incident (see Fig. 3(a));

(ii) $H = K_5$ with at most one vertex to which a blue edge is incident (see Fig. 3(b));

(iii) $H = K_5^*$ with blue edges incident to only the split vertices (see Fig. 3(c));

(iv) H consists of four red internally disjoint u, v-paths P_1, P_2, P_3 and P_4 together with three blue edges u_1u_2, v_2v_3 and w_3w_1 such that $u_1, w_1 \in V_{\rm in}(P_1)$, $u_2, v_2 \in V_{\rm in}(P_2)$, $v_3, w_3 \in V_{\rm in}(P_3)$ (see Fig. 3(d)); and

(v) H consists of a red circular chain of length at least 3 with $p \geq 2$ factors $C_1, C_2, \ldots, C_p$ (which may not appear in this order along the circular chain) and p blue edges u_iv_{i+1}, $i = 1, 2, \ldots, p-1$ and u_pv_1 such that for each $i = 1, 2, \ldots, p-1$, it holds $u_i \in V_{\rm in}(P_i^k)$ and $v_{i+1} \in V_{\rm in}(P_{i+1}^k)$ for the same $k \in \{1, 2\}$, but $u_p \in V_{\rm in}(P_p^k)$ and $v_{i+1} \in V_{\rm in}(P_1^\ell)$ for $k \neq \ell$ (see Fig. 3(e)).

We call a graph with red and blue edges *forbidden* if it is primely forbidden or it is obtained from a primely forbidden graph by subdividing some edges such that each red edge uv is replaced with a red u, v-path and each blue edge uv is replaced with a gray u, v-path. We say that a forbidden graph is of type (i) (resp., (ii), (iii), (iv) and (v)) if it is obtained from a primely forbidden graph in (i) (resp., (ii), (iii), (iv) and (v)). We call a red edge uv in a primely forbidden graph (or a red u, v-path obtained by subdividing it) a *primely red path*. The *core* of a forbidden graph is defined to be the red graph that consists of primely red paths, where we observe that the core is biconnected.

An embedding γ of a graph H is called an E*-planar* embedding for a subset E of edges in H if no edge in E crosses any other edge. An instance (H, E) that admits an E-planar embedding is called E*-planar*.

Lemma 3 *Let $F = (H, E)$ be a forbidden graph, where E is the set of red edges in the graph H. Then F admits no E-planar embedding.*

Proof. When F is of type (iv), the four internally disjoint red u, v-paths divides the plane into four faces in any planar embedding of the four paths, and some of the three gray paths in F always crosses one of the four red paths.

Let F be of type (i), (ii) or (iii). To derive a contradiction, assume that F admits an E-planar embedding γ, where we choose γ so that the number of crossings is minimized. Then we see that no two gray paths with the common end-vertex v_1 have any crossing between them, since otherwise we could switch some parts of these paths to get another E-planar embedding γ' with a smaller number of crossings. When F is of type (i) or (ii), this means that γ is a planar embedding. contradicting that no subdivision of $K_{3,3}$ and K_5 admits a planar embedding. When F is of type (iii), the embedding induced from γ by the red paths in F has a face f whose facial cycle C_f contains the edges in the five gray paths of F, and this implies that a planar embedding of a subdivision of K_5 can be obtained by replacing the five gray paths with four red paths, a contradiction.

Let F be of type (v). Let γ_Q be a planar embedding of the red circular chain Q such that the outer boundary is not any factor cycle in Q. Since the length of the red circular chain is at least 3, no two factor cycles are drawn as four internally disjoint x, y-paths for some vertices x and y in γ_Q. Hence one of the two sides of each factor cycle appears along the interior $f_{\rm in}$ of γ_Q and the other along the exterior $f_{\rm ex}$ of γ_Q (see Fig. 3(e)). When γ_Q can be extended to an E-planar embedding, each of the first $p-1$ gray paths connects sides P_i^k and P_{i+1}^k for the same $k \in \{1, 2\}$ so that both P_i^k and P_{i+1}^k appear in $f_{\rm in}$ or $f_{\rm ex}$. This, however, implies that the p-th path cannot join side P_p^k and P_1^ℓ with $k \neq \ell$ without making a crossing with the red graph Q. □

The main result of this paper is described as follows.

Theorem 4 *Every instance (H, E) of a graph H and a subset $E \subseteq E(H)$ either admits an E-planar embedding or contains a forbidden subgraph. Finding an E-planar embedding or a forbidden subgraph of H can be done in linear time.*

We prove the theorem by a constructive proof with an algorithm that actually finds an E-planar embedding or a forbidden subgraph, showing that it can be implemented to run in linear time. For this, we distinguish instances (H, E) depending on the vertex-connectivity of the red graph $(V(E), E)$. More specifically, we reduce an instance with a red graph of connectivity 0 to that with a red graph of connectivity 1, and reduce an instance with a red graph of connectivity 1 to those with biconnected red graphs. We then design an algorithm that finds either an E-planar embedding or a forbidden graph for each of the cases where the vertex-connectivity of a red graph is 2 and at least 3.

Our forbidden graph characterization enables us to easily reduce an instance with a red graph of connectivity 0 to that with a red graph of connectivity 1. Since the core of a forbidden graph of any type is biconnected, we see that regarding any blue edge e as a red edge does not change the E-planarity (or $(E \cup \{e\})$-planarity) if edge e is not in any cycle of the new red graph $(V(E \cup \{e\}), E \cup \{e\})$.

Lemma 5 *Assume that Theorem 4 is true for any instance such that the red graph is connected. Given an instance $(H = (V_H, E_H), E)$ with a connected graph H, let $\Delta E \subseteq E_H - E$ be a set of blue edges such that no edge in ΔE is contained in a cycle of the graph $(V_H, E' = E \cup \Delta E)$. Then H is E-planar if and only if H is E'-planar. Moreover,*
(a) *if (H, E') admits an E'-planar embedding, then (H, E) admits an E-planar embedding (trivially); and*
(b) *if (H, E') contains a forbidden subgraph F, then regarding the color of edges in $\Delta E \cap E(F)$ as the original red, F is a forbidden subgraph to (H, E).*

Proof. By the assumption on Theorem 4, the instance (H, E') either admits an E'-planar embedding or contains a forbidden subgraph F. Hence (a) and (b) imply that H is E-planar if and only if H is E'-planar. Since (a) is trivial, we show (b). Let F be a forbidden subgraph in (H, E'). Now blue edges in ΔE are regarded as red edges in (H, E'). However, by the choice of ΔE, the new red graph $(V \cup V(\Delta E), E' = E \cup \Delta E)$ contains no red cycle which passes through some edge in ΔE. On the other hand the core of a forbidden graph of any type of (i)-(v) is biconnected. This means that the core of F cannot contain any edge in ΔE; only a gray path of F can contain some edges in ΔE. Hence even after changing the color of edges in ΔE contained in F from red to blue, the graph F is a forbidden graph, which is now a subgraph of (H, E). □

In particular, given an instance $(H = (V_H, E_H), E)$, we choose ΔE as a minimal set of blue edges such that $(V_H, E \cup \Delta E)$ is connected. By the lemma, the new instance $(H, E' = E \cup \Delta E)$ has a special structure that the red graph $(V(E'), E')$ is a connected spanning subgraph of H. See Fig. 4(a) for an illustration of an instance $(H = (V_H, E_H), E)$, where such a minimal set $\{e_1, e_2, \ldots, e_{18}\}$ is chosen, and the new instance $(H, E' = E \cup \Delta E)$ is illustrated in Fig. 4(b).

Now we show how to restrict given instances of the problem to instances with special structure called "star instances." For an instance (H, E), denote the red graph $(V(E), E)$ by $G = (V, E)$. We call an instance $(H = (V_H, E_H), E)$ a *star instance* if it satisfies the following:
(i) The entire graph H is biconnected;
(ii) The red graph G is a connected planar graph but is not outerplanar (hence $E \neq \emptyset$); and
(iii) The set $V(H) - V(E)$ is a nonempty independent set in H, and each vertex in $V(H) - V(E)$ has at least two neighbors (hence the blue edges form a collection of stars).

Lemma 6 *Theorem 4 is true if the statement of the theorem holds for star instances.*

Proof. We prove the lemma by the following five steps (1)-(4).

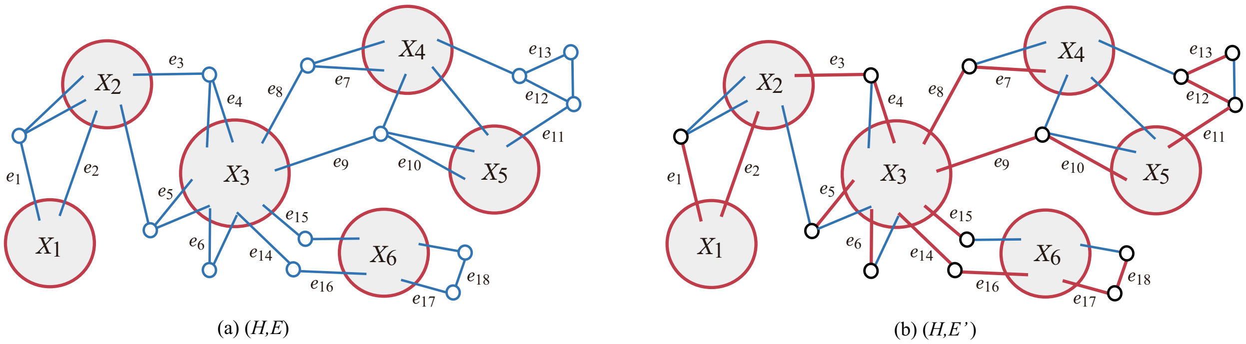


Figure 4: (a) An instance $(H = (V_H, E_H), E)$ such that the red graph $G = (V, E)$ is not connected, where $X_1, X_2, \ldots, X_6$ indicate the nontrivial components in the graph (V_H, E); (b) An instance $(H, E' = E \cup \Delta E)$ obtained from (H, E) in (a) by choosing a minimal set $\Delta E = \{e_1, \ldots, e_{18}\}$ of blue edges such that $(V_H, E \cup \Delta E)$ is connected and regarding the edges in ΔE as red edges.

(1) Without loss of generality we can assume that a given instance (H, E) has a connected graph H. If a given instance (H, E) is not biconnected, then we decompose H into the blocks H_j, $j = 1, 2, \ldots, q$, each of which induces an instance $(H_j, E_j = E(H_j) \cap E)$. We easily see that (a) If an E_j-planar embedding of H_j is given for each j, then an E-planar embedding of H can be obtained by combining them; and (b) If a forbidden subgraph F_j of H_j is given for some j, then it is also a forbidden subgraph F of H. Also all the above tasks can be done in linear time. Now we assume that a given instance (H, E) has a biconnected graph H, satisfying the condition (i) of star instances.

(2) Next find a minimal set ΔE of blues edges that makes (V_H, E) connected in linear time and regard the edges in ΔE as red edges. By Lemma 5, the E-planarity of (H, E) is equivalent with the E'-planarity of $(H, E \cup \Delta E)$, where the new instance has a special structure that the red graph $(V(E'), E')$ is a connected spanning graph of H. Hence each of the end-vertices of any blue edge $uv \in E_H - E'$ is adjacent to a red edge. Then we subdivide each blue edge $e = uv$ into uw_e and $w_e v$ with a new degree-2 vertex w_e so that the condition (iii) of star instances is satisfied.

(3) If the red graph G is not planar, then the original instance (H, E) has no E-planar embedding, and a forbidden subgraph that is a subdivision of K_5 or $K_{3,3}$ can be found in linear time. Note that such a forbidden subgraph is a special case of the primely forbidden subgraph. Hence we can assume that the red graph G is planar in the following.

(4) If the red graph G is outerplanar, then clearly the instance admits an E-planar embedding, which can be obtained from an outerplanar embedding of G by placing all blue edges in the outer face. Hence we can assume that the red graph G is not outerplanar. Finally the condition (ii) of star instances is satisfied. □

In fact, we can also assume that H is not planar when we regard all edges in H red, since otherwise we are done. When H is not planar, it contains a subdivision of K_5 or $K_{3,3}$, which can be obtained in linear time. In general, such a subgraph does not mean the non-E-planarity of (H, E) since it may be drawn in the plane with some crossings between only blue edges.

Given a star instance (H, E), we denote the vertex set $V(H) - V(E)$ by $A = \{s_1, s_2, \ldots, s_k\}$, the set of neighbors of s_i by S_i for each i, where $|S_i| \geq 2$ for each i. We also denote the star instance (H, E) by (H, G, A) or (G, A). For simplicity, we may call $s_i \in A$ a *star*. Let $n = |V|$ and $m = \sum_{1 \leq i \leq k} |S_i|$.

Planarizing Star Partitions

A partition $\mathcal{A}$ of a star set A is called *planarizing* if the graph $H/\mathcal{A}$ obtained from H by contracting each set $A \in \mathcal{A}$ into a single vertex s_A is planar. If a planarizing partition $\mathcal{A} = \{A_1, A_2, \ldots, A_h\}$ exists, then an E-planar embedding of H can be obtained in linear time from a planar embedding $\gamma_{H/\mathcal{A}}$. In fact, let $\gamma^{\text{red}}_{H/\mathcal{A}}$

denote the embedding of the red graph G obtained from $\gamma_{H/\mathcal{A}}$ by deleting all contracted stars, and for each set $A_j \in \mathcal{A}$, let f_j denote the face of $\gamma^{\rm red}_{H/\mathcal{A}}$ that contain the contracted star s_{A_j}. Then by placing the blue edges incident to $s \in A_j$ in the face f_j of $\gamma^{\rm red}_{H/\mathcal{A}}$ for all $j = 1, 2, \ldots, h$, we obtain an E-planar embedding γ_H of H. In what follows, we construct a planarizing partition of A instead of an E-planar embedding of H if one exists.

4 Case of Connectivity 1

This section treats a star instance with a red graph of connectivity 1, and describes how to reduce such an instance to star instances with red biconnected graphs in linear time.

Let (H, G, A) be a star instance with a red connected graph G and a set A of blue stars. By definition, H is biconnected and G is connected. In this section, we assume that a given red graph is not biconnected, i.e., G has $p \geq 2$ blocks $B^1, B^2, \ldots, B^p$, and decompose the instance into p new instances $(B^j = (V^j, E^j), A^j)$, $j = 1, 2, \ldots, p$ by setting each star set A^i adequately so that (G, A) is E-planar if and only if (B^j, A^j) is E^j-planar for all j.

Let $C(G)$ denote the set of cut-vertices in G, let $\mathcal{B}(G)$ denote the set of blocks in G, and let X, Y be subsets of V.

We call set X *pendant* if exactly one vertex $c \in X$ is adjacent to a vertex in $V - X \neq \emptyset$, where $c \in C(G)$ holds. A block B is called *pendant* if $V(B)$ is pendant. The unique cut-vertex $c \in C(G)$ in a pendant set X (resp., block B) is denoted by c_X (resp., c_B).

We say that a star $s \in A$ *links* set $X \subseteq V$ (or a block B with $X = V(B)$) to set $Y \subseteq V$ (or a block B' with $Y = V(B')$) if

$$N(s; H) \cap (X - C(G)) \neq \emptyset \neq N(s; H) \cap (Y - X).$$

A star $s \in A$ is called X-*inter* if it links X to $V - X$, i.e., $N(s; H) \cap (X - C(G)) \neq \emptyset \neq N(s; H) - X$; and a star $s \in A$ is called X-*intra* (or B-intra for a block B with $X = V(B)$) if it does not link X to $V - X$, i.e., $N(s; H) \subseteq X$. Let $A^{\rm inter}(X)$ and $A^{\rm intra}(X)$ denote the sets of X-inter stars $s \in A$ and X-intra stars $s \in A$. Note that when X is pendant, $X' = V - (X - \{c_X\})$ is also pendant and hence $A^{\rm inter}(X) = A^{\rm inter}(X')$.

When $X = V(B)$ for a block B, we may use B instead of X in the above notation X-inter, X-intra, $A^{\rm inter}(X)$ and $A^{\rm intra}(X)$.

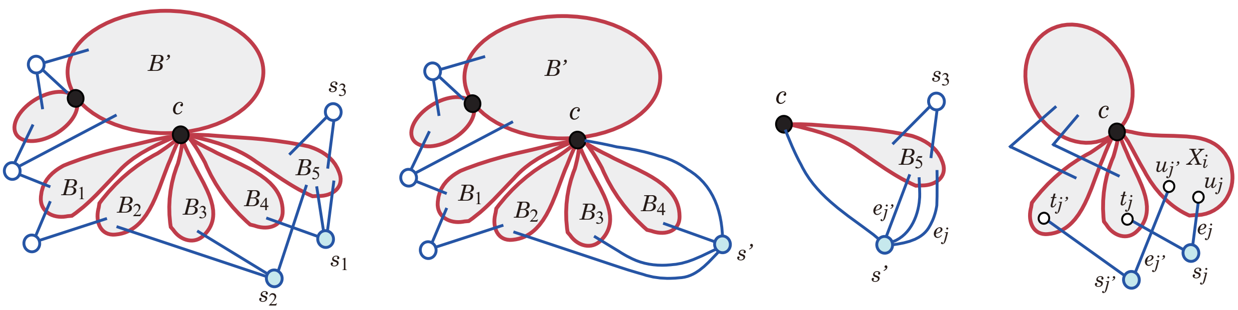
Let X_1 be a pendant set in H, and denote $c = c_{X_1}$ and $X_2 = V - (X - \{c\})$. Then we define two instances $(H_i, G[X_i], A_i)$, $i = 1, 2$ *split* from (G, A) as follows: For each $i = 1, 2$, let H_i denote the union of the induced red graph $G[X_i]$ and blue stars $A_i = A^{\rm intra}(X_i) \cup \{s'\}$, where s' is a new star whose neighbor set $N(s'; H_i) \subseteq X_i$ in H_i is obtained by merging the neighbors in X_i of all X_i-inter stars s and including c as a neighbor; i.e.,

$$N(s'; H_i) := \{c\} \cup \bigcup_{s \in A^{\rm inter}(X_i)} N(s; H) \cap X_i,$$

where $|N(s'; H_i)| \geq 2$. See Fig. 5(a)-(c) for a process of splitting an instance (H, G, A) into instances $(H_i, G[X_i], A_i)$, $i = 1, 2$.

We call a pendant block $B \in \mathcal{B}(G)$ *removable* if any two vertices $u, v \in V - V(B)$ are connected in the graph $H - V(B)$.

Lemma 7 *Let B be a removable block in a star instance (G, A), and let $(H_i, G[X_i] = (X_i, E_i), A_i)$, $i = 1, 2$ be the instances defined for $c = c_B$, $X_1 = V(B)$, $X_2 = V - (X_1 - \{c_B\})$, $E_1 = E(B)$ and $E_2 = E - E_1$.*



(a) (H,G,A) (b) (H_2,G_2,A_2) (c) (H_1,G_1,A_1) (d)

Figure 5: Illustration of splitting a star instance (H, G, A) into two star instances (H_i, G_i, A_i), $i = 1, 2$: (a) A given instance (H, G, A) with a pendant block B_5 at a cut-vertex $c = c_{B_5}$, where $X_1 = V(B_5)$ and $X_2 = V - (X - \{c\})$ are pendent sets, and s_1 and s_2 are B_5-inter stars while s_3 is a B_5-intra star; (b) The resulting instance (H_2, G_2, A_2), where $H(s'; H_2)$ includes c and all neighbors of $s_1, s_2 \in A^{\text{inter}}(B_5)$ not in B_5; and (c) The resulting instance (H_1, G_1, A_1), where $H(s'; H_1)$ includes c and all neighbors of $s_1, s_2 \in A^{\text{inter}}(B_5)$ in B_5; and (d) A pendant set X_i with $c = c_{X_i}$ and edges $e_j, e_{j'} \in \delta(X_i; H)$ which are incident to the merged star s'.

(i) *If $(G[X_i], A_i)$ admits a planarizing partition $\mathcal{A}_i$ of A_i for both $i = 1, 2$, then, for the sets $A_i^0 \in \mathcal{A}_i$, $i = 1, 2$ with $s' \in A_i^0$ and the merged set $A' = (A_1^0 \cup A_2^0 - \{s'\}) \cup A^{\text{inter}}(B)$, the partition $(\mathcal{A}_1 \cup \mathcal{A}_2 - \{A_1^0, A_2^0\}) \cup \{A'\}$ of A is a planarizing partition in (G, A);*

(ii) *If $(G[X_i], A_i)$ for some $i = 1, 2$ contains a forbidden subgraph F, then (G, A) also contains a forbidden subgraph F', which can be constructed from F in $O(n + m)$ time; and*

(iii) *For each $i = 1, 2$, graph H_i is biconnected.*

Proof. Note that $A^{\text{inter}}(B) = A^{\text{inter}}(X_1) = A^{\text{inter}}(X_2)$.

(i) For each $i = 1, 2$, the star set A_i contains a star s' with $N(s'; H_i) = \{c\} \cup \bigcup_{s \in A^{\text{inter}}(B)} N(s; H) \cap V(B)$. Since $\mathcal{A}_i$ is a planarizing partition of A_i, the graph $H_i/\mathcal{A}_i$ has a planar embedding γ_i, where we denote by γ_i^{red} the embedding induced from γ_i by G_i, and we denote by f_i the face of γ_i^{red} in which the edge cs' between the merged star s' and the cut-vertex c is placed. Without loss of generality assume that f_1 is the outer face of γ_1^{red}. Therefore, by placing γ_1^{red} in the face f_2 of γ_2^{red}, we obtain a planar embedding γ_G for the red graph G of H. In the face f_1 of γ_G, we can place all stars $s \in A_1^0 \cup A_2^0 - \{s'\}$ and all stars $s \in A^{\text{inter}}(B)$ merged to s', without creating any new crossing with red edges in G. Clearly the contracted star s_A for any other set $A \in \mathcal{A}_1 \cup \mathcal{A}_2 - \{A_1^0, A_2^0\}$ still can be placed in a face of γ_G, without creating any new crossing with red edges. This proves (i).

(ii) First we claim that for each $i = 1, 2$, every pair of vertices $u, v \in V - X_i$ admits a u, v-path in the graph $H - X_i$. The claim for $i = 1$ is immediate from the definition of removable blocks. Since $V - X_2$ induces a connected graph $B - \{c\}$, the claim for $i = 2$ also holds.

Let F be a forbidden subgraph in $(G[X_i], A_i)$ for some $i = 1, 2$. We show that a forbidden subgraph in H can be obtained from F by replacing some blue edges with paths in $H - (X_i - \{c\})$.

If F does not contain the new star s', then we are done since F is a subgraph of the original instance H. Assume that F contains s', and let $e_j = u_j s'$, $j = 1, \ldots, h$ be the blue edges incident to s' in F, where $h = \deg(s'; F)$ is 2, 3 or 4, since in any type of a forbidden subgraph, the number of blue edges incident to a vertex is 2, 3 or 4. There may be a new blue edge $e_j = cs' \in \delta(s'; H_i)$ in H_i such that c is not adjacent to any B-inter star in H. We call such an edge a *supporting edge* in H_i. Let e_j be a non-supporting edge in $\delta(s'; H_i)$. Since the new star s' is adjacent to all vertices in $N(s; H) \cap X_i$ of an X_i-inter star s, we see that the edge e_j was a blue edge incident to an X_i-inter star, say s_j in H, where s_j has a neighbor

$t_j \in N(s_j; H) - X_i$ in H, which is connected to vertex c by

$$\text{a red } t_j, c\text{-path } P_{t_j,c}. \tag{1}$$

Analogously for any other non-supporting edge $e_{j'} \in \delta(s'; H_i)$ was a blue edge incident to an X_i-inter star, say $s_{j'}$ in H, which is adjacent to $t_j \in N(s_j; H) - X_i$ in H. By the above claim, t_j and $t_{j'}$ are connected by

$$\text{a } t_j, t_{j'}\text{-path } P_{t_j,t_{j'}} \text{ in } H - X_i, \tag{2}$$

and hence H has an $s_j, s_{j'}$-path $P_{s_j,s_{j'}}$ that does not pass through any vertex in X_i, where possibly $s_j = s_{j'}$. See Fig. 5(d) for an illustration of set X_i and edges $e_j, e_{j'} \in \delta(X_i; H)$. We distinguish three cases of $h = 2, 3$ and 4.

(a) $h = 2$: First assume that some $e_j = u_j s' \in \delta(s'; H_i)$ is an edge cs', say $u_2 s' = cs'$. We replace the blue edges $e_1 = u_1 s', e_2 = cs' \in \delta(s'; H_i)$ with a u_1, c-path in H that consists of blue edges $u_1 s_1$, $s_1 t_1$ and the above red t_1, c-path $P_{t_1,c}$ in (1). Then the resulting graph F' is a forbidden subgraph to H.

Next assume that $c \neq u_j$ for each $j = 1, 2$. If $s_1 = s_2$, i.e., e_1 and e_2 are adjacent to the same star $s_1 = s_2$ in H, then we are done. Let $s_1 \neq s_2$. We replace the blue edges $e_1, e_2 \in \delta(s'; H_i)$ in F with a u_1, u_2-path in H that consists of blue edges $u_j s_j$, $s_j t_j$, $j = 1, 2$ and the above t_1, t_2-path P_{t_1,t_2} in (2). Then the resulting graph F' is a forbidden subgraph to H.

(b) $h = 3$: Without loss of generality assume that $u_3 = c$ if some edge $e_j \in \delta(s'; H_i)$ is an edge cs'. Analogously with the case of $h = 2$, we see that H has a u_1, u_2-path P_{u_1,u_2} which passes through s_1 and contains no vertex in $X_i - \{u_1, u_2\}$ and no blue edges other than $u_1 s_1$ and $u_2 s_2$ in H_i, where possibly $s_1 = s_2$. Graph H also contains an s_1, u_3-path P_{s_1,u_3} in H that consists of blue edges $s_1 t_1$, $t_3 s_3$ and $s_3 u_3$ and the above t_1, t_3-path P_{t_1,t_3} in (2) (or of blue edge $s_1 t_1$ and the above red t_1, c-path $P_{t_1,c}$ in (1) when $c = u_3$). Let $x \in V(P_{u_1,u_2}) - \{u_1, u_2\}$ be the first vertex that appears for the first time when we traverse P_{s_1,u_3} from u_3 to s_1. Let $P_{u_3,x}$ be the subpath of P_{s_1,u_3} from u_3 to x. Then we see that the graph F' obtained from F by replacing the blue edges $e_1, e_2, e_3 \in \delta(s'; H_i)$ with paths P_{u_1,u_2} and $P_{u_3,x}$ is a forbidden subgraph to H.

(c) $h = 4$: Now F is of types (ii). Without loss of generality assume that $u_3 = c$ if some edge $e_j \in \delta(s'; H_i)$ is an edge cs'. Let P_{u_1,u_2} and and $P_{u_3,x}$ be the paths defined for edges $e_1, e_2, e_3 \in \delta(s'; H_i)$, as in the case of $h = 3$. Analogously with case of $h = 3$, there is an s_1, u_4-path P_{s_1,u_4} in H that contains no vertex in $X_i - \{u_4\}$ and no blue edges other than $t_4 s_4$ and $s_4 u_4$ in H. Let $y \in V(P_{u_1,u_2}) \cup V(P_{u_3,x}) - \{u_1, u_2, u_3\}$ be the first vertex that appears for the first time when we traverse P_{s_1,u_4} from u_4 to s_1. Let $P_{u_4,y}$ be the subpath of P_{s_1,u_4} from u_4 to y. Then we replace the blue edges $e_1, e_2, e_3, e_4 \in \delta(s'; H_i)$ in F with paths P_{u_1,u_2}, $P_{u_3,x}$ and $P_{u_4,y}$ to obtain a graph F'. Although possibly F' becomes of type (iii) when F is of type (ii) and $x \neq y$, we see that F' is a forbidden subgraph to H.

We easily see that the above construction of F' from F can be executed in $O(n + m)$ time.

(iii) Note that $|N(s'; H_i)| \geq 2$ for each $i = 1, 2$. Since each star $s \in A_i$ in instance (H_i, A_i) is adjacent to at least two vertices in X_i but no other vertices in $V - X_i$, no star $s \in A_i$ can be a cut-vertex in $C(H_i)$. Hence it suffices to show that no vertex $c \in C(H) \cap X_i$ is a cut-vertex in H_i. By claim in (ii), the vertex $c \in C(H) \cap X_1 \cap X_2$ is not a cut-vertex in H_i. Assume that H_i contains a cut-vertex $c' \in C(H_i)$, which separates the vertex c and a block B' with $V(B') \subseteq X_i - \{c, c'\}$. Since the new star s' is adjacent to vertex c, we see that no star $s \in A^{\text{inter}}(B)$ has any neighbor in B'. This, however, implies that the vertex c' was a cut-vertex in H, contradicting the biconnectivity of H. □

Next we show that every star instance (G, A) such that $|\mathcal{B}(G)| = p \geq 2$ has a removable block, based on which we can decompose a given instance (G, A) into p new star instances (B^j, A^j), $j = 1, 2, \ldots, p$. For this, we introduce a parent-child order among cut-vertices and blocks in G.

For two sets X_1 and $X_2 \subseteq V$, where $X_i = \{c\}$ or $V(B)$ for $c \in C(G)$ or $B \in \mathcal{B}(G)$, the *distance* between them is defined to be the number of cut-vertices in an X_1, X_2-path P, where $|P \cap X_i| = 1$ by the definition of S, T-paths.

For each cut-vertex $c \in C(G)$, let $\mathcal{B}(c; G)$ denote the set of blocks $B \in \mathcal{B}(G)$ such that $c \in V(B)$. For each block $B \in \mathcal{B}(G)$, let $C(B; G)$ denote the set of cut-vertices $c \in C(G)$ such that $c \in V(B)$.

Choose a block B^1 as a root, where we keep B^1 as the root even after we remove some pendant blocks B ($\neq B^1$) from G. For each cut-vertex $c \in C(G)$, let $B(c; G)$ be the block $B \in \mathcal{B}(c; G)$ with minimum distance to the root B^1, and denote $\mathcal{B}^-(c; G) = \mathcal{B}(c; G) - \{B(c; G)\}$, where the block $B(c; G)$ is called the *parent* of a block in $\mathcal{B}^-(c; G)$, and each block in $\mathcal{B}^-(c; G)$ is called a *child* of $B(c; G)$. For each non-root block $B \in \mathcal{B}(G)$, let c_B denote the cut-vertex c such that $B \in \mathcal{B}^-(c; G)$.

For each block $B \in \mathcal{B}(G)$, let $c(B; G)$ denote the cut-vertex $c \in C(B; G)$ with minimum distance to the root B^1, and denote $C^-(B; G) = C(B; G) - \{c(B; G)\}$, where the cut-vertex $c(B; G)$ is called the *parent cut-vertex* of a block $B \in \mathcal{B}^-(c; G)$. A cut-vertex c with $\mathcal{B}^-(c; G) = \emptyset$ is called a *leaf cut-vertex*. A non-root block B is *outer-linked* if some star $s \in A$ links B to a block B' which is not a descendant of B. We say that two blocks B and B' are *star-connected* if $B = B'$ or there is an alternating sequence $(B_1 = B, s_1, B_2, \ldots, s_{h-1}, B_h = B')$ of blocks in $\mathcal{B}^-(c; G)$ and stars such that each s_i links B_i and B_{i+1}.

An order $B_1, B_2, \ldots, B_h$ of blocks is called *proper* to the set $\{B_1, B_2, \ldots, B_h\}$ of blocks if for each $i = 1, 2, \ldots, h$, block B_h is star-connected to an outer-linked block $B_{j'}$ with $j' \leq j$. By definition, the order $B_1, B_2, \ldots, B_j$ for any $j \leq h$ is also proper to the set $\{B_1, B_2, \ldots, B_j\}$.

Lemma 8 *For a leaf cut-vertex $c \in C(G)$ in a star instance (H, G, A) such that H is biconnected, there always exists a proper sequence $B_1, B_2, \ldots, B_q$ of blocks in $\mathcal{B}^-(c; G)$, and the last block B_q is removable.*

Proof. Since c is not a cut-vertex in the biconnected graph H, there is at least one outer-linked block in $\mathcal{B}^-(c; G)$ and each non-outer-linked block $B \in \mathcal{B}^-(c; G)$ is star-connected to an outer-linked block. This implies that there always exists a proper sequence of blocks in $\mathcal{B}^-(c; G)$. Let B_q be the last block in a proper sequence $B_1, B_2, \ldots, B_q$ of blocks in $\mathcal{B}^-(c; G)$. Then any other block $B_j \in \mathcal{B}^-(c; G) - \{B_q\}$ is still star-connected to an outer-linked block. This means that any two vertices $u, v \in V - V(B_q)$ admit a u, v-path in H that does not pass through any vertex in $V(B_q)$, i.e., B_q is removable. □

Given a proper sequence $B_1, B_2, \ldots, B_q$ of blocks in $\mathcal{B}^-(c; G)$ at a leaf-cut-vertex c, we can repeatedly apply the lemma to each B_i, $i = q, q-1, \ldots, 1$. More formally for each $i = q, q-1, \ldots, 1$, let $(H'_{i+1}, G'_{i+1}, A'_{i+1})$ denote the current instance after the first $q - i$ instances

$$(H_q, B_q, A_q), (H_{q-1}, B_{q-1}, A_{q-1}), \ldots (H_{i+1}, B_{i+1}, A_{i+1})$$

are generated, where $(H'_{q+1}, G'_{q+1}, A'_{q+1}) = (H, G, A)$. Since each subsequence $B_1, B_2, \ldots, B_i$ with $i < q$ is also proper, the last block B_i in the subsequence is removable in H'_{i+1} by Lemma 8 and the resulting graph H'_i obtained by splitting B_i off is biconnected by Lemma 7(iii). Therefore by Lemma 7(i)-(ii), the original instance $(H, G = (V, E), A)$ is E-planar if and only if the instance (H_i, G_i, A_i) is E_i-planar for each $i = 0, 1, \ldots, q$, where $G_i = B_i$, $E_i = E(B_i)$ for $i = 1, \ldots, q$, and $(H_0, G_0 = (V_0, E_0), A_0)$ denotes the remaining instance (H'_1, G'_1, A'_1). Note that $V_0 = V - (\cup_{1 \leq i \leq q} V(B_i) - \{c\})$.

All the red graphs $G[V_i] = B_i$, $i = 1, 2, \ldots, q$ and $G[V_0]$ in these $q+1$ instances are simply determined by the set of blocks in G. Let us show how each star set A_i will be constructed by the repeated application of the lemma. The star set A_i of the i-th instance is constructed as follows. Let $A^{\rm intra}(B_i) \subseteq A'_{i+1}$ be the set of B_i-intar stars in H'_{i+1}, and let $A^{\rm inter}(B_i) \subseteq A'_{i+1}$ be the set of B_i-inter stars in H'_{i+1}. Clearly any B_i-intra star s will be included in A_i. Hence $A^{\rm intra}(B_i) \subseteq A_i$. We merge all B_i-inter stars into a single star s' such that

$$\begin{aligned} N(s'; H_i) &:= \{c\} \bigcup \{N(s; H'_{i+1}) \cap V(B_i) \mid s \in A^{\rm inter}(B_i)\}, \\ N(s'; H'_i) &:= \{c\} \bigcup \{N(s; H'_{i+1}) - (V(B_i) - \{c\}) \mid s \in A^{\rm inter}(B_i)\}. \end{aligned} \tag{3}$$

Then let

$$A_i := A^{\rm intra}(B_i) \cup \{s'\}, \;\; A'_i := (A'_{i+1} - A^{\rm intra}(B_i) - A^{\rm inter}(B_i)) \cup \{s'\}, \tag{4}$$

where we call the star s' the *preceding star* of H_i and the *succeeding star* of H'_i.

The procedure for splitting all blocks in $\mathcal{B}^-(c;G')$ off at a leaf-cut-vertex c is summarized as follows.
SPLIT(c)
1. Find a proper order sequence $B_1, B_2, \ldots, B_q$ of blocks in $\mathcal{B}^-(c;G')$;
2. Construct the red graphs $G[V_i] = B_i$, $i = 1, 2, \ldots, q$ and $G[V_0]$ with $V_0 = V - (\cup_{1\leq i\leq q} V(B_i) - \{c\})$;
3. Compute the star sets A_i, $i = q, q-1, \ldots, 1$ and $A_0 = A'_1$ according to (3) and (4).

To reduce a given instance (H, G, A) into $p = |\mathcal{B}(G)|$ instances (B, A_B), $B \in \mathcal{B}(G)$, we repeatedly choose a leaf-cut-vertex c in the current instance (H', G', A') and apply the above procedure SPLIT(c) to the blocks in $\mathcal{B}^-(c;G')$. We show that generating star sets A_B of (B, A_B) for all blocks $B \in \mathcal{B}(G)$ can be implemented to run in $O(n+m)$ time.

The adjacency between blocks and cut-vertices in G can be represented by a tree structure, called the *block-cut-vertex tree* $\mathrm{BC}(G) = (\mathcal{B}(G) \cup C(G), E_{\mathrm{BC}})$, a bipartite tree between two vertex sets $\mathcal{B}(G)$ and $C(G)$ such that the edge set E_{BC} contains an edge Bc if and only if $c \in V(B)$. Regard $\mathrm{BC}(G)$ as an ordered tree rooted at the root B^1, and let $\mathrm{ld} : \mathcal{B}(G) \cup C(G) \to \{1, 2, \ldots, |\mathcal{B}(G) \cup C(G)|\}$ be the left depth-first order. We choose cut-vertices $c \in C(G)$ in the decreasing order of $\mathrm{ld}(c)$ and apply SPLIT(c).

Let E_c^{blue} be the set of blue edges incident to a block $B \in \mathcal{B}^-(c;G')$ in the current instance (H', G', A') when c is selected, i.e., $E_c^{\mathrm{blue}} = \cup\{\delta(B;H') - E(G) \mid B \in \mathcal{B}^-(c;G')\}$. We show the next lemma, which implies that the entire algorithm of generating all $p = |\mathcal{B}(G)|$ instances runs in $O(n+m)$ time.

Lemma 9 *Let (H', G', A') be in the current instance when a leaf-cut-vertex c is selected. Then* SPLIT(c) *can be implement to run in $O(|E_c^{\mathrm{blue}}|)$ time.*

Proof. Before we execute the entire algorithm, we first prepare the following data structure in a given instance (H, G, A). Define a mapping $\psi : V \to \mathcal{B}(G)$ such that ψ maps a vertex $v \in V$ to a block B closest to the root B^1 among blocks B with $v \in V(B)$.

We prepare a list $L(s)$ of all edges incident to each star $s \in A$ so that an edge $su \in \delta(s;H)$ appears before any edge $sv \in \delta(s;H)$ with $\mathrm{ld}(\psi(u)) \leq \mathrm{ld}(\psi(u))$. Let $\mathrm{pL}(s)$ denote a pointer that indicates the address of $L(s)$. By visiting all blocks $B \in \mathcal{B}(G)$ in the left depth-first search manner, such lists $L(s)$ for all stars $s \in A$ can be constructed in $O(n+m)$ time. By the above way of storing vertices in the list $L(s)$ for each star $s \in A$, when the blocks in $\mathcal{B}^-(c;G)$ at a leaf-cut-vertex c are removed by the procedure SPLIT(c), the vertices in $N(s;H) \cap (\cup_{B\in\mathcal{B}^-(c;G)} V(B))$ appear consecutively in the list $L(s)$, and there is no need to access any other vertices stored in the list $L(s)$ before we proceed to other leaf-cut-vertices.

When we merge several stars, say $s_1, s_2, \ldots, s_h$ into a single star s^*, we do not directly merge their lists $L(s_i)$, $i = 1, 2, \ldots, h$. Instead, we link their pointers $\mathrm{pL}(s_i)$, $i = 1, 2, \ldots, h$ with a doubly-linked list $\mathrm{dll}(s^*)$. In fact, each star s_i may have consisted of several stars whose pointers pL are linked by $\mathrm{dll}(s_i)$. In this case, merging star $s_1, s_2, \ldots, s_h$ is executed by joining doubly-linked lists $\mathrm{dll}(s_i)$, $i = 1, 2, \ldots, h$ into a single doubly-linked list $\mathrm{dll}(s^*)$ in $O(h)$ time. Note that when some list $L(s_i)$ becomes empty, it can be removed from $\mathrm{dll}(s^*)$ in $O(1)$ time. Hence updating lists $L(s)$ of all B-inter stars s for some $B \in \mathcal{B}^-(c;G)$ still can be executed in $O(|E_c^{\mathrm{blue}}|)$ time.

When a leaf-cut-vertex c in the current instance (H', G', A') is selected, we execute SPLIT(c) as follows. Let $V_0 = V - (\cup\{V(B) \mid B \in \mathcal{B}^-(c;G')\} - \{c\})$. First compute the set A_c of stars s that are B-inters for some block $B \in \mathcal{B}^-(c;G')$ in the current instance (H', G', A'), and construct a bipartite graph $W = (\mathcal{B}^-(c;G') \cup A_c, E_W)$ between two sets $\mathcal{B}^-(c;G')$ and A_c such that E_W contains an edge Bs if and only if $N(s;H') \cap (V(B) - \{c\}) \neq \emptyset$. This can be constructed by scanning the edge set $\delta(B;H') - E(G')$ for all blocks in $B \in \mathcal{B}^-(c;G')$ in $O(|E_c^{\mathrm{blue}}|)$ time.

The set of outer-linked blocks in $\mathcal{B}^-(c;G')$ can be computed as follows. By checking the list $L(s)$ for each star $s \in A_c$, it takes $O(|E_c^{\mathrm{blue}}|)$ time to find the set $A^{\mathrm{inter}}(V_0)$ of all stars $s \in A_c$ such that s has a neighbor $z \in V_0 - \{c\}$, since the edges su in $L(s)$ are stored in the order of $\mathrm{ld}(\psi(u))$ and all edges

$su \in L(s)$ adjacent to a block in $\mathcal{B}^-(c; G')$ appear last. Then we can conclude that any block B that is adjacent to a star in $A^{\rm inter}(V_0)$ is outer-linked.

By a graph search procedure starting from outer-linked blocks in the graph W, we can construct a spanning forest of W, from which a proper sequence $B_1, B_2, \ldots, B_q$ of blocks in $\mathcal{B}^-(c; G')$ is obtained based on the distance from the outer-linked blocks in W. This again takes $O(|E_c^{\rm blue}|)$ time.

Finally we examine each iteration of SPLIT(c) which constructs A_i and A'_i for each $i = q, q-1, \ldots, 1$ according to the formula (3) and (4). From the way of constructing new star sets A_i and A'_i, we observe that once a blue edge su with $u \neq c$ is scanned as an edge incident to a star $s \in A^{\rm intra}(B_i) \cup A^{\rm inter}(B_i)$, the same edge will never be scanned in any instance (H'_j, G'_j, A'_j) with $j < i$. Since A_i and A'_i can be constructed in time linear to the number of blue edges incident to a star $s \in A^{\rm intra}(B_i) \cup A^{\rm inter}(B_i)$ in H'_i, the total time for constructing all star sets $A_q, A_{q-1}, \ldots, A_1$ and $A_0 = A'_1$ is $O(|E_c^{\rm blue}|)$. □

Finally we give an entire algorithm for a star instance (H, G, A) such that G is not biconnected. As observed in the above, we generate in $O(n+m)$ time $p = |\mathcal{B}(G)|$ instances (H^j, B^j, A^j), $j = p, p-1, \ldots, 1$ by a repeated application of SPLIT, where (H^j, B^j, A^j) means the $(p-j+1)$-st instance generated by the algorithm and $(\overline{H}^j, \overline{G}^j, \overline{A}^j)$, $j = p-1, p-2, \ldots, 0$ denotes the instance obtained from (H, G, A) by splitting off the first j blocks $B^p, B^{p-1}, \ldots, B^{p-j+1}$. During this execution, we store the set of stars s'_j, $j = 1, 2, \ldots, p-1$ such that s'_j is the preceding star of H^j and is the succeeding star of $\overline{H}^j$ and the set $A^{\rm intra}(B_i) \subseteq \overline{A}_{i+1}$ of B_i-intar stars in $\overline{H}_{i+1}$ for $j = 1, 2, \ldots, p-1$, where $\sum_i |A^{\rm intra}(B_i)| = O(m)$. Assume that Theorem 4 is true for instances with red biconnected graphs, as will be shown in the following sections. Then either there is an instance (H^j, B^j, A^j) which contains a forbidden subgraph F^j or each instance (H^j, B^j, A^j) admits a planarizing partition $\mathcal{A}^j$ of A^j. In the former, a forbidden subgraph F' in the given instance can be obtained from F^j in $O(n+m)$ time by Lemma 7(ii), and we are done. In the latter, we construct a planarizing partition $\overline{\mathcal{A}}^j$ to instance $(\overline{H}^j, \overline{G}^j, \overline{A}^j)$ in the order of $j = 1, 2, \ldots, p$. For $j = 1$, a planarizing partition $\mathcal{A}^1$ to instance $(H^1, B^1, A^1) = (\overline{H}^1, \overline{G}^1, \overline{A}^1)$ is obtained by assumption. Assume that for some j, a planarizing partition $\overline{\mathcal{A}}^j$ of $\overline{A}^j$ is obtained. For the succeeding star s'_j of $\overline{H}^j$, there is a set $\overline{A}_0^j \in \overline{\mathcal{A}}^j$ with $s'_j \in \overline{A}_0^j$, and the preceding star s'_j of H^j, there is a set $A_0^j \in \mathcal{A}^j$ with $s'_j \in A_0^j$. Then by Lemma 7(i), for the merged set $A' = (\overline{A}_0^j \cup A_0^j - \{s'_j\}) \cup A^{\rm inter}(B_j)$, the partition $(\overline{\mathcal{A}}^j \cup \mathcal{A}^j - \{\overline{A}_0^j, A_0^j\}) \cup \{A'\}$ of A is a planarizing partition $\overline{\mathcal{A}}^{j-1}$ to $(\overline{H}^{j-1}, \overline{G}^{j-1}, \overline{A}^{j-1})$. This can be executed in $O(1)$ time if we store stars in a partition $\mathcal{A}$ of sets as a doubly-linked list in which the stars in the same set $A \in \mathcal{A}$ appear consecutively. Therefore a planarizing partition $\overline{A}$ of (H, G, A) can be obtained in linear time.

5 Case of Connectivity at Least 3

This section treats a star instance with a red graph of vertex-connectivity at least 3, before Sections 6, 7 and 8 handle a star instance with a red graph of vertex-connectivity 2. Since we do not modify a given instance any more in the following sections, we denote the set A of stars by $\{s_1, s_2, \ldots, s_k\}$ and the set $N(s_i; H)$ for each star $s_i \in A$ by S_i. We may denote by $G + \delta(s_i)$ the graph $(V \cup \{s_i\}, E \cup \{s_i t \mid t \in S_i\})$.

This section presents an algorithm for testing whether a given instance (H, E) with a triconnected red graph is E-planar, where we use a geometric argument based on convex grid drawings to make a naive quadratic time algorithm run in linear time.

To apply the result in this section to "triconnected components" in Section 7, we here assume that $G = (V, E)$ is a pseudo-triconnected planar graph obtained from a triconnected planar graph $G' = (V', E')$ by subdividing some edges in E' inserting one vertex per edge so that no two degree-2 vertices are adjacent in G. Fix a planar embedding γ_G of G, and for each star $s_i \in A$, let $\beta(s_i)$ denote the set of faces f in γ_G whose facial cycle C_f contains all vertices in S_i, where $|\beta(s_i)| \leq 2$ since $|S_i| \geq 2$ is assumed.

Theorem 10 *Let $(H, G = (V, E), A)$ be a star instance with a pseudo-triconnected planar graph $G = (V, E)$ obtained from a triconnected planar graph $G' = (V', E')$ by subdividing some edges in E' inserting exactly one degree-2 vertex. Let γ_G be a planar embedding of G.*

(i) *Then (H, G, A) is E-planar if and only if $\beta(s_i) \neq \emptyset$ for each $s_i \in A$;*

(ii) *Computing $\beta(s)$ for all stars $s \in A$ can be done in $O(n + m)$ time.*

Proof. For each star $s_i \in A$, let G_i denote the graph $G + \delta(s_i)$.

(i) From the fact that a combinatorial planar embedding is unique up to reversal, it is immediate to see that, for each i, the embedding γ_G has no facial cycle that contains all the vertices in S_i if and only if G_i is not planar and has no E-planar embedding. If for each i, the embedding γ_G has a facial cycle that contains all the vertices in S_i, then clearly the current embedding γ_G provides an E-planar embedding of (G, A), where we draw each star $\{s_i t \mid t \in S_i\}$ within some face $f \in \beta(s_i)$ of γ_G.

(ii) Let $\gamma_{G'}$ be a planar embedding of triconnected graph G', and γ_G be the planar embedding of G obtained from $\gamma_{G'}$ by inserting degree-2 vertices of $V(G) - C(G')$ in the subdivided edges in G. See Fig. 6(a) for an illustration of a planar embedding of a red pseudo-triconnected graph G.

Assume that for each vertex $v \in V$, we have an index set $I(v) = \{i \mid v \in S_i\}$ and the rotation system $\rho(v)$ of v as an alternating sequence of the neighbors of v in γ_G and the faces incident to v:

$$\rho(v) = (u_1, f_{1,2}, u_2, f_{2,3}, \ldots, u_d, f_{d,1}),$$

where $N(v; G) = \{u_1, u_2, \ldots, u_d\}$, $u_1, u_2, \ldots, u_d$ appear in this order around v in the anti-clockwise way, and $f_{j,j+1}$ is the face incident to v, u_j and u_{j+1} (see vertex v in Fig. 6(a), where $I(v) = \{1, 2, 3, 4, 5, 6\}$).

To obtain $\beta(s)$ for all $s \in A$, we initialize $\beta_s := \emptyset$ for all $s \in A$, where $\beta(s)$ will be given by a final β_s. For the outer face f^o of γ_G, we traverse the facial cycle C_{f^o} to count $c(f^o, i) = |\{v \in V(C_{f^o}) \mid i \in I(v)\}|$ for each i such that $i \in I(v)$ for some vertex v in C_{f^o}. Clearly for each i, $|S_i| = c(f^o, i)$ if and only if C_{f^o} contains all the vertices in S_i. Let $\beta_{s_i} := \{f^o\}$ for each $s_i \in A$ with $c(f^o, i) = |S_i|$. For example, $|S_8| = 3 = c(f^o, 8)$ holds for star s_8 in Fig. 6(a).

We can apply the same procedure to each of inner faces in γ_G to test whether each S_i is contained in some facial cycle. However, this would take $\Omega(nm)$ time, since the same vertex $t \in S_i$ will be counted $\deg(t; G)$ times in total. To avoid this, we prepare the following data structure:

(a) For each vertex $v \in V$ and a star s_i with $v \in S_i$, we "guess" at most two inner faces $f(i; v)$ and $f'(i; v)$ in γ_G so that no other faces can contain S_i (where we call such faces *inevitable*). For example, we see that for vertex v and star s_5 in Fig. 6(a), the inevitable faces $f(5; v)$ and $f'(5; v)$ are faces $f_{5,1}$ and $f_{1,2}$;

(b) We then modify in linear time the rotation system $\rho(v) = (u_1, f_{1,2}, u_2, f_{2,3}, \ldots, u_d, f_{d,1})$ of each vertex v as follows: At a vertex $v \in V$, each index $i \in I(v)$ has one inevitable face $f(i; v)$ or two inevitable faces $f(i; v)$ and $f'(i; v)$ around v. We let $I(v; f_{j,j+1})$ store all indices $i \in I(v)$ such that $f_{j,j+1} = f(i; v)$ or $f'(i; v)$. Then modify $\rho(v)$ into an alternating sequence of the neighbors of v and the pairs $\{f_{j,j+1}, I(v; f_{j,j+1})\}$ of faces incident to v and the index sets; i.e.,

$$\rho^*(v) = (u_1, \{f_{1,2}, I(v; f_{1,2})\}, u_2, \{f_{2,3}, I(v; f_{2,3})\}, u_3, \ldots, u_d, \{f_{d,1}, I(v; f_{d,1})\}).$$

For example, vertex v in Fig. 6(a) is adjacent to six stars s_i with $i \in I(v) = \{1, 2, 3, 4, 5, 6\}$, and we see that inevitable faces are given by $f(1, v) = f_{1,2}$, $f(2, v) = f_{5,1}$, $f(3, v) = f_{2,3}$, $f'(3, v) = f_{3,4}$, $f(4, v) = f_{3,4}$, $f(5, v) = f_{5,1}$ and $f(6, v) = f_{5,1}$, from which we have $\rho^*(v) = (u_1, \{f_{1,2}, \{1\}\}, u_2, \{f_{2,3}, \{3\}\}, u_3, \{f_{3,4}, \{3, 4\}\}, u_4, \{f_{4,5}, \emptyset\}, u_5, \{f_{5,1}, \{2, 5, 6\}\})$.

With this data $\{\rho^*(v) \mid v \in V\}$, we traverse the facial cycle C_f of each inner face f in γ_G in the clockwise order as follows. Let $v_1, v_2, \ldots, v_q$ be the vertices in $V(C_f)$ appearing in the clockwise along C_f. Then we visit $\rho^*(v_1), \rho^*(v_2), \ldots, \rho^*(v_q)$ in this order, during which we collect the indices in $I(v_1; f)$, $I(v_2; f)$, $I(v_3; f), \ldots, I(v_q; f)$. For each index $i \in I(v_1; f) \cup I(v_2; f) \cup I(v_3; f) \cup \cdots \cup I(v_q; f)$, we can test whether C_f contains S_i by counting $c(f, i)$, how many times i appears in the union, and execute $\beta_{s_i} := \beta_{s_i} \cup \{f\}$ for each $s_i \in A$ with $c(f, i) = |S_i|$. After traversing all inner faces, the resulting β_s is equal to $\beta(s)$. The total time for applying the procedure over all inner faces is $O(n+m)$, since each index i of some S_i appears in at most two sets among the sets $I(v; f_{1,2}), I(v; f_{2,3}), \ldots, I(v; f_{d,1})$ in each sequence $\rho^*(v)$.

Finally we show how to attain the conditions (a) and (b) above. For this, we use the fact that every triconnected planar graph admits a convex drawing, in which each edge is drawn as a straight-line segment and each facial cycle forms a convex polygon. Moreover, we can restrict the position of each vertex as a grid point in a grid space of $(n-2) \times (n-2)$, and such a convex grid drawing can be constructed in linear time [13]. We first construct a convex drawing $D_{G'}$ of $\gamma_{G'}$ in a grid space of $(n-2) \times (n-2)$ in $O(n)$ time. Next double the scale to obtain a grid space of $(2n-4) \times (2n-4)$ so that each vertex in G' is on a grid point with even integers. To obtain a convex drawing of γ_G in the grid space of $(2n-4) \times (2n-4)$, we insert a degree-2 vertex $w \in V(G) - C(G')$ for each edge uv that is subdivided into uw and wv in G in such a way that w is placed in the middle point of between the points for vertices u and v. Note that such a middle point is a grid point in the space $(2n-4) \times (2n-4)$. Let D_G be the resulting convex grid drawing of γ_G. For each vertex $u \in V$, let $(x(u), y(u))$ be the grid point on which u is drawn in D_G. See Fig. 6(b) for an illustration of a grid drawing D_G of γ_G in Fig. 6(a).

We fix a vertex $v \in V$ with a rotation system $\rho(v) = (u_1, f_{1,2}, u_2, f_{2,3}, \ldots, u_d, f_{d,1})$. Regarding $(x(v), y(v))$ as the origin of the grid xy-plane, denote $x(u) - x(v)$ and $y(u) - y(v)$ by $\bar{x}(u)$ and $\bar{y}(u)$, respectively, for a notational convenience. Then for example, the angle $\theta \in [0, 2\pi)$ of line segment (v, u_1) from the horizontal line is $\theta(x(u_1)-x(v), y(u_1)-y(v)) = \theta(\bar{x}(u_1), \bar{y}(u_1))$, where we denote $\theta(\bar{x}(u), \bar{y}(u))$ by $\theta_v(u)$ for simplicity. For each index $i \in I(v)$, we choose a vertex $w_i \in S_i - \{v\}$ to determine the direction from v to w_i in the convex grid drawing. For example, a vertex $w_i \in S_i$ is chosen for each star s_i with $i \in I(v)$ in Fig. 6(a).

Since each face f is drawn as a convex polygon P_f, if v and w_i are contained in the same facial cycle C_f, then the line-segment from $(x(v), y(v))$ to $(x(w_i), y(w_i))$ must be contained in the polygon P_f. Hence, if the angle $\theta_v(w_i)$ satisfies

$$\theta_v(u_j) < \theta_v(w_i) < \theta_v(u_{j+1})$$

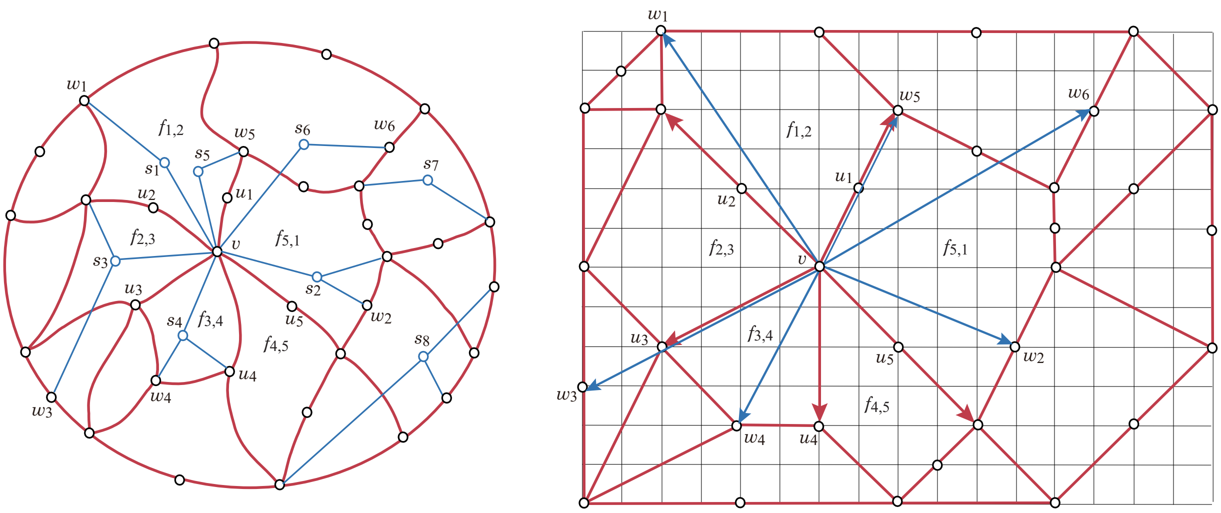
then we can conclude that the face $f_{j,j+1}$ between u_j and u_{j+1} at v must be the inevitable face $f(i; v)$, where no other facial can contain S_i. Analogously, if

$$\theta_v(w_i) = \theta_v(u_j)$$

then the faces $f_{j-1,j}$ and $f_{j,j+1}$, one between u_{j-1} and u_j and the other between u_j and u_{j+1} at v must be the inevitable faces $f(i; v)$ and $f'(i; v)$.

This proves that we can meet the condition (a) in the above. To construct the modified rotation system $\rho^*(v)$ in the condition (b), we sort the vertices u in $\{u_1, u_2, \ldots, u_d\} \cup \{w_i \mid i \in I(v)\}$ in a non-decreasing order of their angles $\theta_v(u)$. This can be done in $O(\deg(v) + |I(v)|)$ time, as will be shown by Lemma 11 in this section. Based on the sorted list, we can easily determine the inevitable faces of each $i \in I(v)$ and construct $\rho^*(v)$ in $O(\deg(v) + |I(v)|)$ time. The total time for constructing the modified rotation system $\rho^*(v)$ over all vertices $v \in V$ is $O(\sum_{v \in V}(\deg(v) + |I(v)|) = O(n+m)$. This proves that we can meet the condition (b) in the above. □

To complete the above proof of Theorem 10, we present a technical lemma on how to approximate angles between grid points in the x, y-grid plane. Let $[a, b]_{\mathbb{Z}}$ denote the set of integers x with $a \leq x \leq b$.



(a) (b)

Figure 6: (a) An instance (H, G, A) with a red pseudo-triconnected planar graph $G = (V, E)$ and stars $A = \{s_1, s_2, \ldots, s_8\}$ and a planar embedding γ_G of G; (b) A grid convex drawing D_G of G, where w_i is another neighbor of each star s_i adjacent to vertex v, and the direction from vertex v to each vertex in $N(v; G) = \{u_1, \ldots, u_5\}$ or in the set $\{w_1, w_2, \ldots, w_6\}$ is indicated by an arrow. The order of the angles $\theta_v(u)$, $u \in N(v; G) \cup \{w_i \mid i \in I(v)\}$ is determined as follows. $\theta_v(w_6) < \theta_v(w_5) = \theta_v(u_1) < \theta_v(w_1) < \theta_v(u_2) < \theta_v(w_3) = \theta_v(u_3) < \theta_v(w_4) < \theta_v(u_4) < \theta_v(u_5) < \theta_v(w_2)$. Then a modified rotation system at v is given by $\rho^*(v) = (u_1, \{f_{1,2}, \{1\}\}, u_2, \{f_{2,3}, \{3\}\}, u_3, \{f_{3,4}, \{3, 4\}\}, u_4, \{f_{4,5}, \emptyset\}, u_5, \{f_{5,1}, \{2, 5, 6\}\})$.

When a sequence $(a_1, a_2, \ldots, a_p)$ is lexicographically smaller than a sequence $(b_1, b_2, \ldots, b_p)$, we write $(a_1, a_2, \ldots, a_p) \prec (b_1, b_2, \ldots, b_p)$. Let $\mathrm{GS}(n)$ denote the set of grid points (x, y) with $x, y \in [-n, n]_{\mathbb{Z}}$ and $(x, y) \neq (0, 0)$ in the grid plane. For a grid point (x, y) in the grid plane, let $\theta(x, y) \in [0, 2\pi)$ denote the angle made by two vectors $(1, 0)$ and (x, y). The next lemma tells that given a set of p points $(x_1, y_1), \ldots, (x_p, y_p) \in \mathrm{GS}(n-1)$, sorting these p points in a non-decreasing order of angles $\theta(x_i, y_i)$ can be executed in $O(p)$ time by the radix sort after an $O(n)$-time preprocessing.

Lemma 11 *There is a function* $\mathrm{code} : \mathrm{GS}(n-1) \to [0, 7]_{\mathbb{Z}} \times [-n, n]_{\mathbb{Z}} \times [-n, n]_{\mathbb{Z}}$ *such that*
(i) *Given a point* $(x, y) \in \mathrm{GS}(n-1)$, $\mathrm{code}(x, y)$ *can be computed in* $O(1)$ *time; and*
(ii) *for any two points* $(x, y), (x', y') \in \mathrm{GS}(n-1)$, *it holds that* $\theta(x, y) < \theta(x', y')$ *if and only if* $\mathrm{code}(x, y) \prec \mathrm{code}(x', y')$; *and* $\theta(x, y) = \theta(x', y')$ *if and only if* $\mathrm{code}(x, y) = \mathrm{code}(x', y')$.

Proof. For a point $(x, y) \in \mathrm{GS}(n-1)$, define $\sigma(x, y)$ to be $\lfloor \theta(x, y)/(\pi/4) \rfloor$, where $\sigma(x, y)$ can be obtained as follows:

$\sigma(x, y) = 0$ if $x \geq y \geq 0$; $\sigma(x, y) = 1$ if $y > x > 0$;
$\sigma(x, y) = 2$ if $y > -x \geq 0$; $\sigma(x, y) = 3$ if $-x \geq -y > 0$;
$\sigma(x, y) = 4$ if $-x > y \geq 0$; $\sigma(x, y) = 5$ if $-y \geq -x > 0$;
$\sigma(x, y) = 6$ if $-y > x \geq 0$; $\sigma(x, y) = 7$ otherwise, i.e., $x \geq -y > 0$.

Note that

$$\text{if } \sigma(x, y) < \sigma(x', y') \text{ then } \theta(x, y) < \theta(x', y'). \tag{5}$$

Let $(x, y), (x', y') \in \mathrm{GS}(n-1)$ be points with $k = \sigma(x, y) = \sigma(x', y')$. For $k \in \{0, 4\}$, where $|y/x|, |y'/x'| \in [0, 1]$,

$$\theta(x, y) < \theta(x', y') \text{ if and only if } |y/x| < |y'/x'|; \tag{6}$$

For $k \in \{1, 5\}$, where $|x/y|, |x'/y'| \in [0, 1]$,

$$\theta(x, y) < \theta(x', y') \text{ if and only if } -(x/y) < -(x'/y'); \tag{7}$$

For $k \in \{2, 6\}$, where $|x/y|, |x'/y'| \in [0, 1]$,

$$\theta(x, y) < \theta(x', y') \text{ if and only if } x/y < x'/y'; \tag{8}$$

For $k \in \{3, 7\}$, where $|y/x|, |y'/x'| \in [0, 1]$,

$$\theta(x, y) < \theta(x', y') \text{ if and only if } -(y/x) < -(y'/x'). \tag{9}$$

For two integers $a \in \{1, 2, \ldots, n-1\}$ and $b \in \{0, 1, \ldots, n-1\}$, we approximate b/a with an n-adic number $\delta_1\delta_2$ with two digits defined to be

$\delta_1(a, b) \triangleq \lfloor nb/a \rfloor$, and

$\delta_2(a, b) \triangleq \lfloor n^2 b/a - n\delta_1(a, b) \rfloor = \lfloor n(nb/a - \delta_1(a, b)) \rfloor$.

Note that $|\delta_1(a, b)| \in [0, n]_{\mathbb{Z}}$ if $|b/a| \leq 1$. It holds that $|\delta_2(a, b)| \in [0, n]_{\mathbb{Z}}$ since $|nb/a - \delta_1(a, b)| \leq 1$. Then we define $\mathrm{code} : \mathrm{GS}(n-1) \to [0, 7]_{\mathbb{Z}} \times [-n, n]_{\mathbb{Z}} \times [-n, n]_{\mathbb{Z}}$ to be:

$$\begin{array}{ll}
\mathrm{code}(x, y) \triangleq (\sigma(x, y), \delta_1(|x|, |y|), \delta_2(|x|, |y|)) & \text{if } \sigma(x, y) \in \{0, 4\}; \\
\mathrm{code}(x, y) \triangleq (\sigma(x, y), -\delta_1(|y|, |x|), -\delta_2(|y|, |x|)) & \text{if } \sigma(x, y) \in \{1, 5\}; \\
\mathrm{code}(x, y) \triangleq (\sigma(x, y), \delta_1(|y|, |x|), \delta_2(|y|, |x|)) & \text{if } \sigma(x, y) \in \{2, 6\}; \text{ and} \\
\mathrm{code}(x, y) \triangleq (\sigma(x, y), -\delta_1(|x|, |y|), -\delta_2(|x|, |y|)) & \text{if } \sigma(x, y) \in \{3, 7\}.
\end{array}$$

We see that given a point $(x, y) \in \mathrm{GS}(n-1)$, $\mathrm{code}(x, y)$ can be computed in $O(1)$ time. To prove the lemma, it suffices to show the next.

Claim 1. *For two points $(x, y), (x', y') \in \mathrm{GS}(n-1)$, it holds that*
(a) $\theta(x, y) < \theta(x', y')$ *if and only if* $\mathrm{code}(x, y) \prec \mathrm{code}(x', y')$*;*
(b) $\theta(x, y) = \theta(x', y')$ *if and only if* $\mathrm{code}(x, y) = \mathrm{code}(x', y')$.

PROOF. Note that (a) implies (b) because one of $\mathrm{code}(x, y) \prec \mathrm{code}(x', y')$, $\mathrm{code}(x', y') \prec \mathrm{code}(x, y)$ and $\mathrm{code}(x, y) = \mathrm{code}(x', y')$ always holds. We show (a).

The claim (a) is clear when $\sigma(x, y) \neq \sigma(x', y')$ by definition of function σ. Assume that $\sigma(x, y) = \sigma(x', y')$. We show the case of $\sigma(x, y) = \sigma(x', y') = 0$ (the other case can be treated analogously). Note that $0 < y/x, y'/x' \leq 1$. Recall that $\theta(x, y) < \theta(x', y')$ if and only if $y/x < y'/x'$. Hence we see that $\mathrm{code}(x, y) \prec \mathrm{code}(x', y')$ implies $y/x < y'/x'$, i.e., $\theta(x, y) < \theta(x', y')$. To show the converse, assume that $y/x < y'/x'$. If $\delta_1(x, y) < \delta_1(x', y')$ then $\mathrm{code}(x, y) \prec \mathrm{code}(x', y')$. Let us assume that $\delta_1(x, y) = \delta_1(x', y')$. We show that the difference between $n(ny'/x' - \delta_1(x, y))$ and $n(ny/x - \delta_1(x, y))$ is greater than 1. In fact, we have $n(ny'/x' - \delta_1(x, y)) - n(ny/x - \delta_1(x, y)) = n^2(y'/x' - y/x) = n^2(y'x - x'y)/(xx') \geq n^2/(n-1)^2 > 1$. This implies that $\delta_2(x, y) = \lfloor n(ny/x - \delta_1(x, y)) \rfloor < \lfloor n(ny'/x' - \delta_1(x, y)) \rfloor = \delta_2(x', y')$, as required. □

We use the lemma with $\mathrm{GS}(2n-1)$ in the proof of Theorem 10.

6 Case of Connectivity 2

Sections 6, 7 and 8 handle the last case where the vertex-connectivity of a red graph in a star instance 2. Our algorithm for this case consists of two major phases, which are presented in the next two sections, respectively. This section first reviews a method of decomposing a red biconnected graph G of an instance (H, G, A) into triconnected components, and observes some structure of triconnected components of G which indicates the existence a forbidden graph.

6.1 SPR-tree Decomposition

To consider all the possible planar embeddings of a biconnected planar graph, we use a decomposition of a biconnected graph into *triconnected components*, defined by Hopcroft and Tarjan (for details, see [27]), which can be computed in linear time [23, 27]. More specifically, we use the SPR-tree, a simplified version of the SPQR-tree defined by Di Battista and Tamassia (for details, see [5]), without Q-nodes. Here we give a brief description on the definition of the SPR-tree using the terminology from [27].

Let $G = (V, E)$ be a biconnected graph. Let $\mathcal{V}$ be the set of triconnected components ν of G, where each triconnected component ν is represented by a multigraph, called the *skeleton* $\mathrm{skl}(\nu) = (V(\nu), E(\nu))$ of ν with the following property:

- $V(\nu) \subseteq V$; $E(\nu)$ consists of some edges in E, called *real edges* and *virtual edges* such that each virtual edge $e = uv$ is associated with a connected subgraph G_e of G with the following property:

  $G_e - \{u, v\}$ remains connected;

  for any two distinct virtual edges $e = uv, e' = u'v' \in E(\nu)$,
  $E(G_e) \cap E(G_{e'}) = \emptyset$ and $V(G_e) \cap V(G_{e'}) = \{u, v\} \cap \{u', v'\}$; and

- $V = \cup_{\nu \in \mathcal{V}} V(\nu)$ and $E \subseteq \cup_{\nu \in \mathcal{V}} E(\nu)$ (i.e., each edge in E is contained in $E(\nu)$ as a real edge for some node $\nu \in \mathcal{V}$).

There are three types of graph structure of skeletons $\mathrm{skl}(\nu)$:

1. S-type: $\mathrm{skl}(\nu)$ is a simple cycle with at least three vertices;

2. P-type: $\mathrm{skl}(\nu)$ consists of two vertices joined by at least three edges; and

3. R-type: $\mathrm{skl}(\nu)$ is a simple triconnected graph with at least four vertices.

For a virtual edge $e = uv \in E(\nu)$, a u, v-path in the associated graph G_e is called a *representing path* of e. A graph obtained from the skeleton $\mathrm{skl}(\nu)$ by replacing each virtual edge e with a representing path P_e is called a *representing graph* of the skeleton. As a graph structure, a representing graph of a skeleton is obtained by subdividing virtual edges in the skeleton. From the definition of skeletons, we observe the next.

Lemma 12 *For a triconnected component $\nu \in \mathcal{V}$ for a biconnected graph $G = (V, E)$, let $e = uv \in E(\nu)$ be a virtual edge in the skeleton $\mathrm{skl}(\nu)$. Any cut-vertex in the graph G_e separates u and v. For a representing u, v-path P_e of e and a vertex $w \in V(G_e) - \{u, v\}$, there is a $w, (V(P_e) - \{u, v\})$-path, which can be found in $O(|V(G_e)| + |E(G_e)|)$ time.*

The *SPR-tree* $\mathcal{T}$ of G is a tree constructed on the set $\mathcal{V}$ of triconnected components of G that represents the adjacency among triconnected components. We call a triconnected component $\nu \in \mathcal{V}$ as a node in $\mathcal{T}$. A node ν with an S-type (resp., P-type and R-type) skeleton $\mathrm{skl}(\nu)$ is called an S-node (resp., P-node and R-node). Then the SPR-tree is a tree $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ with an edge set $\mathcal{E}$ such that
(i) $\nu\mu \in \mathcal{E}$ only if the skeletons $\mathrm{skl}(\nu) = (V(\nu), E(\nu))$ and $\mathrm{skl}(\mu) = (V(\mu), E(\mu))$ have exactly two common vertices;
(ii) two nodes which are both S-nodes or P-nodes are not adjacent in $\mathcal{T}$; and
(iii) any node of degree 1 in $\mathcal{T}$ is an S- or R-node.
It is known that $\sum_{\nu \in \mathcal{V}} (|V(\nu)| + |E(\nu)|) = O(|V| + |E|)$ [5].

6.2 Rooted SPR-trees

In the SPR-tree $\mathcal{T}$ of a biconnected graph G, we choose a node $\nu_{\rm root}$ and regard $\mathcal{T}$ as a tree rooted at $\nu_{\rm root}$. The rooted tree $\mathcal{T}$ defines a parent-child order, where ${\rm Ch}(\nu)$ denotes the set of all children of a node ν and ${\rm pt}(\nu)$ denotes the parent of a non-root ν. Let ν be a non-root node in $\mathcal{T}$, and $\eta = {\rm pt}(\nu)$ be the parent of ν, where the skeleton ${\rm skl}(\nu)$ contains a virtual edge $e = st \in E(\nu)$ such that the skeleton ${\rm skl}(\eta)$ of the parent also contains a virtual edge $e' = st \in E(\eta)$. Such a virtual edge $e = st \in E(\nu)$ is called the *parent virtual edge* in ${\rm skl}(\nu)$, and is denoted by ${\rm pe}(\nu)$. Denote $E^-(\nu) = E(\nu) - \{{\rm pe}(\nu)\}$, and call a virtual edge in $E^-(\nu)$ a *child virtual edge* in ${\rm skl}(\eta)$. Let ${\rm skl}^-(\nu)$ denote the graph $(V(\nu), E^-(\nu))$ obtained from ${\rm skl}(\nu)$ by deleting its parent virtual edge.

For the root $\nu_{\rm root}$, let ${\rm skl}^-(\nu_{\rm root}) = {\rm skl}(\nu_{\rm root})$.

6.3 Some Forbidden Configurations over Skeletons

Given an instance (H, G, A) with a red biconnected graph $G = (V, E)$, let $\mathcal{V}$ be the set of nodes (i.e., triconnected components) of G. Note that the skeleton ${\rm skl}(\nu)$ of a P- or S-node $\nu \in \mathcal{V}$ is a planar graph. In particular, ${\rm skl}(\nu)$ of an S-node ν has a planar embedding γ_ν, which is a unique combinatorial embedding. The skeleton ${\rm skl}(\nu)$ of a P-node ν with p edges admits $p!$ possible planar embeddings.

Since G is planar, the triconnected skeleton ${\rm skl}(\nu) = (V(\nu), E(\nu))$ of each R-node $\nu \in \mathcal{V}$ admits a planar embedding γ_ν, which is unique combinatorial embedding, but $O(|V(\nu)|)$ planare embedding depending on the choice of the outer face. We assume that planar embeddings γ_ν for all R-nodes $\nu \in \mathcal{V}$ have been computed in $O(\sum_{\nu \in \mathcal{V}}(|V(\nu)| + |E(\nu)|)) = O(|V| + |E|) = O(n)$ time. Let $\Phi(\gamma_\nu)$ denote the set of faces in the planar embedding γ_ν of an S- or R-node ν.

In this subsection, we show how a forbidden subgraph of types (i), (ii) and (iv) in H may appear in the skeletons of P- and R-nodes together with stars in A.

Let ${\rm skl}(\nu) = (V(\nu), E(\nu))$ be the skeleton of a node $\nu \in \mathcal{V}$. We say that a star $s_i \in A$ *touches* an element in $z \in V(\nu) \cup E(\nu)$ if z is a vertex $v \in S_i \cap V(\nu)$ or z is a virtual edge $e = uv \in E(\nu)$ such that the associated graph G_e contains a vertex $w_e \in S_i - \{u, v\}$.

With a single star $s_i \in A$, the following conditions on P- or R-node tells us the existence of forbidden subgraph of type (i) or (ii) in H.

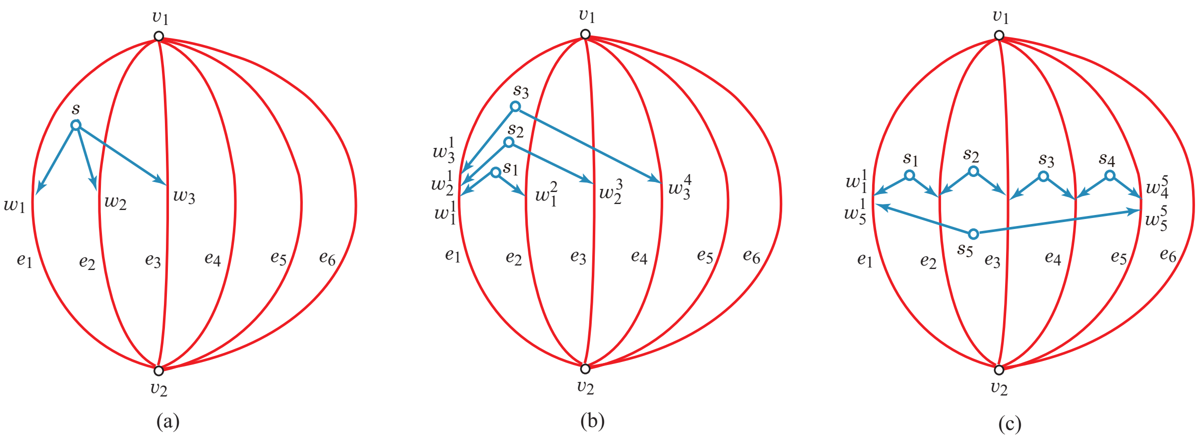


Figure 7: Illustration of forbidden configurations in the skeleton ${\rm skl}(\nu)$ of a P-node $\nu \in \mathcal{V}$ in the SPR-tree of the red graph G: (a) A star s that touches three virtual edges in $E(\nu)$; (b) Three stars s_i, $i = 1, 2, 3$ such that s_i touches virtual edges $e_1, e_{i+1} \in E(\nu)$; (a) A set of $p = 5$ star s_i, $i = 1, 2, \ldots, p$ such that s_i touches virtual edges $e_i, e_{i+1} \in E(\nu)$, where $e_{p+1} = e_1$.

Lemma 13 (i) *For a P-node $\nu \in \mathcal{V}$, if there is a star $s_i \in A$ that touches three virtual edges in $E(\nu)$, as shown in Fig. 7(a), then the graph $G + \delta(s_i)$ contains a forbidden subgraph F of type (i), which can be found in $O(n + m)$ time.*

(ii) *Let $\nu \in \mathcal{V}$ be an R-node and γ_ν be a planar embedding of the skeleton $\mathrm{skl}(\nu)$. For a star $s_i \in A$, let $V_1 = S_i \cap V(\nu)$ and E_1 be the set of all virtual edges in $E(\nu)$ touched by s_i. If γ_ν has no facial cycle that contains V_1 and E_1, then the graph $G + \delta(s_i)$ contains a forbidden subgraph F of type (i) or (ii), which can be found in $O(n + m)$ time.*

Proof. (i) Let $V(\nu) = \{u, v\}$, and let $e_j \in E(\nu)$, $j = 1, 2, 3$ be three virtual edges touched by a star s_i, where the associated graph G_{e_j} contains a vertex $w_j \in S_i - \{u, v\}$. Choose a representing u, v-path P_j of each virtual edge e_j. Then we can find a $w_j, (V(P_j) - \{u, v\})$-path P_{w_j,z_j} with an end-vertex $z_j \in V(P_j) - \{u, v\}$ in time linear to the size of G_{e_j} by Lemma 12, where possibly $w_j = z_j$. Then we see that the six red paths P_j and P_{w_j,z_j}, $j = 1, 2, 3$ and three blue edges $s_i w_j$, $j = 1, 2, 3$ form a forbidden subgraph F of type (i) in H. The above construction can be executed in $O(n + m)$ time.

(ii) Let $E_1 = \{e_j = u_j v_j \mid j = 1, 2, \ldots, q\}$, where the associated graph G_{e_j} contains a vertex $w_j \in S_i - \{u_j, v_j\}$, and skl^* be the pseudo-triconnected graph obtained from $\mathrm{skl}(\nu)$ by subdividing each edge $e_j = u_j v_j \in E_1$ into two edges $u_j w_j$ and $w_j v_j$. By the fact that a planar embedding of a pseudo-triconnected graph is a unique combinatorial embedding, we see that the graph $\mathrm{skl}^* + \delta(s_i)$ is not planar, and hence we see that $G+\delta(s_i)$ is not planar. By Kuratowski's theorem [32], the non-planar graph $G+\delta(s_i)$ contains a subgraph F which is a subdivision of $K_{3,3}$ or K_5, which can be found in $O(n + |S_i|) = O(n)$ time [23]. Since G is assumed to be planar, the subgraph F must contain some blue edge in $\{s_i t \mid t \in S_i\}$. Since all blue edges in $\{s_i t \mid t \in S_i\}$ are incident to s_i, the number ℓ of blue edges contained in F is $\ell \in \{2, 3, 4\}$. When $\ell = 2$, the two blue edges contained in a path between some two vertices of degree 3 or 4 in F. When $\ell = 3$ (resp., $\ell = 4$), graph F is a subdivision F of $K_{3,3}$ (resp., K_5) which has a vertex to which only blue edges are incident. In any case, F is a forbidden subgraph F of type (i) or (ii) in H. □

With a set of several stars in A, the following conditions on P- or R-node tells us the existence of forbidden subgraph of type (i) or (iv) in H.

Lemma 14 *Let $\nu \in \mathcal{V}$ be a P-node.*

(i) *If there are three stars $s_1, s_2, s_3 \in A$ and four virtual edges $e_1, e_2, e_3, e_4 \in E(\nu)$ such that s_j touches $\{e_1, e_{j+1}\}$ for each j, as shown in Fig. 7(b), then the graph $G + \delta(s_1) + \delta(s_2) + \delta(s_3)$ contains a forbidden subgraph F of type (i), which can be found in $O(n + m)$ time.*

(ii) *If there are p stars $s_1, \ldots, s_p \in A$ and p virtual edges $e_1, \ldots, s_p \in E(\nu)$ such that $3 \leq p < |E(\nu)|$ and s_j touches $\{e_j, e_{j+1}\}$ for each j, where e_{p+1} means e_1, as shown in Fig. 7(c), then the graph $G+\delta(s_1)+\cdots+\delta(s_p)$ contains a forbidden subgraph F of type (iv), which can be found in $O(n+m)$ time.*

Proof. Let $V(\nu) = \{u, v\}$ and let w_j^i denote a vertex in $(V(G_{e_j})-V(e_j))\cap S_i$. Choose a representing u, v-path P_j in G_{e_j}. Then we can find a $w_j^i, (V(P_j)-\{u, v\})$-path $P_{w_j^i,z_j^i}$ with an end vertex $z_j^i \in V(P_j)-\{u, v\}$ in time linear to the size of G_{e_j} by Lemma 12, where possibly $w_j^i = z_j^i$.

(i) Without loss of generality assume that vertices z_1^1, z_2^1, z_3^1 appear in this order along path P_1 from u to v, where possibly some of these three vertices may be identical. For each $j = 1$ or 3, let $P_{w_j^1,z_2^1}$ be the w_j^1, z_2^1-path that consists of $P_{w_j^1,z_j^1}$ and the subpath of P_1 from z_1^j to z_1^2. Then we see that the six paths P_j, $j = 2, 3, 4$ and $P_{w_2^1,z_2^1}$ $P_{w_1^1,z_2^1}$ and $P_{w_3^1,z_2^1}$ and six blue edges $s_j w_1^j$, $s_j w_{j+1}^j$, $j = 1, 2, 3$ form a forbidden subgraph F of type (i). The above construction can be executed in $O(n + m)$ time.

(ii) Let $e_0 \in E(\nu) - \{e_j \mid j = 1, 2, \ldots, p\}$, and let P_0 be a representing u, v-path in G_{e_0}. For each $j = 1, 2, \ldots, p$, where $p+1$ means 1, let P'_j be the z_j^j, z_j^{j+1}-path that consists of blue edges s_j, w_j^j and s_j, w_j^{j+1} and red w_j^j, z_j^j-path and w_j^{j+1}, z_j^{j+1}-path. Let P''_3 be a z_2^3, z_p^1-path that consists of P'_j, $j = 3, 4, \ldots, p$ and the subpath of P_j, $j = 3, 4, \ldots, p$ from z_j^{j-1} to z_j^j. Then we see that the four paths P_j, $j = 0, 1, 2, 3$ and the three paths P'_1, P'_2 and P''_3 form a forbidden subgraph F of type (iv). The above construction can be executed in $O(n + m)$ time. □

An Overview of Algorithm We design an algorithm that finds a forbidden graph or a planarizing partition of a star set for a given instance (H, E) with a red biconnected graph. The algorithm consists of two phases. The first phase tests in $O(n + m)$ time whether some condition in Lemmas 13 and 14 holds. To facilitate test of condition in Lemma 13(i) for a non-root R-node ν and a star $s_i \in A$ touching the parent-edge $\mathrm{pe}(\nu)$, we modify the rooted SPR-tree $\mathcal{T}$ by splitting each R-node into four types of nodes in the next section.

Unfortunately, an instance to which none of conditions in Lemmas 13 and 14 holds still may contain a forbidden graph (whose type is (i), (iv) or (v)). This is because each virtual edge $e \in E(\eta)$ of a node η corresponds to the skeleton $\mathrm{skl}(\nu_e)$ of the corresponding node ν_e, which has two possible embeddings in an embedding of η, and there may be no combination of embeddings of skeletons $\mathrm{skl}(\nu_e)$ over all child virtual edges $\nu_e \in E^-(\eta)$ so that all stars in A can be drawn in some face without creating a crossing with a red edge. The second phase examines whether there is a combination of embeddings of skeletons $\mathrm{skl}(\nu)$ over all child virtual edges $\nu \in E^-(\eta)$ for each P- or R-node such that all stars in A can be drawn without creating a crossing with a red edge. To facilitate the examination, we introduce a "simplified structure" of the skeletons $\mathrm{skl}(\nu)$ of each P- or R-node ν, and combine the "simplified skeletons" into the skeleton of the parent node η of ν (or the parent η of parent of ν if the parent of ν is an S-node). This results in a skeleton of η where each virtual edge $e \in E^-(\eta)$ is replaced with a simplified skeleton (or a chain of simplified skeletons if e corresponds to an S-node), which is called "refined skeletons" in Section 8.

In the next two sections, we do not find a forbidden graph of type (iii). In our algorithmic proof for Theorem 4, a forbidden graph of type (iii) is generated only when we construct a forbidden graph F' of type (iii) of an instance (H, G, A) with $|\mathcal{B}(G)| \geq 2$ from a forbidden graph F of type (ii) in an instance (B, A_B) with $B \in \mathcal{B}(G)$ (see the case (c) in the proof of Lemma 7(ii)).

7 Phase 1 for Case of Connectivity 2

This section describes the first phase which tests in $O(n + m)$ time whether some condition in Lemmas 13 and 14 holds.

7.1 Split SPR-tree $\widehat{\mathcal{T}}$

Since G is not outerplanar, the SPR-tree $\mathcal{T}$ has a P- or R-node. We choose a P- or R-node as the root node ν_{root} of $\mathcal{T}$.

If the root of $\mathcal{T}$ is an R-node ν, then let γ_ν denote a planar embedding of the simple triconnected graph $\mathrm{skl}(\nu)$. Let ν be a non-root R-node. Define γ_ν to be a planar embedding of the simple triconnected graph $\mathrm{skl}(\nu)$ such that the parent edge of ν appears as an outer edge of γ_ν, and define γ_ν^- to be the planar embedding of graph $\mathrm{skl}^-(\nu)$ obtained from γ_ν by deleting the parent edge of ν. Since a planar embedding of a planar graph can be constructed in linear time and the total size of all skeletons in $\mathcal{T}$ is $O(n)$, we can obtain γ_ν and γ_ν^- for all R-nodes in $O(n)$ time.

We modify the rooted SPR-tree $\mathcal{T}$ by splitting each non-root R-node ν into four nodes, the cp-R-node ν^{cp}, the o1-R-node ν^{o1}, the o2-R-node ν^{o2} and the in-R-node ν^{in} of ν (called *split* R-node of ν) as follows. Let uv be the parent-edge of ν, and let P_1 and P_2 denote the two internally disjoint u, v-paths which form the outer boundary of $\mathrm{skl}^-(\nu)$. Then partition the vertex set $V(\nu)$ and the edge set $E^-(\nu)$ of the skeleton

$\mathrm{skl}^-(\nu)$ into the vertex and edge sets of the four split nodes of ν as follows.

$V(\nu^{\mathrm{cp}}) \triangleq \{u, v\}, \quad E(\nu^{\mathrm{cp}}) \triangleq \emptyset;$

$V(\nu^{\mathrm{o}j}) \triangleq V(P_j) - \{u, v\}, \quad E(\nu^{\mathrm{o}j}) \triangleq E(P_j)$ for $j = 1, 2;$

$V(\nu^{\mathrm{in}}) \triangleq V(\nu) - \{u, v\} - V(P_1) - V(P_2), \quad E(\nu^{\mathrm{in}}) \triangleq E(\nu) - E(P_1) - E(P_2).$

See Fig. 11(c) for an illustration of the four nodes, cp-R-, o1-R-, o2-R- and in-R-nodes of an R-node η. The virtual edges in $E(\nu^{\mathrm{in}})$ and the vertices in $V(\nu^{\mathrm{in}})$ are called *inner*, whereas those in $E(\nu^{\mathrm{o1}})$, $E(\nu^{\mathrm{o2}})$, $V(\nu^{\mathrm{o1}})$ and $V(\nu^{\mathrm{o2}})$ are called *outer*.

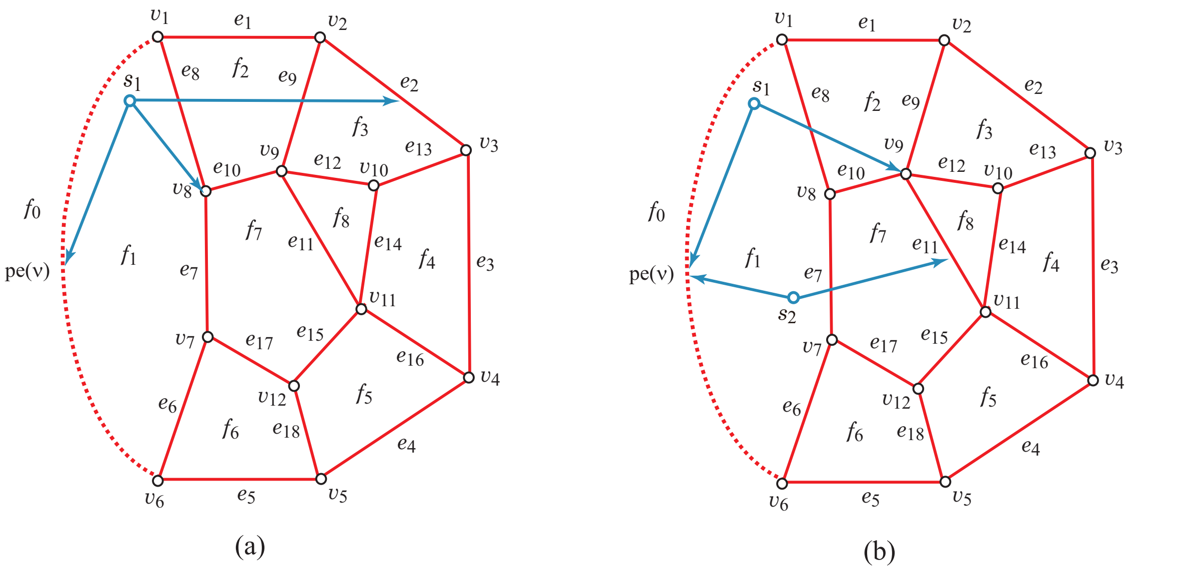


Figure 8: Illustration of forbidden configurations in the skeleton $\mathrm{skl}(\nu)$ of a non-root R-node $\nu \in \mathcal{V}$: (a) A star s_1 which touches the parent-virtual edge $\mathrm{pe}(\nu)$, an element $v_8 \in V(\nu^{\mathrm{o1}}) \cup E(\nu^{\mathrm{o1}})$, and an element $e_2 \in V(\nu^{\mathrm{o2}}) \cup E(\nu^{\mathrm{o2}})$; (b) A star s_1 which touches the parent-virtual edge $\mathrm{pe}(\nu)$ and a vertex $v_9 \in V(\nu^{\mathrm{in}})$, and a star s_2 which touches $\mathrm{pe}(\nu)$ and a virtual edge $e_{11} \in E(\nu^{\mathrm{in}})$.

Replace each non-root R-node ν with ν^{cp}, letting ν^{o1}, ν^{o2} and ν^{in} be the three children of ν^{cp}. Then each child $\mu \in \mathrm{Ch}(\nu)$ of ν will be a child of the node $\nu' \in \{\nu^{\mathrm{o1}}, \nu^{\mathrm{o2}}, \nu^{\mathrm{in}}\}$ such that the parent-edge of μ belongs to $E(\nu')$. Let $\widehat{\mathcal{T}} = (\widehat{\mathcal{V}}, \widehat{\mathcal{E}})$ be the resulting rooted tree obtained from $\mathcal{T}$ by applying the above procedure to all non-root R-nodes ν. We call $\widehat{\mathcal{T}}$ the *split* SPR-tree of G. We regard $\widehat{\mathcal{T}}$ as an ordered tree by introducing an arbitrary sibling order for each child set. In what follows, we define $\mathrm{ld}(\nu)$, $\mathrm{rd}(\nu)$, $\mathrm{dt}(\nu)$, $\mathrm{lp}(\nu)$ and $\mathrm{rp}(\nu)$ for each node $\nu \in \widehat{\mathcal{V}}$ in the ordered tree $\widehat{\mathcal{T}}$.

7.2 Mapping ψ from V to $\widehat{\mathcal{V}}$

For each star $s_i \in A$, any neighbor $t \in S_i$ is a vertex in the vertex set of skeleton of a node ν or the associated graph G_e for some virtual edge of a node ν. Such a node ν may not be unique in general. We here determine uniquely a mapping from a vertex $t \in V$ to a node ν. Let us define a mapping $\psi : V \to \widehat{\mathcal{V}}$ that maps each vertex $v \in V$ to the highest node $\nu \in \widehat{\mathcal{V}}$ with $v \in V(\nu)$, where we see by the construction of $\widehat{\mathcal{T}}$ that such a node $\psi(v)$ is uniquely determined. For each node $\nu \in \widehat{\mathcal{V}}$, let $\psi^{-1}(\nu)$ be the set of vertices $v \in V$ such that $\psi(v) = \nu$. We can construct the mappings ψ and ψ^{-1} in $O(n)$ time by visiting each node ν in the ordered tree $\widehat{\mathcal{V}}$ in a breadth-first search manner and checking for each vertex $v \in V(\nu)$ whether v is scanned for the first time, where if v is scanned for the first time then $\psi(v) = \nu$, and $\psi^{-1}(\nu)$ is the set of such vertices $v \in V(\nu)$.

For each star $s_i \in A$, let $\mathcal{N}_i \triangleq \{\psi(v) \mid v \in S_i\}$. For each $i = 1, 2, \ldots, k$, we construct the mimic tree $\widehat{\mathcal{T}}\langle \mathcal{N}_i \rangle$ obtained from the ordered split SPR-tree $\widehat{\mathcal{T}}$ induced by $\mathcal{N}_i$. For simplicity, we denote $\widehat{\mathcal{T}}\langle \mathcal{N}_i \rangle$ by $\mathcal{T}_i$.

All $\mathcal{T}_i$, $i = 1, 2, \ldots, k$ can be obtained in $O(n + \sum_{1 \le i \le k} |\mathcal{N}_i|) = O(n + m)$ time by Lemma 2, where we see that $\psi_i^{-1}(\nu) = \{t \in S_i \mid \psi(t) = \nu\}$, $\nu \in \mathcal{N}_i^*$ can be computed for all $i = 1, 2, \ldots, k$ in $O(n + m)$ time.

Lemma 15 *For a star $s_i \in A$, let ν be a non-root P-node* (*resp., a non-root R-node) in $\mathcal{V}$ such that $\mathcal{N}_i$ contains* (*resp., one of $\nu^{\rm cp}$, $\nu^{\rm o1}$, $\nu^{\rm o2}$ and $\nu^{\rm in}$*)*. Then it takes $O(1)$ time to test whether star s_i touches the parent edge* $\mathrm{pe}(\nu)$.

Proof. Let $\{u, v\}$ be $V(\nu)$ (resp., $V(\nu^{\rm cp})$) if ν is a P-node (resp., ν is an R-node). Then if s_i touches the parent edge $\mathrm{pe}(\nu)$, then ν (resp., $\nu^{\rm cp}$) has the node $\alpha(s_i)$ as its ancestor in $\mathcal{T}_i$, and "$|\mathrm{Ch}(\alpha(s_i); \mathcal{T}_i)| \ge 2$" or "$|\mathrm{Ch}(\alpha(s_i); \mathcal{T}_i)| = 1$ and $\psi_i^{-1}(\nu) - \{u, v\} \ne \emptyset$." Conversely if ν (resp., $\nu^{\rm cp}$) has the node $\alpha(s_i)$ as such an ancestor, s_i touches the parent edge $\mathrm{pe}(\nu)$. Testing whether ν (resp., $\nu^{\rm cp}$) admits $\alpha(s_i)$ as such an ancestor can be checked in $O(1)$ time. □

For a fixed star $s_i \in A$, if a node ν is not in $\mathcal{T}_i$, then the lemma does not tell that we can test in $O(1)$ time whether s_i touches the parent edge $\mathrm{pe}(\nu)$ (since the node ν may not appear in $\mathcal{T}_i$).

We now prepare a data structure so that we can test in $O(1)$ time whether a given node ν admits some star $s_i \in A$ that touches the parent edge $\mathrm{pe}(\nu)$ and an element in a descendant in $D(\nu; \widehat{\mathcal{T}})$. For a subset $S \subseteq V$ in G and a function $f \in \{\mathrm{dt}, \mathrm{lp}, \mathrm{rp}\}$ over $\widehat{\mathcal{T}}$, let $\mathrm{argmin}_f(S)$ denote the set of all vertices $u \in S$ with minimum value in f; i.e,

$$\mathrm{argmin}_f(S) \triangleq \{u \in S \mid f(\psi(u)) = \min_{t \in S} f(\psi(t))\}.$$

For each integer $j \ge 0$, we let $\mathrm{a}_f^{(j)}(S)$ denote a set of $\min\{j, |\mathrm{argmin}_f(S)|\}$ vertices arbitrarily chosen from $\mathrm{argmin}_f(S)$. For each node ν in $\widehat{\mathcal{T}}$, we let $I_\nu = \{i \mid \mathcal{N}_i \cap D(\nu; \widehat{\mathcal{T}}) \ne \emptyset\}$, and construct $\mathrm{a}_f^{(j)}(\bigcup_{i \in I_\nu} S_i)$ for each function $f \in \{\mathrm{dt}, \mathrm{lp}, \mathrm{rp}\}$. For simplicity, we denote $\mathrm{a}_f^{(j)}(\cup_{i \in I_\nu} S_i)$ by $\mathrm{a}_f^{(j)}\langle\nu\rangle$, and $\mathrm{a}_f^{(1)}\langle\nu\rangle$ by $\mathrm{a}_f\langle\nu\rangle$, where $\mathrm{a}_f^{(j)}\langle\nu\rangle = \emptyset$ if $I_\nu = \emptyset$.

Lemma 16 *The set $\{\mathrm{a}_f^{(3)}\langle\nu\rangle \mid \nu \in \widehat{\mathcal{V}}\}$ for each function $f \in \{\mathrm{dt}, \mathrm{lp}, \mathrm{rp}\}$ can be constructed in $O(n + m)$ time.*

Proof. First for all $i = 1, 2, \ldots, k$, compute $\mathrm{a}_f^{(3)}(S_i)$ in $O(\sum_{1 \le i \le k} |S_i|) = O(m)$ time. Next for each vertex $\nu \in \widehat{\mathcal{V}}$, compute the set $I'_\nu = \{i \mid \nu \in \mathcal{N}_i\}$ of indices from $\psi^{-1}(\nu)$. Finally compute $\mathrm{a}_f^{(3)}\langle\nu\rangle$ for all vertices $\nu \in \widehat{\mathcal{V}}$ in a bottom-up manner along $\widehat{\mathcal{T}}$. (i) For each leaf ν in $\widehat{\mathcal{T}}$, compute $\mathrm{a}_f^{(3)}\langle\nu\rangle$ by choosing at most three vertices $u \in \bigcup_{i \in I'_\nu} \mathrm{a}_f^{(3)}(S_i)$ with the minimum $f(\psi(u))$, which takes $O(|I'_\nu|)$ time. (ii) For each non-leaf vertex v such that $\mathrm{a}_f^{(3)}\langle\mu\rangle$ has been computed for all children $\mu \in \mathrm{Ch}(\nu; \widehat{\mathcal{V}})$, we compute $\mathrm{a}_f^{(3)}\langle\nu\rangle$ by choosing at most three vertices $u \in (\bigcup_{i \in I'_\nu} \mathrm{a}_f^{(3)}(S_i)) \cup \{\mathrm{a}_f^{(3)}\langle\mu\rangle \mid \mu \in \mathrm{Ch}(\nu; \widehat{\mathcal{V}})\}$ with the minimum $f(\psi(u))$ value, which takes $O(|I'_\nu| + |\mathrm{Ch}(\nu; \widehat{\mathcal{V}})|)$ time.

The total time to compute $\mathrm{a}_f^{(3)}\langle\nu\rangle$ for all vertices $\nu \in \widehat{\mathcal{V}}$ is $O(n + \sum_{\nu \in \widehat{\mathcal{V}}}(|I'_\nu| + |\mathrm{Ch}(\nu; \widehat{\mathcal{V}})|)) = O(n + \sum_{1 \le i \le k} |S_i|)$. □

Lemma 17 *Let ν be a non-root node in $\mathcal{V}$, and μ be a P- or S-node in $\widehat{\mathcal{V}}$ such that $\mu \in D(\nu; \mathcal{T})$ or $\mu \in \{\tau^{\rm cp}, \tau^{\rm o1}, \tau^{\rm o2}, \tau^{\rm in}\}$ for some R-node $\tau \in D(\nu; \mathcal{T})$. With data $\{\mathrm{a}_{\rm lp}\langle\nu\rangle, \mathrm{a}_{\rm rp}\langle\nu\rangle, \mathrm{a}_{\rm dt}^{(3)}\langle\nu\rangle \mid$ all nodes ν in $\widehat{\mathcal{V}}\}$, it takes $O(1)$ time to test whether there is a star $s \in A$ that touches the parent edge $\mathrm{pe}(\nu)$ and a vertex z with $\psi(z) \in D(\mu; \widehat{\mathcal{T}})$ and to find one of such stars if one exists.*

Proof. Let $\mathrm{pe}(\nu) = uv$, where $V(\mathrm{pe}(\nu)) = \{u, v\}$. By definition of I_μ, a star s_i touches a vertex z with $\psi(z) \in D(\mu; \widehat{\mathcal{T}})$ if and ony if $i \in I_\mu$.

Some star s_i with $i \in I_\mu$ touches the parent edge $\mathrm{pe}(\nu) - uv$ if and only if a vertex $w \in \cup_{i \in I_\mu} S_i - \{u, v\}$ is mapped to a node $\psi(w)$ in $\widehat{\mathcal{V}} - D(\nu; \widehat{\mathcal{T}})$; i.e., $\bigcup_{i \in I_\mu} \mathcal{N}_i - D(\nu; \widehat{\mathcal{T}})$ contains a node $\eta = \psi(w)$ for some vertex $w \in \cup_{i \in I_\mu} S_i - \{u, v\}$.

To see when $\cup_{i \in I_\mu} \mathcal{N}_i - D(\nu; \widehat{\mathcal{T}})$ contains such a node η, we apply Lemma 1 to $S = \cup_{i \in I_\mu} \mathcal{N}_i - D(\nu; \widehat{\mathcal{T}})$, $v = \nu$ and $T = \widehat{\mathcal{T}}$. By Lemma 1(i), if the set $\cup_{i \in I_\mu} \mathcal{N}_i - D(\nu; \widehat{\mathcal{T}})$ contains a node η on the left side of ν in $\widehat{\mathcal{T}}$, then one of such nodes η is given by the node $\psi(w)$ of the vertex $w \in \mathrm{a_{lp}}\langle \mu \rangle$ $\mathrm{ld}(\psi(w)) < \mathrm{ld}(\nu)$ such that $\mathrm{lp}(\psi(w)) < \mathrm{lp}(\nu)$, where if the vertex $w \in \mathrm{a_{lp}}\langle \mu \rangle$ does not satisfies these inequalities then there is no such node η. Symmetrically for a node η on the right side of ν in $\widehat{\mathcal{T}}$ by Lemma 1(ii).

First assume that the node $\eta = \psi(w)$ with a vertex $w \in \mathrm{a_{lp}}\langle \mu \rangle \cup \mathrm{a_{rp}}\langle \mu \rangle$ on the left or right side of ν in $\widehat{\mathcal{T}}$. Then clearly $w \notin \{u, v\}$ because otherwise $w \in \{u, v\}$ would imply that the parent node $\eta' = \mathrm{pt}(\eta)$ of $\eta = \psi(w)$ also contains the vertex w in the skeleton vertex set $V(\eta')$, contradicting that $\psi(w)$ is chosen as the highest node that contains w in its skeleton.

Next assume that the set $\cup_{i \in I_\mu} \mathcal{N}_i - D(\nu; \widehat{\mathcal{T}})$ contains no node η on the left or right side of ν in $\widehat{\mathcal{T}}$. By Lemma 1(iii), if the set $\cup_{i \in I_\mu} \mathcal{N}_i - D(\nu; \widehat{\mathcal{T}})$ contains an ancestor η of ν, then one of such nodes η is given by the node $\psi(w)$ of any vertex $w \in \mathrm{a}^{(3)}_{\mathrm{dt}}\langle \mu \rangle$ such that $\mathrm{dt}(\psi(w)) < \mathrm{dt}(\nu)$. In this case, the set $\mathrm{a}^{(3)}_{\mathrm{dt}}\langle \mu \rangle$ may contain a vertex in $\{u, v\}$. If a vertex $w^* \in \cup_{i \in I_\mu} S_i - \{u, v\}$ is mapped to an ancestor $\psi(w)$ of ν, then $\mathrm{dt}(\psi(w)) \leq \mathrm{dt}(\psi(w^*))$ for any $w \in \mathrm{a}^{(3)}_{\mathrm{dt}}\langle \mu \rangle$, where $w \neq w^*$ holds only when $w \in \{u, v\}$. Since $|\{u, v\}| < 3$, the set $\mathrm{a}^{(3)}_{\mathrm{dt}}\langle \mu \rangle$ contains a vertex $w \notin \{u, v\}$ if and only if such a vertex w^* exists.

The above procedure of testing the inequalites on $w \in \mathrm{a_{lp}}\langle \mu \rangle \cup \mathrm{a_{rp}}\langle \mu \rangle \cup \mathrm{a}^{(3)}_{\mathrm{dt}}\langle \mu \rangle$ can be executed in $O(1)$ time. □

7.3 Proper Embeddings

For each S- or R-node $\nu \in \mathcal{V}$, any planar embedding γ_ν of the skeleton $\mathrm{skl}(\nu)$ is called *proper*. For a P-node ν, a planar embedding of the skeleton $\mathrm{skl}(\nu)$ is determined by an order $(e_1, e_2, \ldots, e_p)$ of the edges in $E(\nu)$. An order $(e_1, e_2, \ldots, e_p)$ of the edges in $E(\nu)$ is called a *proper embedding* if no star $s \in A$ touches edges e_i and e_j, $i < j$ such that "$i + 1 \neq j$" or "$i = 1$ and $j = p$." For a proper embedding γ_ν of an S-, P- or R-node ν, let $\Phi(\gamma_\nu)$ denote the set of faces in γ_ν, where the two faces whose facial cycles share an edge $e \in E(\nu)$ are denoted by $f_1(e)$ and $f_2(e)$. We call a face $f \in \Phi(\gamma_\nu)$ *genuine* if C_f does not contain the parent-edge $\mathrm{pe}(\nu)$. There are exactly two non-genuine faces $f_1(\mathrm{pe}(\nu)), f_2(\mathrm{pe}(\nu)) \in \Phi(\gamma_\nu)$ if ν is not the root.

Function α For each star $s_i \in A$, define $\alpha(s_i)$ to be the highest node ν in $\mathcal{N}_i^*$ such that $|\mathrm{Ch}(\nu; \mathcal{T}_i)| + |\psi_i^{-1}(\nu)| \geq 2$ (such a node ν exists since $|S_i| \geq 2$). Let A_{R} (resp., A_{P} and A_{S}) be the set of stars $s_i \in A$ such that $\alpha(s_i)$ is a split R-node or the root R-node ν_{root} (resp., $\alpha(s_i)$ is a P-node and an S-node).

Function β For each star $s_i \in A$ with $\alpha(s_i) = \nu$ for a node ν, there are at most two faces in $\Phi(\gamma_\nu)$ in which the star s_i with incident blue edges can be drawn without creating any crossing with a red edge, and we denote by $\beta(s_i)$ the set of such faces in $\Phi(\gamma_\nu)$, where $\beta(s_i) \subseteq \{f_1(e), f_2(e)\}$ for some edges $e \in E^-(\nu)$.

We call a planar embedding of γ_G of the red graph G *proper* if the planar embedding of the skeleton $\mathrm{skl}(\nu)$ of each P-node ν becomes the proper embedding γ_ν. Note that a proper embedding of G is not unique in general. A proper embedding of G is determined by choosing a bijection ϕ_ν for each non-root S-, P- or R-node ν, called a *flip mapping*, such that $\phi_\nu : \{f_1(\mathrm{pe}(\nu)), f_2(\mathrm{pe}(\nu))\} \to \{f_1(e_\nu), f_2(e_\nu)\}$, where e_ν denotes the child virtual edge in $E(\eta)$ of the parent node η.

In a given proper embedding of G, switching a flip mapping from $\phi_\nu(f_i(\mathrm{pe}(\nu))) = f_{j_i}(e_\nu)$, $i = 1, 2$ to $\phi_\nu(f_i(\mathrm{pe}(\nu))) = f_{3-j_i}(e_\nu)$, $i = 1, 2$ is denoted by $\phi_\nu := \overline{\phi_\nu}$.

Note that for a non-genuine face $f = f_i(\mathrm{pe}(\nu))$ in γ_ν of a node ν, the mapped face $\phi_\nu(f)$ may be a non-genuine face again. We denote by $\phi^*(f)$ the genuine face f' mapped from f by repeatedly applying flip mappings; i.e., $f' = \phi_{\nu_q}(\phi_{\nu_{q-1}}(\cdots(\phi_{\nu_1}(f))\cdots))$ is a genuine face in $\Phi(\gamma_{\nu_q})$ for $\nu_1 = \nu$ and the parent ν_{i+1} of ν_i, $i = 1, 2, \ldots, q-1$.

In the following, we fix a proper embedding of γ_G of G by fixing a flip mapping $\phi_\nu(f_i(\mathrm{pe}(\nu))) = f_i(e_\nu)$, $i = 1, 2$ for each non-root S-, P- or R-node ν.

7.4 Testing P-nodes in the First Phase

A P-node ν admits a proper embedding if and only if there are no stars that satisfy the condition in Lemma 13(i), or Lemma 14(i) or (ii) for the P-node ν. We first show how to test if there is a P-node which satisfies the condition in Lemma 13(i).

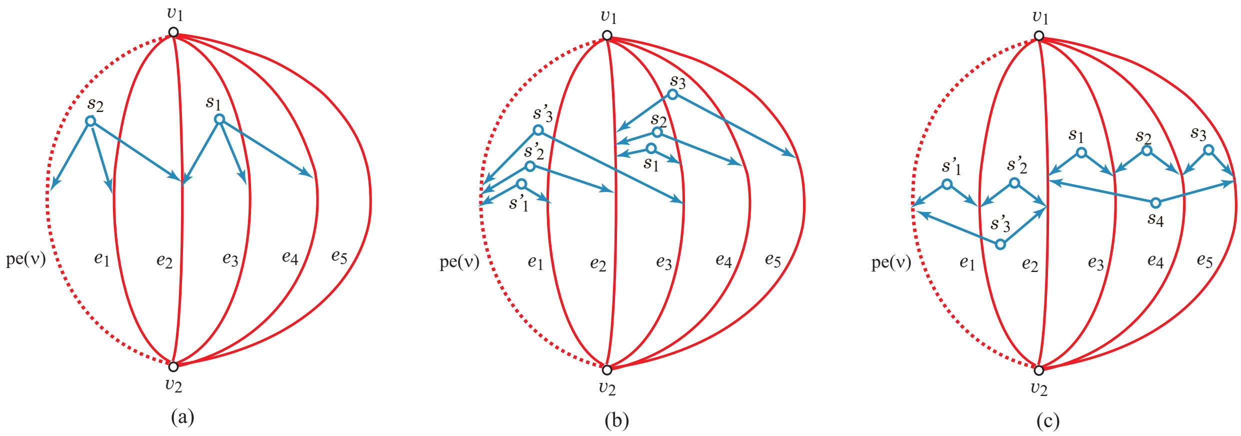


Figure 9: Illustration of forbidden configurations in the skeleton $\mathrm{skl}(\nu)$ of a non-root P-node ν in the rooted SPR-tree of the red graph G: (a) A star s_1 that touches three virtual edges in $E^-(\nu)$ and a star s_2 that touches the parent-virtual edge $\mathrm{pe}(\nu)$ and two virtual edges in $E^-(\nu)$; (b) Three stars s_i, $i = 1, 2, 3$ such that s_i touches virtual edges $e_2, e_{i+2} \in E^-(\nu)$, and three stars s'_i, $i = 1, 2, 3$ such that s'_i touches the parent-virtual edge $\mathrm{pe}(\nu)$ and a virtual edge $e_{i+1} \in E^-(\nu)$; (c) A set of $p = 4$ star s_i, $i = 1, 2, \ldots, p$ such that s_i touches virtual edges $e_{i+1}, e_{i+2} \in E^-(\nu)$, where $e_{p+2} = e_1$, and a set of $p = 3$ star s'_1, s'_2 and s'_3 such that s'_1 touches the parent-virtual edge $\mathrm{pe}(\nu)$ and a virtual edge $e_1 \in E^-(\nu)$, s'_2 touches two virtual edges $e_1, e_2 \in E^-(\nu)$, and s'_3 touches virtual edge $e_2 \in E^-(\nu)$ and $\mathrm{pe}(\nu)$.

Lemma 18 *For each star $s_i \in A$, testing if there exists a P-node ν such that star s_i touches three virtual edges in $E(\nu)$ can be done in $O(|S_i|)$ time. The total time for testing this for all stars s_i is $O(m)$.*

Proof. Fix a star s_i and a P-node ν. See Fig. 9(a) for an illustration of $\mathrm{skl}(\nu)$ of a P-node ν with some star touching three virtual edges. We see that s_i touches three virtual edges in $E^-(\nu)$ if and only if P-node ν appears in the mimic tree $\mathcal{T}_i$ having at least three child nodes. Assume that s_i touches at most two virtual edges in $E^-(\nu)$. Then s_i touches three edges in $E(\nu)$ if and only if P-node ν appears in the mimic tree $\mathcal{T}_i$ having exactly two child nodes. and s_i touches the parent-edge $\mathrm{pe}(\nu)$. For any node ν in $\mathcal{T}_i$, we can test whether s_i touches $\mathrm{pe}(\nu)$ in $O(1)$ time by Lemma 15.

Recall that the number of vertices in the mimic tree $\mathcal{T}_i$ is at most $3|S_i|$. Hence the time for testing for all P-nodes over all stars s_i is $O(\sum_{1\le i\le k}|S_i|) = O(m)$. □

We next show how to test whether there is a P-node which satisfies the condition in Lemma 14(i) or (ii).

Lemma 19 *Assume that for each P-node in $\mathcal{V}$, no star $s \in A$ touches at least three virtual edges in the skeleton of the P-node. Testing whether there is a P-node $\nu \in \mathcal{V}$ which satisfies the condition of Lemma 14(i) or (ii) can be done in $O(n+m)$ time.*

Proof. See Fig. 9(b) and (c) for an illustration of $\mathrm{skl}(\nu)$ of a P-node ν with some stars touching three virtual edges. By the assumption, any P-node in any mimic tree $\mathcal{T}_i$ can have at most two children. We first traverse each mimic tree $\mathcal{T}_i$ to find the set of all P-nodes with exactly two child nodes in $\mathcal{T}_i$ in $O(m)$ time. Then we know that, for each P-node $\nu \in \mathcal{V}$, which pair of virtual edges in $E^-(\nu)$ are touched by some star $s \in A$.

Next select a P-node $\nu \in \mathcal{V}$ and a virtual edge $e_\mu \in E^-(\nu)$, which corresponds to a child node $\mu \in \mathrm{Ch}(\nu; \widehat{\mathcal{T}})$. Then we can test whether there is a star s_i that touches $\mathrm{pe}(\mu)$ and the virtual edge e_μ in $O(1)$ time by Lemma 17. After applying this to all P-nodes and their child nodes in $O(n)$ time, we know that which virtual edge in $E^-(\nu)$ and $\mathrm{pe}(\nu)$ are touched by some star $s \in A$.

Based on the above observation, we can test whether there is a P-node ν that satisfies the condition of Lemma 14(i) or (ii) in $O(n+m)$ time. □

In the following, we assume that each P-node ν admits a proper embedding γ_ν.

7.5 Testing R-nodes in the First Phase

We test whether there is an R-node which satisfies the condition in Lemma 13(ii) by three steps. The first step checks the following condition.

Lemma 20 *It takes $O(m)$ time to test whether there is a non-root R-node $\nu \in \mathcal{V}$ such that some star $s_i \in A$ touches the parent edge $\mathrm{pe}(\nu)$ and two elements z_1 and z_2 such that $V(\nu^{\mathrm{o}j}) \cup E(\nu^{\mathrm{o}j})$ for each $j = 1, 2$,*

Proof. Let s_i be a star in A. Assume that there is a non-root R-node satisfying the condition in the lemma for the star s_i. See Fig. 8(a) for an illustration of $\mathrm{skl}(\nu)$ of an R-node ν with some stars touching three virtual edges. Then for any such R-node ν, the mimic tree $\mathcal{T}_i$ must contain the split nodes ν^{cp}, $\nu^{\mathrm{o}1}$ and $\nu^{\mathrm{o}2}$. When nodes $\nu^{\mathrm{o}1}$ and $\nu^{\mathrm{o}2}$ appear in $\mathcal{T}_i$, it holds $\mathcal{N}_i \cap D(\nu^{\mathrm{o}j}; \widehat{\mathcal{T}}) \neq \emptyset$ for both $j = 1, 2$, i.e., s_i touches an element in $V(\nu^{\mathrm{o}j}) \cup E(\nu^{\mathrm{o}j})$ for each $j = 1, 2$. Testing whether $\mathcal{T}_i$ contains a node ν^{cp} satisfying the above can be done in $O(|S_i|)$ time.

By Lemma 15, we can test in $O(1)$ time whether star $s_i \in A$ touches the parent edge $\mathrm{pe}(\nu)$ and a vertex z with $\psi(z) \in D(\nu^{\mathrm{cp}}; \widehat{\mathcal{T}})$.

Therefore we can test in $O(m)$ time whether there is a non-root R-node $\nu \in \mathcal{V}$ satisfying the condition in the lemma for some star $s_i \in A$. □

The second step checks the following condition.

Lemma 21 *It takes $O(n)$ time to test whether there is a non-root R-node $\nu \in \mathcal{V}$ such that some star $s_i \in A$ touches the parent edge $\mathrm{pe}(\nu)$ and an element in $V(\nu^{\mathrm{in}}) \cup E(\nu^{\mathrm{in}})$.*

Proof. See Fig. 8(b) for an illustration of $\mathrm{skl}(\nu)$ of an R-node ν with some stars touching three virtual edges. For each non-root R-node $\nu \in \mathcal{V}$, we test whether there is a star $s \in A$ that touches the parent edge $\mathrm{pe}(\nu)$ and a vertex z with $\psi(z) \in D(\nu^{\mathrm{in}}; \widehat{\mathcal{T}})$. This can be done in $O(1)$ by Lemma 17. Therefore we can test in $O(n)$ time whether there is a non-root R-node $\nu \in \mathcal{V}$ satisfying the condition in the lemma. □

In the following, we assume that no non-root R-node satisfies the condition in Lemma 20 or Lemma 21. If a star $s_i \in A$ satisfies the condition in Lemma 13(ii) for an R-node $\nu \in \mathcal{V}$, then the star s_i touches two elements in $(V(\nu) - V(\nu^{\mathrm{cp}})) \cup E^-(\nu)$, but does not touch the parent edge $\mathrm{pe}(\nu)$. For such a star s_i and an R-node ν, it holds that $\alpha(s_i) \in \{\nu^{\mathrm{cp}}, \nu^{\mathrm{in}}, \nu^{\mathrm{o}1}, \nu^{\mathrm{o}2}\}$ (or $\alpha(s_i) = \nu$ for the root R-node ν_{root}). Therefore the condition in Lemma 13(ii) for some R-node holds if and only if $\beta(s_i) = \emptyset$ holds for some star $s_i \in A_{\mathrm{R}}$. Finally the third step computes $\beta(s_i)$ for all such stars $s_i \in A_{\mathrm{R}}$.

Lemma 22 *It takes $O(n+m)$ time to compute $\beta(s_i)$ for all such stars $s_i \in A_{\mathrm{R}}$.*

Proof. Let $\nu \in \mathcal{V}$ be an R-node. For each star $s_i \in A_{\mathrm{R}}$ such that $\alpha(s_i)$ is a split node of ν or $\nu = \nu_{\mathrm{root}}$, all virtual edges in $E^-(\nu)$ touched by s_i correspond to the children of nodes ν^{in}, ν^{o1} and ν^{o2} in the mimic tree $\mathcal{T}_i$, while the set of the vertices $V(\nu)$ touched by s_i is the union $\psi^{-1}(\nu^{\mathrm{cp}}) \cup \psi^{-1}(\nu^{\mathrm{o1}}) \cup \psi^{-1}(\nu^{\mathrm{o2}}) \cup \psi^{-1}(\nu^{\mathrm{in}})$. With the planar embedding γ_ν and the elements in $V(\nu) \cup E^-(\nu)$ touched by stars in A_{R}, we can compute $\beta(s)$ for all such stars in $O(|V(\nu)| + |E(\nu)| + |\{s \in A_{\mathrm{R}} \mid \alpha(s) = \nu\}|)$ time by Theorem 10. The total time for computing $\beta(s)$, $s \in A_{\mathrm{R}}$ with $\alpha(s) = \nu$ overall R-nodes $\nu \in \mathcal{V}$ is $O(n + m)$. □

Simplified Skeletons

Before we proceed to the second phase, we define "simplified skeletons" for P- and R-nodes.

For each non-root P-node $\nu \in \mathcal{V}$ with a proper embedding $\gamma_\nu = (\mathrm{pe}(\nu), e_1, e_2, \ldots, e_p)$ of edges in $E(\nu)$, we define the *simplified skeleton* $\mathrm{sskl}(\nu)$ to be a cycle of length 2

$$\mathrm{sskl}(\nu) = (V(\nu), E_s(\nu) = \{e_1, e_p\}).$$

For each non-root R-node $\nu \in \mathcal{V}$, we define the *simplified skeleton* $\mathrm{sskl}(\nu)$ to be a cycle of length 2

$$\mathrm{sskl}(\nu) = (V(\nu^{\mathrm{cp}}), E_s(\nu) = \{e_1, e_2\}),$$

by setting e_1 and e_2 correspond to the o1-R-node and o2-R-node $\nu^{\mathrm{o1}}, \nu^{\mathrm{o2}} \in \widehat{\mathcal{V}}$ of ν, respectively. We say that a star $s_i \in A$ *touches* edge e_j if it touches an element in $V(\nu^{\mathrm{o}j}) \cup E(\nu^{\mathrm{o}j})$.

We do not prepare any simplified skeleton of $\mathrm{skl}(\nu)$ of any S-node. Note that each cycle of a simplified skeleton for a P- or R-node ν has two possible embeddings ϕ_ν when we determine a planar embedding of the entire red graph G. The next section finally tests whether there is a proper embedding of the red graph G that gives an E-planar embedding of (H, G, A).

8 Phase 2 for Case of Connectivity 2

The second phase examines whether there is a combination of embeddings of simplified skeletons $\mathrm{skl}(\nu)$ over all child virtual edges $\nu \in E^-(\eta)$ for each P- or R-node such that all stars in A can be drawn in a face without creating a crossing with a red edge. To facilitate the examination, we combine the simplified skeletons into the skeleton of the parent node η of ν (or the parent η of parent of ν if the parent of ν is an S-node). This results in a skeleton of η where each virtual edge $e \in E^-(\eta)$ is replaced with a simplified skeleton (or a chain of simplified skeletons if e corresponds to an S-node), which is called "refined skeletons."

Refined Skeletons

We define a "refined skeleton" $\mathrm{rskl}(\eta)$ for each node $\nu \in \mathcal{V}$ as follows:

- For each S-node $\eta \in \mathcal{V}$, the *refined skeleton* $\mathrm{rskl}(\eta)$ is defined to be a circular chain obtained from the simple cycle $\mathrm{skl}(\eta)$ by replacing each virtual edge $e \in E^-(\eta)$ corresponding to a child P- or R-node $\nu \in \mathrm{Ch}(\eta; \mathcal{T})$ with the simplified skeleton $\mathrm{sskl}(\nu)$; and
  For each non-root S-node $\eta \in \mathcal{V}$, let $\mathrm{rskl}^-(\eta)$ denote the u, v-chain obtained from $\mathrm{rskl}(\eta)$ by removing the parent-edge $\mathrm{pe}(\eta) = uv$.

- For a P- or R-node $\eta \in \mathcal{V}$, the *refined skeleton* $\mathrm{rskl}(\eta)$ is defined to be the planar embedding obtained from the planar embedding γ_η of skeleton $\mathrm{skl}(\eta)$ by replacing each virtual edge $e = uv \in E^-(\eta)$ with a u, v-chain Q_e such that Q_e is the simplified skeleton $\mathrm{sskl}(\nu)$ if the corresponding child $\nu_e \in \mathrm{Ch}(\eta; \mathcal{T})$ is a P- or R-node; and Q_e is the u, v-chain $\mathrm{rskl}^-(\nu)$ if ν_e is an S-node.

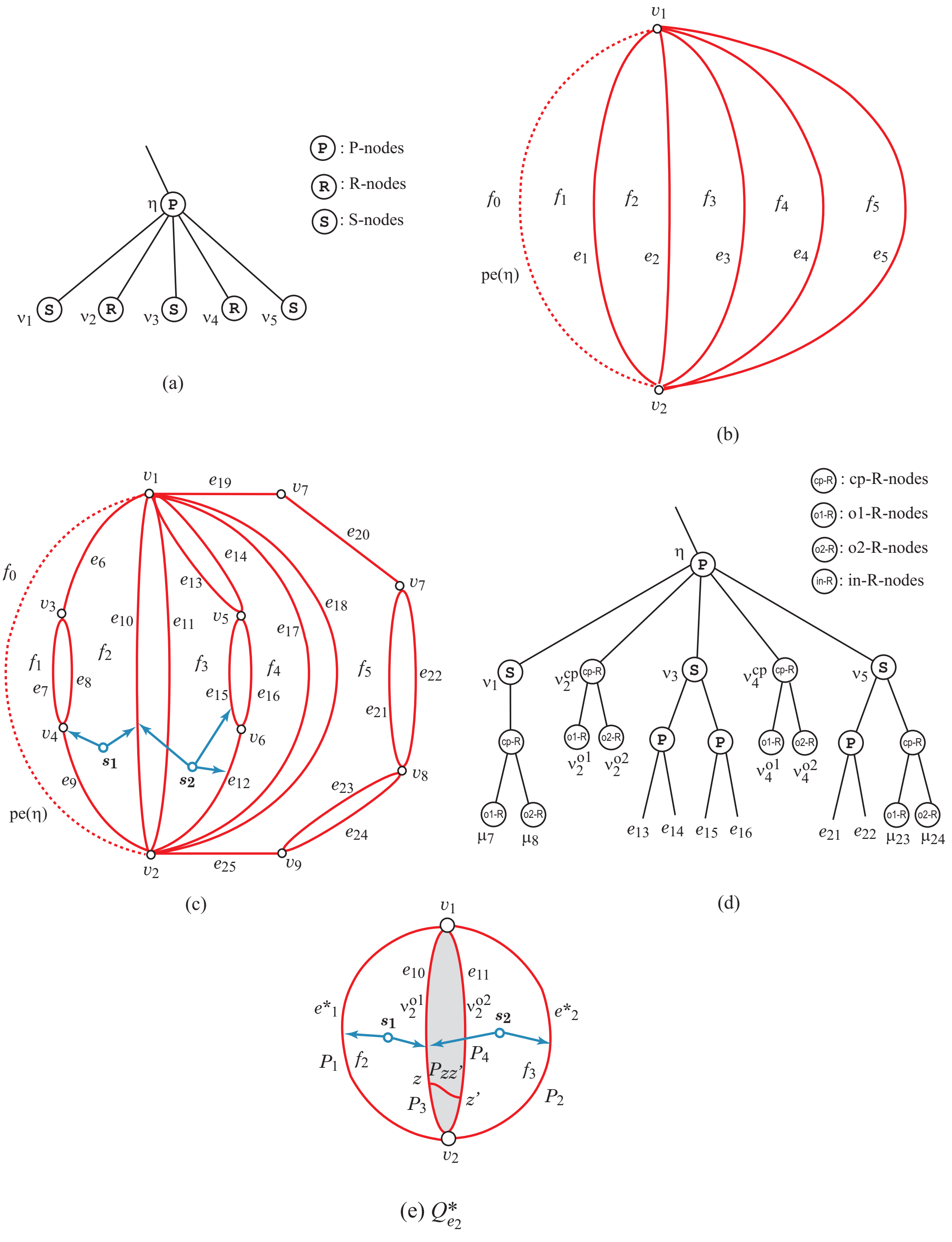


Figure 10: Illustration of skeleton $\mathrm{skl}(\eta)$ and refined skeleton $\mathrm{rskl}(\eta)$ of a P-node: (a) A P-node $\eta \in \mathcal{V}$ in the rooted SPR-tree $\mathcal{T}$ of G, where $\mathrm{Ch}(\eta;\mathcal{T}) = \{\nu_1, \ldots, \nu_5\}$. (b) The skeleton $\mathrm{skl}(\eta) = (V(\eta) = \{v_1, v_2\}, E(\eta) = \{e_1, \ldots, e_5, \mathrm{pe}(\eta)\})$ of P-node η in (a); (c) The refined skeleton $\mathrm{rskl}(\eta)$ with $V(\mathrm{rskl}(\eta)) = \{v_1, v_2, v_3, \ldots, v_9\}$ and $E(\mathrm{rskl}(\eta)) = \{e_6, e_7, \ldots, e_{25}, \mathrm{pe}(\eta)\}$ of P-node η in (a), and some stars touching elements in $\mathrm{rskl}(\eta)$, where $\mathrm{VE}(s_1;\eta) = \{v_4, e_{10}\}$ and $\mathrm{VE}(s_2;\eta) = \{e_{10}, e_{12}, e_{15}\}$; (d) The parent-child relation among the edges in $E(\mathrm{rskl}(\eta))$; and (e) A chain instance $(Q^*_{e_2}, A_{e_2} = \{s_1, s_2\})$ with two factor cycles $C_1 = (e_{10}, e_{11})$ and $C^*_{e_2} = (e^*_1, e^*_2)$, where the instance contains a twisted set that induces a forbidden graph of type (i).

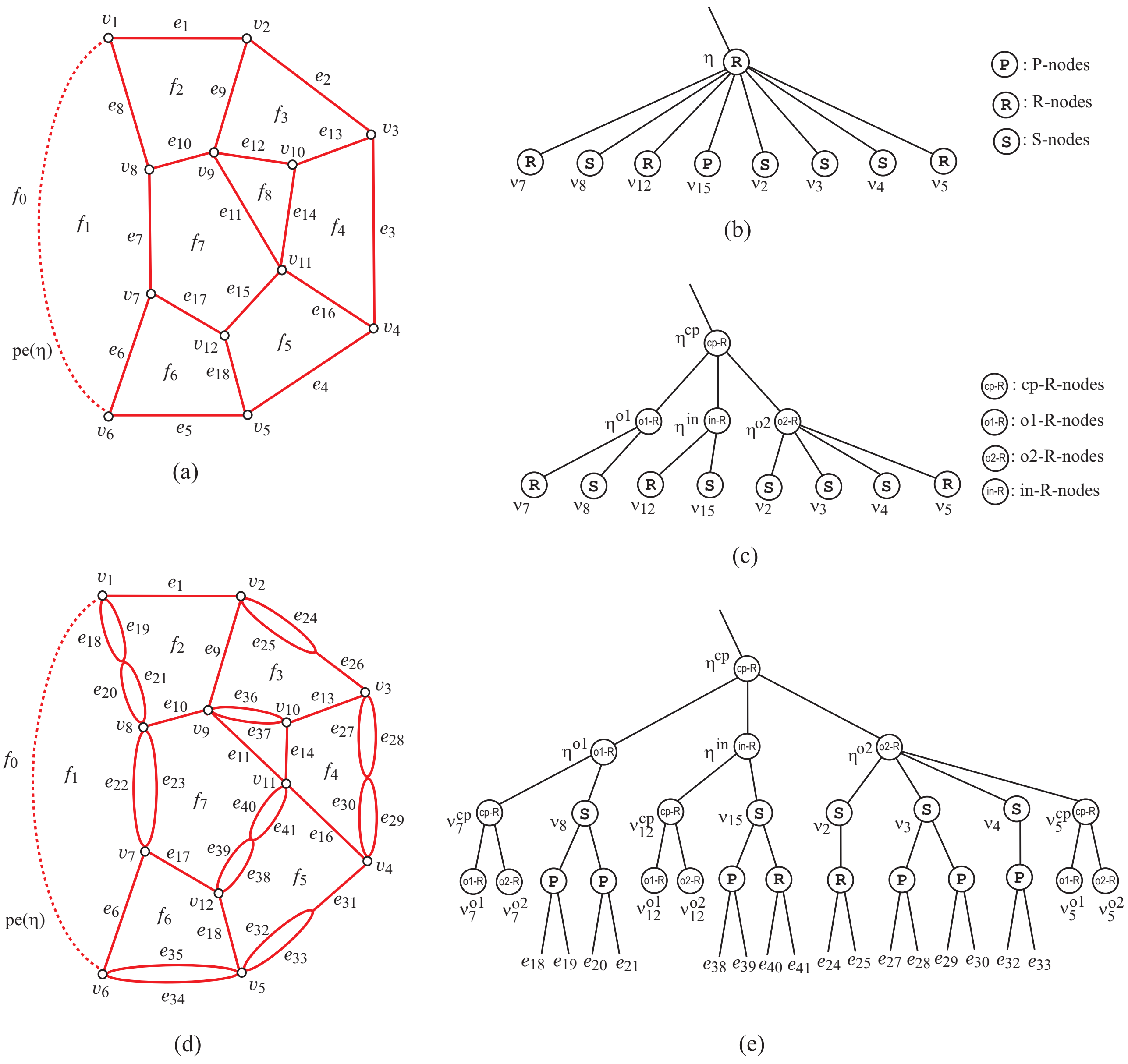


Figure 11: Illustration of skeleton $\mathrm{skl}(\eta)$ and refined skeleton $\mathrm{rskl}(\eta)$ of an R-node: (a) The skeleton $\mathrm{skl}(\eta) = (V(\eta) = \{v_1, v_2, \ldots, v_{12}\}, E(\eta) = \{e_1, \ldots, e_{18}, \mathrm{pe}(\eta)\})$ of an R-node η (b) R-node $\eta \in \mathcal{V}$ in the rooted SPR-tree $\mathcal{T}$ of G, where $\mathrm{Ch}(\eta; \mathcal{T}) = \{\nu_2, \nu_3, \nu_4, \nu_5, \nu_7, \nu_8, \nu_{12}, \nu_{15}\}$; (c) The split nodes of R-node η in (a); and (d) The refined skeleton $\mathrm{rskl}(\eta)$ of R-node η in (a); (e) The parent-child relation among the edges in $E(\mathrm{rskl}(\eta))$.

Note that, when η is not the root, $\mathrm{rskl}(\eta)$ still contains the original parent-virtual edge $\mathrm{pe}(\eta)$, which is not replaced with any chain. See Fig. 10(c)-(d) and Fig. 11(d)-(e) for illustrations of the refined skeleton $\mathrm{rskl}(\eta)$ of a P- or R-node η and the corresponding nodes in $\widehat{\mathcal{T}}$.

The u, v-chains Q_e replaced from child virtual edges $e \in E^-(\nu)$ are called *elementary chains* in $\mathrm{rskl}(\eta)$. For each elementary u, v-chain Q in $\mathrm{rskl}(\eta)$, let $E_{\mathrm{cycle}}(Q)$ denote the set of edges in factor cycles in Q and let $V_{\mathrm{in}}(Q)$ denote the set of vertices in Q other than the terminals u and v of Q. An elementary chain with $E_{\mathrm{cycle}}(Q) \neq \emptyset$ is called *nontrivial*. Let $E(\mathrm{rskl}(\eta))$ be the set of all real and virtual edges over all elementary chains in $\mathrm{rskl}(\eta)$, and $V(\mathrm{rskl}(\eta))$ be the set of all end-vertices of edges in $E(\mathrm{rskl}(\eta))$, where $V(\mathrm{rskl}(\eta)) \subseteq V$.

As for the size of refined skeletons, the refined skeletons of all P- and R-nodes are obtained from skeletons of all nodes by replacing each virtual edge with two multiple edges and by merging the skeleton of each S-node into the skeleton of the parent node of the S-node. Therefore the total size of all refined skeletons remains $O(n)$. Then we see that constructing the refined skeletons $\mathrm{rskl}(\eta)$ for all P- and R-nodes $\eta \in \mathcal{V}$ can be done in $O(n)$ time.

Twistless Embeddings

Let $\eta \in \mathcal{V}$ be a P- or R-node. For each star $s \in A$, let $\mathrm{VE}(s; \eta)$ denote the set of the elements in $V(\mathrm{rskl}(\eta)) \cup E(\mathrm{rskl}(\eta))$ that are touched by s. Let $A(\eta)$ be the set of stars $s \in A$ with $\mathrm{VE}(s; \eta) \neq \emptyset$. See Fig. 12(a) for an illustration of several different types of stars $s \in A$ that touches some vertex or edge in the refined skeleton $\mathrm{rskl}(\eta)$ of an R-node η. Notice that no star $s \in A(\eta)$ touches the both sides of the same factor cycle in any elementary chain, since otheriwise there would be a P-node (or R-node) satisfying the condition in Lemma 18 (or Lemma 20). For example, star s_7 in the refined skeleton $\mathrm{rskl}(\eta)$ in Fig. 12(a) could not exist in fact, since it touches the two sides e_{24} and e_{25} of the same factor cycle.

Fix a virtual edge $e = uv \in E^-(\eta)$ such that Q_e is nontrivial. Let $f_1(e)$ and $f_2(e)$ denote the two faces in $\Phi(\gamma_\eta)$ of the planar embedding γ_η such that their facial cycles $C_{f_1(e)}$ and $C_{f_2}(e)$ share edge e. Note that each factor cycle C in Q_e has two possible embeddings, one of the sides of C is drawn in $f_1(e)$ and the other in $f_2(e)$, which is determined by a choice of the flip mapping ϕ_μ. An embedding γ_e of Q_e is determined by a combination of flip mappings ϕ_μ of all factor cycles $C = \mathrm{sskl}(\mu)$ in Q_e. Since a flip mapping ϕ_μ for each factor cycle $C = \mathrm{sskl}(\mu)$ is currently fixed, we denote by a set $\mathcal{C}_{\mathrm{flip}}$ of factor cycles in an elementary chain Q_e to mean an embedding γ_e of Q_e that is obtained by flipping each cycle $C = \mathrm{sskl}(\mu) \in \mathcal{C}_{\mathrm{flip}}$, i.e., setting $\phi_\mu := \overline{\phi_\mu}$. Let $A(e)$ be the set of stars $s \in A(\eta)$ that touches an element in $V_{\mathrm{in}}(Q_e) \cup E(Q_e)$. An embedding $\gamma_e = \mathcal{C}_{\mathrm{flip}}$ of Q_e is called *twistless* if each star $s \in A(e)$ can be drawn inside one of the two faces $f_1(e), f_2(e) \in \Phi(\gamma_\eta)$ without creating crossing with red edges in $\mathrm{rskl}(\eta)$.

Chain Instances

To test whether Q_e admits a twistless embedding γ_e, we define "chain instances." Construct a circular chain Q_e^* from the elementary u, v-chain Q_e by adding a cycle C_e^* of two new virtual edges $e_1^*, e_2^* = uv$. For $j = 1, 2$, let e_j^* be a new virtual edge uv that corresponds to the set $\mathcal{Q}_j$ of elementary chains $Q_{e'}$ generated from virtual edges e' $(\neq e)$ along facial cycle $C_{f_j(e)}$. We say that a star $s \in A(e)$ *touches* a virtual edge e_j^* if s touches some element in $V_{\mathrm{in}}(Q_{e'}) \cup E(Q_{e'})$ of an elementary chain $Q_{e'} \in \mathcal{Q}_j$. Let Q_e^* be the circular chain obtained from Q_e by adding these virtual edges e_1^* and e_2^*, which form a cycle C_e^*. We call a pair $(Q_e^*, A(e))$ of a red graph Q_e^* and a set $A(e)$ of stars a *chain instance* (defined for a virtual edge $e \in E^-(\eta)$). See Fig. 10(e) and Fig. 12(b) for an illustration of a chain instance $(Q_e^*, A(e))$.

Twisted Sets

A *twisted* set is defined to be a pair $(\{s_1, s_2, \ldots, s_p\}, \{C_1, C_2, \ldots, C_p\})$ of a set of $p \geq 2$ distinct stars in $A(e)$ and a set of p distinct factor cycles in Q_e^* (possibly $C_i = C_e^*$ for some i) with sides P_i^0 and P_i^1 of each cycle C_i such that, for each $i = 1, 2, \ldots, p-1$, star s_i touches an edge in P_i^k and an edge in P_{i+1}^k for

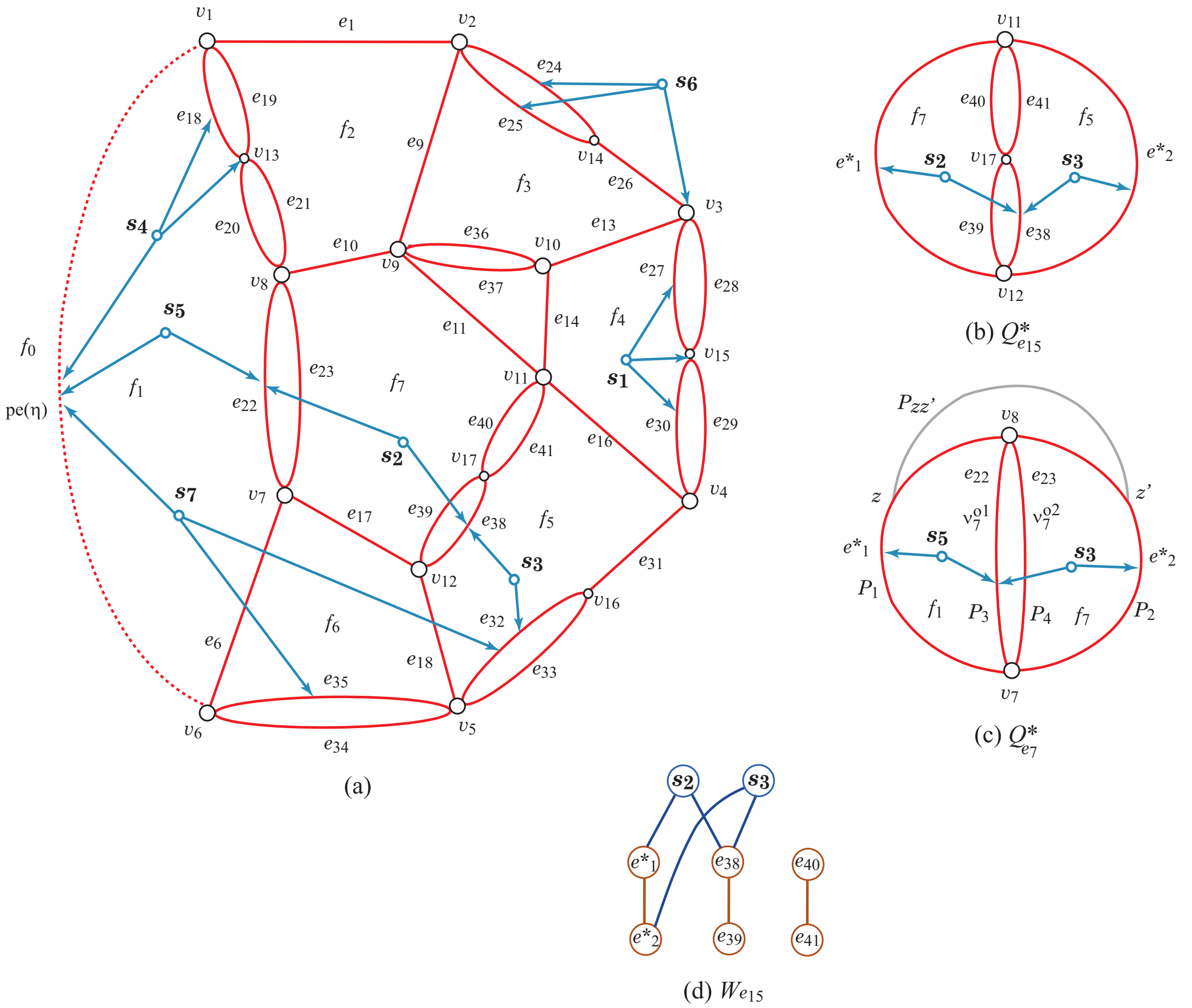


Figure 12: (a) Illustration of stars $s \in A$ that touches some vertex or edge in the refined skeleton $\mathrm{rskl}(\eta)$ of an R-node, where $\mathrm{VE}(s_1;\eta) = \{e_{27}, e_{30}, v_{15}\}$, $\mathrm{VE}(s_2;\eta) = \{e_{22}, e_{39}\}$, $\mathrm{VE}(s_3;\eta) = \{e_{32}, e_{38}\}$, $\mathrm{VE}(s_4;\eta) = \{e_{18}, \mathrm{pe}(\eta), v_{13}\}$, $\mathrm{VE}(s_5;\eta) = \{e_{22}, \mathrm{pe}(\eta)\}$, $\mathrm{VE}(s_6;\eta) = \{e_{24}, e_{25}, v_3\}$ and $\mathrm{VE}(s_7;\eta) = \{e_{32}, e_{35}, \mathrm{pe}(\eta)\}$. It holds that $\{s_1, s_2, \ldots, s_7\} \subseteq A(\eta)$, where s_6 must not exist after the first phase; s_i, $i = 1, 2, \ldots, 6$ are linking stars, and s_5 is an e_{22}-bridging star, while $s_7 \notin A^{\mathrm{link}}(\eta) \cup A^{\mathrm{br}}(\eta)$; (b) A chain instance $(Q^*_{e_{15}}, A_{e_{15}} = \{s_2, s_3\})$ with three factor cycles $C_1 = (e_{38}, e_{39})$, $C_2 = (e_{40}, e_{41})$ and $C^*_{e_{15}} = (e^*_1, e^*_2)$, where the instance contains a twisted set that induces a forbidden graph of type (v); (c) A chain instance $(Q^*_{e_7}, A_{e_7} = \{s_3, s_5\})$ with two factor cycles $C_1 = (e_{22}, e_{23})$ and $C^*_{e_7} = (e^*_1, e^*_2)$, where the instance contains a twisted set that induces a forbidden graph of type (iv); and (d) The auxiliary graph $W_{e_{15}}$ defined to the chain instance $(Q^*_{e_{15}}, A_{e_{15}})$ in (b), where $(s_3, e_{38}, s_3, e^*_2, e^*_1)$ is a chordless odd cycle of length 5.

the same $k \in \{1, 2\}$, but star s_p touches an edge in P_p^k and an edge in P_1^ℓ for $k \neq \ell$. Observe that if there is no twisted set in Q_e^*, then Q_e admits a twistless embedding γ_e.

Lemma 23 *If a chain instance $(Q_e^*, A(e))$ for a virtual edge $e = uv \in E^-(\eta)$ has a twisted set, then $(G, A(e))$ contains a forbidden graph F of type of (i), (iv) or (v), which can be found in $O(n+m)$ time.*

Proof. Let $(\{s_1, s_2, \ldots, s_p\}, \{C_1, C_2, \ldots, C_p\})$ be a twisted set in $(Q_e^*, A(e))$. We distinguish two cases.

Case 1. $p \geq 3$: See Fig. 12(b) for such a circular instance $(Q_e^*, A(e))$. If there is a twisted set, then we see by Lemma 12 that each side P_i^j of C_i has an associated red path P of G and each blue edge touching side P_i^j can be extended to reach a vertex in P by adding some red edges. Hence given a twisted set with $p \geq 3$, we can construct a forbidden subgraph F of type (v) in $O(n+m)$ time.

Case 2. $p = 2$: See Fig. 10(e) and Fig. 12(c) for such a circular instance $(Q_e^*, A(e))$. In this case, $C_p = C_2 = C_e^*$ and $C_1 = (e_1, e_2)$ is a simplified skeleton of a child P- or R-node $\nu \in \mathrm{Ch}(\eta; \mathcal{T})$, which corresponds to the virtual edge in $e = uv \in E^-(\eta)$. Without loss of generality assume that s_j, $j = 1, 2$ touches e_1 and e_j^*. Then the red graph G has four internally disjoint red u, v-paths P_i, $i = 1, 2, 3, 4$ such that P_j is a representing path of virtual edge e_j^*, $j = 1, 2$, and P_3 (resp., P_4) represents to the outer boundary of $\gamma_\nu - \mathrm{pe}(\eta)$ corresponding to edge e_1 (resp., e_2). We distinguish two subcases.

(a) $C_1 = (e_1, e_2)$ is a simplified skeleton of a child R-node. Since the skeleton γ_ν is triconnected, the associated red subgraph $G_{e'}$ has a red z, z'-path $P_{z,z'}$ that joins a vertex $z \in P_1$ and a vertex $z' \in P_4$, as illustrated in Fig. 10(e). Then analogously with Lemma 14(i), we see that the set of these five paths and two stars s_1 and s_2 give rise to a forbidden graph of type (i), which can be obtained in $O(n+m)$ time.

(b) $C_1 = (e_1, e_2)$ is a simplified skeleton of a child P-node: In this case, η is an R-node, and the red graph $G - V(G_e)$ has a red z, z'-path $P_{z,z'}$ that joins a vertex $z \in P_1$ and a vertex $z' \in P_4$, as illustrated in Fig. 12(c). Then analogously with Lemma 14(ii), we see that the set of these five paths and two stars s_1 and s_2 give rise to a forbidden graph of type (iv), which can be obtained in $O(n+m)$ time. □

Valid Sets

However, identifying all stars in $A(\eta)$ and computing the set $\mathrm{VE}(s; \eta)$ for each $s \in A(\eta)$ for all nodes η may take more time than $O(n+m)$.

We here define a subset $A_\eta \subseteq A(\eta)$ as a "valid" set in the sense that if the refined skeleton $\mathrm{rskl}(\eta)$ with $\{\mathrm{VE}(s; \eta) \mid s \in A(\eta)\}$ admits a twisted set then so does $\mathrm{rskl}(\eta)$ with $\{\mathrm{VE}(s; \eta) \mid s \in A_\eta\}$.

- A star $s \in A$ is called *linking* if $\mathrm{VE}(s; \eta)$ contains elements z and z' such that $z = e \in E(C)$ of a factor cycle C in an elementary chain Q and $z \in V(\mathrm{rskl}(\eta)) \cup E(\mathrm{rskl}(\eta)) - V(Q) - E(C) - \{\mathrm{pe}(\eta)\}$. Let $A^{\mathrm{link}}(\eta)$ be the set of all linking stars $s \in A$ in $\mathrm{rskl}(\eta)$.

- A star $s \in A$ is called *bridging* (or *e-bridging*) if
(i) $|\mathrm{VE}(s; \eta)| = 2$ holds, and $\mathrm{VE}(s; \eta) = \{z, z'\}$ consists of the parent edge $z = \mathrm{pe}(\eta)$ and an edge $z' = e \in E(C)$ of a factor cycle C in an elementary chain Q; and
(ii) there is no linking star $s' \in A^{\mathrm{link}}(\eta)$ with $\mathrm{VE}(s'; \eta) \supseteq \mathrm{VE}(s; \eta)$.
Let $A^{\mathrm{br}}(\eta)$ be the set of all bridging stars $s \in A$ in $\mathrm{rskl}(\eta)$.

See Fig. 12(a) for an illustration of linking stars and bridging stars, where s_i, $i = 1, 2, \ldots, 7$ are linking stars, and s_5 is an e_{22}-bridging star.

We easily see that, for each edge $e \in E(\mathrm{rskl}(\eta)) - \{\mathrm{pe}(\eta)\}$ that admits an e-bridging star, only one e-bridging star is enough to detect a possible twisted set. Then we call a subset A_η of $A^{\mathrm{link}}(\eta) \cup A^{\mathrm{br}}(\eta)$ *valid* if
(i) $A^{\mathrm{link}}(\eta) \subseteq A_\eta$; and
(ii) for each edge $e \in E(\mathrm{rskl}(\eta)) - \{\mathrm{pe}(\eta)\}$ that admits an e-bridging star, the set A_η contains at least one e-bridging star $s \in A^{\mathrm{br}}(\eta)$.

Lemma 24 *Valid sets $A_\eta \subseteq A^{\rm link}(\eta) \cup A^{\rm br}(\eta)$ together with $\{\mathrm{VE}(s;\eta) \mid s \in A_\eta\}$ for all P- and R-nodes $\eta \in \mathcal{V}$ can be computed in $O(n+m)$ time. Hence it holds that $\sum_{\text{P-, R-nodes } \eta\in\mathcal{V}} |A_\eta| = O(n+m)$.*

Proof. (I) We first show how to test whether a star $s_i \in A$ belongs the set $A^{\rm link}(\eta)$ for some P- or R-node $\eta \in \mathcal{V}$. We fix a star $s_i \in A$, and distinguish two cases: η is a P- or R-node.

Case of P-nodes: By definition, for a P-node $\eta \in \mathcal{V}$, a star s_i belongs to $A^{\rm link}(\eta)$ if and only if s_i touches an edge $e \in E(C)$ of a factor cycle C in an elementary chain Q in $\mathrm{rskl}(\eta)$ and an element $z \in V(\mathrm{rskl}(\eta)) \cup E(\mathrm{rskl}(\eta)) - V(Q) - E(C) - \{\mathrm{pe}(\eta)\}$. Hence $A^{\rm link}(\eta)$ can contain s_i only when the mimic tree $\mathcal{T}_i = (\mathcal{V}_i, \mathcal{E}_i)$ of s_i satisfies at least one of the following:

(a) the P-node η appears in $\mathcal{T}_i$ having at least two child nodes (i.e., $\eta \in \mathcal{V}_i$ and $|\mathrm{Ch}(\eta;\mathcal{T}_i)| \geq 2$);

(b) a child S-node $\nu \in \mathrm{Ch}(\eta;\widehat{\mathcal{T}})$ appears in $\mathcal{T}_i$ having at least two child nodes (i.e., $\nu \in \mathcal{V}_i$ and $|\mathrm{Ch}(\nu;\mathcal{T}_i)| \geq 2$).

First this implies that all P-nodes η that satisfy (a) or (b) for s_i can be found just by checking $\mathcal{T}_i$ in $O(|\mathcal{V}_i|) = O(|S_i|)$ time. Next it suffices to show that $\mathrm{VE}(s_i;\eta)$ for all P-nodes η satisfying (a) or (b) for s_i can be constructed in $O(|S_i|)$ time, from which we can find all P-nodes η such that $s_i \in A^{\rm link}(\eta)$ in $O(|S_i|)$ time.

Let η be a P-node which satisfies (a) or (b) for s_i. Let $\mathrm{Ch}_\eta = \mathrm{Ch}(\eta;\mathcal{T}_i) \cup \{\text{S-nodes } \nu \in \mathrm{Ch}(\eta;\widehat{\mathcal{T}}) \mid |\mathrm{Ch}(\nu;\mathcal{T}_i)| \geq 2\}$. Then:

- Each edge in $\mathrm{VE}(s_i;\eta) - \{\mathrm{pe}(\eta)\}$ corresponds to a child node $\mu \in \mathrm{Ch}(\nu;\mathcal{T}_i)$ of a node $\nu \in \mathrm{Ch}_\eta$, where ν is an S-node or a split node of an R-node in $\mathrm{Ch}(\eta;\mathcal{T})$;
- Each vertex in $\mathrm{VE}(s_i;\eta)$ corresponds to a vertex u that is mapped to η or a child S-node $\nu \in \mathrm{Ch}_\eta$. Hence the set of vertices in $\mathrm{VE}(s_i;\eta)$ is given by the union of $\psi_i^{-1}(\eta)$ and $\psi_i^{-1}(\nu)$ for all such S-nodes μ; and
- $\mathrm{pe}(\eta) \in \mathrm{VE}(s_i;\eta)$ if and only if s_i touches $\mathrm{pe}(\eta)$, which can be checked in $O(1)$ time by Lemma 15.

Hence computing $\mathrm{VE}(s_i;\eta)$ for all P-nodes η satisfying (a) or (b) for s_i can be executed in the size of the mimic tree $\mathcal{T}_i$ and the total size of $|\psi^{-1}(\nu)|$, $\nu \in \mathcal{V}_i$, which is $O(|S_i|)$.

Case of R-nodes: This case can be treated analogously with case of P-nodes. By definition, for an R-node $\eta \in \mathcal{V}$, $A^{\rm link}(\eta)$ can contain s_i only when the mimic tree $\mathcal{T}_i = (\mathcal{V}_i, \mathcal{E}_i)$ of s_i satisfies at least one of the following:

(a) the cp-R-node $\eta^{\rm cp}$ appears in $\mathcal{T}_i$ having at least two child nodes;

(b) a child S-node $\nu \in \mathrm{Ch}(\eta';\widehat{\mathcal{T}})$ with $\eta' \in \{\eta^{\rm o1}, \eta^{\rm o2}, \eta^{\rm in}\}$ appears in $\mathcal{T}_i$ having at least two child nodes (i.e., $|\mathrm{Ch}(\nu;\mathcal{T}_i)| \geq 2$). This implies that all R-nodes η that satisfy (a) or (b) for s_i can be found just by checking $\mathcal{T}_i$ in $O(|\mathcal{V}_i|) = O(|S_i|)$ time. Analogously with case of P-nodes, we can show that $\mathrm{VE}(s_i;\eta')$ for all R-nodes η' satisfying (a) or (b) for s_i can be constructed in $O(|S_i|)$ time, from which we can find all R-nodes η' such that $s_i \in A^{\rm link}(\eta')$ in $O(|S_i|)$ time.

(II) Next we show how to find bridging stars by using the split SPR-tree $\widehat{\mathcal{T}}$. Let $\eta \in \mathcal{V}$ be a P-node (the case where η is an R-node can be treated analogously). Let $\overline{E}_\eta$ be the set of virtual edges $e \in E(C)$ for some factor cycle C in $\mathrm{rskl}(\eta)$ such that there is no linking star $s' \in A^{\rm link}(\eta)$ with $\{\mathrm{pe}(\eta), e\} \subseteq \mathrm{VE}(s';\eta)$. Since we have computed $\mathrm{VE}(s;\eta)$ for all stars $s \in A^{\rm link}(\eta)$, we can obtain $\overline{E}_\eta$. By definition, a star $s \in A$ is e-bridging if and only if $e \in \overline{E}_\eta$ and $\mathrm{VE}(s;\eta) = \{\mathrm{pe}(\eta), e\}$.

Let $e \in \overline{E}_\eta$ be an edge. Then e is an edge in a simplified skeleton of a P- or R-node ξ (where ξ is a child node of η or a child node of a child S-node ν of η). If ξ is a P-node, then let μ_e be the node corresponding to edge e; and if ξ is an R-node, then let μ_e be the o1- or o2-R-node corresponding to edge e. Hence finding an e-bridging star $s \in A^{\rm br}(\eta)$, i.e., finding a star $s \in A$ that touches the parent edge $\mathrm{pe}(\eta)$ and a vertex z with $\psi(z) \in D(\mu_e;\widehat{\mathcal{T}})$ can be done in $O(1)$ time by Lemma 17. Therefore we can find a subset B_η of bridging stars so that $A_\eta = B_\eta \cup A^{\rm link}(\eta)$ becomes valid in time linear to the size of $\mathrm{rskl}(\eta)$. The total time for constructing valid sets A_η for all P- and R-nodes is $O(n+m)$. □

For each virtual edge $e = uv \in E^-(\eta)$, let $A_e = A(e) \cap A_\eta$. We are ready to detect a possible twisted set in (Q_e^*, A_e) by testing whether an auxiliary graph W_e is bipartite. Let $\mathcal{C}(Q_e^*)$ be the set of factor cycles in the circular chain Q_e^*, where we denote the two sides of each cycle $C \in \mathcal{C}(Q_e^*)$ by P_C^1 and P_C^2. We represent each cycle $C \in \mathcal{C}(Q_e^*)$ as an edge $z_C^1 z_C^2$ and join a star s_i and a vertex z_C^i with an edge sz_C^i if s touches side P_C^i. Let $W_e = (A_e \cup \{z_C^1, z_C^2 \mid C \in \mathcal{C}(Q_e^*)\}, E_e \cup \{z_C^1 z_C^2 \mid C \in \mathcal{C}(Q_e^*)\})$ be the resulting graph, where z_C^i, $i = 1, 2$ denotes a vertex that corresponds to side P_C^i of a cycle C, and E_e contains an edge sz_C^i if s touches side P_C^i. See Fig. 12(d) for an example of graph W_e. Notice that no vertex $s \in A_e$ is adjacent to the both vertcies $z_C^1 z_C^2$ of the same cycle, since no linking star s touches the both sides of the same factor cycle in any elementary chain. Hence W_e contains no cycle of length 3.

Then we see that W_e is a bipartite graph if and only if Q_e admits a twistless embedding $\gamma_e = \mathcal{C}_{\rm flip}$. In fact, a twisted set is given by a chordless odd cycle in W_e, whose length is at least 5. When W_e is bipartite, the set A_e is partitioned into $A_{f_1(e)}$ and $A_{f_2(e)}$ such that a star $s \in A_{f_i(e)}$ is placed in the face $f_i(e)$ in the twistless embedding γ_e. We can test whether W_e contains a chordless odd cycle or Q_e admits a twistless embedding γ_e in time linear to the size of Q_e and A_e by the breadth-first search.

By Lemma 24, we have $\sum_{\text{P-, R-nodes } \eta\in\mathcal{V}} \sum_{e\in E^-(\eta)}(|E(Q_e)|+|A_e|) = \sum_{\text{P-, R-nodes } \eta\in\mathcal{V}}(|E(\mathrm{rskl}(\eta)|+|A_\eta|) = O(n+m)$. Hence it takes $O(n+m)$ time to find a twisted set in a chain instance (Q_e^*, A_e) for some virtual edge $e \in E^-(\eta)$ in a P- or R-node $\eta \in \mathcal{V}$ or construct a twistless embedding $\gamma_e = \mathcal{C}_{\rm flip}$ for all virtual edges $e \in E^-(\eta)$ of all P- and R-nodes $\eta \in \mathcal{V}$.

In the former, the instance (H, G, A) has a forbidden graph F of type of (i), (iv) or (v), which can be found in $O(n+m)$ time by Lemma 23.

In the latter, we change the current proper embedding γ_G of the red graph G by flipping each factor cycle $C = \mathrm{sskl}(\mu)$ in $\gamma_e = \mathcal{C}_{\rm flip}$ by executing $\phi_\mu := \overline{\phi_\mu}$. Let γ'_G denote the resulting proper embedding of G, which we do not need to actually construct, since we instead construct a planarizing partition $\mathcal{A}$ of A as follows. For each star $s \in A$, we assign a genuine face of some proper embedding γ_ν of a node ν so that
(i) if $|\beta(s)| = 1$, then let $\beta_s := f$ for the face $f \in \beta(s)$;
(ii) if $|\beta(s)| = 2$ and $s \in A_e$ for a virtual edge $e \in E^-(\eta)$ for a P- or R-node η, then let $\beta_s := f_i(e)$ for the face $f_i(e)$ with $s \in A_{f_i(e)}$ (i.e., s is placed in face $f_i(e)$ in the twistless embedding γ_e); and
(iii) Otherwise, $|\beta(s)| = 2$ but $s \notin A_e$ for any virtual edge e, where it holds $S_i = \{u, v\}$ for some virtual edge $e = uv \in E^-(\nu)$ for a node $\nu \in \mathcal{V}$, let β_s be any of $f_1(e)$ and $f_2(e)$.
Finally we let $\mathcal{A} = \{A_1, A_2, \ldots, A_h\}$ be a partition of star set A such that $s, s' \in A_i$ if and only if $\phi^*(\beta_s) = \phi^*(\beta_{s'})$. Then $\mathcal{A}$ is planarizing. Since computing the function $\beta^*(f)$ for all non-genuine faces f can be executed in a bottom-up way along the rooted SPR-tree $\mathcal{T}$, the time to construct $\mathcal{A}$ from twistless embeddings is $O(n)$.

References

[1] P. K. Agarwal, B. Aronov, J. Pach, R. Pollack, and M. Sharir, Quasi-planar graphs have a linear number of edges, Combinatorica, 17(1), pp. 1–9, 1997.

[2] 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.

[3] E. N. Argyriou, M. A. Bekos, and A. Symvonis, The straight-line RAC drawing problem is NP-hard, J. Graph Algorithms Appl., 16(2), pp. 569–597, 2012.

[4] C. Auer, C. Bachmaier, F. J. Brandenburg, A. Gleißner, K. Hanauer, D. Neuwirth, and J. Reislhuber, Recognizing outer 1-planar graphs in linear time, Proc. of GD 2013, pp. 107–118, 2014.

[5] G. Di Battista and R. Tamassia, On-line maintenance of triconnected components with SPQR-trees, Algorithmica, 15, pp. 302–318, 1996.

[6] M. A. Bekos, S. Cornelsen, L. Grilli, S. Hong, and M. Kaufmann, On the recognition of fan-planar and maximal outer-fan-planar graphs, Proc. of Graph Drawing 2014, pp. 198–209, 2015.

[7] M. A. Bender, M. Farach-Colton, G. Pemmasani, S. Skiena, and P. Sumazin, Lowest common ancestors in trees and directed acyclic graphs, J. Algorithms, 57, pp. 75–94, 2005.

[8] C. Binucci, E. Di Giacomo, W. Didimo, F. Montecchiani, M. Patrignani, A. Symvonis, and I. G. Tollis, Fan-planarity: properties and complexity, Theor. Comput. Sci., 589, pp. 76–86, 2015.

[9] J. M. Boyer and W. J. Myrvold, Stop minding your p's and q's: A simplified $O(n)$ planar embedding algorithm, SODA, pp. 140–146, 1999

[10] J. M. Boyer and W. J. Myrvold, On the cutting edge: simplified $O(n)$ planarity by edge addition, J. Graph Algorithms Appl., 8(2), pp. 241–273, 2004

[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, Proc. of WADS 2003, pp. 243–255, 2003.

[12] N. Chiba, T. Nishizeki, A. Abe, and T. Ozawa, A linear algorithm for embedding planar graphs using PQ–trees, Journal of Computer and Systems Sciences, 30, pp. 54–76, 1985.

[13] M. Chrobak and G. Kant, Convex grid drawings of 3-connected planar graphs, Int. J. Comp. Geometry and Applications, 7, pp. 211–223, 1997

[14] W. Didimo, P. Eades, and G. Liotta, Drawing graphs with right angle crossings, Theor. Comput. Sci., 412(39), pp. 5156–5166, 2011.

[15] P. Eades, S-H. Hong, N. Katoh, G. Liotta, P. Schweitzer, and Y. Suzuki, A linear time algorithm for testing maximal 1-planarity of graphs with a rotation system, Theor. Comput. Sci., 513, pp. 65–76, 2013.

[16] A. Estrella-Balderrama, E. Gassner, M. Jűnger, M. Percan, M. Schaefer, and M. Schulz, Simultaneous geometric graph embeddings, Proc. of. Graph Drawing 2007, pp. 280–290, 2008.

[17] H. de Fraysseix and P. Rosenstiehl, A characterization of planar graphs by trémaux orders, Combinatorica, 5(2), pp. 127–135, 1985.

[18] H. de Fraysseix, P. O. de Mendez, and P. Rosenstiehl, Trémaux trees and planarity, Int. J. Found. Comput. Sci., 17(5), pp. 1017–1030, 2006

[19] Q-W. Feng, R. F. Cohen, and P. Eades, Planarity for clustered graphs, Proc. of ESA 1995, pp. 213–226, 1995.

[20] A. Garg and R. Tamassia, On the computational complexity of upward and rectilinear planarity testing, Proc. of Graph Drawing 1994, pp. 286–297, 1995.

[21] A. Grigoriev and H. L. Bodlaender, Algorithms for graphs embeddable with few crossings per edge, Algorithmica, 49(1), pp. 1–11, 2007.

[22] C. Gutwenger, K. Klein, and P. Mutzel, Planarity testing and optimal edge insertion with embedding constraints, J. Graph Algorithms Appl., 12(1), pp. 73–95, 2008.

[23] C. Gutwenger and P. Mutzel, A linear time implementation of SPQR-trees, Proc. of Graph Drawing 2000, Lecture Notes in Computer Science 1984, pp. 77–90, 2001.

[24] D. Harel and R. E. Tarjan, Fast algorithms for finding nearest common ancestors, SIAM J. Computing, 13(2), pp. 338–355, 1984.

[25] S-H. Hong, P. Eades, N. Katoh, G. Liotta, P. Schweitzer, and Y. Suzuki, A linear-time algorithm for testing outer-1-planarity, Algorithmica, 72(4), pp. 1033–1054, 2015.

[26] S-H. Hong and H. Nagamochi, Testing full-outer-2-planarity in linear time, Proc. of WG 2015, to appear.

[27] J. Hopcroft and R. Tarjan, Dividing a graph into triconnected components, SIAM J. on Computing, 2(3), pp. 135–158, 1973.

[28] J. Hopcroft and R. Tarjan, Efficient planarity testing, Journal of the Association for Computing Machinery, 21(4), pp. 549–568, 1974

[29] M. Jűnger, S. Leipert, and P. Mutzel, Pitfalls of using PQ-trees in automatic graph drawing, Prof. of Graph Drawing 1997, pp.193–204 , 1998.

[30] M. Jűnger, S. Leipert, and P. Mutzel, Level planarity testing in linear time, Proc. of Graph Drawing 1998, pp. 224–237, 1999.

[31] M. Kaufmann and T. Ueckerdt, The density of fan-planar graphs, CoRR abs/1403.6184, 2014.

[32] V. P. Korzhik and B. Mohar, Minimal obstructions for 1-immersions and hardness of 1-planarity testing, Journal of Graph Theory, 72(1), pp. 30–71, 2013.

[33] K. Kuratowski, Sur le problème des courbes gauches en topologie., Fund. Math., 15, pp. 271–283, 1930.

[34] A. Lempel, S. Even, and I. Cederbaum, An algorithm for planarity testing of graphs, In P. Rosenstiehl, editor, Theory of Graphs, New York, Gordon and Breach, pp. 215–232 ,1967

[35] K. Mehlhorn and P. Mutzel, On the embedding phase of the Hopcroft and Tarjan planarity testing algorithm, Algorithmica, 16, pp. 233–242, 1996.

[36] J. Pach and G. Toth, Graphs drawn with few crossings per edge, Combinatorica, 17(3), pp. 427–439, 1997.

[37] W.-K. Shih and W.-L. Hsu, In Proceedings of the International Workshop on Discrete Mathematics and Algori A simple test for planar graphs,thms, pp. 110–122, 1993.

[38] W.-K. Shih and W.-L. Hsu, A new planarity test, Theor. Comput. Sci., 223, pp. 179–191, 1999.