Star-shaped Drawings of Graphs with Fixed Embedding and Concave Corner Constraints

Seok-Hee Hong1  Hiroshi Nagamochi2

1School of Information Technologies, University of Sydney, shhong@it.usyd.edu.au
2Department of Applied Mathematics and Physics, Kyoto University, nag@amp.i.kyoto-u.ac.jp

Abstract: A star-shaped drawing of a graph is a straight-line drawing such that each inner facial cycle is drawn as a star-shaped polygon, and the outer facial cycle is drawn as a convex polygon. In this paper, given a biconnected planar graph $G$ with fixed plane embedding and a subset $A$ of corners of $G$, we consider the problem of finding a star-shaped drawing $D$ of $G$ such that only corners in $A$ are allowed to become concave corners in $D$. We first characterize a necessary and sufficient condition for a subset $A$ of corners to admit such a star-shaped drawing $D$. Then we present a linear time algorithm for finding such a star-shaped drawing $D$. Our characterization includes Thomassen’s classical characterization of biconnected plane graphs with a prescribed boundary that have convex drawings [11].

1 Introduction

Graph drawing has attracted much attention over the last twenty years due to its wide range of applications, such as VLSI design, software engineering and bioinformatics. Two or three dimensional drawings of graphs with a variety of aesthetics and edge representations have been extensively studied [2]. One of the most popular drawing conventions is the straight-line drawing, where all the edges of a graph are drawn as straight-line segments. Every planar graph is known to have a planar straight-line drawing [4]. A straight-line drawing is called a convex drawing if every facial cycle is drawn as a convex polygon. Note that not all planar graphs admit a convex drawing.

In general, the convex drawing problem has been well investigated. Tutte [12] showed that every triconnected plane graph with a given boundary drawn as a convex polygon admits a convex drawing using the polygonal boundary. However, not all biconnected planes graphs admit convex drawings. Thomassen [11] gave a necessary and sufficient condition for a biconnected planar graph with a prescribed convex boundary to have a convex drawing. Chiba et al. [1] presented a linear time algorithm for finding a convex drawing (if any) for a biconnected plane graph with a specified convex boundary. Miura et al. [10] gave a linear time algorithm for finding a convex drawing with minimum outer apices for an internally triconnected plane graph. Hong and Nagamochi gave conditions for hierarchical plane graphs to admit a convex drawing [6], and for $e$-planar clustered graphs to admit a convex drawing in which every cluster is also drawn as a convex polygon [7].

However, not much attention has been paid to the problem of finding a convex drawing with a non-convex boundary or non-convex faces. Recently, Hong and Nagamochi [5] proved that every triconnected plane graph with a fixed star-shaped polygon boundary has an inner-convex drawing (a drawing in which every inner face is drawn as a convex polygon), if its kernel has a positive area. Note that this is an extension of the classical result by Tutte [12], since any convex polygon is a star-shaped polygon.

To draw biconnected graphs which do not admit convex drawings in a convex way as much as possible, it is natural to minimize the number of convex faces, concave vertices or concave corners in a drawing. However, Kant [9] already proved the NP-completeness of the problem of

---

deciding whether a biconnected planar/plane graph can be drawn with at most k non-convex faces.

Recently, in our companion paper [8], we initiated a new notion of star-shaped drawing of a graph as a straight-line drawing such that each inner facial cycle is drawn as a star-shaped polygon, and the outer facial cycle is drawn as a convex polygon. Note that there is a biconnected plane graph which needs a concave corner in any of its straight-line drawings (including outer apices as concave corners). We proved that, given a biconnected planar graph \( G \), a star-shaped drawing of \( G \) with the minimum number of concave corners can be found in linear time, where we are allowed to choose the plane embedding and the concave corners of \( G \), based on the effective use of lower bounds [8].

In this paper, we deal with a start-shaped drawing of graphs with two given constraints: a fixed plane embedding constraints and a set of concave corner constraints.

Let \( G \) be a biconnected plane graph. We denote a corner \( \lambda \) around a vertex \( v \) by pair \((v, f)\) of the vertex \( v \) and the facial cycle \( f \) whose interior contains the corner. Let \( \Lambda(v) \) denote the set of all corners around a vertex \( v \) in \( G \), and \( \Lambda(G) \) denote the set of all corners in \( G \). For a straight-line drawing \( D \) of a plane graph \( G \), let \( \Lambda^c(D) \) denote the set of concave corners in \( D \). For a given subset \( A \subseteq \Lambda(G) \), we consider the problem of whether \( G \) has a star-shaped drawing \( D^* \) such that \( \Lambda^c(D^*) \subseteq A \). In this paper, we characterize when \( G \) admits such a drawing \( D^* \), and give a linear time algorithm for testing the conditions in the characterization.

A corner \((v, f^o)\) of a vertex \( v \) in the outer facial cycle \( f^o \) is an outer corner of \( f^o \). We denote \( \Lambda^o(f^o(G)) \) the set of the outer corners of the outer facial cycle \( f^o(G) \). We call a cycle \( C \) in \( G \) a cut-cycle if a cut-pair \( \{u, v\} \subseteq V(C) \) separates the vertices outside \( C \) from those along \( C \) (including those inside \( C \)). A corner \((v, f)\) of a vertex \( v \) in a cut-cycle \( C \) is an outer corner of \( C \) if \( v \) is not in the cut-pair of \( C \), and \( f \) is one of the two facial cycles outside \( C \) that share the cut-pair of \( C \). We denote by \( \Lambda^o(C) \) the set of the outer corners of a cut-cycle (or the outer facial cycle) \( C \), and \( \Lambda^o(G) \) denote the set of all outer corners in \( G \). For example, \( C_1 = (u_6, u_9, u_2, u_{18}, u_8) \) of graph \( G \) in Fig. 1 is a cut-cycle, where \( \Lambda^o(C_1) = \{(u_9, f_{12}), (u_{18}, f_9), (u_8, f_9)\} \) (see also Fig. 3(a)). More formally, we prove the next result.

**Theorem 1** Let \( G \) a biconnected plane graph \( G \), \( f^o \) be the outer facial cycle of \( G \), and \( A \) be a subset of \( \Lambda(G) \).

(i) There exists a straight-line drawing \( D \) of \( G \) such that \( \Lambda^c(D) \subseteq A \) if and only if \[
\left| A \cap \Lambda^o(f^o) \right| \geq 3 \quad \text{and} \quad \left| A \cap \Lambda^o(C) \right| \geq 1 \quad \text{for all cut-cycle} \ C \text{ in} \ G.
\]

Testing whether \( A \) satisfies the condition or not can be done in linear time.

(ii) Let \( A \) satisfy the condition in (i), and \( f^o \) be drawn as a convex polygon \( P \) with \( \Lambda^c(P) = A \cap \Lambda^o(f^o) \). Then \( P \) can be extended to a star-shaped drawing \( D^* \) of \( G \) with \( \Lambda^c(D^*) \subseteq A \cap \Lambda^o(G) \). Such a drawing \( D^* \) can be obtained in linear time.

From the above, we see that, for any straight-line drawing \( D \), there exists a star-shaped drawing \( D^* \) of \( G \) with \( \Lambda^c(D^*) \subseteq \Lambda^c(D) \cap \Lambda^o(G) \subseteq \Lambda^c(D) \). Theorem 1 also implies that any convex polygon \( P \) drawn for the boundary of a triconnected plane graph \( G \) can be extended to a convex drawing of \( G \), since \( G \) has no cut-cycle. Moreover, Theorem 1 also contains as special cases Thomassen’s characterization of biconnected plane graphs that have convex drawings [11], and the characterization of convex drawings with minimum outer apices by Miura et al. [10].

This paper is organized as follows. Section 2 reviews basic terminology and background. Section 3 characterizes “proper drawings,” and show how to check the necessity of Theorem 1 efficiently. Section 4 shows the sufficiency of Theorem 1 by using a divide-and-conquer algorithm for constructing a desired star-shaped drawing. Section 5 makes concluding remarks.
2 Preliminaries

Throughout the paper, a graph \( G = (V, E) \) stands for a simple undirected graph. The set of vertices and set of edges of a graph \( G \) may be denoted by \( V(G) \) and \( E(G) \), resp. The set of edges incident to a vertex \( v \in V \) is denoted by \( E(v) \). The degree of a vertex \( v \) in \( G \) is denoted by \( d_G(v) \) (i.e., \( d_G(v) = |E(v)| \)). For a subset \( X \subseteq E \) (resp., \( X \subseteq V \)), let \( G - X \) denote the graph obtained from \( G \) by removing the edges in \( X \) (resp., the vertices in \( X \) together with the edges in \( \cup_{v \in X} E(v) \)).

2.1 Plane Graphs and Biconnected Plane Graphs

A graph \( G \) is called planar if its vertices and edges are drawn as points and curves in the plane so that no two curves intersect except for their end points, where no two vertices are drawn at the same point. In such a drawing, the plane is divided into several connected regions, each of which is called a face. A face is characterized by the cycle of \( G \) that surrounds the region. Such a cycle is called a facial cycle. A plane embedding of a planar graph \( G \) consists of an ordering of edges around each vertex and the outer face. A planar graph with a fixed plane embedding is called a plane graph. Let \( f^o(G) \) denote the outer facial cycle of a plane graph \( G \), and \( V^o(G) \) denote \( V(f^o(G)) \). The set of faces of a plane graph \( G \) is denoted by \( F(G) \). A vertex (resp., an edge) in the outer facial cycle is called an outer vertex (resp., an outer edge), while a vertex (resp., an edge) not in the outer facial cycle is called an inner vertex (resp., an inner edge).

Let \( G = (V, E, F) \) be a biconnected plane graph. Let \( C \) be a cut-cycle and \( \{u, v\} \in V(C) \) be the cut-pair that separates the vertices in \( V(C) - \{u, v\} \) and inside \( C \) from the vertices outside \( C \) in \( G \). We consider a subgraph \( H \) of \( G \) such that the boundary \( f^o(H) \) is a cut-cycle in \( G \), where we treat \( H \) as a plane graph under the same embedding of \( G \). For such a plane graph \( H \), we define the \( u, v \)-boundary path \( f^o_{uv}(H) \) of \( H \) to be the path obtained by traversing the boundary \( f^o(H) \) of \( H \) from \( u \) to \( v \) in the clockwise order. We denote \( V(f^o_{uv}(H)) - \{u, v\} \) by \( V^o_{uv}(H) \), and denote by \( \Lambda^o_{uv}(H) \) the set of outer corners of \( f^o_{uv}(H) \), i.e., \( \Lambda^o_{uv}(H) = \Lambda^o(f^o(H)) \cap (\bigcup_{u \in V^o_{uv}(H)} \Lambda^o(u)) \). For example, cut-cycle \( C_1 = (u_6, u_9, u_2, u_{18}, u_8) \) in Fig. 1 has subgraph \( H_1 \) with edges \((u_6, u_9), (u_9, u_2), (u_9, u_{10}), (u_{10}, u_6), (u_2, u_{18}), (u_{18}, u_8)\), \((u_8, u_6)\) such that \( C_1 = f^o(H_1) \), where \( \Lambda^o_{u_9, u_2}(C_1) = \{(u_9, f_1)\} \) and \( \Lambda^o_{u_2, u_9}(C_1) = \{(u_{18}, f_9), (u_8, f_9)\} \) hold.

For a cut-pair \( \{u, v\} \) of a biconnected plane graph \( G \), a \( u, v \)-component \( H \) is a connected subgraph of \( G \) that either consists of a single edge \((u, v)\) or is a maximal subgraph such that \( H - \{u, v\} \) remains connected. We may treat a \( u, v \)-component \( H \) of a plane graph \( G \) as a plane graph under the same embedding of \( G \). In this case, the boundary \( f^o(H) \) of \( H \) is a cut-cycle. For example, the subgraph \( H \) consisting of edges \((u_6, u_9), (u_9, u_2), (u_9, u_{10}), (u_{10}, u_6), (u_2, u_{18}), (u_{18}, u_8)\), \((u_8, u_6)\) in graph \( G \) has no such \( u, v \)-component \( H \). A simple path with end vertices \( u \) and \( v \) of a graph \( G \) is called a \( u, v \)-path, and is called an induced \( u, v \)-path if every internal vertex (i.e., non end vertex) is of degree 2.

A biconnected plane graph \( G \) is called internally triconnected if, (i) for each inner vertex \( v \) with \( d_G(v) \geq 3 \), there exist three paths disjoint except for \( v \), each connecting \( v \) and an outer vertex; and (ii) every cycle of \( G \) which has no outer edge has at least three vertices \( v \) with \( d_G(v) \geq 3 \).

Define the contracted graph \( G' \) of a biconnected plane graph \( G \) as the graph obtained from \( G \) by replacing each \( u, v \)-induced path \( Q \) in \( G \) with a single edge, where we replace \( Q \) with a path of length two if \((u, v) \in E \) and \( E(Q) \subseteq E(f^o(G)) \). Then, \( G \) is internally triconnected if and only if \( G' \) has no multiple edges and for every cut-pair \( \{u, v\} \) in \( G' \), \( u, v \in V^o(G') \) holds and each component in \( G' - \{u, v\} \) contains an outer vertex. In other words, \( G \) is internally
triconnected if and only if it is a biconnected graph in which

\[ |\Lambda^\circ(C) \cap \Lambda^\circ(f^\circ(G))| \geq 1 \]

holds for every cut-cycle \( C \) in \( G \).

(2)

From this, we can observe that, for two distinct cut-cycles \( C \) and \( C' \) sharing an outer edge, either \( \Lambda^\circ(C) \cap \Lambda^\circ(f^\circ(G)) \subset \Lambda^\circ(C') \cap \Lambda^\circ(f^\circ(G)) \) or \( \Lambda^\circ(C') \cap \Lambda^\circ(f^\circ(G)) \subset \Lambda^\circ(C) \cap \Lambda^\circ(f^\circ(G)) \) holds. In an internally triconnected plane graph \( G \), a cut-cycle \( C \) is minimal if there is no other cut-cycle \( C' \) with \( \Lambda^\circ(C') \cap \Lambda^\circ(f^\circ(G)) \subset \Lambda^\circ(C) \cap \Lambda^\circ(f^\circ(G)) \).

2.2 The SPQR tree of a Biconnected Planar Graph

The SPQR tree of a biconnected planar graph \( G \) represents the decomposition of \( G \) into the triconnected components \([3]\). Each node \( \nu \) in the SPQR tree is associated with a graph \( \sigma(\nu) = (V_\nu, E_\nu) \) \( (V_\nu \subseteq V) \), called the skeleton of \( \nu \). In fact, we use a modified SPQR tree without Q-nodes. Thus, there are three types of nodes in the SPQR tree: S-node (the skeleton is a simple cycle), P-node (the skeleton consists of two vertices with at least 3 edges), and R-node (the skeleton is a triconnected simple graph). Fig. 4 shows the SPQR tree of the biconnected planar graph in Fig. 1.

We treat the SPQR tree as a rooted tree \( T \) by choosing a node \( \nu^* \) as its root. For a node \( \nu \), let \( Ch(\nu) \) denote the set of all children of \( \nu \) and \( \mu \) be the parent of \( \nu \). The graph \( \sigma(\mu) \) has exactly one virtual edge \( e \) in common with \( \sigma(\nu) \). The edge \( e \) is called the parent virtual edge \( parent(\nu) \) of \( \sigma(\nu) \), and the child virtual edge of \( \sigma(\mu) \). We define a parent cut-pair of \( \nu \) as the two end points of \( parent(\nu) \). We denote the graph formed from \( \sigma(\nu) \) by deleting its parent virtual edge as \( \sigma^- (\nu) = (V_\nu, E_\nu) \), \( (E_\nu^- = E_\nu - \{parent(\nu)\}) \). Let \( G^-(\nu) \) denote the subgraph of \( G \) which consists of the vertices and real edges in the graphs \( \sigma^- (\mu) \) for all descendants \( \mu \) of \( \nu \), including \( \nu \) itself.

Consider the case where \( G \) is a plane graph. For a face \( f \in F \) of \( G \), we say that a node \( \nu \) in the SPQR tree is incident to \( f \) if \( \sigma(\nu) \) contains the face corresponding to \( f \) (not that there may exist more than one such node \( \nu \)). We choose a node incident to the outer face \( f^\circ(G) \) as the root \( \nu^* \) of the SPQR tree. In particular, we choose a P- or S-node incident to \( f^\circ(G) \) (if any) as the root \( \nu^* \), and choose an R-node incident to \( f^\circ(G) \) only when there is no such P- or S-node. Hence, we can assume that if the root \( \nu^* \) is an R-node then no outer edge in \( \sigma(\nu^*) \) is a virtual edge. When \( G \) is a plane graph, we also treat graphs \( \sigma^- (\nu) \) and \( G^-(\nu) \) as plane graphs induced from the embedding of \( G \). For a non-root node \( \nu \) with \( (u, v) = parent(\nu) \), two plane drawings for \( \sigma(\nu) \) can be obtained from the plane graph \( \sigma^- (\nu) \) by drawing the parent edge \( e = (u, v) \) outside \( \sigma^- (\nu) \); one has \( f^\circ(\sigma^- (\nu)) \) plus \( e \) as its boundary, and the other has \( f^\circ(\sigma^- (\nu)) \) plus \( e \) as its boundary, where we denote the former and latter plane graphs by \( \sigma_{uv}(\nu) \) and \( \sigma_{em}(\nu) \), respectively. Fig. 2 illustrates examples of \( \sigma_{uv}(\nu) \) and \( \sigma_{em}(\nu) \) of an R-node \( \nu \).

If a subgraph \( H \) of \( G \) is the graph \( G^-(\nu) \) for a node \( \nu \) in the SPQR tree and \( (u, v) = parent(\nu) \), then we denote \( V^\circ_{uv}(H) \) by \( V^\circ_{uv}(\nu) \), \( \Lambda^\circ_{uv}(H) \) by \( \Lambda^\circ_{uv}(\nu) \), \( V^\circ_{uv}(H) \cup V^\circ_{em}(\nu) \) by \( V^\circ(\nu) \), and \( \Lambda^\circ_{uv}(H) \cup \Lambda^\circ_{em}(\nu) \) by \( \Lambda^\circ(\nu) \), respectively. For the root \( \nu^* \), let \( V^\circ(\nu^*) = V(f^\circ(G)) \) and \( \Lambda^\circ(\nu^*) = \Lambda^\circ(f^\circ(G)) \). For example, R-node \( \eta \) in Fig. 2 has \( \Lambda^\circ_{uv}(\eta) = \{(u_{13}, f_7)\} \) and \( \Lambda^\circ_{uv}(\eta) = \{(u_{14}, f_1)\} \).

2.3 Straight-line Drawings, Convex Drawings and Star-Shaped Drawings

For two points \( p_1, p_2 \) in the plane, \( [p_1, p_2] \) denotes the line segment with end points \( p_1 \) and \( p_2 \), and for three points \( p_1, p_2, p_3 \), \( [p_1, p_2, p_3] \) denotes the triangle with three corners \( p_1, p_2, p_3 \). A kernel \( K(P) \) of a polygon \( P \) is the set of all points from which all points in \( P \) are visible. The boundary of a kernel, if any, is a convex polygon. A polygon \( P \) is called star-shaped if \( K(P) \neq \emptyset \).
A straight-line drawing $D$ of a graph $G = (V, E)$ in the plane is an embedding of $G$ in the two dimensional space $\mathbb{R}^2$, such that each vertex $v \in V$ is drawn as a point $\tau_D(v) \in \mathbb{R}^2$, and each edge $(u, v) \in E$ is drawn as a straight-line segment $[\tau_D(u), \tau_D(v)]$, where $\mathbb{R}$ is the set of reals. Let $D$ be a straight-line planar drawing of a biconnected plane graph $G$. A corner of $G$ is called concave in $D$ if its angle in $D$ is greater than $\pi$. A vertex $v$ in a straight-line drawing $D$ is called concave if one of the corners around $v$ is concave in $D$. For a straight-line drawing $D$ of a biconnected plane graph $G$, let $\Lambda'(D)$ denote the set of all concave corners in $D$.

A star-shaped drawing of a plane graph is a straight-line drawing such that each inner facial cycle is drawn as a star-shaped polygon and the outer facial cycle is drawn as a convex polygon. An outer vertex in a straight-line drawing of a plane graph is called an apex if it is concave in the drawing and its concave corner appears in the outer face. Fig. 1(b) shows a star-shaped drawing of the plane graph $G$ in Fig. 1(a), where $(u_1, f^0), (u_4, f_{13}), (u_6, f_8), (u_7, f^0), (u_8, f_0), (u_{10}, f_{10}), (u_{11}, f^0), (u_{14}, f_1), (u_{15}, f_5)$, and $(u_{17}, f_1)$ are the concave corners and $u_1, u_7$, and $u_{11}$ are the apices.

A straight-line drawing $D$ of a plane graph $G = (V, E, F)$ is called a convex drawing, if every facial cycle is drawn as a convex polygon, equivalently if $\Lambda'(D) = \emptyset$. We say that a drawing $D$ of a graph $G$ is extended from a drawing $D'$ of a subgraph $G'$ of $G$, if $\tau_D(v) = \tau_D(v)$ for all $v \in V(G')$. A convex polygon drawn for the outer facial cycle in a biconnected plane graph can be extended to a convex drawing when the next condition holds.

**Theorem 2** [1, 11] Let $G = (V, E, F)$ be a biconnected plane graph. Then a drawing $D$ of $f^0(G)$ on a convex polygon $P$ can be extended to a convex drawing of $G$ if and only if:

(i) $G$ is internally triconnected; and

(ii) Let $Q_1, Q_2, \ldots, Q_k$ be the subpaths of $f^0(G)$, each corresponding to a side of $P$. The graph $G - V^0(G)$ has no component $H$ such that all the outer vertices adjacent to vertices in $H$ are contained in a single path $Q_l$, and there is no inner edge $(u, v)$ whose end vertices are contained in a single path $Q_l$.

We can observe that Theorem 1 contains Theorem 2 as a special case. Let $A$ be the set of the outer corners of $f^0(G)$ that correspond to the apices of a given convex polygon $P$ in Theorem 2. By observation (2), the pair of conditions (i) and (ii) in Theorem 2 is equivalent to condition (1) in Theorem 1(i). Then Theorem 1(ii) tells that if (1) holds, $P$ can be extended to a star-shaped drawing $D'$ with $\Lambda'(D') - A = \emptyset$ of $G$, i.e., a convex drawing of $G$.

**Theorem 3** [10] Let $G$ be an internally triconnected plane graph, $T$ be the SPQR tree of the contracted graph $G'$ of $G$, and $n_\ell$ be the number of leaves of $T$. Then one can find a convex drawing of $G$ having the minimum number of outer apices in linear time, and the minimum number is equal to $\text{max}\{3, n_\ell\}$.

We informally show that Theorem 1 implies Theorem 3. Each leaf in the SPQR tree $T$ of $G'$ corresponds to a minimal cut-cycle $C$ in $G$. Hence $n_\ell$ is the number of minimal cut-cycles. Let $D$ be a convex drawing of $G$. By Theorem 1(i), $\Lambda'(D)$ must contain at least three corners from $\Lambda^0(f^0(G))$ and at least one corner from $\Lambda^0(C') \cap \Lambda^0(f^0(G))$ of each minimal cut-cycle. Since two minimal cut-cycles $C$ and $C'$ satisfy $\Lambda^0(C') \cap \Lambda^0(C) = \emptyset$, we see that the number of concave corners in a convex drawing is at least $\text{max}\{3, n_\ell\}$. Conversely, we choose a set $A$ of $\text{max}\{3, n_\ell\}$ corners of $G$ such that $A$ contains at least one from $\Lambda^0(C') \cap \Lambda^0(f^0(G))$ of each minimal cut-cycle and at least three corners from $\Lambda^0(f^0(G))$. Then by the minimality of cut-cycles, $A \cap \Lambda^0(C) \neq \emptyset$ holds for all cut-cycles. Hence, condition (1) in Theorem 1 holds, and thereby $G$ has a straight-line drawing $D$ with $\Lambda'(D) \subseteq A \subseteq \Lambda^0(f^0(G))$ (a convex drawing). The SPQR tree can be computed in linear time [3]. By Theorem 1(ii), we can construct a convex drawing $D$ with the minimum number of outer apices in linear time.
3 Proper Drawings and Configurations

This section investigates the structural property of the necessary condition (1) in Theorem 1.

**Lemma 4** Let $G$ be a biconnected plane graph. Condition (1) is necessary for a subset $A \subseteq \Lambda(G)$ to admit a straight-line drawing $D$ of $G$ such that $\Lambda'(D) \subseteq A$.

**Proof:** See Appendix 1.

To show that (1) is also a sufficient condition, we define *proper drawings*, a restricted class of straight-line drawings of a given plane graph $G$. Let $D$ be a straight-line drawing of $G$, and $\nu$ be a P-node of the SPQR tree of $G$, where $(u, v)$ denotes $\text{parent}(\nu)$. Then we say that an edge $e$ in the skeleton $\sigma(\nu)^{-}$ intersects with P-node $\nu$ in $D$ if the drawing of $G^{-}(\mu)$ for the child node $\mu \in Ch(\nu)$ corresponding to the virtual edge $e$ (or the drawing of the real edge $e$) intersects with line-segment $[u, v]$ (if we additionally draw $[u, v]$). We call such an edge the *central edge*.

We call a straight-line drawing $D$ *proper* if for each P-node $\nu$, at most one edge in $\sigma(\nu)^{-}$ intersects with $\nu$. We call the set $\psi$ of all central edges in a proper straight-line drawing $D$ the *configuration* of $D$. Such a drawing $D$ is also called *$\psi$-proper*. Fig. 3(b) illustrates a drawing of $G^{-}(\nu)$ in which two edges in $\sigma(\nu)^{-}$ intersect with $\nu$, while Fig. 3(c) illustrates a drawing of $G^{-}(\nu)$ in which only one edge in $\sigma(\nu)^{-}$ intersects with $\nu$. We will show that a configuration determines the structure of concave vertices in a proper straight-line drawing.

![Figure 1](image1.png)

Figure 1: (a) A biconnected plane graph $G$ (b) A star-shaped drawing $D$ of $G$.

![Figure 2](image2.png)

Figure 2: (a) Subgraph $G^{-}(\nu_1)$ of R-node $\nu_1$ in the SPQR tree of $G$ (b) Skeleton $\sigma^{-}(\nu_1)$ of R-node $\nu_1$ (c) Skeleton $\sigma_{u_1, u_12}(\nu_1)$ of R-node $\nu_1$ (d) Skeleton $\sigma_{u_12, u_1}(\nu_1)$ of R-node $\nu_1$. 

6
In the next section, we will prove that if (1) holds, then there exists a proper star-shaped
drawing $D$ of $G$ such that $A'(D) \subseteq A$. In this section, we show how to efficiently test whether
a given subset $A \subseteq A(G)$ satisfies (1). For this, we investigate the structural property of (1) in
terms of the SPQR tree of $G$.

For convenience, we fix a total ordering of all vertices in a given graph $G$, and we denote
by $u < v$ if $u$ is smaller than $v$ in the ordering. For the parent cut-pair $\{u, v\}$ of a P-node $\nu$,
where we assume $u < v$, we number the child edges of $\nu$ as $e_1, e_2, \ldots, e_{k(\nu)}$ by traversing these
edges from left to right, where $k(\nu)$ denotes the number of children of $\nu$ (see Fig. 5(a)).

![Figure 3](image1.png)

Figure 3: (a) Subgraph $G^-(\nu_g)$ of P-node $\nu_g$ in the SPQR tree of $G$ (b) A straight-line drawing
of $G^-(\nu)$ in which two edges $e_1, e_2 \in E_{\nu_1}$ intersect $\nu_1$ (c) A straight-line drawing of $G^-(\nu)$ in
which only edge $e_2 \in E_{\nu_1}$ intersects $\nu_1$ (d) Skeleton $\sigma^-(\nu_g)$ of P-node $\nu_g$.

![Figure 4](image2.png)

Figure 4: The SPQR tree of the biconnected planar graph $G$ in Fig. 1.

We are ready to formally define “central edge” and “configuration.” For each P-node $\nu$ of
$T$, we choose an edge $e_j$ ($1 \leq j^* \leq k(\nu)$) of $\sigma^-(\nu)$, which we call the central edge of $\nu$ and
denote it by $c(\nu)$. If $\nu$ has a real edge among its children, then we always choose the real edge
as $c(\nu)$. Other virtual edge $e_i$ in $\sigma^-(\nu)$ is called a left edge (resp., right edge) if $i < j^*$ (resp.,
$i > j^*$). For each child $\mu_i \in Ch(\nu)$ corresponding to edge $e_i$, we call subgraph $G^-(\mu_i)$ with
$i < j^*$ (resp., $i > j^*$) a left component (resp., a right component) of $\nu$. If $e_j$ is a real edge in
$\sigma^-(\nu)$, then the edge is called the central component of $\nu$; otherwise subgraph $G^-(\mu_{j^*})$ is called
the central component of $\nu$.

A configuration of a node $\nu$ is defined by a set $\psi$ of central edges $c(\nu')$ for all descendants
$\nu'$ of $\nu$, and let $\Psi(\nu)$ denote the set of all configurations $\psi$ of $\nu$. Thus, for the root $\nu^*$, a
configuration $\psi \in \Psi(\nu^s)$ defines a central edge for each P-node in the SPQR tree of $G$. Now we characterize a subset $A \subseteq \Lambda^o(G)$ that satisfies (1) using the SPQR tree.

**Definition 5** For a biconnected plane graph $G$, and a configuration $\psi \in \Psi(\nu^s)$, let $A$ be a subset of $\Lambda^o(G)$, and $\nu$ be a node in the SPQR-tree.

(i) Let $\nu$ be the root $\nu^s$. Subset $A$ is called proper with respect to $\nu^s$ if $|A \cap \Lambda^o(\nu^s)| \geq 3$;

(ii) Let $\nu$ be an R-node which is a child of an R-node or S-node, and $(u, v) = \text{parent}(\nu)$. Subset $A$ is called proper with respect to $\nu$ if $A \cap \Lambda^o(\nu) \neq \emptyset$;

(iii) Let $\nu$ be a non-root P-node, $(u, v) = \text{parent}(\nu)$, and $E^-_\nu = \{e_1, e_2, \ldots, e_k(\nu)\}$, where $e_1, e_2, \ldots, e_k(\nu)$ appear from left to right and $e_{j^*} = \epsilon(\nu) \in \psi$. Let $\mu_j$ denote the child of $\nu$ corresponding to a virtual edge $e_j$. Subset $A$ is called proper with respect to $\nu$ if it satisfies the following:

\[
A \cap \Lambda^o_{\nu^s}(\mu_j) \neq \emptyset \quad (1 \leq j \leq j^* - 1), \quad A \cap \Lambda^o_{\nu^s}(\mu_j) \neq \emptyset \quad (j^* + 1 \leq j \leq k(\nu)),
\]

and if $\mu_{j^*}$ is not an induced path, then $A \cap \Lambda^o(\mu_{j^*}) \neq \emptyset$ (see Fig. 5(b)).

A subset $A \subseteq \Lambda^o(G)$ is called proper with respect to $(\nu^s, \psi)$ if $A$ is proper with respect to all nodes $\nu$ in (i)-(ii) of Definition 5, and all P-nodes $\nu$ in (iii) of Definition 5.

**Lemma 6** Let $G$ be a biconnected plane graph, $\nu^s$ be the root of the SPQR-tree, and $A$ be a subset of $\Lambda^o(G)$. Then $A$ satisfies (1) if and only if there is a configuration $\psi \in \Psi(G)$ such that $A$ is proper with respect to $(\nu^s, \psi)$.

**Proof:** See Appendix 2.

**Lemma 7** Let $G$ be a biconnected plane graph, and $A$ be a subset of $\Lambda^o(G)$. Then testing whether $A$ satisfies (1) or not can be done in linear time.

**Proof:** See Appendix 3.

We define **fringe corners** as those corners which are used as concave corners of any $\psi$-proper straight-line drawing. For a given configuration $\psi \in \Psi(\nu^s)$, Lemma 6 tells how to characterize “fringe corners.” Given a node $\nu$ with a configuration $\psi \in \Psi(\nu^s)$, we define fringe corners in $G^-(\nu)$ as follows. Let $\mu$ be an arbitrary descendents of $\nu$ (including $\mu = \nu$).
Definition 8 (1) \( \mu \) is an R-node that is not a child of a P-node: Let \( C = f^\circ(G^-(\mu)) \) of graph \( G^-(\mu) \) if \( \mu \) is not the root (where \( C \) is a cut-cycle of \( G \)); let \( C = f^\circ(G) \) if \( \mu \) is the root. Then any outer corner of the cut-cycle \( C \) is defined as a fringe corner of \( \nu \).

(2) \( \mu \) is a P-node: Then a corner \( \lambda \) in graph \( G^-(\mu) \) is called a fringe corner of \( \nu \) if it satisfies one of the following: \( \lambda \in \Lambda^\circ(H) \) for the central component \( H \) of \( \mu \); \( \lambda \in \Lambda^\circ_{\text{in}}(H) \) for a right component \( H \) of \( \mu \); and \( \lambda \in \Lambda^\circ_{\text{ex}}(H) \) for a left component \( H \) of \( \mu \).

For a node \( \nu \) and a configuration \( \psi \in \Psi(\nu) \), we denote by \( \Lambda^\circ(\nu, \psi) \) the set of all fringe corners of \( \nu \). Note that \( \Lambda^\circ(\nu, \psi) \subseteq \Lambda^\circ(G) \) holds.

Lemma 9 Let \( G \) be a biconnected plane graph, \( \nu^* \) be the root of the SPQR-tree, and \( D \) be a straight-line drawing of \( G \). Then there is a configuration \( \psi \in \Psi(G) \) such that \( A = \Lambda^\circ(D) \cap \Lambda^\circ(\nu^*, \psi) \) is proper with respect to \( (\nu^*, \psi) \).

Proof: See Appendix 4.

Lemmas 4 and 7 imply the necessity and the time complexity in Theorem 1(i).

We now prove Theorem 1(ii), which also implies the sufficiency of Theorem 1(i).

4 Constructing Star-shaped Drawings

Conversely, we can prove that, given a proper subset \( A \), there exists a star-shaped drawing \( D \) such that \( \Lambda^\circ(D) = A \).

Theorem 10 Let \( G \) be a biconnected plane graph, \( \nu^* \) be the root of the SPQR-tree of \( G \), and \( \psi \in \Psi(\nu^*) \) be a configuration of \( G \). For any subset \( A \subseteq \Lambda^\circ(\nu^*, \psi) \) proper with respect to \( (\nu^*, \psi) \), a convex polygon \( P \) with \( \Lambda^\circ(P) = A \cap \Lambda^\circ(\nu^*) \) drawn for \( f^\circ(G) \) can be extended to a \( \psi \)-proper star-shaped drawing \( D \) such that \( \Lambda^\circ(D) = A \). Such a drawing \( D \) can be constructed in linear time.

Observe that Theorem 1(ii) follows from Theorem 10. Note that Theorem 10 is stronger than Theorem 1(ii) in the sense that, for a fixed configuration \( \psi \in \Psi(G) \), the set \( \Lambda^\circ(D) \) of concave corners is completely determined by a prescribed set \( A \subseteq \Lambda^\circ(\nu^*, \psi) \) of corners.

To prove Theorem 10, we design a divide-and-conquer algorithm that computes a star-shaped drawing for a subset \( A \subseteq \Lambda^\circ(\nu^*, \psi) \) in Theorem 10 in a top-down manner along the SPQR tree \( T \). This is a modification of the algorithm for constructing a star-shaped drawing of a plane graph with a set of “concave vertices” [8]. A full description of the algorithm is given in Appendix 5. We here explain the main procedures to construct a star-shaped drawing recursively:

1. For the rool \( \nu^* \), we first fix the boundary \( B_{\nu^*} \) of \( G \) as a prescribed \( |A \cap \Lambda^\circ(\nu^*)| \)-gon \( P \) with \( \Lambda^\circ(P) = A \cap \Lambda^\circ(\nu^*) \). We then start to process all nodes \( \nu \) in \( T \) from the root to the leaves by repeatedly computing a drawing \( D_{\nu} \) of skeleton \( \sigma^-(\nu) \) (or \( \sigma(\nu) \)), where we first fix the boundary \( B_{\nu} \) of \( G^-(\nu) \) (or \( G(\nu) \)) and then extend \( B_{\nu} \) to a convex drawing \( D_{\nu} \) of the skeleton \( \sigma^-(\nu) \) (or \( \sigma(\nu) \)). The line segments in \( D_{\nu} \) for virtual edges will be replaced with new drawings \( D_{\mu} \) of the nodes \( \mu \) corresponding to the virtual edges, where a virtual edge in the boundary of the skeleton may be a sequence of curve segments which forms part of the convex boundary \( B_{\nu} \).

2. When we process a non-root R-node \( \nu \) whose parent is an R-node, we fix the boundary of \( G^-(\nu) \) as an \((|A \cap \Lambda^\circ(\nu)| + 2)|-gon \( B_{\nu} \), and then compute a convex drawing \( D_{\nu} \) of skeleton \( \sigma^-(\nu) \) as an extension of \( B_{\nu} \) (we use the linear time convex drawing algorithm of Chiba et al. [1] to compute such a convex drawing).
3. When we process a non-root R-node \( \nu \) whose parent is an S-node (resp., P-node), we compute a convex drawing \( D_\nu \) of skeleton \( \sigma^- (\nu) \) (resp., \( D_\nu \) of skeleton \( \sigma (\nu) \)), where the boundary of the skeleton has been fixed as a convex polygon \( B_\nu \), and we extend \( B_\nu \) to such a drawing \( D_\nu \).

4. When we process a non-root S-node \( \nu \) whose parent is an R-node (resp., a P-node), the virtual edge corresponding to \( \nu \) is drawn as a line segment \( L_\nu \) (resp., a convex polygon \( B_\nu \)), and we then compute an \( (|A \cap \Lambda^o(\mu)| + 2) \)-gon \( B_\mu \) of its child node \( \mu \in Ch(\nu) \), and replace \( L_\nu \) with (resp., extend \( B_\nu \) to) the sequence of these convex polygons \( B_\mu \), \( \mu \in Ch(\nu) \).

5. When we process a non-root P-node \( \nu \) with \( (u, \nu) = parent(\nu) \), the boundary of \( G^- (\nu) \) has been fixed as \( B_\nu \), and we fix the boundary \( G^- (\mu) \) of each child node \( \mu \in Ch(\nu) \) as an \( (|A \cap \Lambda^o(\mu)| + 2) \)-gon, \( (|A \cap \Lambda^o(\mu)| + 2) \)-gon \( B_\mu \) if \( \mu \) corresponds to a left, right, and central edge of \( \nu \) with respect to configuration \( \psi \), respectively.

6. When new inner faces are introduced after computing a drawing \( D_\nu \) of \( \sigma^- (\nu) \) (or \( \sigma (\nu) \)) of an R-node \( \nu \), we choose a point \( r_f \) inside each new face \( f \) as the view point of \( f \), which will be kept as a visible point in the kernel of the face \( f \) until a final drawing is obtained.

7. A convex polygon \( B_\nu \) for a node \( \nu \) will be chosen so that the view point(s) \( r_f \) of the face(s) adjacent to the corresponding virtual edge remain visible from everywhere in the face(s).

The correctness follows from the property that desired polygons \( B_\nu \) in 1, \( B_\nu \) in 2, \( B_\mu \) in 4, and \( B_\mu \) in 5 exist (i.e., \( |A \cap \Lambda^o(\nu^*)| \geq 3 \) holds for \( B_\nu \) in 1, \( |A \cap \Lambda^o(\nu)| \geq 1 \) holds for \( B_\nu \) in 2, and so on). This will be guaranteed by the fact that \( A \subseteq \Lambda^o(\nu^*, \psi) \) is proper. Since we can show that a boundary of \( B_\nu \) of a node \( \nu \) can be fixed in linear time of the size of \( B_\nu \), the entire algorithm can be implemented to run in \( O(|V| + |E|) \) time. This proves Theorem 10.

5 Concluding Remarks

In this paper, we defined star-shaped drawings for biconnected plane graphs as an extension of convex drawings of triconnected plane graphs, and derived a characterization for a subset \( A \) of corners to admit a star-shaped drawing. The characterization includes Thomassen’s characterization of biconnected plane graphs that admits convex drawings as a special case. We also designed a linear-time algorithm to construct a star-shaped drawing for a given set \( A \) of corners.

By modifying the dynamic programming algorithm used in the proof of Lemma 7, it is possible to design a linear time algorithm for finding a smallest subset \( A' \) of a given set \( A \) of corners such that \( A' \) remains to admit a star-shaped drawing. Thus, we can find a star-shaped drawing with the minimum number of concave corners for a given plane graph.

It is left open to investigate the complexity status of the problem of finding a straight-line drawing \( D \) of a triconnected plane graph \( G \) with \( \Lambda^e(D) = A \) for a prescribed set \( A \) of corners. Note that a configuration for a triconnected plane graph \( G \) is empty. So we need to establish a new argument to deal with this problem.

References

Appendix 1: Proof of Lemma 4

Let $D$ be a straight-line drawing of $G$ such that $\Lambda^r(D) \subseteq A$. The boundary of $G$ is a convex polygon, which has at least three faces. Hence $3 \leq |\Lambda^r(D) \cap \Lambda^r(f^r(G))| \leq |A \cap \Lambda^r(f^r(G))|$ must hold. Analogously, for any cut-cycle $C$, at least one outer corner must be concave so that $C$ encloses its interior without creating edge crossings in $D$. Thus, $1 \leq |\Lambda^r(D) \cap \Lambda^r(C)| \leq |A \cap \Lambda^r(C)|$ must hold.

Appendix 2: Proof of Lemma 6

We say that a cut-cycle of $G$ is bad if $A \cap \Lambda^r(C) = \emptyset$.

We first show the only-if part. To show this by a contradiction, we assume that there is no such configuration, i.e., $A$ does not satisfy one of (i)-(iii) in Definition 5. If $A$ violates (i) (resp., (ii)), then it would hold $|A \cap \Lambda^r(f^r(G))| < 3$ (resp., the boundary of graph $G^{-}(\nu)$ of a non-root R-node $\nu$ would be a bad cut-cycle). Hence $A$ cannot be proper for any choice of the central edge $e(\nu)$ in (iii) for some non-root P-node $\nu$. First consider the case where $E^{-}_A$ contains the real edge $e_{j\nu}$. Then $e_{j\nu}$ is the central edge $e(\nu)$ and $\nu$ has a left component $G^{-}(\nu)$ with $A \cap \Lambda^r(\nu) = \emptyset$ or a right component $G^{-}(\nu)$ with $A \cap \Lambda^{\nu}(\nu) = \emptyset$. Hence the cycle $C$ consisting of $e_{j\nu}$ and $f^g_{\nu}(G^{-}(\nu))$ (or $f^{g}_{\nu}(G^{-}(\nu))$) would be a bad cut-cycle, a contradiction. We can treat the case where $E^{-}_A$ contains no real edge $e_{j\nu}$ analogously.

First assume that there is a component $G^{-}(\nu_j)$ that is not an induced path, but satisfies $A \cap \Lambda^r(\nu_j) = \emptyset$. If $\nu_j$ is an R-node, then the boundary of $G^{-}(\nu_j)$ would be a bad cut-cycle. Otherwise, if $\nu_j$ is an S-node, then $\nu^{-}(\nu_j)$ has at least one non-real edge $\ell$ (since $G^{-}(\nu_j)$ is not an induced path), and the node $\mu$ corresponding to $\ell$ would have a bad cut-cycle as its boundary of $G^{-}(\mu)$, a contradiction. The remaining case is that there are two components $G^{-}(\nu_j)$ and $G^{-}(\mu)$ with $j < \mu$ such that $A \cap \Lambda^r(\nu_j) = \emptyset$ and $A \cap \Lambda^{\nu}(\nu_j) = \emptyset$. In this case, the two paths $f^g_{\nu}(G^{-}(\nu_j))$ and $f^{g}_{\nu}(G^{-}(\nu_j))$ would give rise to a bad cut-cycle. This proves the only-if part.

We next show the if part. For this, we prove that $A$ cannot be proper for any choice of a configuration $\psi \in \Psi(G)$, assuming that (i) does not hold for $A$. Clearly $|A \cap \Lambda^r(f^r(G))| < 3$ violates (i) of Definition 5. Then $G$ has a bad cut-cycle $C$. Let $\{u, v\} \subseteq V(C)$ ($u < v$) be the cut-pair in $G$. First consider the case where $\{u, v\}$ is the parent edge of an R-node $\nu$. If $v$ is not a child of a P-node, then (ii) of Definition 5 cannot hold for the R-node $\nu$. Otherwise, if $\nu$ is a child of a P-node $\nu'$, then (iii) of Definition 5 cannot hold for any choice of the central edge $e(\nu')$ due to the bad boundary $C$ of $G^{-}(\nu')$. The remaining
case is that \( \{u, v\} \) is the parent edge of a P-node \( \nu \) and \( C \) consists of the two paths \( f^\nu_{uv}(G^-(\nu_j)) \) and \( f^\nu_{uv}(G^-([\nu_j]) \) for some two components \( G^-([\nu]) \) and \( G^-([\nu_j]) \) of the P-node. Again in this case, (iii) of Definition 5 cannot hold for any choice of the central edge \( e(\nu) \). This proves the if part.

**Appendix 3: Proof of Lemma 7**

We can easily check whether (i) of Definition 5 holds or not. To check (ii) and (iii) of Definition 5, we first compute

\[
a_{uv}(\nu) = |A \cap \Lambda^u_{uv}(G^-([\nu]))|,
\]

\[
a_{uv}(\nu) = |A \cap \Lambda^u_{uv}(G^-([\nu]))|,
\]

where parent(\( \nu \)) = \( \{u, v\} \), for all non-root nodes \( \nu \) in a bottom-up manner along the rooted SPQR tree. For a non-root node \( \nu \), where parent(\( \nu \)) = \( \{u, v\} \), let \( Ch_{uv}(\nu) \) denote the set of children \( \mu \in Ch(\nu) \) that correspond to virtual edges along \( f^\nu_{uv}(\sigma^-([\nu])) \). We define \( Ch_{uv}(\nu) \) analogously.

For each leaf R-node \( \nu \), we compute \( a_{uv}(\nu) \) and \( a_{uv}(\nu) \) as defined (note that \( \sigma^-([\nu]) = G^-([\nu]) \)). For each non-leaf node \( \nu \), where parent(\( \nu \)) = \( \{u, v\} \), \( a_{uv}(\nu) \) is given as the summation of \( a_{uv}(\nu) \) for all children \( \nu \in Ch_{uv}(\nu) \), where \( u', v' \in parent(\nu') \) appear in this order \( f^\nu_{uv}([\nu^-([\nu])) \) from \( u \) to \( v \) and the number of edges in \( A \) that appear along \( f^\nu_{uv}([\nu^-([\nu])) \) in this order \( f^\nu_{uv}([\nu^-([\nu])) \). It is not difficult to see that \( a_{uv}(\nu) \), \( a_{uv}(\nu) \) can be computed in \( O([\nu^-([\nu])) + |E([\nu^-([\nu]))]) \) time. Hence \( a_{uv}(\nu) \), \( a_{uv}(\nu) \) for all non-root nodes \( \nu \) can be computed in \( O(\sum_{\nu \in Ch(\nu)} [\nu^-([\nu])) + |E([\nu^-([\nu]))]) \) time. This proves the lemma.

**Appendix 4: Proof of Lemma 9**

Since \( D \) is a straight-line drawing of \( G \), the set \( W = A^c(D) \) of its concave corners satisfies condition (1). Hence by Lemma 6, there is a configuration \( \psi \in \Psi(G) \) such that \( W \) is proper with respect to \( (\nu^*, \psi) \). By the definition of fringe corners, \( A = W \cap \Lambda^c(\nu^*, \psi) \) remains proper with respect to \( (\nu^*, \psi) \). This proves the lemma.

**Appendix 5: Algorithm for Star-shaped Drawings**

To give a full description of the algorithm (Algorithms 1-4 below), we distinguish type I with type II for each of S- and R-nodes. We briefly explain what we need to compute for each type of nodes. When a vertex \( u \) is drawn as a point in the plane, the point may be denoted by \( u \) for notational simplicity.

**type I S-node**

A non-root S-node \( \nu \) is called type I if the virtual edge \( e(\nu) \) corresponding \( \nu \) is an outer edge (resp., an edge) in the drawing for its parent R-node (resp., its parent P-node).

Input: The view point \( r_f \) of the face incident to \( e(\nu) \) and a convex polygon \( B_\nu \) drawn for \( f^\nu_{uv}(G^-([\nu])) \) (or \( f^\nu_{uv}(G^-([\nu])) \)) are given.

Output: For each child node \( \mu \in Ch(\nu) \), place the parent cut-pair \( (s, t) \) on \( B_\mu \) and fix the boundary of \( G^-([\mu]) \) as a convex \( (|A \cap \Lambda^c([\mu]) + 2 \)-gon \( B_\mu \) with \( \Lambda^c([\mu]) = A \cap \Lambda^c(\mu) \cup \{s, t\} \) by combining the line segments of \( B_\mu \) between \( s \) and \( t \) and a new line segment between \( s \) and \( t \) inside \( B_\mu \) (see Fig. 6, the virtual edge \( e_1 \) in Fig. 7, and Algorithm 2).

**type II S-node**

A non-root S-node \( \nu \) is called type II if its parent node is an R-node, and the virtual edge \( e(\nu) \) corresponding \( \nu \) is an inner edge in the drawing of the parent node.

Input: The view points \( r_1 \) and \( r_2 \) of the two faces incident to \( e(\nu) \) and a line segment \( B_\nu = [u, v] \) drawn for parent(\( \nu \)) = \( \{u, v\} \) are given.

Output: For each (R- or P-node) \( \mu \), place the parent cut-pair \( (s, t) \) on \( B_\mu \) and fix the boundary of \( G^-([\mu]) \) as a convex \( (|A \cap \Lambda^c([\mu]) + 2 \)-gon \( B_\mu \) with \( \Lambda^c([\mu]) = A \cap \Lambda^c(\mu) \cup \{s, t\} \) inside \( [s, t, r_1] \cup [s, t, r_2] \) (see the virtual edge \( e_2 \) in Fig. 7, and Algorithm 2).
**type I R-node** A non-root R-node $\nu$ is called type I if a convex drawing of plane graph $\sigma_{vu}(\nu)$ (resp., $\sigma_{wv}(\nu)$) is required to be computed. Let $(u, v) = parent(\nu)$.

Input: A convex $(|A \cap \Lambda^o_{vu}(\nu)| + 2)$-gon $B_\nu$ for $f^v_{vu}(\nu)$ (resp., $(|A \cap \Lambda^o_{wv}(\nu)| + 2)$-gon $B_\nu$ for $f^v_{wu}(\nu)$) is given (see Fig. 8(a)).

Output: A convex drawing of $\sigma_{vu}(\nu)$ (resp., $\sigma_{wv}(\nu)$) as an extension of $B_\nu$ (see Algorithm 3 and Fig. 8(b)).

**type II R-node** An R-node $\nu$ is called type II if a convex drawing of $\sigma^{-}(\nu)$ is required to be computed.

Input: A convex $(|A \cap \Lambda^o(\nu)| + 2)$-gon $B_\nu$ for the boundary of $G^{-}(\nu)$ is given (see Fig. 9(a)), where a convex $|A \cap \Lambda^o(\nu^{*})|$-gon $B_{\nu^{*}}$ for the boundary of $G$ is given if $\nu$ is the root $\nu^{*}$.

Output: A convex drawing of $\sigma^{-}(\nu)$ as an extension of $B_\nu$ (see Algorithm 3 and Fig. 9(b)).

**P-node** Let $\nu$ be a P-node. Let $u$, $v$ be the vertices in $V_\nu$ if $\nu$ is the root $\nu^{*}$ or the vertices in $parent(\nu)$ otherwise.

Input: A convex boundary $B_\nu$ of $G^{-}(\nu)$, where $(e_1, e_2, \ldots, e_j, \ldots, e_k)$ denotes the sequence of $E^\nu_{-}$, where $e_j = e(\nu)$ is the central edge, and $f_i$ denotes the face between two edges $e_i$ and $e_{i+1}$ in the plane graph $\sigma^{-}(\nu)$ (see Fig. 10(a)).

Output: Drawings of $\sigma^{-}(\mu)$ of child nodes $\mu \in Ch(\nu)$. We process left edges in $E^\nu_{-}$ from $e_1$ to $e_{j-1}$ and right edges in $E^\nu_{-}$ from $e_k$ to $e_{j+1}$ before the central edge $e_j$ is processed. Left edges: If left edge $e_i$ corresponds to an S-node $\mu_i$, then we choose a view point $r_i$ and treat $\mu_i$ as a type I S-node. If left edge $e_i$ corresponds to an R-node $\mu_i$, then treat $\mu_i$ as a type I R-node, and compute a convex drawing $D_\mu$ of $\sigma_{vu}(\mu_i)$ (including virtual edge $parent(\nu)$), where the view point $r_i$ is computed during the computation of $D_\mu$ (see Fig. 11(a) and (b)).

Right edges: we apply the above procedure symmetrically to right edges $e_k, e_{k-1}, \ldots, e_j$.

We treat the central edge $e_j$ as follows. If $e_j$ corresponds to an R-node $\mu_j$, then we then treat $\mu_j$ as a type II R-node.

![Diagram](image)

Figure 6: (a) Fixing the boundary $B_\nu$ of $G^{-}(\nu)$ of an S-node $\nu$ of type I; (b) Fixing the boundaries $B_{\mu_i}$ of $G^{-}(\mu_i)$ of the child nodes $\mu_i \in Ch(\nu)$ of an S-node $\nu$ of type I.

Given a proper set $A$, we apply Algorithm 1 to the root $\nu^{*}$, and then invoke for its child nodes Algorithms 2-4 recursively. For a subset $A \subseteq \Lambda^o(G)$, let $V(A)$ denote the set of vertices $\nu$ such that $\Lambda(\nu) \cap A \neq \emptyset$. 
Figure 7: (a) Fixing the boundary $B_{\nu^*}$ of $G$ of the root R-node $\nu^*$ as $P$; (b) Fixing the boundaries of $G^-(\mu)$ and $G^-(\mu')$ for the child P- and R-nodes $\mu \in Ch(\nu)$ and the child P- and R-nodes $\mu' \in Ch(\mu)$ with child S-nodes $\mu \in Ch(\nu)$.

Figure 8: (a) For an R-node $\nu$ of type I, a convex $(|A \cap \Lambda^{\nu}(\nu)| + 2)$-gon $B_{\nu}$ is drawn for $f^*_w(\nu)$; (b) Extending $B_{\nu}$ into a convex drawing of $\sigma_{\nu}(\nu)$; (c) Fixing the boundaries of $G^-(\mu)$ and $G^-(\mu')$ for the child P- and R-nodes $\mu \in Ch(\nu)$ and the child P- and R-nodes $\mu' \in Ch(\mu)$ with child S-nodes $\mu \in Ch(\nu)$.

Algorithm 1: RootNode($\nu^*, P$)

1: Fix the boundary of $G$ as a prescribed convex $|A \cap \Lambda(\nu^* )| -$gon $P$ with $\Lambda^c(P) = A \cap \Lambda^c(\nu^*)$; Let $B_{\nu^*} := P$;
2: if $\nu^*$ is an S-node then
3: Choose a point $r_{f}$ in the inner face $f$ of $B_{\nu^*}$ as the view point of $f$, and SNode($\nu^*$, $B_{\nu^*}, r_{f}, \emptyset$)
4: end if;
5: RNode($\nu^*, B_{\nu^*}, \sigma(\nu^*), \Pi$), and PNode($\nu^*, B_{\nu^*}$) if $\nu^*$ is an R- and P-node, respectively.
Figure 9: (a) For an R-node $\nu$ of type II, a convex $([A \cap \Lambda^c(\nu)] + 2)$-gon $B_\nu$ is given as the boundary of $G^-(\nu)$; (b) Extending $B_\nu$ into a convex drawing of $\sigma^-(\nu)$; (c) Fixing the boundaries of $G^-(\mu)$ and $G^-(\mu')$ for the child P- and R-nodes $\mu \in C h(\nu)$ and the child P- and R-nodes $\mu' \in C h(\mu)$ with child S-nodes $\mu \in C h(\nu)$.

Figure 10: (a) $B_\nu$ for the root P-node or an internal P-node $\nu$; (b) $e_1$ corresponds to an S-node $\mu_1$, where $S N O D E(\mu_1, B_{\mu_1}, r_{f_1}, \emptyset, 1)$ is executed after choosing view point $r_{f_1}$.

Figure 11: (a) $e_1$ corresponds to an R-node $\mu$ of type I, where $R N O D E(\mu_1, B_{\mu_1}, \sigma_{e_1}(\mu_1), 1)$ computes a convex drawing $D_\mu$ of $\sigma_{e_1}(\mu)$ and returns a view point $r_{f_1}$; (b) $R N O D E(\mu_1, B_{\mu_1}, \sigma_{e_1}(\mu_1), 1)$ draws the boundaries of child nodes of $\mu_1$. 
Algorithm 2: SNode($\nu$, $B_\nu$, $r_1$, $r_2$, type)
1: Fix the positions of vertices in $V_\nu - V(A \cap \Lambda^\nu(\nu))$ on $B_\nu$;
2: for each virtual edge $e = (s, t) \in E_\nu^-$ and its corresponding child node $\mu \in Ch(\nu)$ do
3: Fix the boundary of $G^-(\mu)$ as a convex $([A \cap \Lambda^\nu(\mu)] + 2)$-gon $B_\mu$ with $\Lambda^\nu(B_\mu) = A \cap \Lambda^\nu(\mu) \cup \{s, t\}$ as follows:
4: if type=1 then
5: Let $B_\mu$ consist of the boundary of $B_\nu$ from $t$ to $s$ and a series of line segments between $s$ and $t$ inside region $[s, t, r_1]$;
6: else /* type=II and $B_\nu = [u, v]$ for parent($\nu$) = ($u, v$) */
7: Choose such a convex $([A \cap \Lambda^\nu(\mu)] + 2)$-gon $B_\mu$ inside $[s, t, r_1] \cup [s, t, r_2]$
8: end if;
9: RNode($\mu$, $B_\mu$, $\sigma^-(\mu)$, II) if $\mu$ is an R-node, and PNode($\mu$, $B_\mu$) otherwise ($\mu$ is a P-node)
10: end for.

Algorithm 3: RNode($\nu$, $B_\nu$, $H$, type)
1: /* $H = \sigma_{vu}(\nu)$ or $\sigma_{uv}(\nu)$ for ($u, v$) = parent($\nu$) if type=I, and $H = \sigma^-(\nu)$ otherwise (type=II) */
2: Extend $B_\nu$ to a convex drawing $D_\nu$ of $H$;
3: Choose a point $r_f$ in each inner face $f$ of $D_\nu$ as the view point of $f$;
4: Return the view point $r_f$ of the inner face $f$ incident to parent($\nu$) if type=I;
5: Fix the positions of vertices in $V_\nu - V(A \cap \Lambda^\nu(\nu))$ in $B_\nu$;
6: for each outer edge $e = (s, t) \in E_\nu^-$ do
7: if $e$ corresponds to a child R- or P-node $\mu \in Ch(\nu)$ then
8: Fix the boundary of $G^-(\nu)$ as a convex $([A \cap \Lambda^\nu(\mu)] + 2)$-gon $B_\mu$ as in line 5 of SNode;
9: RNode($\mu$, $B_\mu$, $\sigma^-(\nu)$, II) if $\mu$ is an R-node, and PNode($\mu$, $B_\mu$) otherwise
10: end if;
11: if $e$ corresponds to a child S-node $\mu \in Ch(\nu)$ then
12: Let $B_\mu$ be the line segment $[s, t]$, $r_f$ be the view point of the inner face $f$ of $B_\nu$ incident to $e$, and SNode($\mu$, $B_\mu$, $r_f$, $\emptyset$, II)
13: end if
14: end for;
15: for each inner edge $e = (s, t) \in E_\nu^-$ do
16: if $e$ corresponds to a child R- or P-node $\mu \in Ch(\nu)$ then
17: Fix the boundary of $G^-(\nu)$ as a convex $([A \cap \Lambda^\nu(\mu)] + 2)$-gon $B_\mu$ as in line 7 of SNode;
18: RNode($\mu$, $B_\mu$, $\sigma^-(\nu)$, II) if $\mu$ is an R-node, and PNode($\mu$, $B_\mu$) otherwise
19: end if;
20: if $e$ corresponds to a child S-node $\mu \in Ch(\nu)$ then
21: Let $B_\mu$ be the line segment $[s, t]$, $r_{f_1}$ and $r_{f_2}$ be the view points of the two inner faces $f_1$ and $f_2$ of $B_\nu$ incident to $e$, and SNode($\mu$, $B_\mu$, $r_{f_1}$, $r_{f_2}$, II)
22: end if
23: end for.
Algorithm 4: PNODE($v, B_v$)

1: Let $u$ and $v$ be the terminals of the P-node $v$, and $B_v^u$ be the ($|A \cap \Lambda^0_{vu}(\mu_1)| + 2$)-gon that consists of side $w$ and the boundary of $B_v$ from $u$ to $v$;
2: Let $(e_1, e_2, \ldots, e_{j^*}, \ldots, e_k)$ denotes the sequence of $E_v^-$, where $e_{j^*} = c(v)$;
3: for $i = 1, 2, \ldots, j^* - 1$ do
4:  Let $\mu_i \in Ch(v)$ denote the child node corresponding to $e_i$;
5:  Construct a convex ($|A \cap \Lambda^0_{vu}(\mu_i)| + 2$)-gon $B_{\mu_i}$ with $\Lambda^i(B_{\mu_i}) = A \cap \Lambda^0_{wu}(\mu_i) \cup \{u, v\}$ within region $[u, v, r_{f_{j+1}}]$ (we set $B_{\mu_i} := B_v^u$);
6:  if $\mu_i$ is an S-node then
7:   Choose a point $r_{f_{j_i}}$ inside region $[u, v, r_{f_{j_{i-1}}}]$ as the view point of $f_{j_i}$ (we choose $r_{f_{j_i}}$ inside $B_v^u$), and SNODE($\mu_i, B_{\mu_i}, r_{f_{j_i}}, \emptyset, I$)
8:  else /* $\mu_i$ is an R-node: $f^\mu_{vu}(G^-(\mu_i))$ is fixed as $B_{\mu_i}$ */
9:   RNODE($\mu_i, B_{\mu_i}, \sigma_{vu}(\mu_i), I$), which will return a view point $r_{f_{j^*}}$, and let $r_{f_{j^*}} := r_{f_{j_i}}$.
10: end if
11: end for;
12: for $i = k, k - 1, \ldots, j^* + 1$ do
13:  Apply the same procedure in lines 4-10 to $e_i$ by exchanging $A_{vu}$ and $\sigma_{vu}$ with $A_{uv}$ and $\sigma_{uv}$, respectively
14: end for;
15: if $e_{j^*}$ corresponds to an S-node $\mu_k \in Ch(v)$ then
16:  Let $B_{\mu_k}$ be the line segment $[u, v]$, and SNODE($\mu, B_{\mu_k}, r_{f_{j_{k-1}}}, r_{f_{j_{k+1}}}, I, \emptyset$), where if $j^* = k$ (resp., $j^* = 1$) then invoke SNODE($\mu, B_{\mu_k}, r_{f_{j_{k-1}}}, \emptyset, I$) (resp., SNODE($\mu, B_{\mu_k}, r_{f_{j_{k+1}}}, \emptyset, I$)).
17: end if;
18: if $e_{j^*}$ corresponds to an R-node $\mu_{j^*} \in Ch(v)$ then
19:  Fix the boundary of $G^-(\mu_{j^*})$ as a convex ($|A \cap \Lambda^0(\mu_{j^*})|$+2)-gon $B_{\mu_{j^*}}$ with $\Lambda^i(B_{\mu_{j^*}}) = A \cap \Lambda^0(\mu_{j^*}) \cup \{u, v\}$ within region $[u, v, r_{f_{j^*+1}}]$ (resp., $[u, v, r_{f_{j_{k+1}}}]$), and RNODE($\mu_{j^*}, B_{\mu_{j^*}}, \sigma^-(\mu_{j^*}), I$), where if $j^* = k$ (resp., $j^* = 1$) then choose $B_{\mu_{j^*}}$ in region $[u, v, r_{f_{j_{k-1}}}]$ (resp., $[u, v, r_{f_{j_{k+1}}}]$).
20: end if