Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 21, no. 2, pp. 155–175 (2017)
DOI: 10.7155/jgaa.00409

Star-Shaped and L-Shaped Orthogonal Drawings

Xin He[1] *Dayu He*[1]

[1]Department of Computer Science and Engineering,
State University of New York at Buffalo, Buffalo, NY 14260, USA.

Abstract

An *orthogonal drawing* of a plane graph G is a planar drawing of G, denoted by $D(G)$, such that each vertex of G is drawn as a point on the plane, and each edge of G is drawn as a sequence of horizontal and vertical line segments with no crossings. An orthogonal polygon P is called *orthogonally convex* if the intersection of any horizontal or vertical line L and P is either a single line segment or empty. An orthogonal drawing $D(G)$ is called *orthogonally convex* if all of its internal faces are orthogonally convex polygons. An orthogonal polygon P is called a *star-shaped polygon* if there is a point $p \in P$ such that the entire P is visible from p. An orthogonal drawing $D(G)$ is called a *star-shaped orthogonal drawing* (SSOD) if all of its internal faces are star-shaped polygons. Every SSOD is an orthogonally convex drawing, but the reverse is not true. SSOD is visually more appealing than orthogonally convex drawings. Recently, Chang et al. gave a necessary and sufficient condition for a plane graph to have an orthogonally convex drawing. In this paper, we show that if G satisfies the same condition given by Chang et al., it not only has an orthogonally convex drawing, but also a SSOD, which can be constructed in linear time.

An orthogonal drawing $D(G)$ is called an L-shaped drawing if each face of $D(G)$ is an L-shaped polygon. In this paper we also show that an L-shaped orthogonal drawing can be constructed in $O(n)$ time. The same algorithmic technique is used for solving both problems. It is based on regular edge labeling and is quite different from the methods used in previous results.

Submitted:	Reviewed:	Revised:	Reviewed:	Revised:
February 2016	June 2016	August 2016	September 2016	December 2016
	Accepted:	Final:	Published:	
	December 2016	December 2016	January 2017	
	Article type:		Communicated by:	
	Regular paper		S.-H. Hong	

Research supported in part by NSF Grant CCR-1319732. An extended abstract of this paper, with title "Star Shaped Orthogonal Drawing", was published in the proceedings of the 12th Annual Conference on Theory and Applications of Models of Computation (TAMC'2015), Lecture Notes in Computer Science 9076, pp. 137 - 149, May 2015.

E-mail addresses: xinhe@buffalo.edu (Xin He) dayuhe@buffalo.edu (Dayu He)

1 Introduction

Among many graph drawing styles, *orthogonal drawing* has attracted much attention due to its various applications in circuit schematics, relationship diagrams, data flow diagrams etc. [3]. An *orthogonal drawing* of a plane graph G is a planar drawing, denoted by $D(G)$, of G such that each vertex of G is drawn as a point on the plane, and each edge is drawn as a sequence of horizontal and vertical line segments with no crossings. A *bend* is a point where an edge changes its direction. (Figure 1 (b) shows an orthogonal drawing of the graph in Figure 1 (a). The point p is a bend.)

Rahman et al. [12] gave a necessary and sufficient condition for a plane graph G of maximum degree 3 to have an orthogonal drawing without bends. A linear-time algorithm to find such a drawing was also obtained in [12]. In the drawings constructed in [12], the faces of $D(G)$ can be of complicated shapes. An orthogonal polygon P is *orthogonally convex* if, for any horizontal or vertical line L, the intersection of L and P is either empty or a single line segment. (Figure 1 (c) shows an orthogonally convex polygon. The face marked by F in Figure 1 (b) is not orthogonally convex.) An orthogonal drawing $D(G)$ is *orthogonally convex* if all faces of $D(G)$ are orthogonally convex polygons. The orthogonally convex drawings are more visually appealing than arbitrary orthogonal drawings.

Chang et al. [2] gave a necessary and sufficient condition (which strengthens the conditions in [12]) for a plane graph G of maximum degree 3 to have an orthogonal convex drawing without bends. A linear-time algorithm to find such a drawing was also obtained in [2].

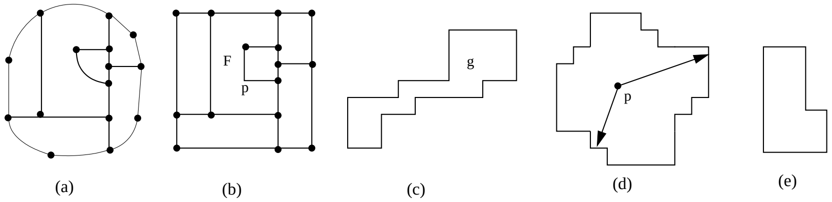


Figure 1: (a) A plane graph G, (b) an orthogonal drawing of G, (c) an orthogonally convex polygon, (d) a star-shaped orthogonal polygon, (e) an L-shaped polygon.

An orthogonal polygon P is called *star-shaped* if there exists a point p in P such that for all other point q in P, the line segment (p, q) is entirely contained in P. (In other words, the entire polygon P is *visible* from p. See Figure 1 (d).) It is easy to see that any star-shaped orthogonal polygon is always orthogonally convex. But the reverse is not true. An orthogonal drawing $D(G)$ is called a *star-shaped orthogonal drawing* (SSOD) if every inner face of $D(G)$ is a star-shaped orthogonal polygon. The star-shaped orthogonal drawings are more visually appealing than orthogonally convex drawings (see Figure 1 (c) and (d)). In this paper, we show that if G satisfies the same conditions given in [2], then G has a SSOD without bends. In addition, such a drawing can be constructed in linear time.

To the best knowledge of the authors, SSOD is a new drawing style. Although star-shaped drawings have been studied before [7, 8], the polygons in their drawings are required to be star-shaped but not orthogonal. In [11, 4], the problem of covering orthogonal polygons by star-shaped orthogonal polygons is studied.

An orthogonal polygon P is called *L-shaped* if P has exactly five convex corners and one concave corner (see Figure 1 (e)). An orthogonal drawing $D(G)$ is called *L-shaped* if every face of $D(G)$ is either a rectangle or an L-shaped polygon. In [13], an algorithm for testing if G has an L-shaped orthogonal drawing was given. The testing is done by checking if an axillary graph has a perfect matching and takes $O(n^{1.5})$ time. If G passes the test, an L-shaped drawing of G can be constructed in $O(n^2)$ time [13]. In [14], it was shown that the construction part can be done in linear time. In this paper, we show that our method can also be used to construct an L-shaped orthogonal drawing of G in linear time. Our algorithm is simpler and quite different from the algorithm in [14].

The paper is organized as follows. In Section 2, we present the definitions and preliminary results. Section 3 describes a special rectangular dual needed by our algorithm. In Section 4, we present our SSOD algorithm. In Section 5, we discuss how to construct an L-shaped orthogonal drawing. Section 6 concludes the paper.

2 Preliminaries

Most definitions in this paper are standard. Let $G = (V, E)$ be a graph with n vertices. Let $N(v)$ denote the set of the neighbors of a vertex v. The *degree* of a vertex v is the number of neighbors of v in G. A vertex of degree 2 (respectively 3) is called a *2-vertex* (respectively *3-vertex*). The maximum degree of G is denoted by $\Delta(G)$. G is called a d-graph if $\Delta(G) \leq d$. A connected graph G is called *biconnected* if it has no cut vertices.

A *planar graph* is a graph G that can be drawn on the plane so that the vertices are drawn at points on the plane and the edges are drawn as curves connecting their end vertices without crossings. Such a drawing is called a *plane embedding* of G. A *plane graph* is a planar graph with a fixed plane embedding. For the problems considered in this paper, the case that G is only connected can be easily reduced to the biconnected case [12, 2]. Thus, as in [12, 2], G always denotes a biconnected plane 3-graph in the rest of this paper.

A fixed embedding of G divides the plane into a set of connected regions called *faces*. The unbounded face of G is called the *exterior face*. The bounded faces are called the *interior faces*. The *contour* of a face f is the cycle formed by the vertices and the edges on the boundary of f. The contour of the exterior face of G is denoted by $C_o(G)$.

A cycle C of G with k edges is called a k-cycle. A *triangle* is a 3-cycle. G is called *internally triangulated* if all of its interior faces are triangles. G is called *triangulated* if all of its faces (both interior and exterior) are triangles. A cycle C of G divides the plane into two regions: the interior and the exterior region. A *separating cycle* of G is a cycle C such that there is at least one vertex in the interior region of C and at least one vertex in the exterior region of C. Note that a separating cycle may be contained in the interior of other separating cycles. A separating cycle C is called *maximal* if it's not contained in any other separating cycle C'.

Let $D(G)$ be an orthogonal drawing of G without bends. Each cycle C of G is drawn as an orthogonal polygon, denoted by $D(C)$, in $D(G)$. Let a be a vertex of C. We will also use a to denote the point in $D(C)$ that corresponds to a. A vertex a of $D(C)$ is called a *corner* of $D(C)$ if the interior angle of $D(C)$ at a is 90° or 270°. A corner with 90° (270°, respectively) interior angle is called a *convex* (*concave*, respectively) corner. For an orthogonal drawing $D(G)$ without bends, any concave corner a of $D(G)$ must be a vertex of G, and since there are only two line segments incident to a, the degree of this vertex must be 2.

The algorithm **Bi-Orthogonal-Draw** in [12] finds an orthogonal drawing of the input graph G. The first step of algorithm **Bi-Orthogonal-Draw** arbitrarily selects four 2-vertices on the exterior face $C_o(G)$ and draw $C_o(G)$ as a rectangle with the selected four vertices as its four corners. The drawing algorithm in [2] is a modified version of the algorithm **Bi-Orthogonal-Draw**. It also draws the exterior face $C_o(G)$ as a rectangle with four selected vertices as the four corners of the exterior rectangle [2]. We will following the same convention as in [12, 2]: the input to our problem is a plane graph H with four specified 2-vertices a, b, c, d on $C_o(H)$ in clockwise order. Our goal is to produce an orthogonal drawing $D(H)$ of H such that $C_o(H)$ is drawn as a rectangle with a, b, c, d as the northwest, northeast, southeast and southwest corner of $D(H)$, respectively.

To further simplify the presentation, we construct a new graph G from H as follows (see Figure 2 (a)):

1. Add eight new vertices $a'', a', b'', b', c'', c', d'', d'$ in the exterior face of G; connect them into a clockwise cycle.
2. Add four new edges $(a, a'), (b, b'), (c, c'), (d, d')$.

Clearly, H has a no-bend orthogonal drawing with a, b, c, d as the four corners if and only if G has a no-bend orthogonal drawing with a'', b'', c'', d'' as the four corners (see Figure 2 (b)). Note that G satisfies the following properties:

Property 1 *1. G is a biconnected plane 3-graph.*

2. G has eight vertices on its exterior face $C_o(G)$; four are 2-vertices and four are 3-vertices; the 2-vertices and 3-vertices alternate on $C_o(G)$.

3. The four 2-vertices on $C_o(G)$ are specified as the northwest, northeast, southeast, southwest vertices.

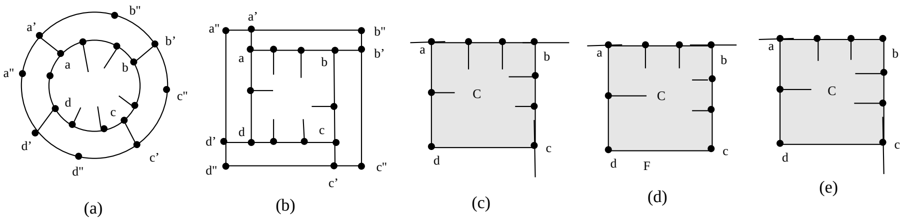


Figure 2: (a) The construction of G from H, (b) drawings of H and G, (c) and (d) conditions in Theorem 1, (e) conditions in Theorem 2.

In the rest of the paper, without loss of generality, we always assume G satisfies Property 1.

Let C be a cycle of G. A *leg* of C is an edge e that is in the exterior of C and has exactly one vertex on C. The vertex of e that is on C is called a *leg vertex* of C. A *k-legged cycle* is a cycle C with exactly k legs. The k leg vertices divide C into k sub-paths. Each sub-path is called a *contour path* of C.

Theorem 1 *[12] Let G be a plane graph that satisfies the conditions as in Property 1. Then G has an orthogonal drawing without bends if and only if the following two conditions hold:*

1. *Every 3-legged cycle C has at least one 2-vertex.*

2. *Every 2-legged cycle C has at least two 2-vertices.*

Figure 2 (c) shows a 3-legged cycle $C = \{a, b, c, d\}$ and an orthogonal drawing of C (d is a 2-vertex.) Figure 2 (d) shows a 2-legged cycle $C = \{a, b, c, d\}$ and an orthogonal drawing of C (c and d are two 2-vertices.)

Theorem 2 *[2] Let G be a plane graph that satisfies the conditions in Property 1. Then G has an orthogonally convex drawing without bends if and only if the following two conditions hold:*

1. *Every 3-legged cycle C has at least one 2-vertex.*

2. *Every 2-legged cycle C has at least two 2-vertices, at least one on each of its two contour paths.*

Figure 2 (e) shows a 2-legged cycle $C = \{a, b, c, d\}$ and an orthogonal drawing of C (b and d are two 2-vertices.) Note that, in Figure 2 (d), the 2-legged cycle C satisfies the condition 2 in Theorem 1, but not the condition 2 in Theorem 2. Hence there exists no orthogonally convex drawing: In any drawing, the face outside of C (marked by F) cannot be orthogonally convex. In Section 4, we will show that any graph satisfying the conditions in Theorem 2 has a SSOD without bends.

Let $G^* = (V^*, E^*)$ be the dual graph of G. To avoid confusion, the members of V^* are called *nodes*. Each node in V^* corresponds to an interior face f of G, and two nodes in V^* are adjacent to each other in G^* if and only if their corresponding faces in G share an edge as common boundary. (Note that there is no node in V^* corresponding to the exterior face of G. This is is slightly different from the normal definition of the dual graph.)

Definition 1 *Let G^* be an internally triangulated plane graph whose exterior face has four nodes. A rectangular dual of G^* is a rectangle R divided into smaller rectangles such that the following hold:*

1. *No four smaller rectangles meet at the same point.*
2. *Each smaller rectangle corresponds to a node of G^*.*
3. *Two nodes of G^* are adjacent in G^* if and only if their corresponding small rectangles share a line segment as their common boundary.*

Figure 3 (a) shows an orthogonal drawing $D(G)$ of a graph G that satisfies the conditions in Property 1. Figure 3 (b) shows the dual graph G^* of G. Note that the orthogonal drawing $D(G)$ in Figure 3 (a) is also a rectangular dual R of G^* (each face of $D(G)$ is a rectangle). Not every internally triangulated graph G^* has a rectangular dual. The following theorem characterizes such graphs.

Theorem 3 *[10] A plane graph $G^* = (V^*, E^*)$ has a rectangular dual with four rectangles on its boundary if and only if:*

1. *Every interior face of G^* is a triangle and the exterior face of G^* is a quadrangle.*
2. *G^* has no separating triangles.*

G is called a *proper triangular plane (PTP) graph* if it satisfies the two conditions in Theorem 3. Our algorithm heavily depends on the following concept:

Definition 2 *A regular edge labeling (***REL***) $\mathcal{R} = \{T_1, T_2\}$ of a PTP graph G^* is a partition of the interior edges of G^* into two subsets T_1, T_2 of directed edges such that the following conditions hold:*

1. *For each interior node v, the edges incident to v appear in clockwise order around v as follows: a set of edges in T_1 leaving v; a set of edges in T_2 leaving v; a set of edges in T_1 entering v; a set of edges in T_2 entering v. (All four sets are not empty.)*
2. *Let v_N, v_E, v_S, v_W be the four exterior nodes of G in clockwise order. All interior edges incident to v_N are in T_1 and entering v_N. All interior edges incident to v_E are in T_2 and entering v_E. All interior edges incident to v_S are in T_1 and leaving v_S. All interior edges incident to v_W are in T_2 and leaving v_W.*

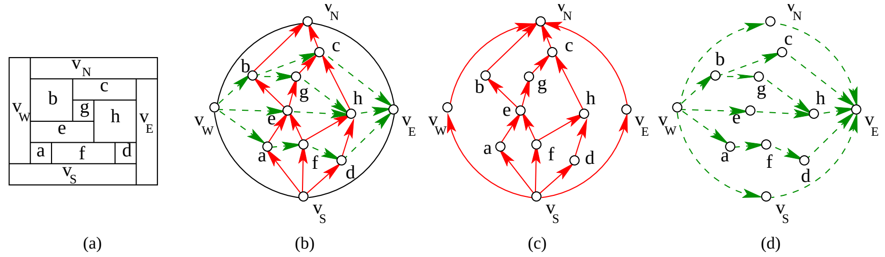


Figure 3: (a) An orthogonal drawing $D(G)$ of a graph G, which is also a rectangular dual of G^*, (b) the dual graph G^* of G with an **REL** $\mathcal{R} = \{T_1, T_2\}$, (c) the subgraph G^*_{SN}, (d) the subgraph G^*_{WE}.

Figure 3 (b) shows an example of **REL** of a PTP graph. The red solid lines are edges in T_1. The green dashed lines are edges in T_2.

Theorem 4 *[5, 6] Every PTP graph G^* has an* **REL** *which can be constructed in linear time. From an* **REL** *of G^*, a rectangular dual of G^* can be constructed in linear time.*

3 A Special Rectangular Dual

A PTP graph G^* may have many different **REL**s. From the same **REL** of G^*, we may obtain different rectangular duals. In this section, we describe a rectangular dual of G^* with special properties, which is needed by our SSOD construction. First we outline the rectangular dual algorithm **DUAL** in [5].

Let G^* be a PTP graph with an **REL** $\mathcal{R} = \{T_1, T_2\}$. The *SN subgraph* G^*_{SN} is the directed subgraph of G^* consisting of the edges in T_1 and the four exterior edges directed as $v_S \to v_W \to v_N, v_S \to v_E \to v_N$ (see Figure 3 (c)). The *WE subgraph* G^*_{WE} is the directed subgraph of G^* consisting of the edges in T_2 and the four exterior edges directed as $v_W \to v_S \to v_E, v_W \to v_N \to v_r$ (see Figure 3 (d)).

Define the dual graph G_{SN} of G^*_{SN} as follows. The vertex set of G_{SN} is the set of the interior faces of G^*_{SN} plus two exterior faces f_W and f_E. For each directed edge e^* in G^*_{SN}, the *left face of* e^*, denoted by left(e^*), is the face in G^*_{SN} that is located to the left of e^* when walking along e^*. Similarly, the *right face of* e^*, denoted by right(e^*), is the face in G^*_{SN} that is located to the right of e^* when walking along e^*. For each directed edge e^* in G^*_{SN}, there is a corresponding edge e in G_{SN} directed from the face left(e^*) to the face right(e^*). G_{SN} is an acyclic directed graph with f_W as the only source and f_E as the only sink [5]. A *consistent numbering* of order k_1 of G_{SN} is a mapping F_1 from the vertex set of G_{SN} to the set of integers $\{0, 1, ..., k_1\}$ such that (1) $F_1(f_W) = 0$ and $F_1(f_E) = k_1$, and (2) if there is an edge from the vertex f to the vertex g in G_{SN}, then $F_1(f) < F_1(g)$. (For example, a *topological numbering* of G_{SN} is a consistent numbering.) (Figure 4 (a) shows an example: The small circles are nodes in G^*_{SN}, the solid lines are the edges in G^*_{SN}, the blue dashed lines are the edges in G_{SN}, the small squares are the vertices in G_{SN}, the number inside a square is the number assigned to the vertex of G_{SN}.) The dual graph G_{WE} of G^*_{WE} and the consistent numbering F_2 of G_{WE} are defined similarly.

For any node u in G^*_{SN}, the *left face* of u, denoted by $\text{left}(u)$, is the face in G^*_{SN} bounded by the left most edge entering u and the left most edge leaving u. The *right face* of u, denoted by $\text{right}(u)$, is the face in G^*_{SN} bounded by the right most edge entering u and the right most edge leaving u (see Figure 4 (b)). Define $l(u) = F_1(\text{left}(u))$ and $r(u) = F_1(\text{right}(u))$. Clearly, $l(u) < r(u)$.

For any node u in G^*_{WE}, the *top face* of u, denoted by $\text{top}(u)$, is the face in G^*_{WE} bounded by the left most edge entering u and the left most edge leaving u. The *bottom face* of u, denoted by $\text{bottom}(u)$, is the face in G^*_{WE} bounded by the right most edge entering u and the right most edge leaving u. Define $t(u) = F_2(\text{top}(u))$ and $b(u) = F_2(\text{bottom}(u))$. Clearly, $b(u) < t(u)$.

The algorithm **DUAL** in [5] assigns each node u in G^* a rectangle with the left boundary $x = l(u)$, the right boundary $x = r(u)$, the bottom boundary $y = b(u)$, and the top boundary $y = t(u)$. It was shown in [5] these rectangles constitute a rectangular dual of G^*.

The subgraphs G^*_{SN} and G^*_{WE} are completely defined by the given **REL** $\mathcal{R} = \{T_1, T_2\}$. However, there may be many consistent numberings for G_{SN} (and G_{WE}.) Different numberings will result in different rectangular duals. In the following lemma, we show that G^* has a rectangular dual with special properties. This special rectangular dual is important for our SSOD drawing and may have applications in solving other drawing problems. For a rectangular dual R of G^*, let r_u denote the rectangle in R corresponding to the node u.

Lemma 1 *Any PTP graph G^* has a rectangular dual R such that the following properties hold for any node u in G^*.*

1. *Let $v_1 \to u$ be the first clockwise T_1 edge entering u and $u \to v_2$ the first clockwise T_1 edge leaving u. Then there exists a vertical strip in R that intersects r_{v_1}, r_u, r_{v_2}.*

2. *Let $w_1 \to u$ be the first clockwise T_2 edge entering u and $u \to w_2$ the first clockwise T_2 edge leaving u. Then there exists a horizontal strip in R that intersects r_{w_1}, r_u, r_{w_2}.*

Proof: Let $\mathcal{R} = \{T_1, T_2\}$ be any **REL** of G^*. We use the *right-most topological sort* to calculate a consistent numbering F_1 of G_{SN} which is defined as follows. We run the topological sort algorithm on G_{SN} starting from the (only) source vertex f_W of G_{SN}. Once a vertex u in G_{SN} has been numbered, we number the outgoing neighbors of u in G_{SN}. Among all neighbors of u, we always visit the right-most outgoing neighbor v_1 first. After all vertices that can be numbered from v_1 have been numbered, we continue to number the next right-most outgoing neighbor of u, and so on. Figure 4 (a) shows the consistent numbering obtained by the right-most topological sort of G_{SN}.

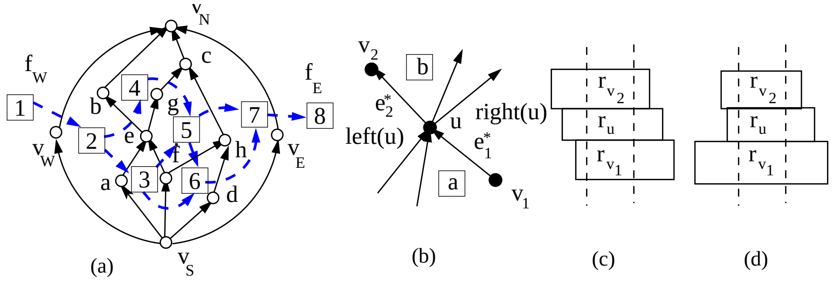


Figure 4: (a) The graph G^*_{SN}, its dual graph G_{SN}, and a consistent numbering of G_{SN}, (b) the pattern of the edges around a node u. The first clockwise edge entering u is e^*_1. The first clockwise edge leaving u is e^*_2, (c) and (d) two examples of the rectangles corresponding to the nodes v_1, u, v_2.

Consider any node u in G^*. Let $e^*_1 = v_1 \rightarrow u$ be the first clockwise T_1 edge entering u and $e^*_2 = u \rightarrow v_2$ be the first clockwise T_1 edge leaving u. See Figure 4 (b). Let $l(u) = F_1(\text{left}(u))$, $r(u) = F_1(\text{right}(u))$, $l(v_1) = F_1(\text{left}(v_1))$, $r(v_1) = F_1(\text{right}(v_1))$, $l(v_2) = F_1(\text{left}(v_2))$, $r(v_2) = F_1(\text{right}(v_2))$. Let a be the number assigned to the left face of e^*_1 and b be the number assigned to the right face of e^*_2. By the construction of **DUAL** algorithm in [5], the rectangle r_u shares a boundary with and is located above the rectangle r_{v_1}, and the rectangle r_{v_2} shares a boundary with and is located above the rectangle r_u.

By the definition of consistent numbering, the following hold: (1) $l(v_1) \leq a < r(v_1)$, (2) $l(v_2) < b \leq r(v_2)$, and (3) $l(v_2) \leq l(u) \leq a < r(u) \leq r(v_1)$.

Because F_1 is obtained by using the rightmost topological sort of G_{SN}, we have $a < b$. This implies (4) $l(v_1) \leq a < b \leq r(v_2)$. By (1) through (4), $l(v_1), l(u), l(v_2)$ are all $\leq a$ and $r(v_1), r(u), r(v_2)$ are all $> a$. Because all these quantities are integers, the vertical strip with the left vertical boundary line $x = a$ and the right vertical boundary line $x = a + 1$ intersects the three rectangles r_{v_1}, r_u, r_{v_2}.

So Statement 1 holds. Figure 4 (c) and (d) show two possible relative positions of r_{v_1}, r_u, r_{v_2}.

Similarly, we can use the right-most topological sort to calculate a consistent numbering F_2 of G_{WE}. By using the same argument, we can show Statement 2 holds. □

4 Star-shaped Orthogonal Convex Drawing

Let G be a plane graph that satisfies the conditions in Theorem 2. In this section, we describe how to find a SSOD without bends for G.

Let v be a 2-vertex in G with two neighbors u, w. The operation *contracting* v is defined as follows: delete v and replace the two edges (u, v) and (v, w) by a single edge (u, w).

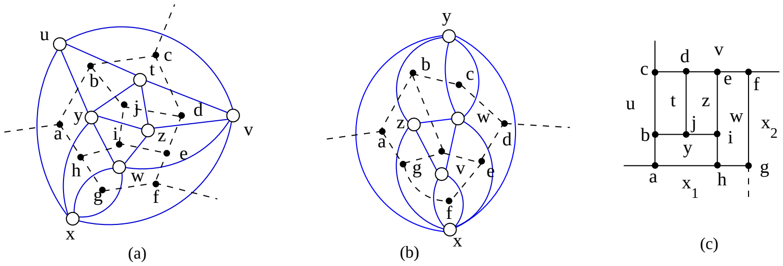


Figure 5: (a) A 3-legged cycle $C = \{a, b, c, d, e, f, g, h\}$ and the corresponding dual separating triangle $C^* = \{u, v, x\}$, (b) a 2-legged cycle $C = \{a, b, c, d, e, f, g\}$ and the corresponding dual separating 2-cycle $C^* = \{x, y\}$, (c) the drawing of the subgraph in (a).

First we modify G as follows. For every 3-legged cycle C in G with more than one 2-vertex on C, we arbitrarily choose one 2-vertex and contract every other 2-vertices on C. For every 2-legged cycle C in G with more than two 2-vertices on C, we arbitrarily choose one 2-vertex on each contour path of C and contract every other 2-vertices on C. After this modification, the resulting graph H has the following properties:

Property 2 *1. Each 3-legged cycle C of H has exactly one 2-vertex on C.*

2. Each 2-legged cycle C of H has exactly one 2-vertex on each of the two contour paths of C.

After we construct a SSOD $D(H)$ of H, we can obtain a SSOD $D(G)$ of G as follows: Consider any 2-vertex v that was contracted from G. Let u, w be the two neighbors of v in G. In the drawing $D(H)$, the edge (u, w) is drawn as a line segment L. We simply draw v in the middle of L. After doing this for every contracted vertex v, we get a SSOD $D(G)$ for G. Thus, without loss of generality, we assume G satisfies the conditions in Property 2 from now on.

Let G^* be the dual graph of G, but without the node corresponding to the exterior face of G. So G^* has exactly four nodes on its exterior face. Each 2-vertex of G corresponds to a pair of parallel edges in G^*. We only keep one of them in G^*. These edges in G^* are called *marked edges*.

Note that every 3-legged cycle C in G corresponds to a separating triangle C^* in G^*, and every 2-legged cycle C in G corresponds to a separating 2-cycle C^* in G^*. A 3-legged cycle C is shown in Figure 5 (a). The edges in G are drawn as dashed lines, the edges in G^* are drawn as solid lines. The nodes in G^* are drawn as empty circles. The 2-vertex g in G corresponds to two parallel edges (w, x) in G^*. We keep only one of them in G^* and (w, x) is a *marked edge*. Figure 5 (b) shows a 2-legged cycle and its corresponding separating 2-cycle in G^*.

We first outline the main ideas of our algorithm. Basically, we want to construct a rectangular dual of G^* which will be the "skeleton" of the drawing $D(G)$. However, because G^* has separating 2-cycles and 3-cycles, it is not a PTP graph and hence has no rectangular dual. We have to modify G^* to get a PTP graph $G^{*\prime}$ as follows. For each separating 2-cycle or 3-cycle C^* in G^* incident to a node x, we perform a *node split* operation on x (which will be defined later.) This operation "splits" x into two nodes and "destroys" C^*. After all separating 2-cycles and 3-cycles in G^* are destroyed, the resulting graph $G^{*\prime}$ is a PTP graph. Each node x in G^* either corresponds to a node in $G^{*\prime}$ (if x is not split); or a set of nodes in $G^{*\prime}$ (since there may be multiple separating cycles incident to x, we may have to split x multiple times.) We then find an **REL** $\mathcal{R}'$ of $G^{*\prime}$ and construct a rectangular dual $D(G^{*\prime})$ of $G^{*\prime}$ by Lemma 1. $D(G^{*\prime})$ is a "skeleton" of a SSOD $D(G)$ of G. Each face f of $D(G)$ corresponds to a node x in G^*, which either corresponds to a single rectangle in $D(G^{*\prime})$ (if x is not split), or an orthogonal polygon F that is the union of several rectangles in $D(G^{*\prime})$ (each rectangle corresponds to a split node of x.) Figure 5 (c) illustrates the drawing $D(G)$ for the graph G in Figure 5 (a) by using this process. We split the node x into two nodes x_1 and x_2 in order to destroy the separating triangle $C^* = \{u, v, x\}$. In Figure 5 (c), each rectangle corresponds to a node in G^*. The union of the two rectangles marked by x_1 and x_2 corresponds to the node x. The drawing in Figure 5 (c) is an orthogonal drawing of the graph G in Figure 5 (a). Note the location of the 2-vertex g in $D(G)$.

Since we want $D(G)$ to be a SSOD of G, we must make sure each face F in $D(G)$ is star-shaped. This is done by carefully constructing the **REL** $\mathcal{R}'$ so that certain properties are satisfied (which are discussed later.)

4.1 Node Split Operation

In this subsection, we describe the details of our algorithm based on the ideas outlined above. Let G_1^* be the graph obtained from G^* as follows:

- For each maximal separating triangle C^* in G^*, delete all interior nodes of C^*.
- For each maximal separating 2-cycle C^* in G^*, delete all interior nodes of C^*, and replace the two edges of C^* by a single edge. We call these edges the *merged 2-cycle edges*.

Clearly G_1^* is a PTP graph. By Theorem 4, G_1^* has an **REL** $\mathcal{R}_1 = \{T_1, T_2\}$. We now need to add the deleted nodes back into G_1^*. We process the separating cycles of G^* one by one.

First consider a separating triangle C^* in G^*. Let $G^*(C^*)$ denote the induced subgraph of G^* consisting of the nodes on and in the interior of C^*. Let $G_1^* \cup G^*(C^*)$ be the graph obtained by adding the interior nodes of C^* back into G_1^*. We want to construct an **REL** for $G_1^* \cup G^*(C^*)$. However, $G_1^* \cup G^*(C^*)$ is not a PTP graph because C^* is a separating triangle. We must modify $G_1^* \cup G^*(C^*)$ so that C^* is not a separating triangle in it.

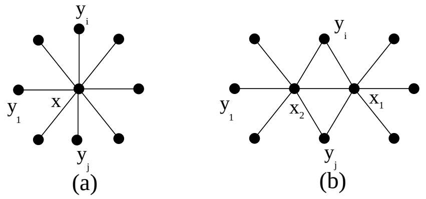


Figure 6: Node split operation. (a) Before split, (b) after split.

Let C be the 3-legged cycle in G corresponding to C^*. By Property 2, there is exactly one 2-vertex a in G on C. The vertex a corresponds to a marked edge e^*_a in G^*. e^*_a must be incident to one of the three nodes of C^*. Let x denote this node. We say the separating triangle C^* is *assigned to* x. (In Figure 5 (a), the marked edge $e^* = (x, w)$ in G^* corresponds to the 2-vertex g in G. e^* is incident to the node x. So the separating triangle $C^* = \{u, v, x\}$ is assigned to x.) We perform the *node split* operation at x defined below:

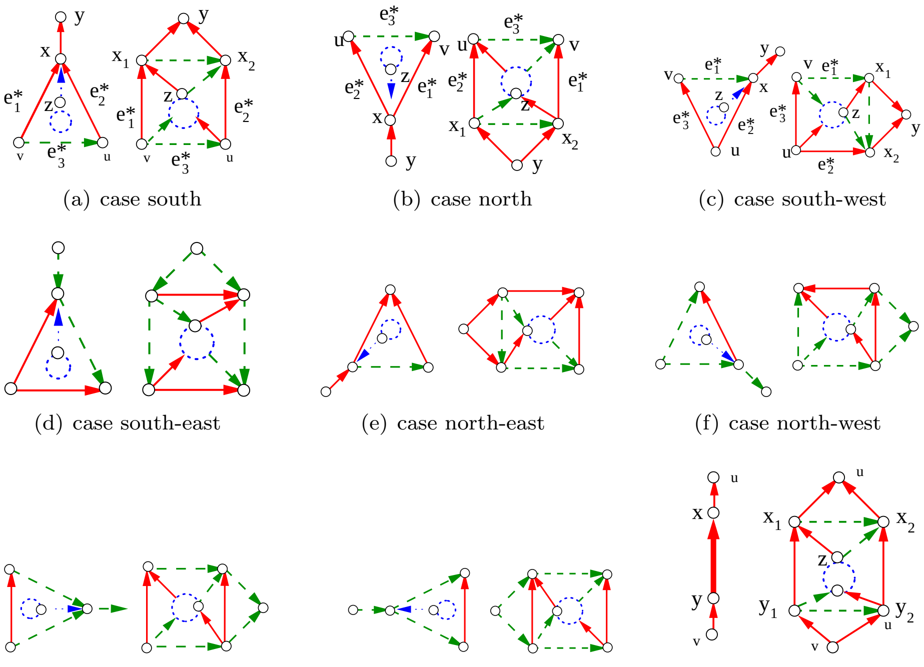
Definition 3 *Let x be an interior node of G^* with neighbors $y_1, ..., y_p$ in clockwise order (see Figure 6.) The node split operation at x with respect to the two edges (x, y_i) and (x, y_j) $(1 \leq i < j \leq p)$ is:*

1. *Delete x. Create two new nodes x_1 and x_2. Add a new edge (x_1, x_2).*
2. *For $t = i + 1, ..., j - 1$, replace the edge (x, y_t) by (x_1, y_t). For $t = j + 1, j + 2, ..., p, 1, ..., i - 1$, replace the edge (x, y_t) by (x_2, y_t).*
3. *Replace the two edges (x, y_i) and (x, y_j) by four new edges: (x_1, y_i), (x_2, y_i) and (x_1, y_j), (x_2, y_j).*

Let e^*_1, e^*_2 and e^*_3 be the three edges of C^*. Two of them, say e^*_1 and e^*_2, are incident to x. Depending on the pattern of these two edges in $\mathcal{R}_1$, there are eight cases for node split operation at x (see Figure 7 (a) - (h).) If both e^*_1 and e^*_2 are T_1 edges entering x, we call it the case south. If e^*_1 is a T_2 edge entering x and e^*_2 is a T_1 edge entering x, we call it the case southwest. The other six cases are respectively called the case southeast, east, northeast, north, northwest and west.

For example, consider the case south. We split x with respect to two edges: (z, x) is the marked edge in G^* that is in the interior of C^*; and (x, y) is a T_1 edge in the exterior of C^* leaving x (we will specify how to pick the edge (x, y) later.) In Figure 7 (a), the left figure for the case south shows the edge pattern of C^* before the node split operation. The right figure shows the edge pattern of C^* after the node split operation. In Figure 7 (a), a blue dotted circle indicates the component inside C^* that was deleted. The blue dotted arrow (z, x) indicates the marked edge inside C^*.

Note that when looking from outside of C^*, the patterns of the involved edges are identical before and after the node split operation. After splitting



(a) case south (b) case north (c) case south-west

(d) case south-east (e) case north-east (f) case north-west

(g) case west (h) case east (i) case 2-cycle

Figure 7: Cases of node split operation.

x into two nodes x_1 and x_2, each of the two edges (z, x) and (x, y) is split into two edges. C^* becomes a quadrangle with four exterior nodes x_1, x_2, u, v in clockwise order. We recursively construct an **REL** $\mathcal{R}(C^*)$ for $G^*(C^*)$ with x_1, x_2, u, v as the north, east, south and west node respectively. Now we put the nodes and the edges in the interior of the subgraph $G^*(C^*)$ back into G_1^*, together with the edge pattern specified in $\mathcal{R}(C^*)$. It is easy to see that after these operations, we get a valid **REL** of the graph $G_1^* \cup G^*(C^*)$.

The other cases are similar as shown in Figure 7 (b) - (h). Readers are invited to verify that, for each of the eight cases, we get a valid **REL** of the graph $G_1^* \cup G^*(C^*)$ after splitting x and putting $\mathcal{R}(C^*)$ back into $\mathcal{R}_1$. For the cases south, east, north and west, the pattern of the third edge e_3^* of C^* is unique (as shown in Figure 7 (a), (b), (g) and (h).) For the other four cases, there are two possible patterns for the edge e_3^*. Only one pattern of e_3^* is shown in Figure 7 (c), (d), (e) and (f). The other pattern of e_3^* is similar. (For example, for the case southwest, the edge e_3^* can also be in T_2 directed from v to u. In this case, e_3^* remains in T_2 directed from v to u after the node split operation.)

Next consider a separating 2-cycle C^* in G^*. C^* corresponds to a merged 2-cycle edge $e^* = (x, y)$ for some nodes x and y in G_1^*. Let C be the 2-legged cycle in G corresponding to C^*. By Property 2, C has two 2-vertices, a and b, one on each of its two contour paths. a and b correspond to two marked edges e^*_a and e^*_b in G^*. One of them, say e^*_a, is incident to the node x. The other (e^*_b) is incident to the node y. We say e^* is *assigned to* both x and y. Or equivalently, we say the separating 2-cycle C^* is assigned to both x and y. (In Figure 5 (b), the edges (x, v) and (y, w) are two marked edges in G^*. They are incident to x and y, respectively. So the separating 2-cycle $C^* = \{x, y\}$ is assigned to both x and y.) The processing of C^* is similar to a separating triangle. The only difference is that we need to split both x and y. Depending on the pattern of $e^* = (x, y)$ in $\mathcal{R}_1$, there are four cases. For example, if $e^* = y \rightarrow x$ is in T_1, then we split x according to the case south, and split y according to the case north. (See Figure 7 (i), case 2-cycle.) After performing these two node split operations, C^* becomes a quadrangle with four exterior nodes x_1, x_2, y_2, y_1 in clockwise order. We recursively construct an **REL** $\mathcal{R}(C^*)$ for $G^*(C^*)$ with x_1, x_2, y_2, y_1 as the north, east, south and west nodes respectively. Putting $\mathcal{R}_1$ and $\mathcal{R}(C^*)$ together, we get a valid **REL** of $G^*_1 \cup G^*(C^*)$.

4.2 The Edge Pattern around a Node

Although we can process the separating cycles of G^* in arbitrary order to add all deleted nodes back into G^*_1, doing so does not guarantee a SSOD of G at the end. Consider a node x in G^*_1. Let $\mathcal{C}$ be the set of all separating cycles of G^* assigned to x. If $\mathcal{C}$ contains several separating cycles, x must be split multiple times in order to destroy all separating cycles in $\mathcal{C}$. The problem of breaking all separating triangles has been studied in [1]. However, our goal here is different. To make sure the union of the rectangles corresponding to these split nodes constitutes a star-shaped orthogonal polygon, we must split the node x carefully as described below.

Figure 8 (a) shows the general pattern of the edges in G^*_1 around x with respect to the **REL** $\mathcal{R}_1 = \{T_1, T_2\}$. (In Figure 8 (a), a blue dotted circle indicates the component inside a separating triangle C^* assigned to x. The blue dotted arrow indicates the marked edge inside C^*. A thick line indicates a merged 2-cycle edge assigned to x.) We partition $\mathcal{C}$ into four subsets (some subsets may be empty):

- $\mathcal{C}_S = \{C^* \in \mathcal{C} \mid C^*$ is a case south or southwest separating triangle, or a case south separating 2-cycle $\}$

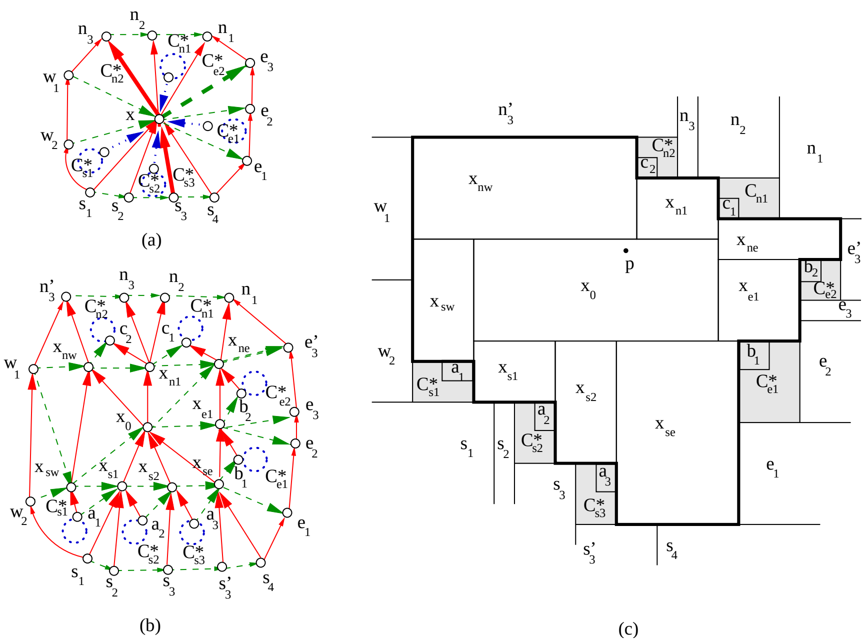
  Let $m_S = |\mathcal{C}_S|$. Denote the separating cycles in $\mathcal{C}_S$ by C^*_{si} $(1 \leq i \leq m_S.)$

- $\mathcal{C}_E = \{C^* \in \mathcal{C} \mid C^*$ is a case east or southeast separating triangle or a case east separating 2-cycle $\}$

  Let $m_E = |\mathcal{C}_E|$. Denote the separating cycles in $\mathcal{C}_E$ by C^*_{ei} $(1 \leq i \leq m_E.)$

- $\mathcal{C}_N = \{C^* \in \mathcal{C} \mid C^*$ is a case north or northeast separating triangle or a case north separating 2-cycle $\}$

  Let $m_N = |\mathcal{C}_N|$. Denote the separating cycles in $\mathcal{C}_N$ by C^*_{ni} $(1 \leq i \leq m_N.)$



(a)

(b)

(c)

Figure 8: (a) The edge pattern around a node x, (b) the subgraph created for x, (c) the orthogonal drawing of the subgraph in (b).

- $\mathcal{C}_W = \{C^* \in \mathcal{C} \mid C^*$ is a case west or northwest separating triangle or a case west separating 2-cycle $\}$

  Let $m_S = |\mathcal{C}_S|$. Denote the separating cycles in $\mathcal{C}_S$ by C^*_{wi} $(1 \le i \le m_W.)$

We create a subgraph around x as follows (see Figure 8 (b)):

- Replace x by a new node x_0 and create a cycle K around x_0. K contains four corner nodes $x_{sw}, x_{se}, x_{ne}, x_{nw}$. The edge $x_{sw} \to x_0$ is in T_2. The edge $x_{se} \to x_0$ is in T_1. The edge $x_0 \to x_{ne}$ is in T_2. The edge $x_0 \to x_{nw}$ is in T_1.

- Between x_{sw} and x_{se}, K has a sub-path K_S containing $\max\{1, m_S\}$ edges. All edges in K_S are in T_2 directed counterclockwise. The nodes on K_S are named as x_{si} $(1 \le i \le m_S - 1)$ counterclockwise. For $1 \le i < m_S$, the edge $x_{si} \to x_0$ is in T_1. For $1 \le i \le m_S$, the edge $(x_{s(i-1)}, x_{si})$ is used to destroy the separating cycle C^*_{si}. Namely, $(x_{s(i-1)}, x_{si})$ is an edge of the quadrangle obtained from C^*_{si}. Here $x_{s0} = x_{sw}$ and $x_{sm_S} = x_{se}$.

- Between x_{se} and x_{ne}, K has a sub-path K_E containing $\max\{1, m_E\}$ edges. All edges in K_E are in T_1 directed counterclockwise. The nodes on K_E are named as x_{ei} $(1 \le i \le m_E - 1)$ counterclockwise. For $1 \le i < m_E$, the edge $x_0 \to x_{ei}$ is in T_2. For $1 \leq i \leq m_E$, the edge $(x_{e(i-1)}, x_{ei})$ is used to destroy the separating cycle C^*_{ei}. Namely, $(x_{e(i-1)}, x_{ei})$ is an edge of the quadrangle obtained from C^*_{ei}. Here $x_{e0} = x_{se}$ and $x_{em_E} = x_{ne}$.

- Between x_{ne} and x_{nw}, K has a sub-path K_N containing $\max\{1, m_N\}$ edges. All edges in K_N are in T_2 directed clockwise. The nodes on K_N are named as x_{ni} $(1 \leq i \leq m_N - 1)$ counterclockwise. For $1 \leq i < m_N$, the edge $x_0 \to x_{ni}$ is in T_1. For $1 \leq i \leq m_N$, the edge $(x_{n(i-1)}, x_{ni})$ is used to destroy the separating cycle C^*_{ni}. Namely, $(x_{n(i-1)}, x_{ni})$ is an edge of the quadrangle obtained from C^*_{ni}. Here $x_{n0} = x_{ne}$ and $x_{nm_N} = x_{nw}$.
- Between x_{nw} and x_{sw}, K has a sub-path K_W containing $\max\{1, m_W\}$ edges. All edges in K_W are in T_1 directed clockwise. The nodes on K_W are named as x_{wi} $(1 \leq i \leq m_W - 1)$ counterclockwise. For $1 \leq i < m_W$, the edge $x_{wi} \to x_0$ is in T_2. For $1 \leq i \leq m_W$, the edge $(x_{w(i-1)}, x_{wi})$ is used to destroy the separating cycle C^*_{wi}. Namely, $(x_{w(i-1)}, x_{wi})$ is an edge of the quadrangle obtained from C^*_{wi}. Here $x_{w0} = x_{nw}$ and $x_{wm_W} = x_{sw}$.

When some of the sets $\mathcal{C}_S, \mathcal{C}_E, \mathcal{C}_N, \mathcal{C}_W$ are empty, they are treated as a special case. For example, suppose $\mathcal{C}_W = \emptyset$. Then K_W just contains one T_1 edge $x_{sw} \to x_{nw}$, and we split the T_2 edge $w_1 \to x$ into two T_2 edges $w_1 \to x_{sw}$ and $w_1 \to x_{nw}$ (see Figure 8 (b)).

Figure 8 (a) and (b) show an example of this construction. The south neighbors of x are s_1, s_2, s_3, s_4. The east neighbors of x are e_1, e_2, e_3. The north neighbors of x are n_1, n_2, n_3. The west neighbors of x are w_1, w_2.

$\mathcal{C}_S = \{C^*_{s1}, C^*_{s2}, C^*_{s3}\}$ (C^*_{s3} is the separating 2-cycle represented by the merged 2-cycle edge (s_3, x)). For $i = 1, 2, 3$, C^*_{si} contains a marked edge (a_i, x) in its interior.

$\mathcal{C}_E = \{C^*_{e1}, C^*_{e2}\}$ (C^*_{e2} is the separating 2-cycle represented by the merged 2-cycle edge (x, e_3)). For $i = 1, 2$, C^*_{ei} contains a marked edge (b_i, x) in its interior.

$\mathcal{C}_N = \{C^*_{n1}, C^*_{n2}\}$ (C^*_{n2} is the separating 2-cycle represented by the merged 2-cycle edge (x, n_3)). For $i = 1, 2$, C^*_{ni} contains a marked edge (c_i, x) in its interior.

$\mathcal{C}_W = \emptyset$.

Note that, for each separating 2-cycle represented by a merged 2-cycle edge e^*, both end nodes of e^* are split after this operation. For example, the separating 2-cycle C^*_{s3} (represented by $e^* = (s_3, x)$) becomes a quadrangle with nodes $x_{s2}, x_{se}, s'_3, s_3$ (s'_3 is a split node from s_3.)

This construction deals with the most general case. If some of the sets $\mathcal{C}_S, \mathcal{C}_E, \mathcal{C}_N, \mathcal{C}_W$ are empty, the construction can be simplified. For example, if only $\mathcal{C}_S$ is not empty, we can simply pick a T_1 edge $x \to y$, and split all separating cycles C^*_{si} with respect to this edge.

Figure 8 (c) shows the portion of the rectangular dual R' for the nodes in the subgraph shown in Figure 8 (b). Each small rectangle in R' is labeled by the node it represents. Let r_x be the union of the rectangle x_0 and all rectangles $x_{\alpha i}$ ($\alpha \in \{s, e, n, w\}$ and $1 \leq i \leq m_\alpha$.) This orthogonal polygon r_x is the face in the drawing $D(G)$ corresponding to the node x in G_1^*. In Figure 8 (c), r_x is outlined by the thick line segments. A shaded rectangle in R' indicates the region to draw the interior nodes in a separating cycle $C_{\alpha i}^*$. To illustrate the location of 2-vertices, consider the cycle C_{s2}^*. The node a_2 is in the interior of C_{s2}^*. The edge (a_2, x) is a marked edge in G^* and corresponds to a 2-vertex in G. The northeast corner of the rectangle a_2 in Figure 8 (c) is this 2-vertex.

Lemma 2 *For any node x in G_1^*, the orthogonal polygon r_x is star-shaped.*

Proof: If x is not split, then r_x is a rectangle and the claim is trivially true. Suppose x is split. r_x is obtained by adding the rectangles $x_{\alpha i}$ ($\alpha \in \{s, e, n, w\}$ and $1 \leq i \leq m_\alpha$) to the rectangle x_0. Let P_S be the lower envelop of r_x. P_S consists of the lower boundary of the rectangles $x_{s0}, x_{s1}, \ldots, x_{sm_S-1}, x_{sm_S}$ (where $x_{s0} = x_{sw}$ and $x_{sm_S} = x_{se}$.) For $1 \leq i \leq m_S$, there is a marked edge (a_i, x) in the interior of the separating cycle C_{si}^*. Note that $a_i \to x_{s(i-1)}$ is a T_1 edge and $a_i \to x_{si}$ is a T_2 edge. So the rectangle x_{a_i} must touch the lower side of the rectangle $x_{s(i-1)}$ and touch the left side of the rectangle x_{si}. So the lower side of x_{si} must be lower than the lower side of $x_{s(i-1)}$. Since this is true for any $1 \leq i \leq m_S$, the lower envelop P_S of r_x must be a downward staircase-like poly-line, with the lower side of x_{se} as its lowest horizontal segment.

Similarly, the upper envelop P_N of r_x must be an upward staircase-like poly-line (from right to left, namely from x_{ne} to x_{nw}) with the upper side of x_{nw} as the highest horizontal segment. Because $x_{se} \to x_0$ is the first clockwise T_1 edge entering x_0 and $x_0 \to x_{nw}$ is the first clockwise T_1 edge leaving x_0, by Lemma 1, there is a vertical strip L_v in the drawing R' that intersects x_{se}, x_0, x_{nw}. Any point p in the region $x_0 \cap L_v$ can see the entire lower envelop P_S and the entire upper envelop P_N. (See Figure 8 (c)).

Similarly, we can show the left envelop P_W of r_x is a staircase-like poly-line (from the left side of x_{nw} to the left side of x_{sw}), with the left side of x_{sw} as the leftmost vertical segment. The right envelop P_E of r_x is a staircase-like poly-line (from the right side of x_{se} to the right side of x_{ne}), with the right side of x_{ne} as the rightmost vertical segment. Because $x_{sw} \to x_0$ is the first clockwise T_2 edge entering x_0 and $x_0 \to x_{ne}$ is the first clockwise T_2 edge leaving x_0, by Lemma 1, there is a horizontal strip L_h in the drawing R' that intersects x_{sw}, x_0, x_{ne}. Any point p in the region $x_0 \cap L_h$ can see the entire left envelop P_W and the entire right envelop P_E. (See Figure 8 (c)).

Then the entire polygon r_x is visible from any point p in the region $x_0 \cap L_v \cap L_h$. □

4.3 Algorithm

Now we can describe our algorithm:

Algorithm SSOD-Draw:
Input: A plane graph G_0 that satisfies the conditions in Theorem 2.

1. Contract unneeded 2-vertices from G_0 so that the resulting graph satisfies the conditions in Property 2. Denote the resulting graph by G.

2. Construct the dual graph G^* of G.

3. Construct the graph G^*_1, by deleting all nodes in the interior of maximal separating cycles in G^*.

4. Construct a **REL** $\mathcal{R}_1$ of G^*_1.

5. By using the procedures described Subsections 4.1 and 4.2, perform node split operation for all nodes x with at least one maximal separating cycle C^* assigned to it. Then C^* becomes a quadrangle. Make recursive call to construct a **REL** $\mathcal{R}(C^*)$ for $G^*(C^*)$. Let $G^{*\prime}$ be the PTP graph obtained from G^*_1 by adding all deleted nodes back into G^*_1. Let $\mathcal{R}'$ be the **REL** of $G^{*\prime}$ obtained after this process.

6. Construct a rectangular dual R' of $G^{*\prime}$ by using $\mathcal{R}'$, as described in Lemma 1.

7. Let $D(G)$ be the orthogonal drawing of G obtained from R' as described above.

8. Add the 2-vertices contracted during Step 1 back into $D(G_0)$.

Corollary 1 *The drawing $D(G)$ constructed by Algorithm* **SSOD-Draw** *is a SSOD without bends for G.*

Proof: In Lemma 2, we have shown that for any node x in G^*_1, the orthogonal polygon r_x corresponding to x is star-shaped. Any node y of G^* not in G^*_1 is in the interior of a maximal separating cycle C^*. The orthogonal polygon r_y for y in $D(G)$ is contained in the drawing for $G^*(C^*)$. Our argument can be recursively applied to the drawing of $G^*(C^*)$ to show r_y is star-shaped. Hence $D(G)$ is a SSOD of G. During the entire process, no bends are introduced. □

Next we briefly analyze the run time of Algorithm **SSOD-Draw**. Let n_1 be the size (namely the number of nodes plus the number of edges) of G^*_1. All steps in Algorithm **SSOD-Draw** involving G^*_1 can be done in $O(n_1)$ time by Theorem 4 and basic algorithmic techniques for planar graphs.

Let $\mathcal{S}$ denote the set of separating cycles in G^*. We can organize $\mathcal{S}$ into a tree like structure $\mathcal{S}_T$ as follows [13]. The root t_0 of $\mathcal{S}_T$ corresponds to the entire graph. Let $C^*_1, \ldots C^*_k$ be all maximal separating cycles of G^*. Then t_0 has k children $t_1, \ldots, t_k$. Each t_i corresponds to C^*_i $(1 \leq i \leq k.)$ Below each t_i, we recursively construct the tree representing the separating cycles within the subgraph $G^*(C^*_i)$. The tree structure $\mathcal{S}_T$ can be constructed in linear time [13, 9]. By using this data structure, we can show that, for each maximal separating cycle C^*, the time needed to process $G^*(C^*)$ is linear of the size of $G^*(C^*)$. Therefore the entire algorithm takes linear time. In summary we have:

Theorem 5 *Let G be a graph that satisfies the conditions in Theorem 2. Then G has a SSOD without bends, which can be constructed in linear time.*

5 L-shaped Orthogonal Drawing

Let G^* be an internally triangulated plane graph with a quadrangular exterior face. If G^* has separating triangles, then it does not have a rectangular dual. However, it is possible to represent G^* by an *L-shaped modules representation* (LSMR) as follows [13]: An LSMR of G^* is a rectangle R subdivided into a set of orthogonal *modules*. Each module is either a rectangle or an L-shaped polygon. Each node u of G^* corresponds to a module r_u in R. Two nodes u, v in G^* are adjacent in G^* if and only if their corresponding modules r_u and r_v share a common boundary.

Not every internally triangulated plane graph G^* has LSMR. A necessary and sufficient condition for G^* to have such a representation was obtained in [13], which is described below.

Let $\mathcal{S}$ be the set of all separating triangles in G^*. A *nested separating triangle sequence* (NSTS) assigned to a node x in G^* is a sequence of separating triangles $\mathcal{T} = \{C_1, C_2, \ldots, C_k\} \subset \mathcal{S}$ such that: (1) C_i is contained in the interior of C_{i+1} $(1 \leq i < k)$; and (2) all C_i are incident to the node x. A *valid assignment of* $\mathcal{S}$ is a partition of $\mathcal{S}$ into a set of NSTS $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_q$ such that each node x in G^* is assigned at most one $\mathcal{T}_i$. The following results are obtained in [13].

Theorem 6 *[13] Let G^* be an internally triangulated plane graph with a quadrangular exterior face.*

1. *G^* has a LSMR if and only if there exists a valid assignment of $\mathcal{S}$;*
2. *The existence of a valid assignment of $\mathcal{S}$ can be tested in $O(n^{1.5})$ time by finding a perfect matching in an auxiliary graph.*
3. *If a valid assignment of $\mathcal{S}$ is given, a LSMR of G^* can be constructed in $O(n^2)$ time.*

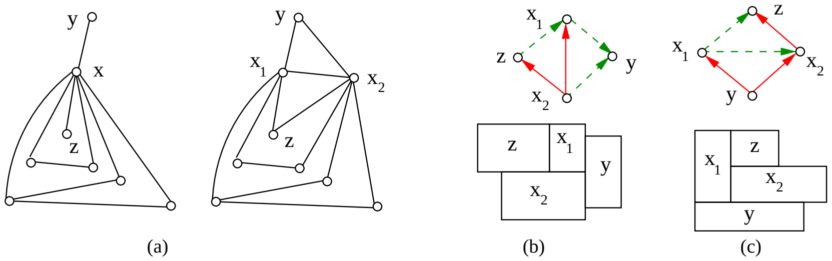


Figure 9: (a) Splitting x destroys all three separating triangles assigned to x, (b) and (c) two examples of L-shaped polygon for x.

In [14], it was shown that if a valid assignment of $\mathcal{S}$ is given, then a LSMR can be constructed in linear time. In this section, we show that if a valid assignment of $\mathcal{S}$ is given, a LSMR of G^* can be constructed in $O(n)$ time. So our algorithm is an alternative of the algorithm in [14]. Our method is totally different from the methods used in [14].

The basic idea is that we can destroy all separating triangles in $\mathcal{S}$ by using node split operation. Let $\mathcal{T} = \{C_1, \ldots, C_k\}$ be a NSTS assigned to a node x in G^*. If we split the node x with respect to an edge (z, x) in the interior of C_1 and an edge (x, y) in the exterior of C_k, then all C_i $(1 \leq i \leq k)$ are destroyed. See Figure 9 (a).

We apply the same method in Section 4 to solve this problem. Suppose that we are given a valid assignment of $\mathcal{S}$. Let G_1^* be the graph obtained from G^* by deleting all nodes in the interior of all maximal separating triangles in G^*. Then G_1^* is a PTP graph. So we can find a **REL** $\mathcal{R}_1 = \{T_1, T_2\}$ for G_1^*. Now we want to put the deleted nodes back into G_1^*. By the definition of valid assignment, at most one maximal separating triangle C is assigned to each node x in G_1^*. So there is no need to use the complicated scheme to split x. We simply apply one of the cases shown in Figure 7. Suppose that x is split with respect to the edges (z, x) and (x, y) (where (x, y) is in the exterior of C.) After the split operation, the edge (x, y) becomes two edges (x_1, y) and (x_2, y). In all eight cases, the type of (x_1, y) and (x_2, y) are the same. Namely they must be one of the following: (1) they are both in T_1 and entering y; (2) both in T_1 and leaving y; (3) both in T_2 and entering y; (4) both in T_2 and leaving y.

Let $D(G)$ be the drawing of G obtained by Algorithm **SSOD-Draw**. Each node x is split at most once. So the module r_x is the union of at most two rectangles. Because of the special edge pattern around the nodes x_1 and x_2, r_x is always an L-shaped polygon. (Figure 9 (b) and (c) show two examples.) Thus we have:

Theorem 7 *Let G^* be an internally triangulated plane graph with a quadrangular exterior face. Given a valid assignment of $\mathcal{S}$, a LSMR of G^* can be constructed in linear time.*

6 Conclusion

In this paper, we strengthen one of the results in the recent paper [2]. We show that if G satisfies the same conditions in [2], it not only has an orthogonally convex drawing, but also a stronger star-shaped orthogonal drawing. The method we use is quite different from the methods in [2, 12]. By using the same technique, we show that an L-shaped module representation can be constructed in $O(n)$ time. It will be interesting to see if this method can be used to solve other orthogonal drawing problems.

7 Acknowledgments

The authors would like to thank anonymous reviewers for suggesting possible improvements of our paper.

References

[1] A. Accornero, M. Ancona, and S. Varini. All separating triangles in a plane graph can be optimally "broken" in polynomial time. *Int. J. Found. Comput. Sci.*, 11(3):405–421, 2000. doi:10.1142/S012905410000020X.

[2] Y. Chang and H. Yen. On orthogonally convex drawings of plane graphs. In *21st Int. Symp. on Graph Drawing, GD 2013*, volume 8242 of *LNCS*, pages 400–411. Springer, 2013. doi:10.1007/978-3-319-03841-4_35.

[3] C. A. Duncan and M. T. Goodrich. Planar orthogonal and polyline drawing algorithms. In R. Tammassia, editor, *Handbook of Graph Drawings and Visualization*, pages 223 – 246. CRC Press, 2013.

[4] L. Gewali, M. Keil, and S. C. Ntafos. On covering orthogonal polygons with star-shaped polygons. *Inf. Sci.*, 65(1-2):45–63, 1992. doi:10.1016/0020-0255(92)90077-L.

[5] X. He. On finding the rectangular duals of planar triangular graphs. *SIAM J. Comput.*, 22(6):1218–1226, 1993. doi:10.1137/0222072.

[6] X. He. On floor-plan of plane graphs. *SIAM J. Comput.*, 28(6):2150–2167, 1999. doi:10.1137/S0097539796308874.

[7] S. Hong and H. Nagamochi. A linear-time algorithm for star-shaped drawings of planar graphs with the minimum number of concave corners. *Algorithmica*, 62(3-4):1122–1158, 2012. doi:10.1007/s00453-011-9513-6.

[8] S. Hong and H. Nagamochi. Minimum cost star-shaped drawings of plane graphs with a fixed embedding and concave corner constraints. *Theor. Comput. Sci.*, 445:36–51, 2012. doi:10.1016/j.tcs.2012.05.011.

[9] A. Itai and M. Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978. doi:10.1137/0207033.

[10] K. Koźmiński and E. Kinnen. Rectangular duals of planar graphs. *Networks*, 15(2):145–157, 1985. doi:10.1002/net.3230150202.

[11] A. Lingas, A. Wasylewicz, and P. Żyliński. Note on covering monotone orthogonal polygons with star-shaped polygons. *Information Processing Letters*, 104 (6):220 – 227, 2007.

[12] M. Rahman, T. Nishizeki, and M. Naznin. Orthogonal drawings of plane graphs without bends. *Journal of Graph Algorithms and Applications*, 7(4):335–362, 2003. doi:10.7155/jgaa.00074.

[13] Y. Sun and M. Sarrafzadeh. Floorplanning by graph dualization: L-shaped modules. *Algorithmica*, 10(6):429–456, 1993. doi:10.1007/BF01891831.

[14] T. Ueckerdt. *Geometric Representation of Graphs with Low Polygonal Complexity.* PhD thesis, Technischen Universität Berlin, 2012. doi:10.14279/depositonce-3190.