Star Shaped Orthogonal Drawing

Xin He(✉) and Dayu He

Department of Computer Science and Engineering,
State University of New York at Buffalo, Buffalo, NY 14260, USA
{xinhe,dayuhe}@buffalo.edu

Abstract. 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. $D(G)$ is called *orthogonally convex* if each of its faces is an *orthogonally convex polygon* P. (Namely, for any horizontal or vertical line L, the intersection of L and P is a single line segment or empty). Recently, Chang et al. [1] gave a necessary and sufficient condition for a plane graph to have such a drawing.

$D(G)$ is called a *star-shaped orthogonal drawing* (SSOD) if each of its faces is a star-shaped polygon P. (Namely there is a point $p \in P$ such that the entire P is visible from p). Every SSOD is an orthogonally convex drawing, but the reverse is false. SSOD is visually more appealing than orthogonally convex drawings. In this paper, we show that if G satisfies the same conditions as in [1], it not only has an orthogonally convex drawing, but also a SSOD, which can be constructed in linear time.

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. [2]. 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. (See Fig. 1 (1) and (2). The point p is a bend).

Rahman et al. [8] 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 presented in [8]. In the drawing obtained in [8], 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. (Fig. 1 (3) shows an orthogonally convex polygon. The face marked by F in Fig. 1 (2) 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.

Research supported in part by NSF Grant CCR-1319732.

R. Jain et al. (Eds.): TAMC 2015, LNCS 9076, pp. 137–149, 2015.
DOI: 10.1007/978-3-319-17142-5_13

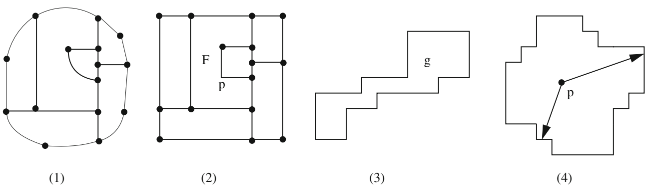

Fig. 1. (1) A plane graph G; (2) An orthogonal drawing of G; (3) An orthogonally convex polygon; (4) A star-shaped orthogonal polygon.

Chang et al. [1] gave a necessary and sufficient condition (which strengthens the conditions in [8]) for a plane graph G of maximum degree 3 to have an orthogonally convex drawing without bends. A linear time algorithm to find such a drawing was also obtained in [1].

An orthogonal polygon P is called *star-shaped* if there exists a point p in P such that the entire polygon P is *visible* from p. (See Fig. 1 (4)). 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. In this paper, we show that if G satisfies the same conditions as in [1], 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 [5], the polygons in their drawings are required to be star-shaped but not orthogonal. In [7], the problem of covering orthogonal polygons by star-shaped orthogonal polygons is studied.

The paper is organized as follows. In Sect. 2, we present the definitions and preliminary results. Section 3 describes a special rectangular dual needed by our algorithm. In Sect. 4, we present our SSOD algorithm. Section 5 concludes the paper.

2 Preliminaries

Let $G = (V, E)$ be a graph with n vertices. The *degree* of a vertex v is the number of neighbors of v in G. A vertex of degree 2 is called a 2-vertex. G is called a d-graph if the maximum degree of vertices of G is $\leq d$. A *planar graph* is a graph G that can be drawn on the plane without edge crossings. A *plane graph* is a planar graph with a fixed plane embedding. For the rest of this paper, as in [1,8], G always denotes a biconnected plane 3-graph.

The embedding of G divides the plane into a set of connected regions called *faces*. The unbounded face of G is called the *exterior face*. Other faces are called *interior faces*. The *contour* of a face is the cycle formed by the vertices and edges on the boundary of the face. The contour of the exterior face of G is denoted by $C_o(G)$. If a vertex a is on the contour of a face f, we say f is *incident to a*.

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. A cycle

C divides the plane into its interior and exterior regions. A *separating cycle* of G is a cycle C such that there are vertices in both its interior and exterior. A separating cycle may be contained in other separating cycles. A separating cycle C is called *maximal* if it's not contained in other separating cycles.

Let $D(G)$ be an orthogonal drawing of G without bends. Each cycle C of G is drawn as an orthogonal polygon $D(C)$ 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 correspond to a 2-vertex in G.

In the definition of the orthogonal drawing of G, the exterior face $C_o(G)$ is not necessarily drawn as a rectangle. However, the algorithm **Bi-Orthogonal-Draw** in [8] (which finds an orthogonal drawing of G) produces an orthogonal drawing such that $C_o(G)$ is actually a rectangle. The first step of algorithm **Bi-Orthogonal-Draw** arbitrarily selects four degree-2 vertices on $C_o(G)$ as the four corners of the exterior rectangle of the drawing. Since the drawing in [1] is produced by a modified version of the algorithm **Bi-Orthogonal-Draw**, this is also true for the drawing in [1]. Thus, without loss of generality, we assume the input to our problem is a plane graph H with four specified degree-2 vertices a, b, c, d on $C_o(H)$ in clockwise order. Our goal is to produce an orthogonal drawing $D(G)$ of G 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 simplify the presentation, we construct a graph G from H (Fig. 2 (1)):

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 an orthogonal drawing with no bends (with four corners a, b, c, d) if and only if G has an orthogonal drawing with no bends (with four corners a'', b'', c'', d'', see Fig. 2 (2)). Note that G satisfies the following properties:

Property 1.

- G is a biconnected plane 3-graph; On the exterior face $C_o(G)$, there are four degree-2 vertices and four degree-3 vertices; the degree-2 and degree-3 vertices alternate on $C_o(G)$;
- The four degree-2 vertices on $C_o(G)$ are specified as the northwest, northeast, southeast, southwest vertices.

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. C is a *k-legged cycle* if it has exactly k legs. The k leg vertices divide C into k sub-paths. Each sub-path is called a *contour path* of C.

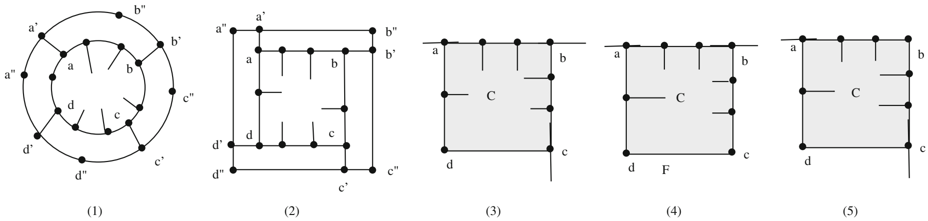

Fig. 2. (1) The construction of G from H; (2) drawings of H and G; (3) and (4) Conditions in Theorem 1; (5) Conditions in Theorem 2.

Theorem 1 [8]. *Let G be a plane graph that satisfies the conditions 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 and (2) Every 2-legged cycle C has at least two 2-vertices.*

Figure 2 (3) shows a 3-legged cycle $C = \{a, b, c, d\}$ and its orthogonal drawing. Figure 2 (4) shows a 2-legged cycle $C = \{a, b, c, d\}$ and its orthogonal drawing.

Theorem 2 [1]. *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 and (2) Every 2-legged cycle C has at least two 2-vertices, at least one on each of its two contour paths.*

Figure 2 (5) 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 Fig. 2 (4), 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 Sect. 4, we will show that if G satisfies the conditions in Theorem 2, then G 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 if and only if their corresponding faces in G share an edge as common boundary. Note that G^* is an internally triangulated plane graph and the exterior face of G^* has four nodes. A *rectangular dual* of such a graph G^* is a rectangle R divided into smaller rectangles such that the following hold:

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

See Fig. 3 (a) for an example. It's easy to see that a rectangular dual R of G^* is an orthogonal drawing $D(G)$ of the original graph G, and each face of $D(G)$ is a rectangle. Not every internally triangulated plane graph G^* has a rectangular dual. The following theorem characterizes such graphs.

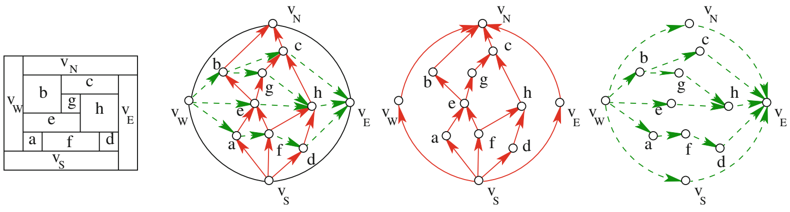

Fig. 3. (a) A rectangular dual of the graph shown in (b); (b) an **REL** $\mathcal{R} = \{T_1, T_2\}$; (c) the subgraph consisting of edges in T_1 and the 4 exterior edges oriented from v_S to v_N; (d) the subgraph consisting of edges in T_2 and the 4 exterior edges oriented from v_W to v_E (Color figure online).

Theorem 3 [6]**.** *A plane graph G^* 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; and (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 1. *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 entering v_N. All interior edges incident to v_E are in T_2 entering v_E. All interior edges incident to v_S are in T_1 leaving v_S. All interior edges incident to v_W are in T_2 leaving v_W.*

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 [3,4]**.** *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.

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 stripe 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 stripe in* R *that intersects* r_{w_1}, r_u, r_{w_2}*.*

The proof is omitted due to space limitation.

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). 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.

- Each 3-legged cycle C of H has exactly one 2-vertex on C.
- 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. 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 Fig. 4 (1). 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 cycles. g is a 2-vertex in G. It 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 4 (2) 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 R of G^* which will be the "skeleton" of the drawing

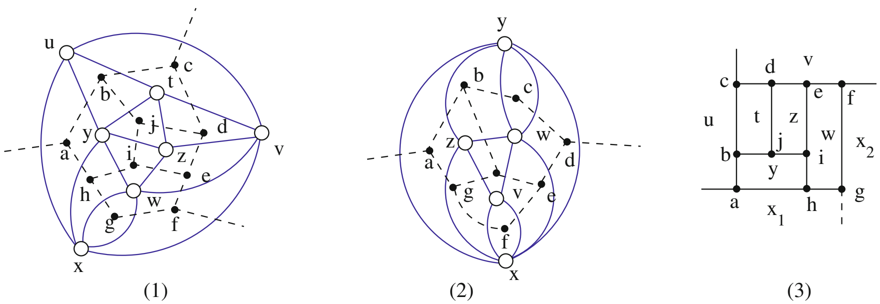

Fig. 4. (1) A 3-legged cycle $C = \{a, b, c, d, e, f, g, h\}$ and the dual separating triangle $C^* = \{u, v, x\}$; (2) A 2-legged cycle $C = \{a, b, c, d, e, f, g\}$ and the dual separating 2-cycle $C^* = \{x, y\}$. (3) The drawing of the graph in (1).

$D(G)$. 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 as follows: 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 4 (3) illustrates the drawing $D(G)$ for the graph G in Fig. 4 (1) 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 Fig. 4 (3), each rectangle corresponds to a node in $G^{*\prime}$. The union of the two rectangles marked by x_1 and x_2 corresponds to the node x. The drawing in Fig. 4 (3) is an orthogonal drawing of the graph G in Fig. 4 (1). Note the location of the 2-vertex g in $D(G)$.

4.1 Node Split Operation

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 (to be defined later). Next we describe the details of our algorithm. Let G_1^* be the graph obtained from G^* as follows:

- For each maximal separating triangle C^*, delete all interior nodes of C^*.
- For each maximal separating 2-cycle C^*, 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. Consider a maximal 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.

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 a node on C^*. Let x be this node. We say the separating triangle C^* is *assigned to* x. (In Fig. 4 (1), 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). The *node split* operation at x with respect to two specified edges (x, y_i) and (x, y_j) is illustrated in Fig. 5.

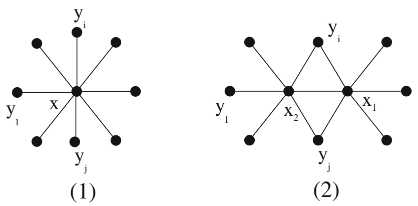

Fig. 5. Node split operation. (a) Before split; (b) After split.

Consider a separating triangle C^* assigned to x. After splitting x into two nodes, C^* becomes a quadrangle. Then we can add back the deleted interior nodes of C^*. 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 (see Fig. 6). 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 shown in Fig. 6.

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 Fig. 6, 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 Fig. 6, 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 the node split operation, x is split 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** $\mathcal{R}(C^*)$

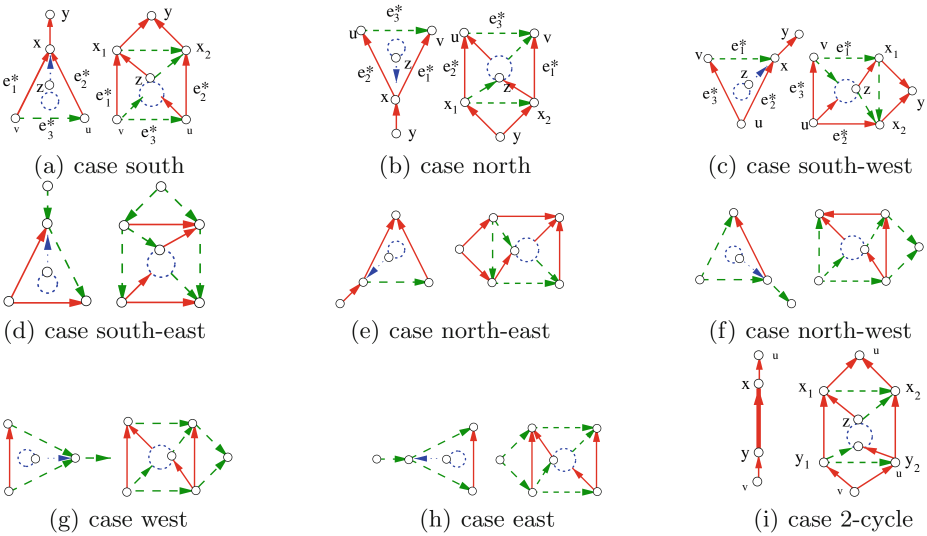

Fig. 6. Cases of node split operation (Color figure online).

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 Fig. 6. For each of the eight cases, we get a valid **REL** of the graph $G_1^* \cup G^*(C^*)$ after the node split operation.

Now consider a separating 2-cycle C^* in G^*. We want to add the interior nodes of C^* back into G_1^*. 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 Fig. 4 (2), 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 $e^* = y \to x$ is in T_1, then we split x according to the case south, and split y according to the case north. (See Fig. 6 (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 $\mathcal{C}$. 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 7 (1) 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 Fig. 7 (1), 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 cycle$\}$.
  Let $m_S = |\mathcal{C}_S|$. Denote the separating cycles in $\mathcal{C}_S$ by C_{si}^* $(1 \le i \le m_S)$.
- $\mathcal{C}_E = \{C^* \in \mathcal{C} \mid C^*$ is a case east or southeast separating cycle$\}$.
  Let $m_E = |\mathcal{C}_E|$. Denote the separating cycles in $\mathcal{C}_E$ by C_{ei}^* $(1 \le i \le m_E)$.
- $\mathcal{C}_N = \{C^* \in \mathcal{C} \mid C^*$ is a case north or northeast separating cycle$\}$.
  Let $m_N = |\mathcal{C}_N|$. Denote the separating cycles in $\mathcal{C}_N$ by C_{ni}^* $(1 \le i \le m_N)$.
- $\mathcal{C}_W = \{C^* \in \mathcal{C} \mid C^*$ is a case west or northwest separating 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 (Fig. 7 (1) and (2)):

- 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 $x_{sm_S} = x_{se}$.
- The other sides of K are similar.

When some of $\mathcal{C}_S, \mathcal{C}_E, \mathcal{C}_N, \mathcal{C}_W$ are empty, they are treated as a special case. For example, when $\mathcal{C}_W = \emptyset$, K_W just contains one T_1 edge $x_{sw} \to x_{nw}$. Then we split the edge (w_1, x) into two edges (w_1, x_{sw}) and (w_1, x_{nw}). (See Fig. 7 (2)).

Note that, for each separating 2-cycle, both end nodes of e^* are split. For example, for the separating 2-cycle C_{s3}^* represented by $e^* = (s_3, x)$, C_{s3}^* 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.

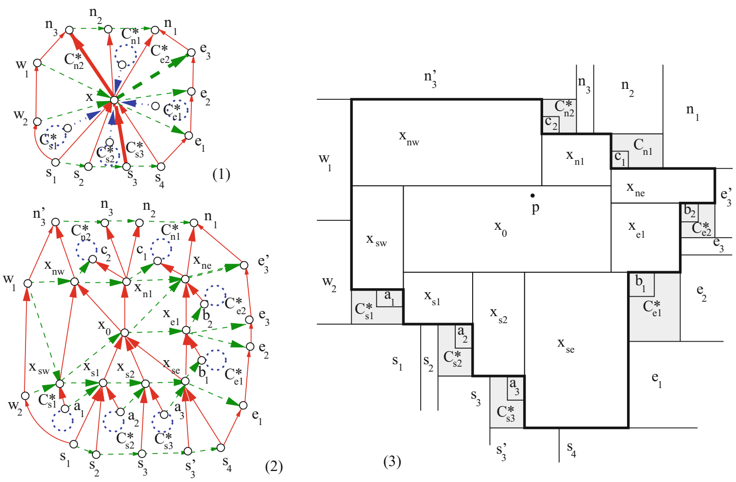

Fig. 7. (1) The edge pattern around a node x; (2) The subgraph created for x; (3) the orthogonal drawing of the subgraph in (2).

Figure 7 (3) shows an orthogonal drawing D of the nodes in the subgraph shown in Fig. 7 (2). Let r_x be the union of the rectangle x_0 and all rectangles $x_{\alpha i}$ ($\alpha \in \{s, e, n, w\}$ and $1 \le i \le m_\alpha$). This orthogonal polygon r_x is the face in the drawing $D(G)$ corresponding to the node x in G_1^*. In Fig. 7 (3), r_x is outlined by the thick line segments. A shaded rectangle indicates the region to draw interior nodes in a separating cycle $C^*_{\alpha i}$. Look at 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 it corresponds to a 2-vertex in G. The northeast corner of the rectangle a_2 in Fig. 7 (3) is this 2-vertex.

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

Proof. r_x is obtained by adding the rectangles $x_{\alpha i}$ ($\alpha \in \{s, e, n, w\}$ and $1 \le i \le m_\alpha$) 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 $1 \le i \le m_S$, 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 below the lower side of $x_{s(i-1)}$. Since this is true for any $1 \le i \le m_S$, 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, we can show that 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 stripe L_v in the drawing D 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 Fig. 7 (3)).

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 $x_{sw} \to x_0$ 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 stripe L_h in the drawing D 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 Fig. 7 (3)).

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

4.3 Algorithm

Algorithm SSOD-Draw:
Input: A graph G that satisfies the conditions in Theorem 2 and Property 2.

1. Construct the dual graph G^* of G.
2. Construct the graph G_1^*, by deleting all nodes in the interior of maximal separating cycles in G^*.
3. Construct a **REL** $\mathcal{R}_1$ of G_1^*.
4. By using the procedure described above, perform node split operation for all nodes x with at least one maximal separating cycle C^* assigned to it. When C^* is destroyed, 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 in this process.
5. Construct a rectangular dual R' of $G^{*\prime}$ by using $\mathcal{R}'$ as in Lemma 1.
6. Let $D(G)$ be the orthogonal drawing of G obtained from R' as above.

By Lemma 2, for any node x in G_1^*, the orthogonal polygon r_x corresponding to x is star-shaped. Any node y 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 a star-shaped orthogonal polygon. Hence $D(G)$ is a SSOD of G. All steps in Algorithm **SSOD-Draw** can be done in linear time by Theorem 4 and basic algorithmic techniques for planar graphs. In summary:

Theorem 5. *Let G be a graph that satisfies the conditions in Theorem 2. Then G has a SSOD drawing, which can be constructed in linear time.*

5 Conclusion

In this paper, we strengthen the result in [1]. We show that if G satisfies the same conditions as in [1], 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 used in [1,8]. It will be interesting to see if this method can be used to solve other orthogonal drawing problems.

References

1. Chang, Y.-J., Yen, H.-C.: On orthogonally convex drawings of plane graphs. In: Wismath, S., Wolff, A. (eds.) GD 2013. LNCS, vol. 8242, pp. 400–411. Springer, Heidelberg (2013)
2. Duncan, C.A., Goodrich, M.T.: Planar orthogonal and polyline drawing algorithms. In: Tammassia, R. (ed.) Handbook of Graph Drawings and Visualization, Chap. 7, pp. 223–246. CRC Press (2013)
3. He, X.: On finding the rectangular duals of planar triangular graphs. SIAM J. Comput. **22**, 1218–1226 (1993)
4. He, X.: On floor-plan of plane graphs. SIAM J. Comput. **28**, 2150–2167 (1999)
5. Hong, S.-H., Nagamochi, H.: A linear-time algorithm for star-shaped drawings of planar graphs with the minimum number of concave corners. Algorithmica **62**, 1122–1158 (2012)
6. Koźmiński, K., Kinnen, E.: Rectangular duals of planar graphs. Networks **5**, 145–157 (1985)
7. Lingas, A., Wasylewicz, A., Żyliński, P.: Note on covering monotone orthogonal polygons with star-shaped polygons. Info. Proc. Lett. **104**, 220–227 (2007)
8. Rahman, M., Nishizeki, T.: Orthogonal drawings of plane graphs without bends. J. Gr. Algorithms Appl. **7**, 335–362 (2003)