Vol.2009-AL-125 No.2
2009/7/21

Convex Drawings of Internally Triconnected Plane Graphs on $O(n^2)$ Grids

Xiao Zhou and Takao Nishizeki

In a convex grid drawing of a plane graph, every edge is drawn as a straight-line segment without any edge-intersection, every vertex is located at a grid point, and every facial cycle is drawn as a convex polygon. A plane graph G has a convex drawing if and only if G is internally triconnected. It has been known that an internally triconnected plane graph G of n vertices has a convex grid drawing on a grid of $O(n^3)$ area if the triconnected component decomposition tree of G has at most four leaves. In this paper, we improve the area bound $O(n^3)$ to $O(n^2)$, which is optimal within a coefficient. More precisely, we show that G has a convex grid drawing on a $2n \times 4n$ grid. We also present an algorithm to find such a drawing in linear time.

1. Introduction

Recently automatic aesthetic drawing of graphs has created intense interest due to their broad applications, and as a consequence, a number of drawing methods have come out[12]. The most typical drawing of a plane graph is a *straight line drawing*, in which all edges are drawn as straight line segments without any edge-intersection. A straight line drawing is called a *convex drawing* if every facial cycle is drawn as a convex polygon. One can find a convex drawing of a plane graph G in linear time if G has one[3),4),12)].

A straight line drawing of a plane graph is called a *grid drawing* if all vertices are put on grid points of integer coordinates. This paper deals with a *convex grid drawing* of a plane graph. Throughout the paper we assume for simplicity that every vertex of a plane graph G has degree three or more. Then G has a convex drawing if and only if G is "internally triconnected"[2),9),10)]. One may thus assume that G is internally triconnected. If either G is triconnected[1),2)] or the "triconnected component decomposition tree" $T(G)$ of G has two or three leaves[9)], then G has a convex grid drawing on an $(n-1) \times (n-1)$ grid, where n is the number of vertices in G. If $T(G)$ has exactly four leaves, then G has a convex grid drawing on a $2n \times n^2$ grid[11)]. Thus, G has a convex grid drawing of $O(n^3)$ area if $T(G)$ has at most four leaves.

In this paper, we improve the area bound $O(n^3)$ above to $O(n^2)$, which is optimal within a coefficent because a plane graph of nested triangles needs an $\Omega(n^2)$ area in any straight line drawing[5)]. More precisely, we show that an internally triconnected plane graph G has a convex grid drawing on a $2n \times 4n = O(n^2)$ grid if $T(G)$ has exactly four leaves, and present an algorithm to find such a drawing in linear time.

†1 Graduate School of Information Sciences, Tohoku University, Sendai 980-8579, Japan.
This work is supported in part by a Grant-in-Aid for Scientific Research (C) 19500001 from Japan Society for the Promotion of Science (JSPS).

2. Outline of our algorithm

In this section, we outline our algorithm, which is a modification of the algorithm in[11)].

The plane graph G in Fig. 1(a) is internally triconnected, the triconnected components of G are depicted in Fig. 2(b), and the triconnected component decomposition tree $T(G)$ of G having four leaves l_1, l_2, l_3 and l_4 is depicted in Fig. 2(c). We draw G so that the contour of the outer face of G is a rectangle, as illustrated in Fig. 1(e). We first appropriately choose four vertices a_1, a_2, a_3 and a_4 as the four apices of the rectangular contour. We then divide G into an upper subgraph G_u and a lower subgraph G_d, as illustrated in Fig. 1(b), so that G_u contains a_1 and a_2 and G_d contains a_3 and a_4. Using the "pentagon algorithm" in[11)], we then obtain "inner convex" grid drawings D_u of G_u and D_d of G_d, both of $O(n^2)$ area, as illustrated in Figs. 1(c) and (d). More precisely, D_u has width $W(D_\mathrm{u}) \leq 2n_\mathrm{u} - 2$ and height $H(D_\mathrm{u}) \leq 2n_\mathrm{u} - 2$, and D_d has width $W(D_\mathrm{d}) \leq 2n_\mathrm{d} - 2$ and height $H(D_\mathrm{d}) \leq 2n_\mathrm{d} - 2$, where n_u and n_d are the numbers of vertices in G_u and G_d, respectively, and hence $n_\mathrm{u} + n_\mathrm{d} = n$. We then shift either vertex a_1 to the left or a_3 to the right so that these two drawings have the same width $\max\{2n_\mathrm{d} - 2, 2n_\mathrm{u} - 2\}$. We next arrange D_d and D_u so that $y(a_3) = y(a_4) = 0$ and $y(a_1) = y(a_2) = H(D_\mathrm{d}) + H(D_\mathrm{u}) + \max\{2n_\mathrm{d} - 2, 2n_\mathrm{u} - 2\} + 1$, as illustrated in Fig. 1(e), where $y(a_1)$, $y(a_2)$, $y(a_3)$ and $y(a_4)$ are the y-coordinates of a_1, a_2,

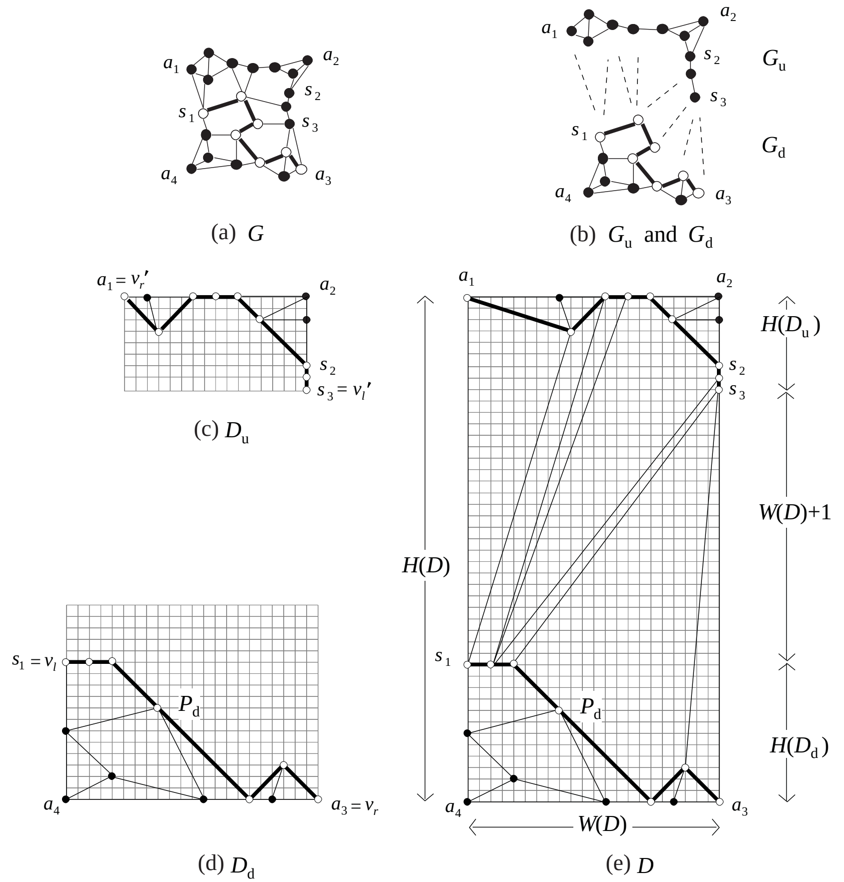

Fig. 1 (a) A plane graph G, (b) subgraphs G_{u} and G_{d}, (c) a drawing D_{u} of G_{u}, (d) a drawing D_{d} of G_{d}, and (e) a convex grid drawing D of G.

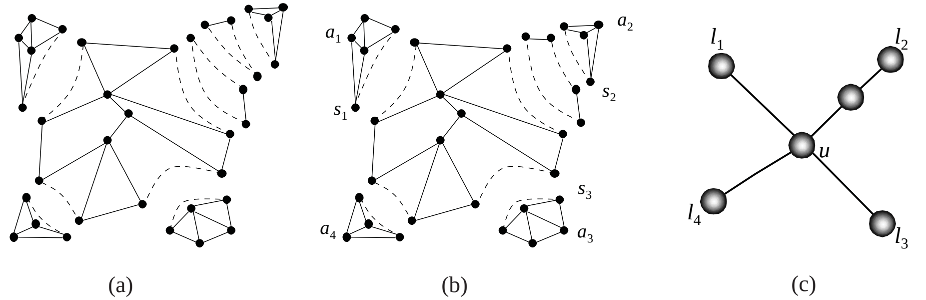

Fig. 2 (a) Split components of the graph G in Fig. 1(a), (b) triconnected components of G, and (c) a decomposition tree $T(G)$.

a_3 and a_4, respectively. We finally draw, by straight line segments, all the edges of G that are contained in neither G_{u} nor G_{d}. Thus, the width $W(D)$ of the resulting drawing D of G is

$$W(D) \leq \max\{2n_{\mathrm{d}} - 2, 2n_{\mathrm{u}} - 2\} < 2n,$$

and the height $H(D)$ of D is

$$H(D) \leq 2n_{\mathrm{d}} - 2 + 2n_{\mathrm{u}} - 2 + \max\{2n_{\mathrm{d}} - 2, 2n_{\mathrm{u}} - 2\} + 1 < 4n.$$

Hence, the area of the drawing D is $2n \times 4n = O(n^2)$. The selection of apices a_1, a_2, a_3 and a_4, the division of G to G_{u} and G_{d} and some others are different from those in[11].

3. Preliminaries

In this section, we give some definitions, and outline the pentagon algorithm in[11] which is used to draw G_{d} and G_{u}.

We denote by $G = (V, E)$ an undirected connected simple graph with vertex set V and edge set E. We often denote the set of vertices of G by $V(G)$ and the set of edges by $E(G)$. An edge joining vertices u and v is denoted by (u, v).

A $W \times H$ *integer grid* consists of $W + 1$ vertical grid lines and $H + 1$ horizontal grid lines, and has a rectangular contour. We call W and H the *width* and *height* of the integer grid, respectively. We denote by $W(D)$ the width of the minimum integer grid enclosing a grid drawing D of a graph, and by $H(D)$ the height of D.

A plane graph G divides the plane into connected regions, called *faces*. The boundary of a face is called a *facial cycle*. We denote by $F_o(G)$ the outer facial cycle of G. A vertex on $F_o(G)$ is called an *outer vertex*, while a vertex not on $F_o(G)$ is called an *inner vertex*. In a convex drawing D of a plane graph G, all facial cycles must be drawn as convex polygons. The convex polygonal drawing of $F_o(G)$ is called the *outer polygon* of D. We call a vertex of a polygon an *apex* in order to avoid the confusion with a vertex of a graph.

We call a vertex v of a connected graph G a *cut vertex* if its removal from G results in a disconnected graph, that is, $G - v$ is not connected. A connected graph G is *biconnected* if G has no cut vertex. We call a pair $\{u, v\}$ of vertices in a biconnected graph G a *separation pair* if its removal from G results in a disconnected graph, that is, $G - \{u, v\}$ is not connected. A biconnected graph G is *triconnected* if G has no separation pair. A biconnected plane graph G is *internally triconnected* if, for any separation pair $\{u, v\}$ of G, both u and v are outer vertices and each connected component of $G - \{u, v\}$ contains an outer vertex.

Let $G = (V, E)$ be a biconnected graph, and let $\{u, v\}$ be a separation pair of G. Then, G has two subgraphs $G'_1 = (V_1, E'_1)$ and $G'_2 = (V_2, E'_2)$ satisfying the following two conditions.

(a) $V = V_1 \bigcup V_2$, $V_1 \bigcap V_2 = \{u, v\}$; and

(b) $E = E'_1 \bigcup E'_2$, $E'_1 \bigcap E'_2 = \emptyset$, $|E'_1| \geq 2$, $|E'_2| \geq 2$.

For a separation pair $\{u, v\}$ of G, $G_1 = (V_1, E'_1 + (u, v))$ and $G_2 = (V_2, E'_2 + (u, v))$ are called the *split graphs* of G with respect to $\{u, v\}$. The new edges (u, v) added to G_1 and G_2 are called the *virtual edges*. Even if G has no multiple edges, G_1 and G_2 may have. Dividing a graph G into two split graphs G_1 and G_2 is called *splitting*. Reassembling the two split graphs G_1 and G_2 into G is called *merging*. Merging is the inverse of splitting. Suppose that a graph G is split, the split graphs are split, and so on, until no more splits are possible, as illustrated in Fig. 2(a) for the graph in Fig. 1(a) where virtual edges are drawn by dotted lines. The graphs constructed in this way are called the *split components* of G. The split components are of three types: triconnected graphs; triple bonds (i.e. a set of three multiple edges); and triangles (i.e. a cycle of length three). The *triconnected components* of G are obtained from the split components of G by merging triple bonds into a bond and triangles into a ring, as far as possible, where a *bond* is a set of multiple edges and a *ring* is a cycle. Thus the triconnected components of G are of three types: (a) triconnected graphs; (b) bonds; and (c) rings. Two triangles in Fig. 2(a) are merged into a single ring, and hence the graph in Fig. 1(a) has six triconnected components as illustrated in Fig. 2(b).

Let $T(G)$ be a tree such that each node corresponds to a triconnected component H_i of G and there is an edge (H_i, H_j), $i \neq j$, in $T(G)$ if and only if H_i and H_j are triconnected components with respect to the same separation pair, as illustrated in Fig. 2(c). We call $T(G)$ a *triconnected component decomposition tree* or simply a *decomposition tree* of G[8]. $T(G)$ has four leaves for the graph G in Fig. 1(a). (See Fig. 2(c).) If G is triconnected, then $T(G)$ consists of a single isolated node and hence $T(G)$ has exactly one leaf.

Let G be an internally triconnected plane graph such that $T(G)$ has exactly four leaves. Then every leaf of $T(G)$ is a triconnected graph and the outer polygon of every convex drawing of G must have four or more apices[11]. Our algorithm obtains a convex grid drawing of G whose outer polygon has exactly four apices and is a rectangle in particular, as illustrated in Fig. 1(e).

In Section 4, we will present an algorithm to draw G, which uses the following "canonical decomposition"[2),12)]. Let $G = (V, E)$ be an internally triconnected plane graph, and let $V = \{v_1, v_2, \cdots, v_n\}$. Let v_1, v_2 and v_n be three arbitrary outer vertices appearing counterclockwise on $F_o(G)$ in this order. We may assume that v_1 and v_2 are consecutive on $F_o(G)$; otherwise, add a virtual edge (v_1, v_2) to the original graph, and let G be the resulting graph. Let $\Pi = (U_1, U_2, \cdots, U_m)$ be an ordered partition of V into nonempty subsets $U_1, U_2, \cdots, U_m$. We denote by G_k, $1 \leq k \leq m$, the subgraph of G induced by $U_1 \bigcup U_2 \bigcup \cdots \bigcup U_k$, and denote by $\overline{G_k}$, $0 \leq k \leq m - 1$, the subgraph of G induced by $U_{k+1} \bigcup U_{k+2} \bigcup \cdots \bigcup U_m$. We say that Π is a *canonical decomposition* of G (with respect to vertices v_1, v_2 and v_n) if the following three conditions (cd1)–(cd3) hold:

(cd1) $U_m = \{v_n\}$, and U_1 consists of all the vertices on the inner facial cycle containing edge (v_1, v_2).

(cd2) For each index k, $1 \leq k \leq m$, G_k is internally triconnected.

(cd3) For each index k, $2 \leq k \leq m$, all the vertices in U_k are outer vertices of G_k, and

(a) if $|U_k| = 1$, then the vertex in U_k has two or more neighbors in G_{k-1}

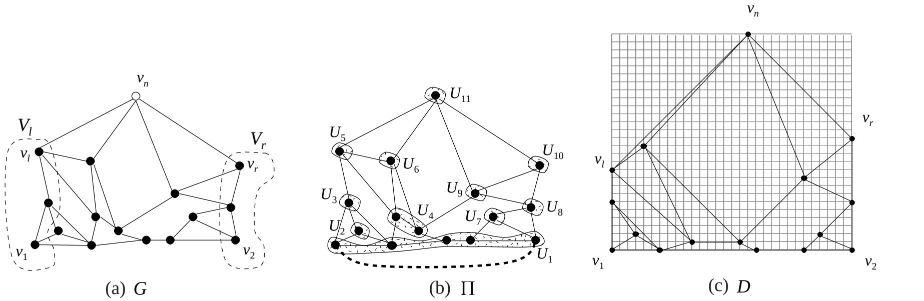

Fig. 3 (a) An internally triconnected plane graph G, (b) a canonical decomposition Π of G, and (c) a pentagonal drawing D of G.

and has one or more neighbors in $\overline{G_k}$ when $k < m$; and

(b) if $|U_k| \geq 2$, then each vertex in U_k has exactly two neighbors in G_k, and has one or more neighbors in $\overline{G_k}$.

A canonical decomposition $\Pi = (U_1, U_2, \cdots, U_{11})$ with respect to vertices v_1, v_2 and v_n of the graph in Fig. 3(a) is illustrated in Fig. 3(b). If $T(G)$ has at most three leaves, then G has a canonical decomposition[11].

Let G be a plane graph having a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to vertices v_1, v_2 and v_n, as illustrated in Fig. 3(b). Miura *et al.*[11] give a linear-time algorithm, called the *pentagon algorithm*, to find a convex grid drawing of G with a pentagonal outer polygon, as illustrated in Fig. 3(c). The algorithm is based on the so-called shift methods given by Chrobak and Kant[2] and de Fraysseix *et al.*[6], and will be used by our convex grid drawing algorithm in Section 4 to draw G_{u} and G_{d}.

We then outline the pentagon algorithm. Let v_l be an arbitrary outer vertex on the path going from v_1 to v_n clockwise on $F_{\mathrm{o}}(G)$, and let $v_r (\neq v_l)$ be an arbitrary outer vertex on the path going from v_2 to v_n counterclockwise on $F_{\mathrm{o}}(G)$, as illustrated in Fig. 3(a). Let V_l be the set of all vertices on the path going from v_1 to v_l clockwise on $F_{\mathrm{o}}(G)$, and let V_r be the set of all vertices on the path going from v_2 to v_r counterclockwise on $F_{\mathrm{o}}(G)$. The pentagon algorithm in[11] obtains a convex grid drawing of G whose outer polygon is a pentagon with apices v_1, v_2, v_r, v_n and v_l, as illustrated in Fig. 3(d).

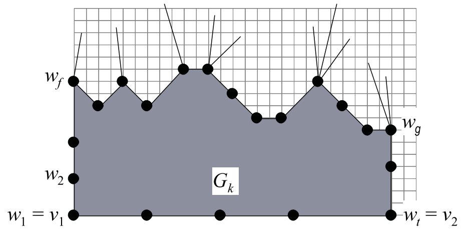

Fig. 4 Drawing D_k of graph G_k.

More precisely, the pentagon algorithm obtains an "inner convex" grid drawing D_k for each k, $1 \leq k \leq m$, in which all inner facial cycles are convex polygons. Let $F_{\mathrm{o}}(G_k) = w_1, w_2, \cdots, w_t$, $w_1 = v_1$, and $w_t = v_2$, as illustrated in Fig. 4. Let w_f be the vertex with the maximum index f among all the vertices w_i, $1 \leq i \leq t$, on $F_{\mathrm{o}}(G_k)$ that are contained in V_l. Let w_g be the vertex with the minimum index g among all the vertices w_i that are contained in V_r. Of course, $1 \leq f < g \leq t$. We denote by $\angle w_i$ the interior angle of apex w_i of the outer polygon of D_k. We call w_i a *convex apex* of the polygon if $\angle w_i < \pi$. The drawing D_k of G_k satisfies the following six conditions (sh1)–(sh6). (See Fig. 4.)

(sh1) w_1 is on the grid point $(0,0)$, and w_t is on the grid point $(2|V(G_k| - 2, 0)$.

(sh2) $x(w_1) = x(w_2) = \cdots = x(w_f)$, $x(w_f) < x(w_{f+1}) < \cdots < x(w_g)$, $x(w_g) = x(w_{g+1}) = \cdots = x(w_t)$, where $x(w_i)$ is the x-coordinate of w_i.

(sh3) Every edge (w_i, w_{i+1}), $f \leq i \leq g-1$, has slope $-1, 0$, or 1.

(sh4) The Manhattan distance between any two grid points w_i and w_j, $f \leq i < j \leq g$, is an even number.

(sh5) Every inner face of G_k is drawn as a convex polygon.

(sh6) Vertex w_i, $f+1 \leq i \leq g-1$, has one or more neighbors in $\overline{G_k}$ if w_i is a convex apex.

The pentagon algorithm obtains a convex grid drawing D of $G = G_m$ on a $W \times H$ grid with $W = 2n-2$ and $H \leq n^2 - n - 2$ in linear time[11]. Thus the area of D is $O(n^3)$. However, we observe that the algorithm obtains a convex grid drawing D of $O(n^2)$ area if one chooses $v_r = v_2$, as follows.

Lemma 3.1 For a plane graph G having a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$, the pentagon algorithm obtains an inner convex grid drawing D_k

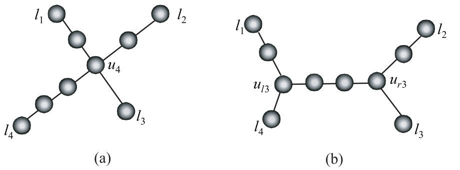

Fig. 5 Decomposition trees $T(G)$ (a) having a node of degree four and (b) having two nodes of degree three.

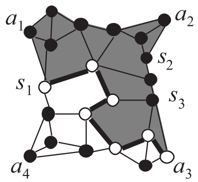

Fig. 6 Graph G.

of G_k, $1 \leq k \leq m$, such that $H(D_k) \leq W(D_k) = 2|V(G_k)| - 2$ if one chooses $v_r = v_2$.

Proof. The condition (sh1) implies $W(D_k) = 2|V(G_k)| - 2$. Therefore, it suffices to prove that $H(D_k) \leq W(D_k)$. Since $w_g = w_t = v_2$, we have $y(w_g) = y(w_t) = 0$. The condition (sh3) implies that each vertex of G_k is below the straight line of slop $-45°$ passing through w_g. Thus we have $H(D_k) \leq W(D_k)$. $\mathcal{Q.E.D.}$

Figure 8(d) depicts the convex drawing of the graph in Fig. 8(a) obtained by the pentagon algorithm with choosing $v_r = v_2$. It should be noted that the outer polygon is not a pentagon but is a quadrangle with apices v_1, v_2, v_n $(= w)$ and v_l.

4. Convex grid drawing algorithm

In this section we present a linear algorithm to find a convex grid drawing D of an internally triconnected plane graph G whose decomposition tree $T(G)$ has exactly four leaves. Such a graph G does not have a canonical decomposition, and hence none of the pentagon algorithm and those in[1),2),7),9)] can find a convex grid drawing of G. Our algorithm draws the outer facial cycle $F_o(G)$ as a rectangle as illustrated in Fig. 1(e). The algorithm first divides G into an upper subgraph G_u and a lower subgraph G_d as illustrated in Fig. 1(b), then draws G_u and G_d by using the pentagon algorithm[11)] with choosing $v_r = v_2$ as illustrated in Figs. 1(c) and (d), and finally combine these two drawings to a convex grid drawing of G as illustrated in Fig. 1(e).

4.1 Division

We first explain how to divide G into G_u and G_d. (See Figs. 1(a) and (b).) One may assume that the four leaves l_1, l_2, l_3 and l_4 of $T(G)$ appear clockwise in $T(G)$ in this order as illustrated in Fig. 5. Clearly, either exactly one internal node u_4 of $T(G)$ has degree four and each of the other internal nodes has degree two as illustrated in Fig. 5(a), or exactly two internal nodes u_{l3} and u_{r3} have degree three and each of the other internal nodes has degree two as illustrated in Fig. 5(b), where node u_{l3} is assumed to be arranged to the left and node u_{r3} to the right.

Since each vertex of G is assumed to have degree three or more, all the four leaves of $T(G)$ are triconnected graphs. Moreover, every triconnected component of G having degree three or four in $T(G)$ is either a triconnected graph or a ring, while every bond has degree two in $T(G)$[11)]. Thus there are the following six cases (a)–(f) to consider.

(a) Node u_4 is a triconnected graph as illustrated in Fig. 7(a);
(b) Node u_4 is a ring as illustrated in Fig. 7(b);
(c) Both of nodes u_{l3} and u_{r3} are triconnected graphs as illustrated in Fig. 7(c);
(d) Node u_{l3} is a triconnected graph and u_{r3} is a ring, as illustrated in Fig. 7(d);
(e) Node u_{l3} is a ring and u_{r3} is a triconnected graph, as illustrated in Fig. 7(e);
(f) Both of nodes u_{l3} and u_{r3} are rings as illustrated in Fig. 7(f).

As the four apices of the rectangular contour of G, the algorithm in[11)] chooses as a_i, $1 \leq i \leq 4$, an arbitrary outer vertex in the triconnected component C_i corresponding to leaf l_i that is not a vertex of the separation pair of C_i. We choose a_2 and a_4 in the same way as in[11)]. However, we choose a_1 and a_3 of G, as follows. Let a_i, $i \in \{1, 3\}$, be the outer vertex in C_i that we encounter second when we traverse $F_o(G)$ clockwise from an outer vertex not in C_i. Thus a_i, $i \in \{1, 2, 3, 4\}$, is not a vertex of the separation pair of the component C_i. Let s_1 be the outer vertex that is counterclockwise next to a_1 on $F_o(G)$, and let s_3 be the outer vertex that is clockwise next to a_3 on $F_o(G)$, as illustrated in Figs. 1(a) and 7. Clearly s_i, $i \in \{1, 3\}$, is a vertex of a separation pair of C_i. The six vertices s_1, a_1, a_2, s_3, a_3 and a_4 appear clockwise on $F_o(G)$ in this order as illustrated in Figs. 1(a) and 7.

We then show how to divide G into G_u and G_d. Our division is different from

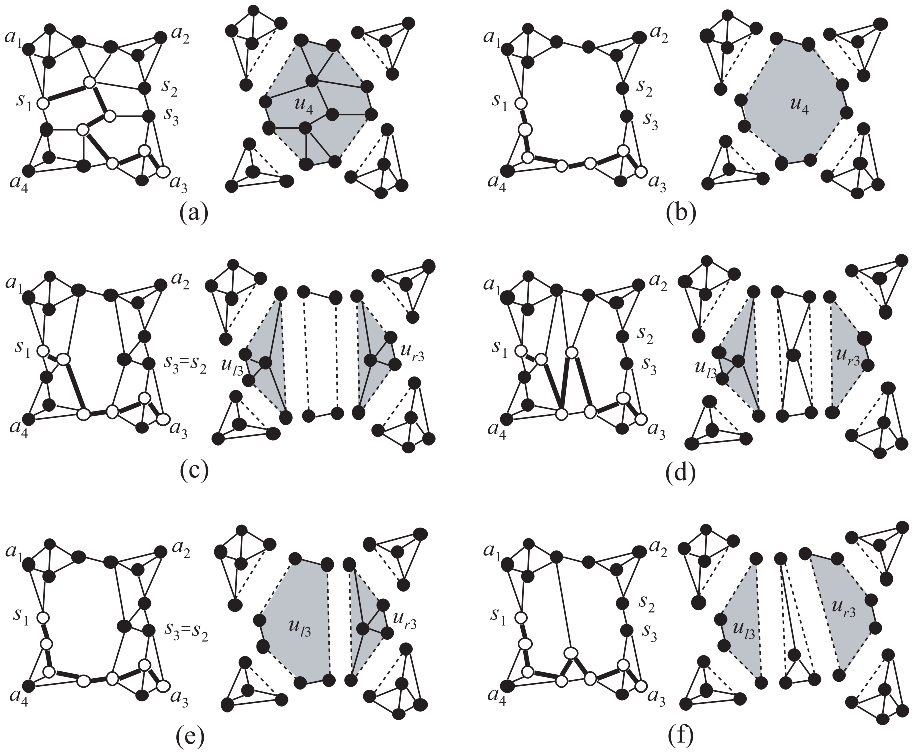

Fig. 7 Graph G, decomposition and path P for Cases (a)–(f).

that in[11]. Consider all the inner faces of G that contain one or more vertices on the path going from a_1 to s_3 clockwise on $F_o(G)$. (All these faces for the graph G in Fig. 1(a) are shaded in Fig. 6.) Let G' be the subgraph of G induced by all the edges on these faces. Then $F_o(G')$ is a simple cycle. Clearly, $F_o(G')$ contains vertices s_1, a_1, a_2, s_3 and a_3 in all Cases (a)–(f). Let P_d be the path going from s_1 to a_3 counterclockwise on $F_o(G')$. P_d is drawn by thick lines in Figs. 1(a), 6 and 7. Let G_d be the subgraph of G induced by all the vertices on P_d or below P_d, and let G_u be the subgraph of G obtained by deleting all vertices in G_d, as illustrated in Fig. 1(b). For every edge e of G that is contained neither in G_u nor in G_d, an end of e is on $F_o(G_u)$ and the other is on $F_o(G_d)$. Let n_d be the number of vertices of G_d, and let n_u be the number of vertices of G_u, then $n_d + n_u = n$.

4.2 Drawing of G_d

We now explain how to draw G_d.

Let G'_d be a graph obtained from G by contracting all the vertices of G_u to a single vertex w, as illustrated in Fig. 8(a) for the graph in Fig. 1(a). Then clearly $G_d = G'_d - w$, and G'_d is biconnected. One can prove, similarly as in[11], that G'_d is internally triconnected and has a canonical decomposition.

The decomposition tree $T(G'_d)$ of G'_d has exactly two leaves l_3 and l_4, and a_3 and a_4 are contained in the triconnected graphs corresponding to the leaves and are not vertices of the separation pairs. Every vertex of G'_d other than w has degree three or more, and w has degree two or more in G'_d. Therefore, G'_d has a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to a_4, a_3 and w, as illustrated in Fig. 8(b), where $U_m = \{w\}$, $v_1 = a_4$ and $v_2 = a_3$. We choose $v_l = s_1$ and $v_r = a_3$, as illustrated in Fig. 8(a). Using the pentagon algorithm in[11], we obtain a convex grid drawing $D_m = D'_d$ of $G_m = G'_d$, in which the outer polygon of D_m is a quadrangle with apices a_4, a_3, w and s_1, as illustrated in Fig. 8(d). Our drawing D_d of G_d is an intermediate drawing of D_m, that is, D_d is the drawing D_{m-1} of G_{m-1} induced by $U_1 \cup U_2 \cup \cdots \cup U_{m-1}$, as illustrated in Fig. 8(c). Note that $G_d = G'_d - w = G_{m-1}$. Since we choose $v_r = v_2$, by Lemma 3.1 we have $H(D_d) \le W(D_d) \le 2n_d - 2$.

4.3 Drawing of G_u

We now explain how to draw G_u.

If the degree of s_3 is one in G_u, then let P_3 be the maximal induced path with

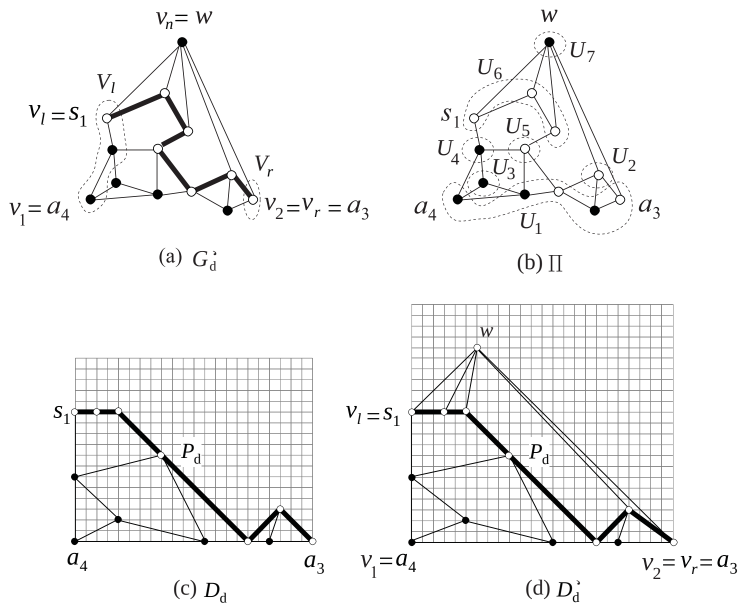

Fig. 8 (a) $G_{\rm d}'$ for the graph G in Fig. 1(a), (b) a canonical decomposition of $G_{\rm d}'$, (c) a drawing $D_{\rm d}$ of $G_{\rm d}$, and (d) a drawing $D_{\rm d}'$ of $G_{\rm d}'$.

an end s_3 such that all the intermediate vertices of P_3 have degree two in $G_{\rm u}$, and let s_2 be the other end of P_3, as illustrated in Fig. 9(a). Otherwise, let P_3 be the trivial path consisting only of s_3 and let $s_2 = s_3$. Let $G_{\rm u}'$ be a graph obtained from G by contracting all the vertices of $G_{\rm d}$ and all the vertices of P_3 except s_2 to a single vertex w, as illustrated in Fig. 9(b) for the graph G in Fig. 1(a). Then clearly $G_{\rm u}'$ is biconnected. Similarly to $G_{\rm d}'$, $G_{\rm u}'$ has a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to a_2, a_1 and w, as illustrated in Fig. 9 (c). We choose $v_l = s_2$ and $v_r = a_1$, as illustrated in Fig. 9(c). Using the pentagonal algorithm in[11], we obtain a convex grid drawing $D_u' = D_m$ of $G_{\rm u}'$, in which the outer polygon is a quadrangle with apices a_2, a_1, w and s_2, as illustrated in Fig. 9(e). Figure 9(d) depicts the drawing D_{m-1} of $G_{m-1} = G_u' - w$ induced by $U_1 \cup U_2 \cup \cdots \cup U_{m-1}$. We obtain a drawing $D_{\rm u}$ of $G_{\rm u}$ from D_{m-1} by putting the vertices of P_3 on grid points having the same x-coordinate as a_2, as illustrated

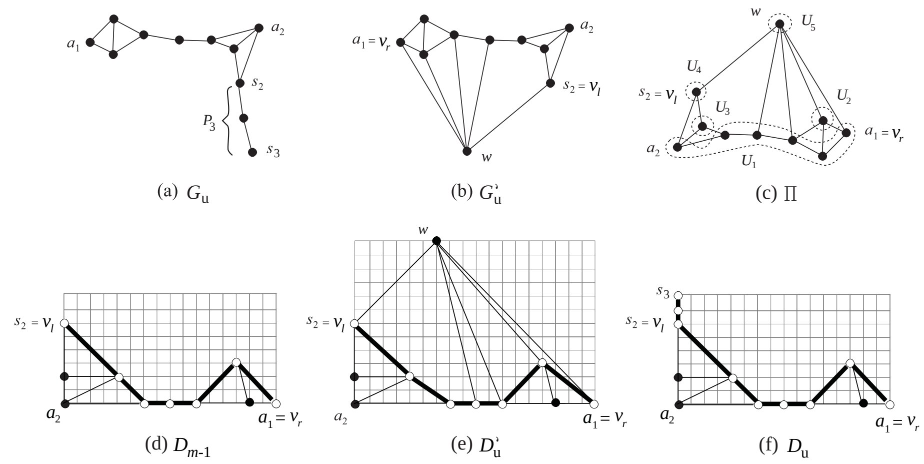

Fig. 9 (a) $G_{\rm u}$, (b) $G_{\rm u}'$, (c) a canonical decomposition of $G_{\rm u}'$, (d) a drawing D_{m-1} of G_{m-1}, (e) a drawing $D_{\rm u}' = D_m$ of $G_{\rm u}'$, and (f) a drawing $D_{\rm u}$ of $G_{\rm u}$.

in Fig. 9(f). Clearly $V(G_{m-1}) = n_{\rm u} - |V(P_3)| + 1$ and $|V(P_3)| \geq 1$. Therefore, by Lemma 3.1,

$$\begin{aligned} W(D_{\rm u}) &= W(D_{m-1}) \\ &\leq 2|V(G_{m-1})| - 2 \\ &\leq 2n_{\rm u} - 2|V(P_3)| \\ &\leq 2n_{\rm u} - 2 \end{aligned}$$

and

$$\begin{aligned} H(D_{\rm u}) &\leq W(D_{m-1}) + |V(P_3)| - 1 \\ &\leq 2n_{\rm u} - 2|V(P_3)| + |V(P_3)| - 1 \\ &\leq 2n_{\rm u} - 2. \end{aligned}$$

4.4 Drawing of G

If $W(D_{\rm d}) \neq W(D_{\rm u})$, then we widen the narrow one of $D_{\rm d}$ and $D_{\rm u}$ by the shift method in[2] so that both have the same width, which is at most $\max\{2n_{\rm d} - 2, 2n_{\rm u} - 2\}$. Since we combine the two drawings $D_{\rm d}$ and $D_{\rm u}$ of the same width to a drawing D of G, we have

$$W(D) \leq \max\{2n_{\rm d} - 2, 2n_{\rm u} - 2\} < 2n.$$

We arrange $D_{\rm d}$ and $D_{\rm u}$ so that $y(a_3) = y(a_4) = 0$ and $y(a_1) = y(a_2) =$

$H(D_\mathrm{d}) + H(D_\mathrm{u}) + W(D) + 1$, as illustrated in Fig. 1(e). Since $n_\mathrm{d} + n_\mathrm{u} = n$, we have

$$\begin{aligned} H(D) &= H(D_\mathrm{d}) + H(D_\mathrm{u}) + W(D) + 1 \\ &< (2n_\mathrm{d} - 2) + (2n_\mathrm{u} - 2) + 2n + 1 \\ &< 4n. \end{aligned}$$

We finally draw, by straight line segments, all the edges of G that are contained in neither G_u nor G_d. This completes the grid drawing D of G. (See Fig. 1(e).)

Since the conditions (sh5) and (sh6) hold for D_d and D_u, one can prove similarly as in[11] that the drawing D obtained above is a convex grid drawing of G. Clearly the algorithm takes linear time. We thus have the following theorem.

Theorem 4.1 Assume that G is an internally triconnected plane graph, every vertex of G has degree three or more, and the triconnected component decomposition tree $T(G)$ has exactly four leaves. Then our algorithm finds a convex grid drawing of G on a $2n \times 4n$ grid in linear time.

5. Conclusions

In this paper, we showed that every internally triconnected plane graph G whose decomposition tree $T(G)$ has exactly four leaves has a convex grid drawing on a $2n \times 4n = O(n^2)$ grid, and we present a linear algorithm to find such a drawing. The area bound $O(n^2)$ is optimal within a coefficient since the nested triangles graph needs $\Omega(n^2)$ area. The remaining problem is to obtain an algorithm for an internally triconnected plane graph whose decomposition tree has five or more leaves.

References

1) N. Bonichon, S. Felsner and M. Mosbah, *Convex drawings of 3-connected plane graphs*, Algorithmica, 47, 4, pp.399–420, 2007.
2) M.Chrobak and G.Kant, *Convex grid drawings of 3-connected planar graphs*, International Journal of Computational Geometry and Applications, 7, pp.211–223, 1997.
3) N.Chiba, K.Onoguchi and T.Nishizeki, *Drawing planar graphs nicely*, Acta Inform., 22, pp.187–201, 1985.
4) N.Chiba, T.Yamanouchi and T.Nishizeki, *Linear algorithms for convex drawings of planar graphs*, in Progress in Graph Theory, J.A.Bondy and U.S.R. Murty (Eds.), Academic Press, pp.153–173, 1984.
5) D. Dolev, F. T. Leighton and H. Trickey, *Planar embedding of planar graphs*, Advances in Computer Research - Vol 2, VLSI Theory, pp.147–161, 1984.
6) H. de Fraysseix, J. Pach and R. Pollack, *How to draw a planar graph on a grid*, Combinatorica, 10, pp.41–51, 1990.
7) S. Felsner, *Convex drawings of plane graphs and the order of dimension of 3-polytopes*, Order, 18, pp.19–37, 2001.
8) J. E. Hopcroft and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM J. Compt.2, 3, pp.135–138, 1973.
9) K.Miura, M.Azuma and T.Nishizeki, *Canonical decomposition, realizer, Schnyder labeling and orderly spanning trees of plane graphs*, International Journal of Foundations of Computer Science, 16, 1, pp.117–141, 2005.
10) K.Miura, M.Azuma and T.Nishizeki, *Convex drawings of plane graphs of minimum outer apices*, International Journal of Foundations of Computer Science, 17, 5, pp. 1115–1127, 2006.
11) K.Miura, A. Kamada and T.Nishizeki, *Convex grid drawings of plane graphs of rectangular contours*, J. Graph Algorithms and Applications, 12, 2, pp.197–224, 2008.
12) T.Nishizeki and M.S.Rahman, *Planar Graph Drawing*, World Scientific, Singapore, 2004.