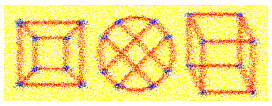
Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 12, no. 2, pp. 197–224 (2008)

Convex Grid Drawings of Plane Graphs with Rectangular Contours

Kazuyuki Miura[1] *Akira Kamada*[2] *Takao Nishizeki*[2]

[1]Faculty of Symbiotic Systems Science, Fukushima University,
Fukushima 960-1296, Japan.
[2]Graduate School of Information Sciences, Tohoku University,
Sendai 980-8579, Japan.

Abstract

In a convex drawing of a plane graph, all edges are drawn as straight-line segments without any edge-intersection and all facial cycles are drawn as convex polygons. In a convex grid drawing, all vertices are put on grid points. A plane graph G has a convex drawing if and only if G is internally triconnected, and an internally triconnected plane graph G has a convex grid drawing on an $(n-1) \times (n-1)$ grid if either G is triconnected or the triconnected component decomposition tree $T(G)$ of G has two or three leaves, where n is the number of vertices in G. In this paper, we show that an internally triconnected plane graph G has a convex grid drawing on a $2n \times n^2$ grid if $T(G)$ has exactly four leaves. We also present an algorithm to find such a drawing in linear time. Our convex grid drawing has a rectangular contour, while most of the known algorithms produce grid drawings having triangular contours.

Submitted:	Reviewed:	Revised:	Accepted:	Final:
August 2007	June 2008	July 2008	July 2008	July 2008

Published:
October 2008

Article type: Regular Paper
Communicated by: X. He

E-mail addresses: miura@sss.fukushima-u.ac.jp (Kazuyuki Miura) kamada@nishizeki.ecei.tohoku.ac.jp (Akira Kamada) nishi@ecei.tohoku.ac.jp (Takao Nishizeki)

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 [13]. 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, 13].

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," that is, G can be extended to a triconnected graph by adding a vertex in the outer face and joining it to all outer vertices [11, 15]. One may thus assume without loss of generality that G is internally triconnected. If either G is triconnected or the "triconnected component decomposition tree" $T(G)$ of G has two or three leaves, then G has a convex grid drawing on an $(n-1) \times (n-1)$ grid and such a drawing can be found in linear time, where n is the number of vertices of G [1, 2, 10]. However, it has not been known whether G has a convex grid drawing of polynomial size if $T(G)$ has four or more leaves. Figure 1(a) depicts an internally triconnected plane graph G, Fig. 2(b) the triconnected components of G, and Fig. 2(c) the triconnected component decomposition tree $T(G)$ of G, which has four leaves l_1, l_2, l_3 and l_4.

In this paper, we show that an internally triconnected plane graph G has a convex grid drawing on a $2n \times n^2$ grid if $T(G)$ has exactly four leaves, and present an algorithm to find such a drawing in linear time. The algorithm is outlined as follows: we first divide a plane graph G into an upper subgraph $G_{\rm u}$ and a lower subgraph $G_{\rm d}$ as illustrated in Fig. 1(b) for the graph in Fig. 1(a); we then construct "inner" convex grid drawings of $G_{\rm u}$ and $G_{\rm d}$ by a so-called shift method as illustrated in Figs. 1(c) and (d); we finally extend these two drawings to a convex grid drawing of G as illustrated in Fig. 1(e). This is the first algorithm that finds a convex grid drawing of such a plane graph G in a grid of polynomial size. The size $2n \times n^2$ of our convex grid drawing is larger than the size $(n-1) \times (n-1)$ of a convex grid drawing of a triconnected plane graph obtained by previously known algorithms [1, 2, 5]. It is known that every triconnected plane graph has a "strict convex grid drawing" of size $O(n^{7/3}) \times O(n^{7/3})$ in which all facial cycles are drawn as strictly convex polygons [14]. It is also known that every internally triconnected plane graph has a grid drawing of size $(n-1) \times (n-2)$ in which all inner facial cycles are drawn as convex polygons although the outer facial cycle is not necessarily drawn as a convex polygon [2, 13]. Our convex grid drawing has a rectangular contour, while most of the previously known algorithms except those in [8, 12] produce a grid drawing having a triangular contour [1, 2, 5, 6, 10, 17].

The remainder of the paper is organized as follows. In Section 2 we give

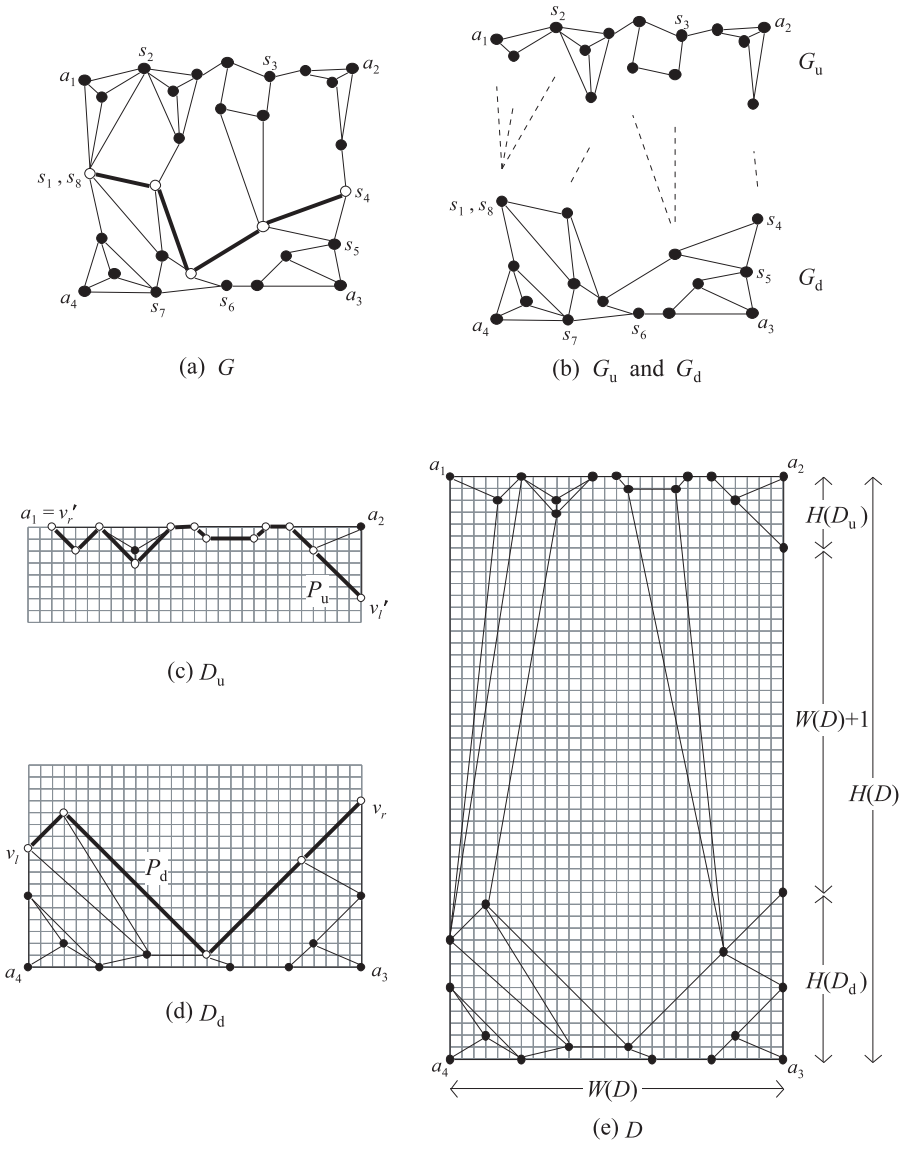

Figure 1: (a) Plane graph G, (b) upper subgraph G_u and lower subgraph G_d, (c) inner convex drawing D_u of G_u, (d) inner convex drawing D_d of G_d, and (e) convex grid drawing D of G.

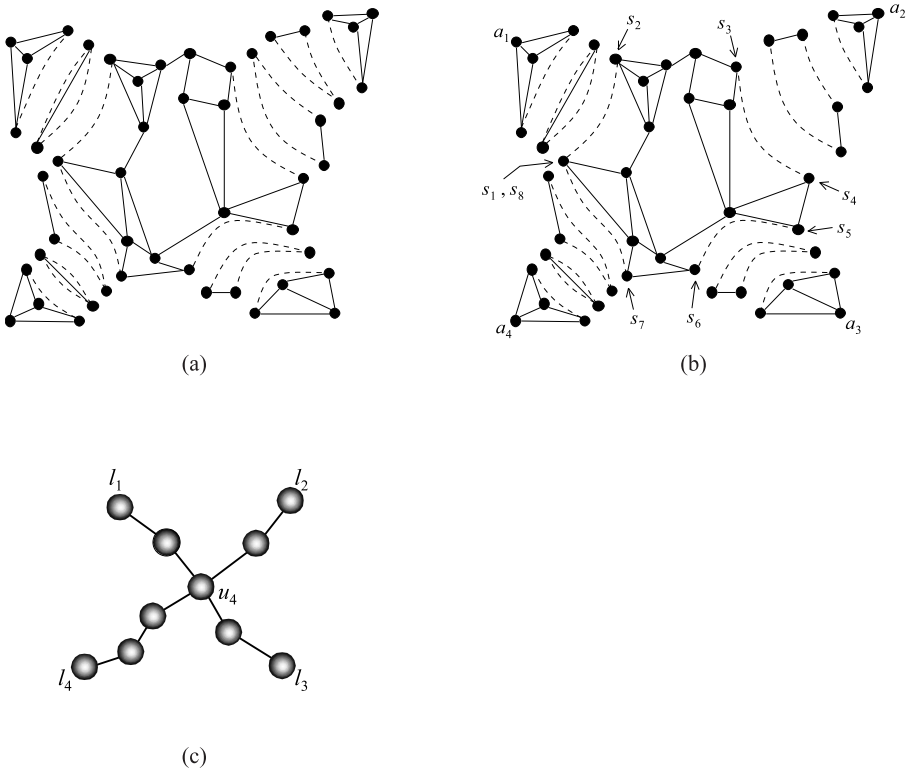

Figure 2: (a) Split components of the graph G in Fig. 1(a), (b) triconnected components of G, and (c) decomposition tree $T(G)$.

some definitions and known lemmas. In Section 3 we explain an algorithm for $G_{\rm u}$ and $G_{\rm d}$. In Section 4 we present our convex grid drawing algorithm. Finally we conclude in Section 5. An early version of the paper was presented at [9].

2 Preliminaries

In this section, we give some definitions and known lemmas.

A $W \times H$ *integer grid* consists of $W+1$ regular vertical grid lines and $H+1$ regular 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 a minimum integer grid enclosing a grid drawing D of a graph, and by $H(D)$ the height of D.

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 $E(G)$. Throughout the paper we denote by n the number of vertices in G. An edge joining vertices u and v is denoted by (u, v). The *degree* of a vertex v in G is the number of neighbors of v in G.

A plane graph G divides the plane into connected regions, called *faces*. The infinite face is called an *outer face*, and the others are called *inner faces*. The boundary of a face is called a *facial cycle*. A cycle is represented by a clockwise sequence of the vertices in the cycle. We denote by $F_{\rm o}(G)$ the outer facial cycle of G. A vertex on $F_{\rm o}(G)$ is called an *outer vertex*, while a vertex not on $F_{\rm o}(G)$ is called an *inner vertex*. In a convex drawing of a plane graph G, all facial cycles must be drawn as convex polygons. The convex polygonal drawing of $F_{\rm o}(G)$ is called the *outer polygon*. 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. In other words, G is internally triconnected if and only if it can be extended to a triconnected graph by adding a vertex in the outer face and joining it to all outer vertices. If a biconnected plane graph G is not internally triconnected, then G has a separation pair $\{u, v\}$ as illustrated in Figs. 3(a)–(c) and a "split graph" H contains an inner vertex other than u and v; in Fig. 3(a) both u and v are inner vertices, in Fig. 3(b) one of u and v, say v, is an inner vertex, and in Fig. 3(c) both u and v are outer vertices but a split graph H contains no outer vertex other than u and v.

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.

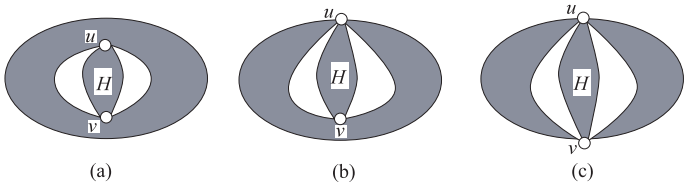

Figure 3: Biconnected plane graphs which are not internally triconnected.

(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 graph in Fig. 1(a) has eleven split components illustrated in Fig. 2(a). The split components are of three types: a triconnected graph; a triple bond (i.e. a set of three multiple edges); and a triangle (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) a triconnected graph;

(b) a bond; and

(c) a ring.

The split components of G are not necessarily unique, but the triconnected components of G are unique [7, 16]. Two triangles in Fig. 2(a) are merged into a single ring, and hence the graph in Fig. 1(a) has ten 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 [7, 16]. We denote by $\ell(G)$ the number of leaves of $T(G)$. Then $\ell(G) = 4$ 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 $\ell(G) = 1$.

The following three lemmas are known.

Lemma 1 *[7] A decomposition tree $T(G)$ of a graph G can be found in linear time.*

Lemma 2 *[11] Let G be a biconnected plane graph in which every vertex has degree three or more. Then the following three statements are equivalent to each other:*

(a) G has a convex drawing;

(b) G is internally triconnected; and

(c) both vertices of every separation pair are outer vertices, and a node of the decomposition tree $T(G)$ of G has degree two if it is a bond.

Lemma 3 *[11] If a plane graph G has a convex drawing D, then the number of apices of the outer polygon of D is no less than* $\max\{3, \ell(G)\}$*, and there is a convex drawing of G whose outer polygon has exactly* $\max\{3, \ell(G)\}$ *apices.*

Since G is an internally triconnected simple graph and every vertex of G has degree three or more, by Lemma 2 every leaf of $T(G)$ is a triconnected graph. Lemmas 2 and 3 imply that if $T(G)$ has exactly four leaves, that is, $\ell(G) = 4$ then the outer polygon of every convex drawing of G must have four or more apices. Our algorithm obtains a convex grid drawing of G whose outer polygon is a rectangle and hence has exactly four apices, as illustrated in Fig. 1(e).

In Section 3, we will present an algorithm to draw the upper subgraph $G_{\rm u}$ and the lower subgraph $G_{\rm d}$. (See Fig. 1(b).) The algorithm uses the following "canonical decomposition." 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 $F_{\rm o}(G)$ in this order. We may assume that v_1 and v_2 are consecutive on $F_{\rm o}(G)$; 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 $U_1, U_2, \cdots, U_m$, where $U_1 \bigcup U_2 \bigcup \cdots \bigcup U_m = V$ and $U_i \bigcap U_j = \emptyset$ for any indices i and j, $1 \leq i < j \leq m$. 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 $U_{k+1} \bigcup U_{k+2} \bigcup \cdots \bigcup U_m$. Clearly, $G_k = G - U_{k+1} \bigcup U_{k+2} \bigcup \cdots \bigcup U_m$ and $G = G_m = \overline{G_0}$. 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} and has one or more neighbors in $\overline{G_k}$ when $k < m$, as illustrated in Fig. 4(a); 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 $\overline{G_k}$, as illustrated in Fig. 4(b).

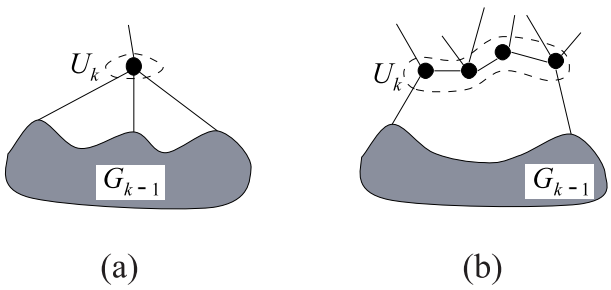

Figure 4: (a) Graphs G_k for which $|U_k| = 1$ and (b) $|U_k| \geq 2$.

Although the definition of a canonical decomposition above is slightly different from the one given in [2], they are effectively equivalent with each other. 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. 5(a) is illustrated in Fig. 5(b).

The following lemma is known.

Lemma 4 *[10] Assume that G is an internally triconnected plane graph and $\ell(G) \leq 3$. Then one can find a canonical decomposition Π of G in linear time if v_1, v_2 and v_n are chosen as follows.*
Case 1: $\ell(G) = 3$.

In this case, from each of the three triconnected components corresponding to leaves of $T(G)$, we choose an arbitrary outer vertex of G which is not a vertex of the separation pair of the component.
Case 2: $\ell(G) = 2$.

In this case, we choose two vertices from the two leaves of $T(G)$, similarly to Case 1 above. We choose an arbitrary outer vertex of G other than them as the third one.
Case 3: $\ell(G) = 1$.

In this case, G is triconnected. We choose three arbitrary outer vertices of G.

One can easily observe that Lemma 4 holds even if exactly one of the outer vertices has degree two and the vertex is chosen as v_n.

3 Pentagonal drawing

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. 5(b). In this section,

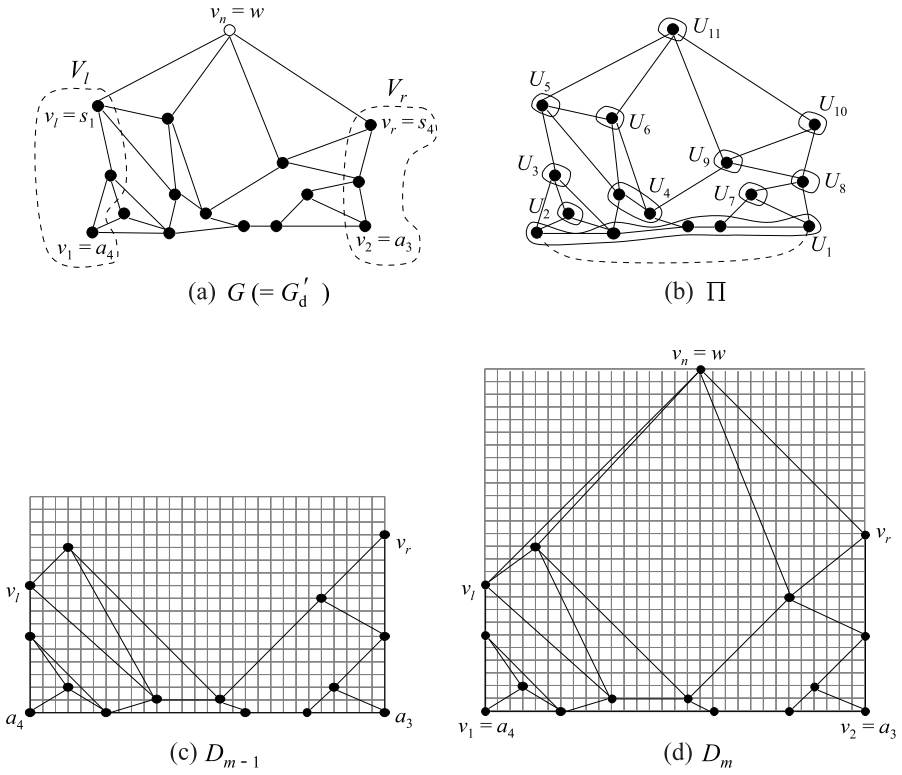

Figure 5: (a) An internally triconnected plane graph $G(= G'_{\rm d})$, (b) a canonical decomposition Π of G, (c) a drawing D_{m-1} of G_{m-1}, and (d) a pentagonal drawing D_m of G.

we present a linear-time algorithm, called a *pentagonal drawing algorithm*, to find a convex grid drawing of G with a pentagonal outer polygon, as illustrated in Fig. 5(d). The algorithm is based on the so-called shift methods given by Chrobak and Kant [2] and de Fraysseix *et al.* [5], and will be used by our convex grid drawing algorithm to draw the lower subgraph $G_{\rm d}$ and the upper subgraph $G_{\rm u}$ of G in Sections 4.2 and 4.3, respectively.

Let v_l be an arbitrary vertex on the path going from v_1 to v_n clockwise on $F_{\rm o}(G)$, and let $v_r (\neq v_l)$ be an arbitrary vertex on the path going from v_2 to v_n counterclockwise on $F_{\rm o}(G)$, as illustrated in Fig. 5(a). Let V_l be the set of all vertices on the path going from v_1 to v_l clockwise on $F_{\rm o}(G)$, and let V_r be the set of all vertices on the path going from v_2 to v_r counterclockwise on $F_{\rm o}(G)$. Our algorithm 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. 5(d). The pentagon has a shape similar to a baseball home plate; the side v_1v_2 is horizontal, and the two sides v_1v_l and v_2v_r are vertical and contain all vertices in V_l and V_r, respectively.

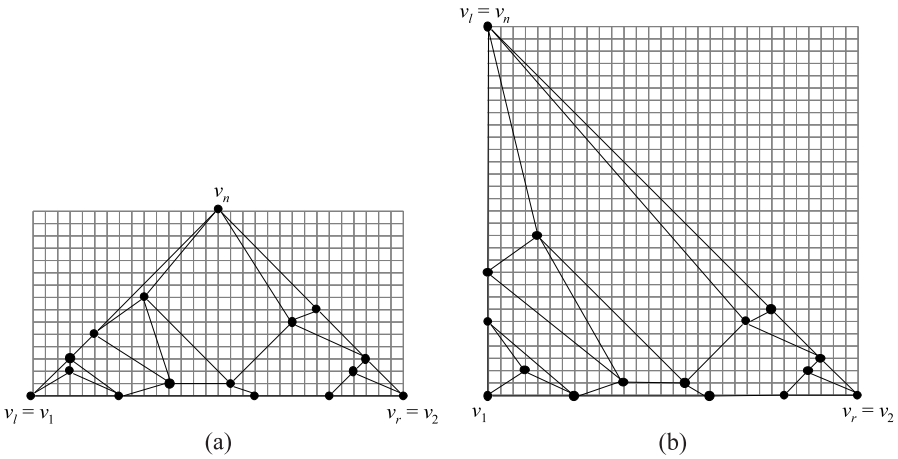

Figure 6: (a) Convex grid drawings of G in Fig. 5(a) for the case where $v_l = v_1$ and $v_r = v_2$, and (b) for the case where $v_l = v_n$ and $v_r = v_2$.

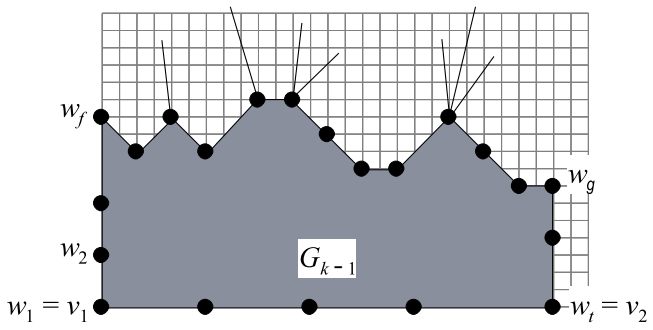

Figure 7: Drawing D_{k-1} of graph G_{k-1}.

However, v_l may be v_1 or v_n, and v_r may be v_2 or v_n, and hence the pentagon may degenerate into a right-angled isosceles triangle as illustrated in Figs. 6(a) and (b).

For each k, $2 \leq k \leq m$, let $F_{\rm o}(G_{k-1}) = w_1, w_2, \cdots, w_t$, where $w_1 = v_1$, $w_t = v_2$, and $w_1, w_2, \cdots, w_t$ appear clockwise on $F_{\rm o}(G_{k-1})$ in this order, as illustrated in Fig. 7.

We first obtain a drawing D_1 of the subgraph G_1 of G induced by all vertices in U_1. Let $F_{\rm o}(G_1) = w_1, w_2, \cdots, w_t$, $w_1 = v_1$, and $w_t = v_2$. We draw G_1 as illustrated in Fig. 8, depending on whether (v_1, v_2) is a real edge or not, $w_2 \in V_l$ or not, and $w_{t-1} \in V_r$ or not. (See Appendix for the details.)

Initialize:

Case 1: v_1 and v_2 are adjacent in an original graph G, that is, (v_1, v_2) is a real

edge.

In this case we draw G_1 as a trapezoid such that the bottom side has length $2t-2$ and the two sides other than the top and bottom have slope ± 1 or $\pm\infty$.

If $w_2 \in V_l$ and $w_{t-1} \notin V_r$, then draw G_1 as in Fig. 8(a);

If $w_2 \notin V_l$ and $w_{t-1} \in V_r$, then draw G_1 as in Fig. 8(b);

If $w_2 \in V_l$ and $w_{t-1} \in V_r$, then draw G_1 as in Fig. 8(c);

If $w_2 \notin V_l$ and $w_{t-1} \notin V_r$, then draw G_1 as in Fig. 8(d);

Case 2: Otherwise, that is, (v_1, v_2) is a virtual edge.

Draw G_1 on a horizontal line segment of length $2t-2$, as in Fig. 8(e).

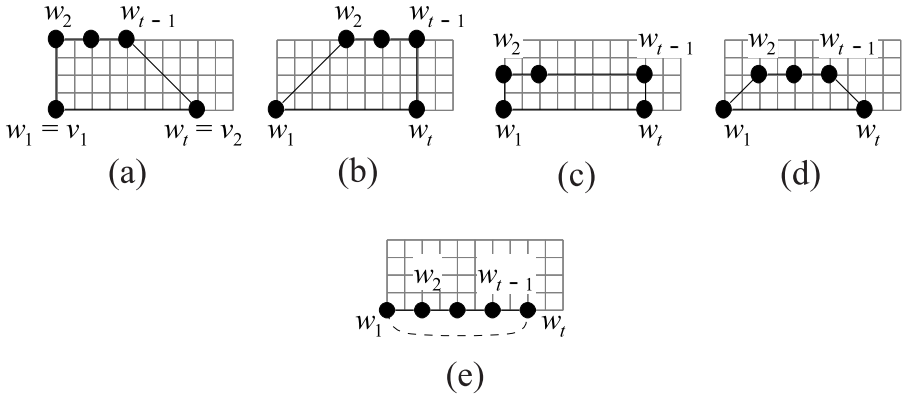

Figure 8: Drawings D_1 of G_1 (a)–(d) for Cases 1 and (e) for Case 2.

We then extend a drawing D_{k-1} of G_{k-1} to a drawing D_k of G_k for each index k, $2 \le k \le m$. Let $F_o(G_{k-1}) = w_1, w_2, \cdots, w_t$, $w_1 = v_1$, $w_t = v_2$, and $U_k = \{u_1, u_2, \cdots, u_r\}$. By the condition (cd3) of a canonical decomposition, one may assume that the vertices $u_1, u_2, \cdots, u_r$ in U_k appear clockwise on $F_o(G_k)$ in this order and that the first vertex u_1 and the last one u_r in U_k have neighbors in G_{k-1}. (See Fig. 4.) Let w_p be the leftmost neighbor of u_1, that is, w_p is the neighbor of u_1 in G_k having the smallest index p, and let w_q be the rightmost neighbor of u_r, as illustrated in Fig. 9.

Let w_f be the vertex with the maximum index f among all the vertices w_i, $1 \le i \le t$, on $F_o(G_{k-1})$ that are contained in V_l. Let w_g be the vertex with the minimum index g among all the vertices w_i, $1 \le i \le t$, on $F_o(G_{k-1})$ that are contained in V_r. Of course, $1 \le f < g \le t$. We denote by $\angle w_i$ the interior angle of apex w_i of the outer polygon of D_{k-1}. We call w_i a *convex apex* of the polygon if $\angle w_i < 180°$. We denote the current position of a vertex v by $P(v)$; $P(v)$ is expressed by its x- and y-coordinates as $(x(v), y(v))$. Assume that a drawing D_{k-1} of G_{k-1} satisfies the following six conditions (sh1)–(sh6).

(sh1) $P(w_1) = (0, 0)$ and $P(w_t) = (2n_{k-1} - 2, 0)$, where $n_{k-1} = |V(G_{k-1})|$.

(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-1} is drawn as a convex polygon.

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

Indeed D_1 satisfies the six conditions above. We extend D_{k-1} to D_k so that D_k satisfies them, as follows.

Before installing U_k to D_{k-1}, we shift (move to the left or right) some vertices of G_{k-1} as illustrated in Figs. 9(a)–(d). We first shift $w_1, w_2, \cdots, w_p$ of G_{k-1} and some inner vertices of G_k to the left by distance $|U_k|$, and then shift $w_q, w_{q+1}, \cdots, w_t$ of G_{k-1} and some inner vertices of G_k to the right by distance $|U_k|$. After the operation, we shift all vertices of G_{k-1} to the right by distance $|U_k|$ so that $P(w_1) = (0, 0)$. See Appendix for the details.

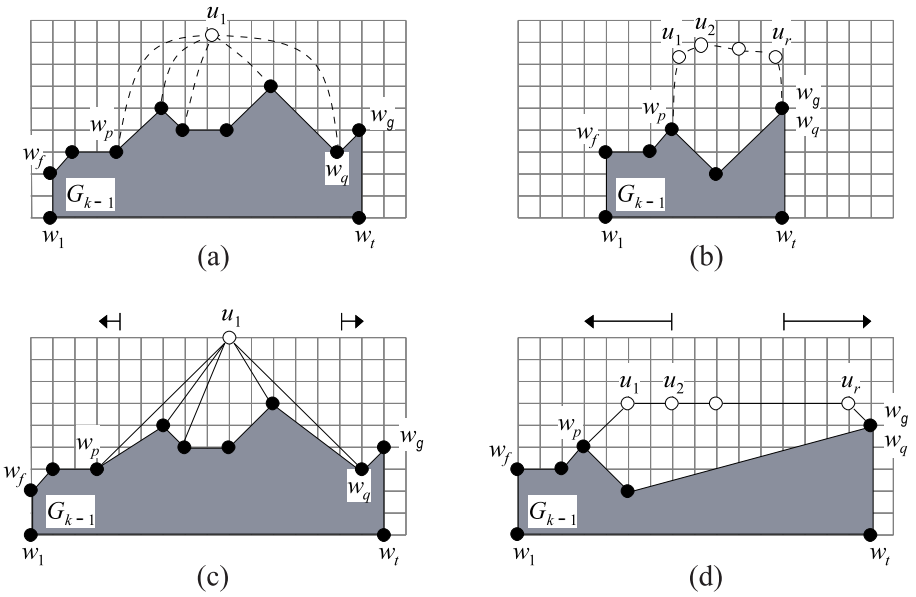

Figure 9: (a) Graphs G_{k-1} for Case $|U_k| = 1$, (b) for Case $|U_k| \geq 2$, (c) graphs G_k for Case $|U_k| = 1$, and (d) for Case $|U_k| \geq 2$.

Then, we install U_k to D_{k-1} as follows. (See Appendix for the details.)

Install U_k:

Case 1: $|U_k| = 1$.

If $u_1 \in V_l$, then put the vertex u_1 in U_k above $w_f = w_p$ so that the edge (u_1, w_q) has slope -1, as in Fig. 10(a);

If $u_1 \in V_r$, then put u_1 above $w_g = w_q$ so that the edge (w_p, u_1) has slope $+1$, as in Fig. 10(b);

If $u_1 \notin V_l$ and $u_1 \notin V_r$, then put u_1 on a grid point so that the edge (w_p, u_1) has slope $+1$ and the edge (u_1, w_q) has slope -1, as in Fig. 10(c);

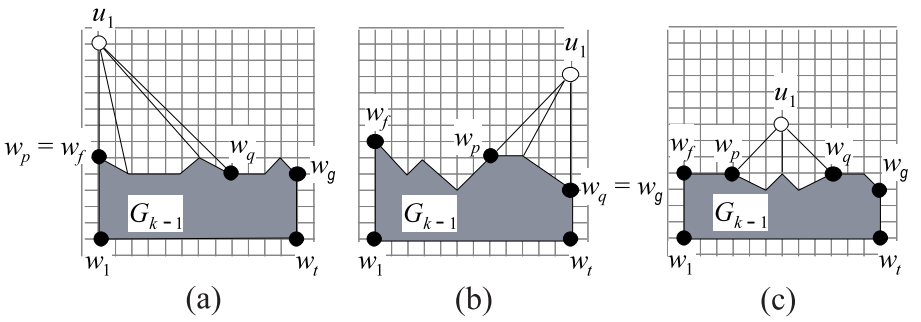

Figure 10: Installing $U_k = \{u_1\}$ to D_{k-1}.

Case 2: $|U_k| \geq 2$.

Case (a): $u_1 \in V_l$ and $u_r \notin V_r$.

If $y(w_q) - y(w_p)$ is an odd number ≥ 1, then put the vertices $u_1, u_2, \cdots, u_r$ in U_k on a horizontal line of y-coordinate $\max\{y(w_p), y(w_q)\}+1$, as in Fig. 11(a);

Otherwise, put the vertices in U_k on a horizontal line of y-coordinate $\max\{y(w_p), y(w_q)\} + 2$, as in Fig. 11(a′);

Case (b): $u_1 \notin V_l$ and $u_r \in V_r$.

If $y(w_p) - y(w_q)$ is an odd number ≥ 1, then put the vertices in U_k on a horizontal line of y-coordinate $\max\{y(w_p), y(w_q)\} + 1$, as in Fig. 11(b);

Otherwise, put the vertices in U_k on a horizontal line of y-coordinate $\max\{y(w_p), y(w_q)\} + 2$, as in Fig. 11(b′);

Case (c): $u_1 \in V_l$ and $u_r \in V_r$.

Put the vertices in U_k on a horizontal line of y-coordinate $\max\{y(w_p), y(w_q)\} + 1$, as in Fig. 11(c);

Case (d): $u_1 \notin V_l$ and $u_r \notin V_r$.

Put the vertices in U_k on a horizontal line of y-coordinate $\max\{y(w_p), y(w_q)\} + 1$, as in Fig. 11(d).

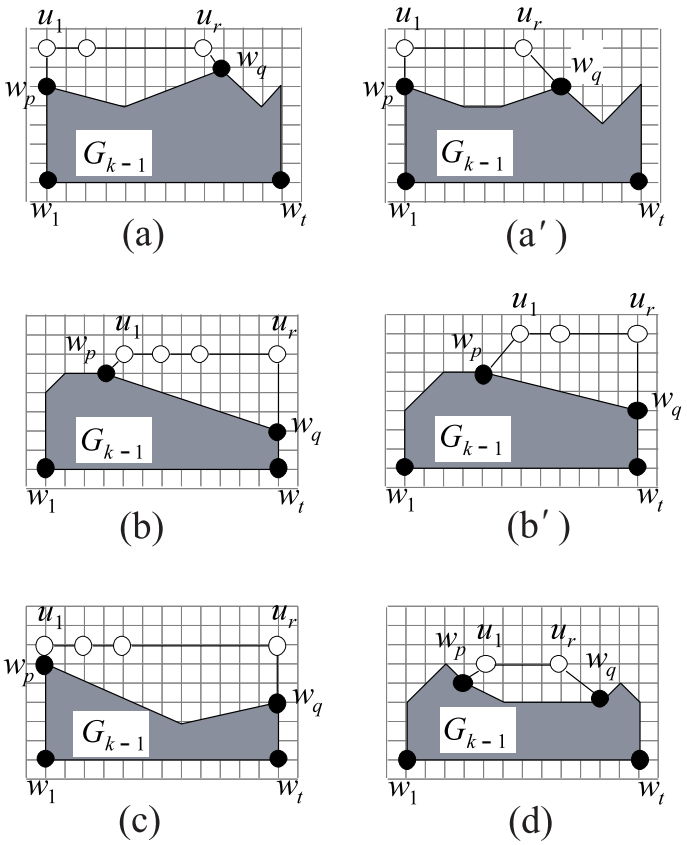

Figure 11: Installing $U_k = \{u_1, u_2, \cdots, u_r\},\ r \geq 2$, to D_{k-1}.

Clearly, the drawing D_k of G_k extended from D_{k-1} satisfies conditions (sh1), (sh2) and (sh3). One can prove similarly as in [5] that D_k satisfies condition (sh4), and prove similarly as in [2] that D_k satisfies conditions (sh5) and (sh6). One can easily show that the pentagonal drawing algorithm takes linear time.

We finally consider the width $W = W(D)$ and height $H = H(D)$ of the final drawing $D = D_m$ of $G = G_m$.

For each k, $1 \leq k \leq m$, the drawing D_k of G_k satisfies conditions (sh1) and (sh2), and hence we have

$$W(D_k) = 2n_k - 2$$

where $n_k = |V(G_k)|$. We thus have

$$W(D) = W(D_m) = 2n - 2.$$

Since $n_1 \geq 3$, we have

$$H(D_1) \leq 4 \leq n_1^2 - n_1 - 2.$$

One can easily observe from Figs. 10 and 11 that $H(D_k) \leq H(D_{k-1}) + W(D_k)$ for each k, $2 \leq k \leq m$. Noting $n_k = n_{k-1} + |U_k| \geq n_{k-1} + 1$, one can inductively prove that

$$H(D_k) \leq n_k^2 - n_k - 2.$$

Therefore we have

$$H(D) \leq n^2 - n - 2.$$

We thus have the following lemma.

Lemma 5 *For a plane graph G having a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to v_1, v_2 and v_n, the pentagonal drawing algorithm yields a convex grid drawing of G on a $W \times H$ grid with $W = 2n - 2$ and $H \leq n^2 - n - 2$ in linear time. Furthermore, $W(D_{m-1}) = 2n_{m-1} - 2$ and $H(D_{m-1}) \leq n_{m-1}^2 - n_{m-1} - 2$.*

If $v_l = v_1$ and $v_r = v_2$, then the outer pentagon degenerates into a right-angled isosceles triangle with $\angle v_1 v_n v_2 = 90^\circ$ and $H = n - 1$, as illustrated in Fig. 6(a). Such a grid drawing is effectively the same as the one obtained by the algorithm in [5].

If $v_l = v_n$ and $v_r = v_2$, then the outer pentagon degenerates into a right-angled isosceles triangle with $\angle v_n v_1 v_2 = 90^\circ$ and $H = 2n - 2$, as illustrated in Fig. 6(b). Such a convex grid drawing is effectively the same as the one obtained by the algorithm in [2].

Thus our pentagonal drawing algorithm is an extension of both the straight-line drawing algorithm in [5] and the convex grid drawing algorithm in [2].

Let $\Pi = (U_1, U_2, \cdots, U_m)$ be a canonical decomposition of the plane graph G in Fig. 12(a). Then $|U_1| = 3$, $|U_2| = |U_3| = \cdots = |U_m| = 1$, and $m = n - 2$. Let n be an odd number, $v_l = v_n$, and $v_r = v_{n-1}$. Then, the pentagonal drawing algorithm draws G_k as illustrated in Fig. 12(b), and the width $W = 2n - 2$ and height $H = n^2 - n - 2$ of G attains the bounds in Lemma 5.

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 algorithms in [1], [2], [6] and [10] and the pentagonal drawing algorithm in Section 3 can find a convex grid drawing of G. Our algorithm draws the outer facial cycle $F_{\rm o}(G)$ as a rectangle as illustrated in Fig. 1(e). The algorithm first divides G into an upper subgraph $G_{\rm u}$ and a lower subgraph $G_{\rm d}$ as illustrated in Fig. 1(b), then draws $G_{\rm u}$ and $G_{\rm d}$ by using the pentagonal drawing algorithm 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).

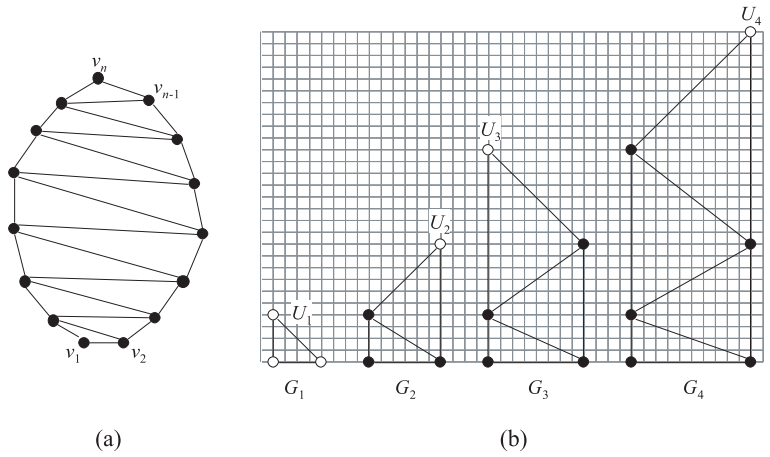

Figure 12: (a) A graph G, and (b) drawings of G_1, G_2, G_3 and G_4.

4.1 Division

We first explain how to divide G into $G_{\rm u}$ and $G_{\rm d}$. (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. 13. Clearly, either exactly one node u_4 of $T(G)$ has degree four and each of the other non-leaf nodes has degree two as illustrated in Fig. 13(a), or exactly two nodes u_{l3} and u_{r3} have degree three and each of the other non-leaf nodes has degree two as illustrated in Fig. 13(b). One may assume that, in Fig. 13(b), node u_{l3} is arranged to the left and node u_{r3} to the right.

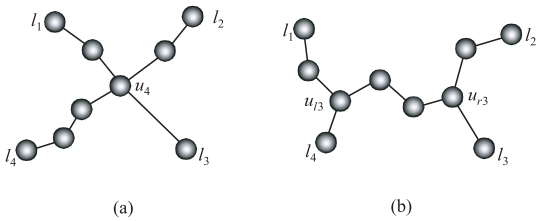

Figure 13: Decomposition trees $T(G)$ (a) having a node of degree four and (b) having two nodes of degree three.

Since each vertex of G is assumed to have degree three or more, all the four leaves of $T(G)$ are triconnected graphs. Moreover, according to Lemma 2, 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 $T(G)$. Thus there are the following six cases (a)–(f) to consider.

(a) Node u_4 is a triconnected graph as illustrated in Fig. 14(a);

(b) Node u_4 is a ring as illustrated in Fig. 14(b);

(c) Both of nodes u_{l3} and u_{r3} are triconnected graphs as illustrated in Fig. 14(c);

(d) Node u_{l3} is a triconnected graph and u_{r3} is a ring, as illustrated in Fig. 14(d);

(e) Node u_{l3} is a ring and u_{r3} is a triconnected graph, as illustrated in Fig. 14(e);

(f) Both of nodes u_{l3} and u_{r3} are rings as illustrated in Fig. 14(f).

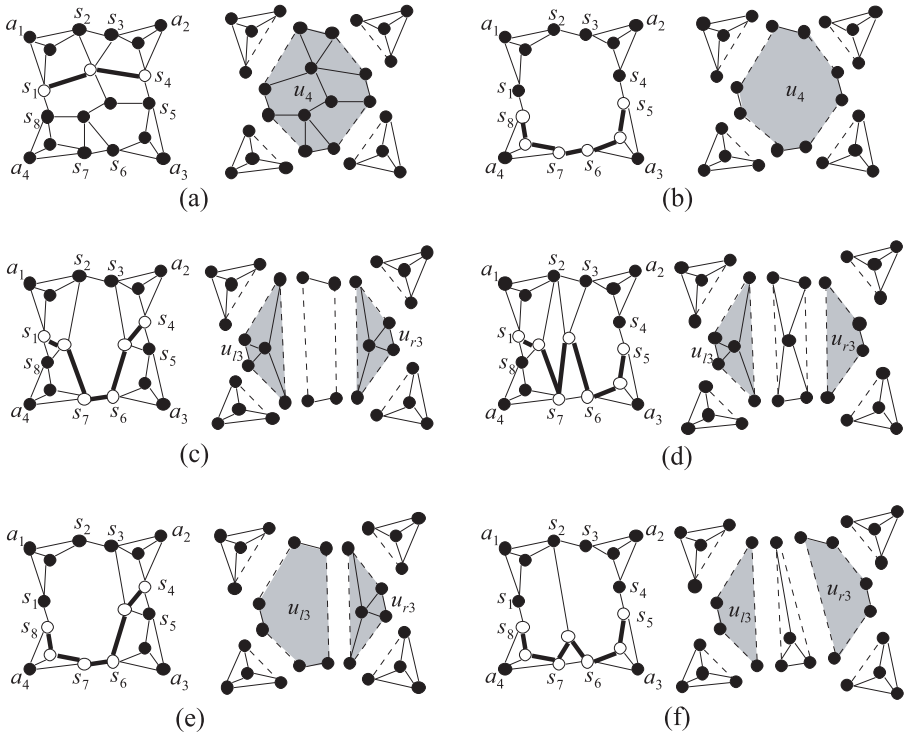

Figure 14: Graph G, decomposition and path P for Cases (a)–(f).

As the four apices of the rectangular contour of G, we choose four outer vertices a_i, $1 \leq i \leq 4$, of G; let a_i be an arbitrary outer vertex in the component corresponding to leaf l_i that is not a vertex of the separation pair of the component. The four vertices a_1, a_2, a_3 and a_4 appear clockwise on $F_{\rm o}(G)$ in this order as illustrated in Fig. 1(a).

We then choose eight vertices $s_1, s_2, \cdots, s_8$ from the outer vertices of the components u_4, u_{l3} and u_{r3}. Among these outer vertices, let s_1 be the vertex that one encounters first when one traverses $F_{\rm o}(G)$ counterclockwise from vertex a_1, and let s_2 be the vertex that one encounters first when one traverses $F_{\rm o}(G)$

clockwise from a_1, as illustrated in Figs. 1(a) and 14. (Thus, $\{s_1, s_2\}$ is a separation pair of G, and corresponds either to the edge of $T(G)$ which is incident to node u_4 and lies on the path between u_4 and leaf l_1 in $T(G)$ or to the edge of $T(G)$ which is incident to node u_{l3} and lies on the path between u_{l3} and l_1.) Similarly, we choose s_3 and s_4 for a_2, s_5 and s_6 for a_3, and s_7 and s_8 for a_4. Possibly $s_2 = s_3$, $s_4 = s_5$, $s_6 = s_7$, and $s_8 = s_1$.

We then show how to divide G into $G_{\rm u}$ and $G_{\rm d}$.

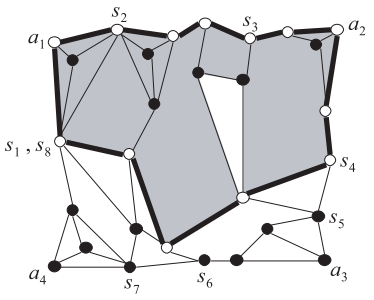

Figure 15: Graph G.

Consider all the inner faces of G that contain one or more vertices on the path going from a_1 to a_2 clockwise on $F_{\rm o}(G)$. (All these faces for the graph G in Fig. 1(a) are shaded in Fig. 15.) Let G' be the subgraph of G induced by all the vertices on these faces. Then $F_{\rm o}(G')$ is a simple cycle, which is drawn by thick lines in Fig. 15. Clearly, $F_{\rm o}(G')$ contains vertices s_1 and s_4 in all Cases (a)–(f), contains vertex s_8 in Cases (b), (e) and (f), and contains vertex s_5 in Cases (b), (d) and (f). For Cases (a) and (c), let P be the path going from s_1 to s_4 counterclockwise on $F_{\rm o}(G')$, as illustrated in Figs. 1(a), 14(a) and 14(c) where P is drawn by thick lines. For Cases (b) and (f), let P be the path going from s_8 to s_5 counterclockwise on $F_{\rm o}(G')$, as illustrated in Figs. 14(b) and (f). For Case (d), let P be the path going from s_1 to s_5 counterclockwise on $F_{\rm o}(G')$, as illustrated in Fig. 14(d). For Case (e), let P be the path going from s_8 to s_4 counterclockwise on $F_{\rm o}(G')$, as illustrated in Fig. 14(e).

Let $G_{\rm d}$ be the subgraph of G induced by all the vertices on P or below P, and let $G_{\rm u}$ be the subgraph of G obtained by deleting all vertices in $G_{\rm d}$ as illustrated in Fig. 1(b). For every edge e of G that is contained neither in $G_{\rm u}$ nor in $G_{\rm d}$, one of the ends of e is on $F_{\rm o}(G_{\rm u})$ and the other is on $F_{\rm o}(G_{\rm d})$. Let $n_{\rm d}$ be the number of vertices of $G_{\rm d}$, and let $n_{\rm u}$ be the number of vertices of $G_{\rm u}$. Then $n_{\rm d} + n_{\rm u} = n$.

4.2 Drawing of $G_{\rm d}$

We now explain how to draw $G_{\rm d}$.

Let $G'_{\rm d}$ be a graph obtained from G by contracting all the vertices of $G_{\rm u}$ to a single vertex w, as illustrated in Fig. 5(a) for the graph G in Fig. 1(a). Then clearly $G'_{\rm d}$ is biconnected. We now claim that $G'_{\rm d}$ is internally triconnected. Suppose for a contradiction that $G'_{\rm d}$ is not internally triconnected. Then $G'_{\rm d}$ has a separation pair $\{u, v\}$ for which there is a split graph H illustrated in Figs. 3(a), (b) and (c). Since G is internally triconnected and all the vertices of $G'_{\rm d}$ except w are vertices of G, one may assume that $u = w$ and H has a structure in Fig. 3(b) or (c). We consider only the case where $G'_{\rm d}$ has a structure in Fig. 3(b), because one can similarly give a proof for the other case where $G'_{\rm d}$ has a structure in Fig. 3(c). Then $G'_{\rm d}$ has a structure as illustrated in Fig. 16(a), and hence G has a structure illustrated in Fig. 16(b) or (c). If G has a structure in Fig. 16(b), then G is not internally triconnected, a contradiction. If G has a structure in Fig. 16(c), then all the vertices above the path P (drawn by thick lines in Fig. 16(c)) should be contracted to a single vertex w in $G'_{\rm d}$ and hence $G'_{\rm d}$ does not have a structure in Fig. 16(a), a contradiction.

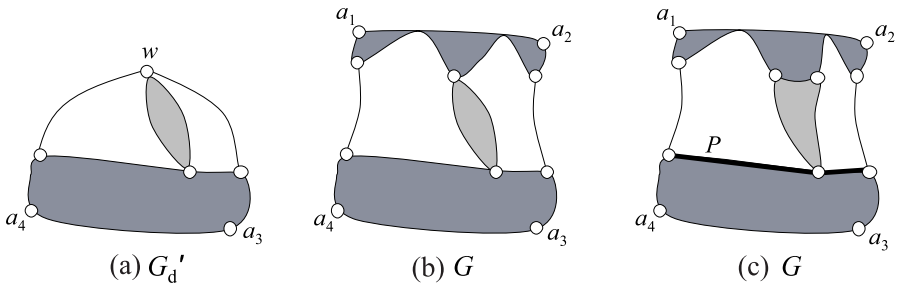

Figure 16: (a) Graph $G'_{\rm d}$ having a structure in Fig. 3(b), (b) G which is not internally triconnected, and (c) G and the path P.

The decomposition tree $T(G'_{\rm d})$ of $G'_{\rm d}$ 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 $G'_{\rm d}$ other than w has degree three or more, and w has degree two or more in $G'_{\rm d}$. Therefore, $G'_{\rm d}$ 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. 5(b), where $U_m = \{w\}$. Let v_l be the vertex preceding w clockwise on the outer face $F_{\rm o}(G'_{\rm d})$, and let v_r be the vertex succeeding w, as illustrated in Fig. 5(a). We obtain a pentagonal drawing D_m of $G'_{\rm d}$ by the algorithm in Section 3, as illustrated in Fig. 5(d). The drawing D_{m-1} of G_{m-1} induced by $U_1 \bigcup U_2 \bigcup \cdots \bigcup U_{m-1}$ is our drawing $D_{\rm d}$ of $G_{\rm d} (= G_{m-1})$. (See Figs. 1(d) and 5(c).) By Lemma 5, we have $W(D_{\rm d}) = 2n_{\rm d} - 2$ and $H(D_{\rm d}) \leq n_{\rm d}^2 - n_{\rm d} - 2$.

4.3 Drawing of $G_{\rm u}$

We now explain how to draw $G_{\rm u}$. Let $G'_{\rm u}$ be a graph obtained from G by contracting all the vertices of $G_{\rm d}$ to a single vertex w', as illustrated in Fig. 17(a).

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. 17(b), where $U_m = \{w'\}$. Let v'_r be the vertex succeeding w' clockwise on the outer face $F_{\rm o}(G'_{\rm u})$, and let v'_l be the vertex preceding w', as illustrated in Fig. 17(a). We rotate $G'_{\rm u}$ by 180° as illustrated in Fig. 17(b). We then obtain a drawing D_{m-1} of $G_{\rm u}(= G_{m-1})$ by the algorithm in Section 3, as illustrated in Fig. 17(c) where the clockwise path from a_2 to v'_l and the clockwise path from v'_r to a_1 on $F_{\rm o}(G'_{\rm u})$ are drawn as vertical line segments. (The path from v'_r to a_1 degenerates into a single path in Fig. 17(c) since $v'_r = a_1$.) We finally rotate it to obtain a drawing $D_{\rm u}$ of $G_{\rm u}$, as illustrated in Fig. 1(c). By Lemma 5, we have $W(D_{\rm u}) = 2n_{\rm u} - 2$ and $H(D_{\rm u}) \leq n_{\rm u}^2 - n_{\rm u} - 2$.

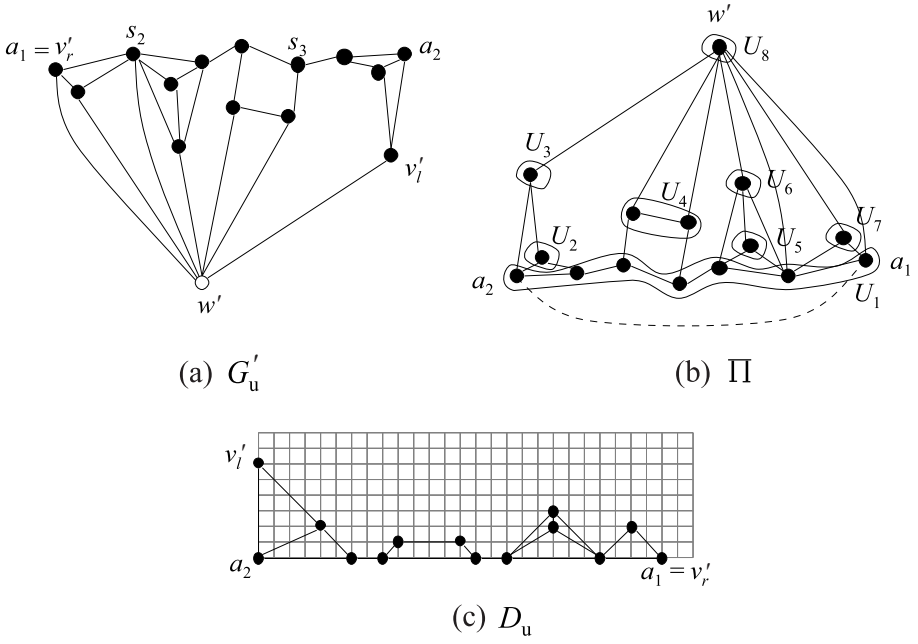

Figure 17: (a) Graph $G'_{\rm u}$, (b) a canonical decomposition of $G'_{\rm u}$, and (c) a drawing $D_{\rm u}$ of $G_{\rm u}$.

4.4 Drawing of G

If $W(D_{\rm d}) \neq W(D_{\rm u})$ as in Figs. 1(c) and (d), then we widen the narrow one of $D_{\rm d}$ and $D_{\rm u}$ by the shift operation in Section 3 so that both have the same width. (Since $W(D_{\rm u}) = W(D_{\rm d}) - 2$ for $D_{\rm u}$ in Fig. 1(c) and $D_{\rm d}$ in Fig. 1(d), we widen $D_{\rm u}$ by shifting vertex a_1 to the left by two as illustrated in Fig. 1(e).) We may thus assume that $W(D_{\rm d}) = W(D_{\rm u}) = \max\{2n_{\rm d} - 2, 2n_{\rm u} - 2\}$. 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

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

We arrange $D_{\rm d}$ and $D_{\rm u}$ so that $y(a_3) = y(a_4) = 0$ and $y(a_1) = y(a_2) = H(D_{\rm d}) + H(D_{\rm u}) + W(D) + 1$, as illustrated in Fig. 1(e). Noting that $n_{\rm d} + n_{\rm u} = n$ and $n_{\rm d}, n_{\rm u} \geq 5$, we have

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

We finally draw, by straight line segment, all the edges of G that are contained in neither $G_{\rm u}$ nor $G_{\rm d}$. This completes the grid drawing D of G. (see Fig. 1(e).)

4.5 Validity of drawing algorithm

In this section, we show that the drawing D obtained above is a convex grid drawing of G. Since both $D_{\rm d}$ and $D_{\rm u}$ satisfy condition (sh5), every inner facial cycle of $G_{\rm d}$ and $G_{\rm u}$ is drawn as a convex polygon in D. Therefore, it suffices to show that the straight line drawings of the edges not contained in $G_{\rm u}$ and $G_{\rm d}$ do not introduce any edge-intersection and that all the faces newly created by these edges are convex polygons.

Since $D_{\rm d}$ satisfies condition (sh3), the absolute value of the slope of every edge on the path $P_{\rm d}$ going from v_l to v_r clockwise on $F_{\rm o}(G_{\rm d})$ is less than or equal to 1. The path $P_{\rm d}$ is drawn by thick lines in Fig. 1(d). (Note that $D_{\rm d}$ may be widen by the shift operation if $W(D_{\rm d}) < W(D_{\rm u})$.) Similarly, the absolute value of the slope of every edge on the path $P_{\rm u}$ going from v'_r to v'_l counterclockwise on $F_{\rm o}(G_{\rm u})$ is less than or equal to 1. Since $H(D) = H(D_{\rm d}) + H(D_{\rm u}) + W(D) + 1$, the absolute value of the slope of every straight line segment that connects a vertex in $G_{\rm u}$ and a vertex in $G_{\rm d}$ is larger than 1. Therefore, all the outer vertices of $G_{\rm d}$ on $P_{\rm d}$ are visible from all the outer vertices of $G_{\rm u}$ on $P_{\rm u}$. Furthermore, G is a plane graph. Thus the addition of all the edges not contained in $G_{\rm u}$ and $G_{\rm d}$ does not introduce any edge-intersection.

Since $D_{\rm d}$ satisfies condition (sh6), every convex apex of the outer polygon of $G_{\rm d}$ on $P_{\rm d}$ has one or more neighbors in $G_{\rm u}$. Similarly, every convex apex of the outer polygon of $G_{\rm u}$ on $P_{\rm u}$ has one or more neighbors in $G_{\rm d}$. Therefore, every interior angle of a newly formed face is smaller than 180°. Thus all the inner faces of G not contained in $G_{\rm u}$ and $G_{\rm d}$ are convex polygons in D.

Thus, D is a convex grid drawing of G. Clearly the algorithm takes linear time. We thus have the following main theorem.

Theorem 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 $2n \times n^2$ grid in linear time.*

We finally remark that the grid size is improved to $2n \times 4n$ for the case where either the node u_4 of degree four in $T(G)$ is a ring as illustrated in Fig. 14(b) or $T(G)$ has two nodes of degree three as illustrated in Figs. 14(c)–(f). In such a case, the paths P and hence $P_{\rm d}$, drawn by thick lines in Figs. 1(a), 1(d) and 14(b)–(f), pass through the outer vertices s_6 and s_7. Clearly $y(s_6) = y(s_7) = 0$ in $D_{\rm d}$. Therefore, condition (sh3) implies that $H(D_{\rm d}) < W(D_{\rm d}) = 2n_{\rm d} - 2$. Similarly we have $H(D_{\rm u}) < W(D_{\rm u}) = 2n_{\rm u} - 2$. Thus we have

$$H(D) = H(D_{\rm d}) + H(D_{\rm u}) + W(D) + 1 < 4n.$$

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 $2n \times n^2$ grid, and we present a linear algorithm to find such a drawing. This is the first algorithm that finds a convex grid drawing of such a graph G on a grid of polynomial size. The remaining problem is to obtain an algorithm for an internally triconnected plane graph whose decomposition tree has five or more leaves.

Acknowledgments

The authors would like to thank the referees for helpful comments.

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] H. de Fraysseix, J. Pach and R. Pollack, *How to draw a planar graph on a grid*, Combinatorica, 10, pp. 41–51, 1990.

[6] S. Felsner, *Convex drawings of plane graphs and the order of dimension of 3-polytopes*, Order, 18, pp. 19–37, 2001.

[7] J. E. Hopcroft and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM J. Compt.2, 3, pp. 135–158, 1973.

[8] É. Fusy, *Transversal structures on triangulations, with application to straight-line drawing*, Proc. of GD2005, LNCS 3843 pp. 177–188, 2006.

[9] A. Kamada, K. Miura and T. Nishizeki, *Convex grid drawings of plane graphs with rectangular contours -Extended Abstract-*, Proc. of ISAAC 2006, Springer Lect. Notes in Computer Science, 4288, pp. 131–140, 2006.

[10] 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.

[11] 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.

[12] K. Miura, S. Nakano and T. Nishizeki, *Convex grid drawings of four-connected plane graphs*, International Journal of Foundations of Computer Science, 17, 5, pp. 1032–1060, 2006.

[13] T. Nishizeki and M. S. Rahman, *Planar Graph Drawing*, World Scientific, Singapore, 2004.

[14] G. Rote, *Strictly convex drawings of planar graphs*, Proc. of 16th Ann. ACM-SIAM Symp. on Discrete Algorithms, pp. 728–734, 2005.

[15] C. Thomassen, *Planarity and duality of finite and infinite graphs*, J. Combinatorial Theory, Series B, 29, pp. 244–271, 1980.

[16] W. T. Tutte, *Connectivity in Graphs*, University of Tronto Press, Tronto, 1966.

[17] W. Schnyder and W. Trotter, *Convex drawings of planar graphs*, Abstracts of the AMS 13, 5, 92T-05-135, 1992.

Appendix

[The details of a pentagonal drawing algorithm]

In the appendix we explain the pentagonal drawing algorithm outlined in Section 3 in detail.

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 Figs. 5(a) and (b).

Whenever a vertex v is shifted, that is, moved to the left or right, during the execution of the algorithm, a set $L(v)$ of vertices need to be moved together with v, similarly as the algorithms in [2] and [5]. Note that $v \in L(v)$. (See [13] for example.)

We denote the current position of a vertex v by $P(v)$; $P(v)$ is expressed by its x- and y-coordinates as $(x(v), y(v))$. If the Manhattan distance between two grid points $P(v_1) = (x_1, y_1)$ and $P(v_2) = (x_2, y_2)$ is an even number, then the intersection $Q(v_1, v_2)$ of the straight line with slope $+1$ through $P(v_1)$ and the straight line with slope -1 through $P(v_2)$ is a grid point. Clearly

$$Q(v_1, v_2) = \left(\frac{1}{2}(x_1 - y_1 + x_2 + y_2), \frac{1}{2}(-x_1 + y_1 + x_2 + y_2)\right).$$

We say that a vertex $v \in U_k$, $1 \leq k \leq m$, has *rank* k. Let $F_o(G_{k-1}) = w_1, w_2, \cdots, w_t$ for $k, 2 \leq k \leq m$. Then the definition of a canonical decomposition implies that there is a pair of indices a and b, $1 \leq a < b \leq t$, such that each of w_a and w_b has a neighbor in $\overline{G_{k-1}}$ but every vertex w_i, $a < i < b$, is an inner vertex of G and has no neighbor in $\overline{G_{k-1}}$. Then the path going from w_a to w_b clockwise on $F_o(G_{k-1})$ is a part of an inner facial cycle F of G; F contains two edges connecting w_a and w_b with $\overline{G_{k-1}}$, plus possibly some edges in $\overline{G_{k-1}}$. Let c, $a \leq c < b$, be an index such that w_c has the smallest rank among the vertices $w_a, w_{a+1}, \cdots, w_{b-1}$. If there are two or more vertices with the smallest rank, then let w_c be the leftmost one, that is, let c be the smallest index of these vertices. We denote the index c for a and b by $\mu_k^+(a)$ and $\mu_k^-(b)$, and hence $c = \mu_k^+(a) = \mu_k^-(b)$. The superscript $+$ indicates $a \leq c$, while the superscript $-$ indicates $c < b$. Note that if $b = a + 1$ then $a = \mu_k^+(a) = \mu_k^-(b)$.

We are now ready to describe the pentagonal drawing algorithm in detail. We first obtain a drawing D_1 of the subgraph G_1 of G induced by all vertices of U_1 as follows. Let $F_o(G_1) = w_1, w_2, \cdots, w_t$, $w_1 = v_1$, and $w_t = v_2$. We draw G_1 as illustrated in Fig. 8, depending on whether (v_1, v_2) is a real edge or not, $w_2 \in V_l$ or not, and $w_{t-1} \in V_r$ or not. More precisely, we draw G_1 as follows.

Initialize:

Set $L(w_i) = \{w_i\}$ for each $i, 1 \leq i \leq t$;

Case 1: v_1 and v_2 are adjacent in an original graph G, that is, (v_1, v_2) is a real edge (see Figs. 8(a)–(d)).

Set $P(w_1) = (0, 0)$;

Set $P(w_t) = (2t - 2, 0)$;

Case (a): $w_2 \in V_l$ and $w_{t-1} \notin V_r$ (see Fig. 8(a)).
Set $P(w_i) = (2i-4, 4)$ for each $i, 2 \leq i \leq t-1$;

Case (b): $w_2 \notin V_l$ and $w_{t-1} \in V_r$ (see Fig. 8(b)).
Set $P(w_i) = (2i, 4)$ for each $i, 2 \leq i \leq t-1$;

Case (c): $w_2 \in V_l$ and $w_{t-1} \in V_r$ (see Fig. 8(c)).
Set $P(w_i) = (2i-4, 2)$ for each $i, 2 \leq i \leq t-2$;
Set $P(w_{t-1}) = (2t-2, 2)$;

Case (d): $w_2 \notin V_l$ and $w_{t-1} \notin V_r$ (see Fig. 8(d)).
Set $P(w_i) = (2i-2, 2)$ for each $i, 2 \leq i \leq t-1$;

Case 2: Otherwise, that is, (v_1, v_2) is a virtual edge (see Fig. 8(e)).
Set $P(w_i) = (2i-2, 0)$ for each $i, 1 \leq i \leq t$.

We then extend a drawing D_{k-1} of G_{k-1} to a drawing D_k of G_k for each index k, $2 \leq k \leq m$, similarly as the algorithm by Chrobak and Kant for finding a convex grid drawing of a triconnected plane graph [2]. Let $F_{\rm o}(G_{k-1}) = w_1, w_2, \cdots, w_t$, $w_1 = v_1$, $w_t = v_2$, and $U_k = \{u_1, u_2, \cdots, u_r\}$. The first vertex u_1 and the last one u_r in U_k have neighbors in G_{k-1}. Let w_p be the leftmost neighbor of u_1, and let w_q be the rightmost neighbor of u_r, as illustrated in Figs. 10 and 11. Let $\alpha = \mu_k^+(p)$, and let $\beta = \mu_k^-(q)$. If $U_k = \{u_1\}$ and u_1 has three or more neighbors in G_{k-1} as illustrated in Fig. 10, then at least one of the vertices $w_{p+1}, w_{p+2}, \cdots, w_{q-1}$ has a neighbor in $\overline{G_{k-1}}$ and hence $\alpha < \beta$; in fact, w_α and w_β will belong to two different inner faces of G_k, to the first and last faces among those that are created when adding u_1 to G_{k-1}. If either $U_k = \{u_1\}$ and u_1 has exactly two neighbors in G_k or $|U_k| \geq 2$ as illustrated in Fig. 11, then all vertices $w_{p+1}, w_{p+2}, \cdots, w_{q-1}$ belong to the same inner face of G_k and none of them has a neighbor in $\overline{G_{k-1}}$ and hence $\alpha = \beta$.

We extend D_{k-1} to D_k, $2 \leq k \leq m$, so that D_k satisfies the six conditions (sh1)–(sh6) in Section 3, as follows.

Update:

$$\begin{aligned}
\text{Set } L(w_p) &= \bigcup_{i=p}^{\alpha} L(w_i); \\
\text{Set } L(w_q) &= \bigcup_{i=\beta+1}^{q} L(w_i); \\
\text{Set } L(u_1) &= \{u_1\} \cup \left(\bigcup_{i=\alpha+1}^{\beta} L(w_i) \right); \\
\text{Set } L(u_i) &= \{u_i\} \text{ for each } i, 2 \leq i \leq r;
\end{aligned}$$

Shift:

For each vertex $v \in \bigcup_{i=1}^{p} L(w_i)$, set $x(v) = x(v) - r$;

For each vertex $v \in \bigcup_{i=q}^{t} L(w_i)$, set $x(v) = x(v) + r$;

For each vertex $v \in \bigcup_{i=1}^{t} L(w_i)$, set $x(v) = x(v) + r$;

Install U_k:

Case 1: $|U_k| = 1$.

Case (a): $u_1 \in V_l$ (see Fig. 10(a)).
Set $P(u_1) = \big(x(w_p), x(w_q) + y(w_q)\big)$;

Case (b): $u_1 \in V_r$ (see Fig. 10(b)).
Set $P(u_1) = \big(x(w_q), -x(w_p) + y(w_p) + x(w_q)\big)$;

Case (c): $u_1 \notin V_l$ and $u_1 \notin V_r$ (see Fig. 10(c)).
Set $P(u_1) = Q(w_p, w_q)$;

Case 2: $|U_k| \geq 2$.

Case (a): $u_1 \in V_l$ and $u_r \notin V_r$D
If $y(w_q) - y(w_p)$ is an odd number ≥ 1 (see Fig. 11(a)),
set $y(u_i) = \max\{y(w_p), y(w_q)\} + 1$ for each $i, 1 \leq i \leq r$;
Otherwise (see Fig. 11(a′)),
set $y(u_i) = \max\{y(w_p), y(w_q)\} + 2$ for each $i, 1 \leq i \leq r$;
Set $x(u_i) = x(w_p) + 2(i-1)$ for each $i, 1 \leq i \leq r-1$;
Set $x(u_r) = x(w_q) - \big\{y(u_r) - y(w_q)\big\}$;

Case (b): $u_1 \notin V_l$ and $u_r \in V_r$D
If $y(w_p) - y(w_q)$ is an odd number ≥ 1 (see Fig. 11(b)),
set $y(u_i) = \max\{y(w_p), y(w_q)\} + 1$ for each $i, 1 \leq i \leq r$;
Otherwise (see Fig. 11(b′)),
set $y(u_i) = \max\{y(w_p), y(w_q)\} + 2$ for each $i, 1 \leq i \leq r$;
Set $x(u_i) = x(w_p) + \big\{y(u_1) - y(w_p)\big\} + 2(i-1)$ for each $i, 1 \leq i \leq r-1$;
Set $x(u_r) = x(w_q)$;

Case (c): $u_1 \in V_l$ and $u_r \in V_r$ (see Fig. 11(c)).
Set $y(u_i) = \max\{y(w_p), y(w_q)\} + 1$ for each $i, 1 \leq i \leq r$;
Set $x(u_i) = x(w_p) + 2(i-1)$ for each $i, 1 \leq i \leq r-1$;
Set $x(u_r) = x(w_q)$;

Case (d): $u_1 \notin V_l$ and $u_r \notin V_r$ (see Fig. 11(d)).

Set $y(u_i) = \max\{y(w_p), y(w_q)\} + 1$ for each $i, 1 \leq i \leq r$;

Set $x(u_i) = x(w_p) + \big\{y(u_1) - y(w_p)\big\} + 2(i-1)$ for each $i, 1 \leq i \leq r-1$;

Set $x(u_r) = x(w_q) - \big\{y(u_r) - y(w_q)\big\}$.