CA-001

Convex Grid Drawings of Internally Triconnected Plane Graphs with Pentagonal Contours

Kazuyuki Miura *

Abstract

In a convex grid drawing of a plane graph, all edges are drawn as straight-line segments without any edge-intersection, all vertices are put on grid points and all facial cycles are drawn as convex polygons. 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. An internally triconnected plane graph G has a convex grid drawing on a $2n \times 2n$ grid if $T(G)$ has exactly four leaves. Furthermore, an internally triconnected plane graph G has a convex grid drawing on a $20n \times 16n$ grid if $T(G)$ has exactly five leaves.

In this paper, we show that an internally triconnected plane graph G has a convex grid drawing on a $10n \times 5n$ grid if $T(G)$ has exactly five leaves. 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 [1, 2, 4, 5, 8, 9, 10, 11, 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 [9].

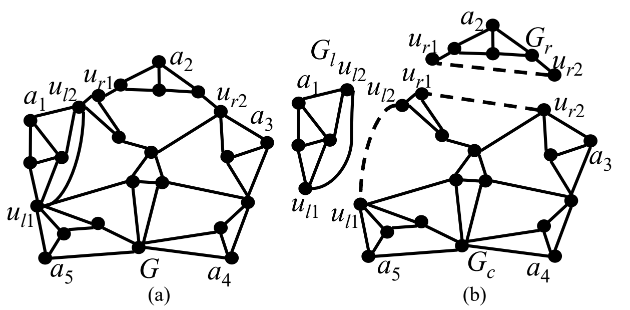

Figure 1: (a) Plane graph G and (b) subgraphs G_l, G_r and G_c.

A convex drawing of a plane graph is called a *convex grid drawing* if all vertices are put on grid points of integer coordinates. 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 [7, 12]. 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, 6]. An internally triconnected plane graph G has a convex grid drawing on a $2n\times 2n$ grid if $T(G)$ has exactly four leaves [4, 8, 13]. Furthermore, an internally triconnected plane graph G has a convex grid drawing on a $20n \times 16n$ grid if $T(G)$ has exactly five leaves [5, 10]. Figure 1(a) depicts an internally triconnected plane graph G, Fig. 3(c) the triconnected component decomposition tree $T(G)$ of G, which has five leaves l_1, l_2, l_3, l_4 and l_5.

In this paper, we improve the area in the case where $T(G)$ has exactly five leaves. More precisely, we show that an internally triconnected plane graph G has a convex grid drawing on a $10n \times 5n$ grid if $T(G)$ has exactly five leaves, and present an algorithm to find such a drawing in linear time.

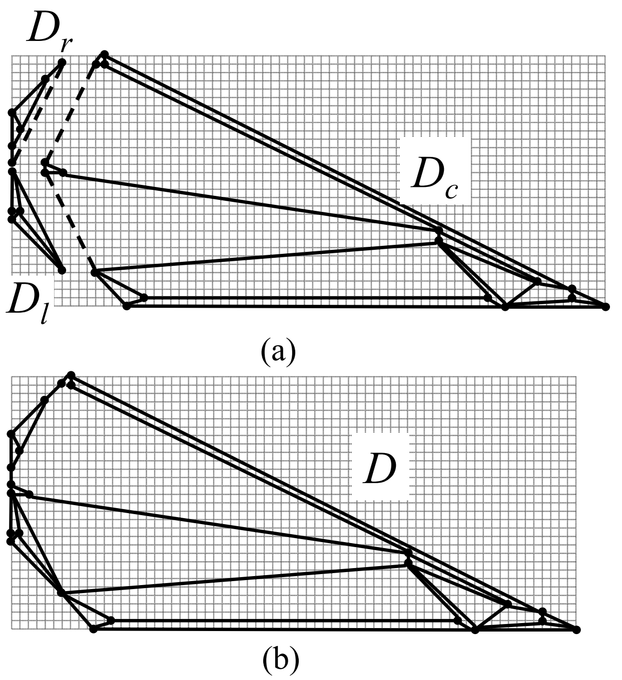

Figure 2: (a) convex grid drawings D_l of G_l, D_r of G_r, and D_c of G_c, and (b) convex grid drawing D of G.

The algorithm is outlined as follows: we first divide a plane graph G into a left subgraph G_l, a right subgraph G_r and a center subgraph G_c as illustrated in Fig. 1(b) for the graph in Fig. 1(a); we then construct convex grid drawings with triangular contours of G_l and G_r and construct a convex grid drawing with heptagonal contour of G_c by a so-called shift method as illustrated in Fig. 2(a); we finally combine these three drawings to a convex grid drawing

*Faculty of Symbiotic Systems Science Fukushima University, Fukushima 960-1296, Japan.

with pentagonal contour of G as illustrated in Fig. 2(b).

The remainder of the paper is organized as follows. In Section 2 we give some definitions and known lemmas. In Section 3 we explain an algorithm for G_l, G_r and G_c. In Section 4 we present our convex grid drawing algorithm. Finally we conclude in Section 5.

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 unbounded 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 $F_o(G)$ the outer facial cycle of G. A vertex on $F_o(G)$ is called an *outer vertex*. In a convex drawing of a plane graph G, all facial cycles must be drawn as convex polygons. The convex polygonal drawing of $F_o(G)$ 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.

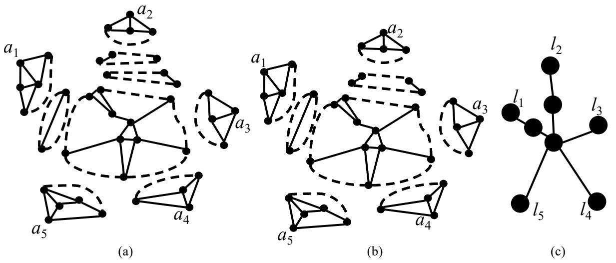

Figure 3: (a) Split components of the graph G in Fig. 1(a), (b) triconnected components of G, and (c) a decomposition tree $T(G)$.

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) and (b).

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

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

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 *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. 3(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 nine split components illustrated in Fig. 3(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 (which is not a triconnected graph, of course). The split components of G are not necessarily unique, but the triconnected components of G are unique [3]. Two triangles in Fig. 3(a) are merged into a single ring, and hence the graph in Fig. 1(a) has eight triconnected components as illustrated in Fig. 3(b).

For a separation pair $\{u, v\}$ of G, two triconnected components H_i and H_j $(i \neq j)$ are called the *triconnected components of G with respect to $\{u, v\}$* if it is possible to merge H_i with H_j at $\{u, v\}$. 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. 3(c). We call $T(G)$ a *triconnected component decomposition tree* or simply a *decomposition tree* of G [3].

We denote by $\ell(G)$ the number of leaves of $T(G)$. Then $\ell(G) = 5$ for the graph G in Fig. 1(a). (See Fig. 3(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 *[3] A decomposition tree $T(G)$ of a graph G can be found in linear time.*

Lemma 2 *[7] 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 *[7] 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 neither a bond nor a ring but a triconnected graph. Lemmas 2 and 3 imply that if $T(G)$ has exactly five leaves, that is, $\ell(G) = 5$ then the outer polygon of every convex drawing of G must have five or more apices. Our algorithm finds a convex grid drawing of G whose outer polygon is a pentagon and hence has exactly five apices, as illustrated in Fig. 2(b).

In Section 3, we will present an algorithm to draw the center subgraph G_c, the left subgraph G_l and the right subgraph G_r. (See Fig. 2(a).) These algorithms use the following "canonical decomposition." Let $G = (V, E)$ be an internally triconnected plane graph, and let

$V = \{v_1, v_2, \ldots, v_n\}$. 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 $U_1, U_2, \cdots, U_m$, where $U_1 \cup U_2 \cup \cdots \cup U_m = V$ and $U_i \cap U_j = \emptyset$ for any indices i and j, $1 \le i < j \le m$. We denote by G_k, $1 \le k \le m$, the subgraph of G induced by $U_1 \cup U_2 \cup \cdots \cup U_k$, and denote by $\overline{G_k}$, $0 \le k \le m-1$, the subgraph of G induced by $U_{k+1} \cup U_{k+2} \cup \cdots \cup U_m$. Clearly, $G_k = G - U_{k+1} \cup U_{k+2} \cup \cdots \cup U_m$ 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, $1 \le k \le m$, G_k is internally triconnected.

(cd3) For each index k, $2 \le k \le m$, 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 $|U_k| \ge 2$, 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).

Although the definition of a canonical decomposition above is slightly different from the one given in [1], they are effectively equivalent to each other. A canonical decomposition $\Pi = (U_1, U_2, \cdots, U_{10})$ with respect to vertices v_1, v_2 and v_n of the graph in Fig. 5(a) is illustrated in Fig. 5(b).

By the condition (cd3), one may assume that all the vertices in U_k, $1 \le k \le m$, consecutively appear clockwise on $F_o(G_k)$. We number all vertices of G by $1, 2, \cdots, n$ so that they appear in $U_1, U_2, \cdots, U_m$ in this order. We call each vertex in G by the number i, $1 \le i \le n$. Thus one can define an order $<$ on the vertices in G. For a vertex u, $1 \le u \le n-1$, we denote by $w^*(u)$ the largest neighbor of u.

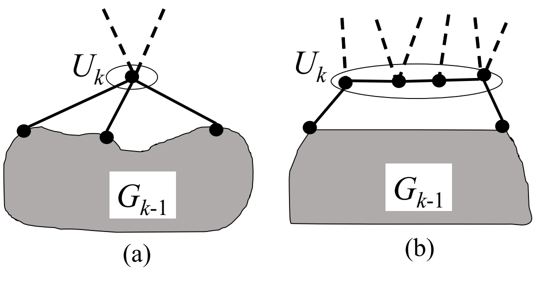

Figure 4: (a) Graphs G_k for which $|U_k| = 1$ and (b) $|U_k| \ge 2$.

The following lemma is known.

Lemma 4 *[6] Assume that G is an internally triconnected plane graph and $\ell(G) \le 3$. 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: $\ell(G) = 2$.

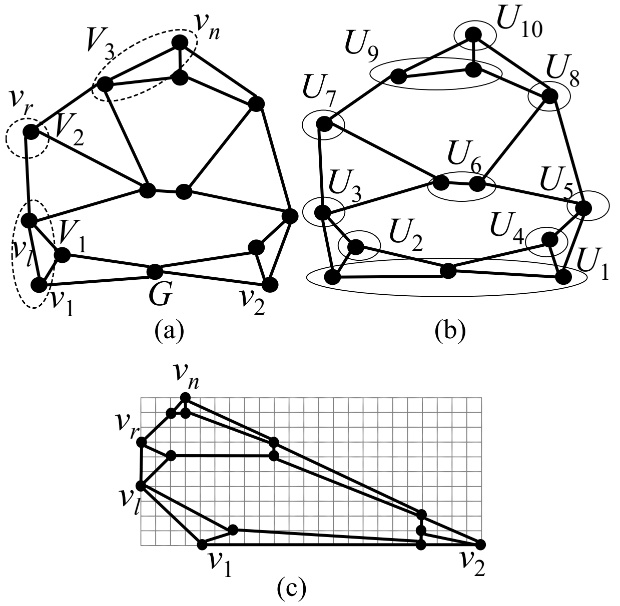

Figure 5: (a) An internally triconnected plane graph G, (b) a canonical decomposition Π of G, and (c) a pentagonal drawing of G.

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 triconneced. We choose three arbitrary outer vertices of G.

One can easily observe that Lemma 4 holds even if exactly one (resp. two) of the outer vertex has (resp. vertices have) degree two and the vertex is (resp. vertices are) chosen as v_n (resp. v_2 and v_n).

3 Extended Triangular 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). Miura [5] give a linear-time algorithm, called the *triangular drawing algorithm*, to find a convex grid drawing of G with a triangular outer polygon. In this section, we present a linear-time algorithm, called a *extended triangular drawing algorithm* extending the triangular drawing algorithm, to find a convex grid drawing of G with a pentagonal outer polygon. The algorithm is based on the so-called shift methods given by Chrobak and Kant [1] and de Fraysseix *et al.* [2]. This algorithm will be used by our convex grid drawing algorithm to draw the left subgraph G_l, the right subgraph G_r and the center subgraph G_c of G in Sections 4.2, 4.3, and 4.4, respectively.

We now outline the extended triangular drawing algorithm. Let v_l be an arbitrary vertex on the path going from v_1 to v_n clockwise on $F_o(G)$, and let v_r be an arbitrary vertex on the path going from v_l to v_n clockwise on $F_o(G)$ as illustrated in Fig. 5(a). Let V_1 be the set of all vertices on the path going from v_1 to v_l clockwise on $F_o(G)$, let V_2 be an arbitrary vertex on the path going from v_l to v_r clockwise on $F_o(G)$ and let V_3 be an arbitrary vertex on the path going from v_r to v_n clockwise on $F_o(G)$, as illustrated in Fig. 5(a). The extended triangular drawing algorithm finds a convex grid drawing of G whose outer polygon is a pentagon with apices v_1, v_l, v_r, v_n and v_2, such that the side $v_1 v_l$ has slope -1 the side $v_l v_r$ has slope ∞, the side $v_r v_n$ has slope $+1$, the side $v_n v_2$ has slope $-1/2$, and the side v_2v_1 has slope 0, respectively, as illustrated in Fig. 5(c).

We are now ready to describe the extended triangular drawing algorithm in detail. We first obtain a drawing D_1 of the subgraph G_1 of G induced by all vertices in U_1 as follows. 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. 6, depending on whether (v_1, v_2) is a real edge or not, and $w_2 \in V_1$ or not.

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 (see Figs. 6(a) and (b)).
Set $P(w_1) = (0, 0)$;

Case 1(a): $w_2 \in V_1$ (see Fig. 6(a)).
Set $P(w_i) = (i - 3, 1)$ for each i, $2 \le i \le t - 1$;
Set $P(w_t) = (t - 2, 0)$;

Case 1(b): $w_2 \notin V_1$ (see Fig. 6(b)).
Set $P(w_i) = (i - 1, 1)$ for each i, $2 \le i \le t - 1$;
Set $P(w_t) = (t, 0)$;

Case 2: Otherwise, that is, (v_1, v_2) is a virtual edge (see Fig. 6(c)).
Set $P(w_i) = (i - 1, 0)$ for each i, $1 \le i \le t$.

We then extend a drawing D_{k-1} of G_{k-1} to a drawing D_k of G_k for each index k, $2 \le k \le m$, similarly as the algorithm by Chrobak and Kant for finding a convex grid drawing of a triconnected plane graph [1]. For each k, $2 \le k \le m$, 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. Let $U_k = \{u_1, u_2, \cdots, u_r\}$. 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 $F_{\rm o}(G_k)$ 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, and let w_q be the rightmost neighbor of u_r.

Let w_f be the vertex with the maximum index f among all the vertices w_i, $1 \le i \le t$, on $F_{\rm o}(G_{k-1})$ that are contained in V_1, let w_g be the vertex with the maximum index g among all the vertices w_i, $1 \le i \le t$, on $F_{\rm o}(G_{k-1})$ that are contained in V_2 (in any) and let w_h be the vertex with the maximum index h among all the vertices w_i, $1 \le i \le t$, on $F_{\rm o}(G_{k-1})$ that are contained in V_3 (in any), respectively Of couse, $1 \le f \le g \le f < t$. 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 five conditions (sh1)–(sh5).

(sh1) $P(w_1) = (0, 0)$, $x(w_t) \le 3|V(G_{k-1})|$ and $y(w_t) = 0$.

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

(sh3) Every edge (w_i, w_{i+1}), $1 \le i \le t - 1$, has slope -1, $-1/2$, 0 or $[1, +\infty]$.

(sh4) Every inner face of G_{k-1} is drawn as a convex polygon.

(sh5) Vertex w_i, $2 \le i \le t - 1$, has one or more neighbors in $\overline{G_{k-1}}$ if w_i is a convex apex.

Indeed D_1 satisfies the five conditions above. We extend D_{k-1} to D_k, $2 \le k \le m$, so that D_k satisfies the five conditions, as follows.

In our algorithm, we wish to put the vertex u_1 of U_k on a grid point so that the edge (w_p, u_1) has slope such that -1 (if $u_1 \in V_1$), $+\infty$ (if $u_1 \in V_2$), $+1$ (if $u_1 \in V_3$), or $[1, +\infty]$ (otherwise), respectively and put the vertex u_r on a grid point so that the edge (u_r, w_q) has slope $-1/2$. Furthermore, if $|U_k| \ge 2$, then we wish to put the vertices $u_2, u_3, \cdots, u_{r-1}$ so that, for each i, $1 \le i \le r - 1$, the edges (u_i, u_{i+1}) has slope 0 and the distance between two vertices u_i and u_{i+1} is equal to 1. For this purpose, before installing $U_k = \{u_1, u_2, \cdots, u_l\}$ to D_{k-1}, we shift some vertices of G_{k-1} to the right as illustrated in Figs. 7(a)–(d), as follows.

Let ϵ be 0 if $u_1 = w^*(w_p)$, and 1 otherwise. If $x(w_q) - x(w_p) + \epsilon$ is an odd number, as illustrated in Fig. 7(a), then we 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 $|U_k|$, as illustrated in Fig. 7(b). Otherwise, $(x(w_q) - x(w_p) + \epsilon$ is an even number, as illustrated in Fig. 7(c),) we 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 $|U_k| + 1$, as illustrated in Fig. 7(d). Furthermore, if $u_1 \in V_3$, then we 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 $|U_k|$.

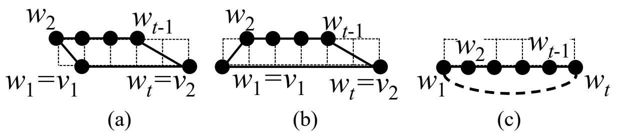

Figure 6: Drawings D_1 of G_1 (a), (b) for Case 1 and (c) for Case 2.

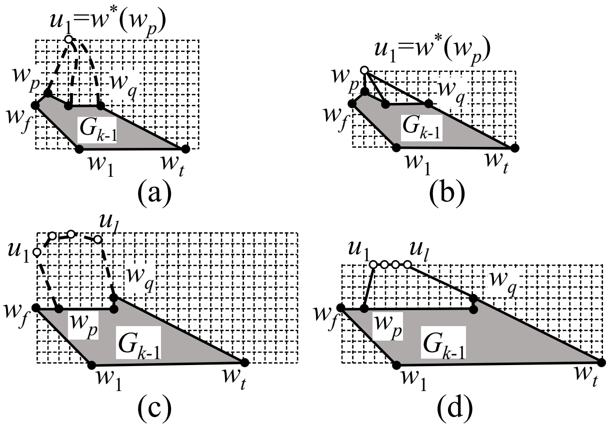

Figure 7: Graphs (a) G_{k-1} and (b) G_k for Case $x(w_q) - x(w_p) + \epsilon$ is an odd number, graphs (c) G_{k-1} and (d) G_k for Case $x(w_q) - x(w_p) + \epsilon$ is an even number.

Then, we install U_k to D_{k-1} as follows.

Install U_k:

Case 1: $u_1 \in V_1$.
For each i, $1 \le i \le r$, we set

$$x(u_i) = 2x(w_p) - x(w_q) + 2y(w_p) - 2y(w_q) + r - 2 + i,$$

and set

$$y(u_i) = -x(w_p) + x(w_q) - y(w_p) + 2y(w_q) - r + 1,$$

as illustrated in Figs. 8(a), (b);

Case 2: $u_1 \in V_2$.
For each i, $1 \le i \le r$, we set

$$x(u_i) = x(w_p) + i - 1,$$

and set

$$y(u_i) = y(w_q) - (x(w_p) - x(w_q) + r - 1) \times 1/2,$$

as illustrated in Figs. 8(c), (d);

Case 3: $u_1 \in V_3$.
we set

$$x(u_i) = (x(w_p)+x(w_q)/2-y(w_p)+y(w_q))\times 2/3+r-2+i,$$

and set

$$y(u_i) = (-x(w_p) + x(w_q) + y(w_p) + 2y(w_q)) \times 1/3,$$

as illustrated in Figs. 8(e), (f);

Case 4: Otherwise.
For each i, $1 \le i \le r$, we set

$$x(u_i) = x(w_p) + i - 1 + \epsilon,$$

and set

$$y(u_i) = y(w_q) - (x(w_p) - x(w_q) + r - 1 + \epsilon) \times 1/2,$$

as illustrated in Figs. 8(g), (h) for the case $\epsilon = 0$ and in Figs. 8(i), (j) for the case $\epsilon = 1$.

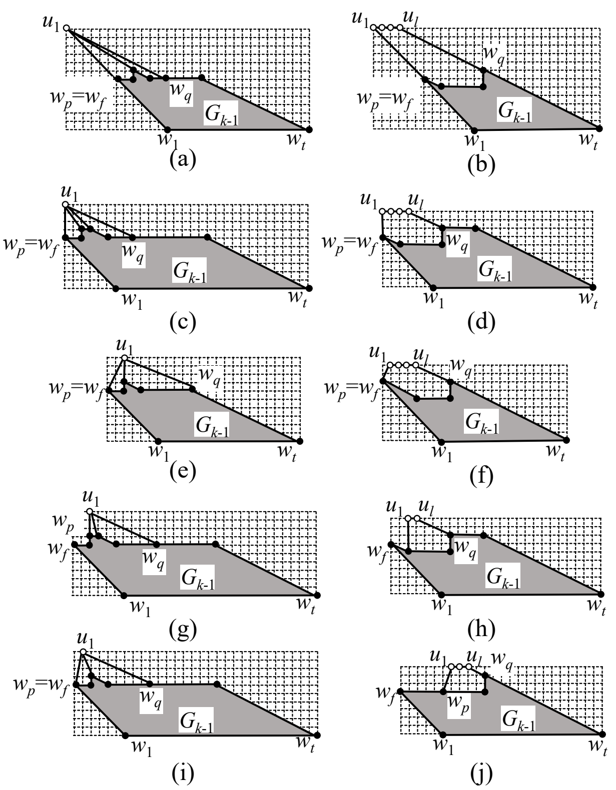

Figure 8: Installing U_k to D_{k-1}.

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 [1] that D_k satisfies conditions (sh4) and (sh5). By the condition (sh1), we have $x(v_2) \le 2n$ in D_m of $G_m = G$ and hence one can easily show that the triangular drawing algorithm yields a convex grid drawing of G on a $W \times H$ grid with $W \le 6n$ and $H \le 3n$. Furthermore, one can easily show that the triangular drawing algorithm takes linear time.

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 extended triangular drawing algorithm yields a convex grid drawing of G on a $W \times H$ grid with $W \le 6n$ and $H \le 3n$ in linear time.*

Actually, the triangular drawing algorithm in [5] is corresponding to a special case of the extended triangular drawing algorithm for the case where $v_l = v_r = v_n$. The following lemma is known.

Lemma 6 *[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 triangular drawing algorithm yields a convex grid drawing of G on a $W \times H$ grid with $W \le 4n$ and $H \le 2n$ in linear time.*

4 Convex Grid Drawing Algorithm

In this section, we present a linear-time algorithm to find a convex grid drawing D of an internally triconnected plane graph G whose decomposition tree $T(G)$ has exactly five leaves. Such a graph G does not have a canonical decomposition, and hence none of the algorithms in [1] and [6] can find a convex grid drawing of G. Our algorithm draws the outer facial cycle $F_{\rm o}(G)$ as a pentagon such that the five sides have slopes -1, ∞, $+1$, -1 and 0, respectively, as illustrated in Fig. 2(b). The algorithm first divides a plane graph G into a left subgraph G_l, a right subgraph G_r and a center subgraph G_c as illustrated in Fig. 1(b), then draw G_l, G_r and G_c by using the extended triangular drawing algorithm in Section 3, respectively, in Fig. 2(a), and finally combine these three drawings to a convex grid drawing of G as illustrated in Fig. 2(b).

4.1 Division

We first explain how to divide G into G_l, G_r and G_c. (See Figs. 1(a) and (b).)

One may assume that the five leaves l_1, l_2, l_3, l_4 and l_5 of $T(G)$ appear clockwise in $T(G)$ in this order, as illustrated in Fig. 3(c). Clearly, there are three cases to consider.

Case a: exactly one node u_5 of $T(G)$ has degree five and each of the other non-leaf nodes has degree two as illustrated in Fig. 9(a).

Case b: exactly one node u_4 has degree four, exactly one node u_3 has degree three and each of the other non-leaf nodes has degree two as illustrated in Fig. 9(b).

Case c: exactly three nodes u_{l3}, u_{c3} and u_{r3} have degree three and each of the other non-leaf nodes has degree two as illustrated in Fig. 9(c).

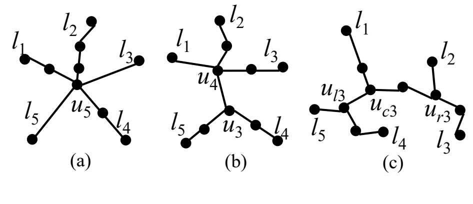

Figure 9: Decomposition trees $T(G)$ (a) having a node of degree five, (b) having a node of degree four and a node of degree three, and (c) having three nodes of degree three.

We only consider Case a, because the other cases are identical.

As the five apices of the pentagonal contour of G, we choose five outer vertices a_i, $1 \le i \le 5$ 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 five vertices a_1, a_2, a_3, a_4 and a_5 appear clockwise on $F_{\rm o}(G)$ in this order as illustrated in Fig. 1(a).

Let $Path(l_i)$, $1 \le i \le 5$, be a path from l_i to u_5 in $T(G)$. We choose arbitrary two consecutive leaves l_i and

l_{i+1}, where indices are computed as modulo 5. We then split the graphs corresponding to $Path(l_i) - u_5$, and corresponding to $Path(l_{i+1}) - u_5$ from G. Let G_l be the graph corresponding to $Path(l_i) - u_5$, let G_r be the graph corresponding to $Path(l_{i+1}) - u_5$ and let G_c be the graph corresponding to $G - G_l - G_r$. In Fig. 1(b), G_l is the graph corresponding to $Path(l_1) - u_5$ and G_r is the graph corresponding to $Path(l_2) - u_5$. Let $\{u_{l_1}, u_{l_2}\}$ be the separation pair for G_l and G_c, let $\{u_{r_1}, u_{r_2}\}$ be the separation pair for G_r and G_c, and let $u_{l_1}, u_{l_2}, u_{r_1}, u_{r_2}$ appear around $F_{\rm o}(G)$ in this order, as illustrated in Fig. 1(a).

If (u_{l_1}, u_{l_2}) (resp. (u_{r_1}, u_{r_2})) is a real edge of G, then one can easily know that the real edge (u_{l_1}, u_{l_2}) (resp. (u_{r_1}, u_{r_2})) is included in G_l (resp. G_r), by the definition of G_l (resp. G_r) and G_c above. Therefore, G_l (resp. G_r) has multiple edges (u_{l_1}, u_{l_2}) (resp. (u_{r_1}, u_{r_2})) (one is real and the other is virtual). In this case, let G_l (resp. G_r) be the graph obtained by deleting the virtual edge (u_{l_1}, u_{l_2}) (resp. (u_{r_1}, u_{r_2})) from the graph defined above, as illustrated in Fig. 1(b).

4.2 Drawing of G_l

By using Lemma 4, one can easily show that G_l has a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to $v_1 = a_i$, $v_2 = u_{l_2}$ and $v_n = u_{l_1}$. Let G'_l be a "mirror" copy of G_l. We first obtain a triangular drawing of G'_l by using the extended triangular drawing algorithm in Section 3 as $v_1 = a_i$, $v_2 = u_{l_2}$, $v_n = u_{l_1}$ and $v_l = v_r = v_n$, respectively. We then modify the drawing of G'_l using the left-right reflection and we obtain a triangular drawing of G_l, as illustrated in Fig. 10(a). We then rotate the drawing by 90° clockwise and obtain a drawing D_l of G_l whose outer polygon is a triangle with apices u_{l_1}, a_i and u_{l_2}, such that the side $a_i u_{l_2}$ has slope ∞, the side $u_{l_2} u_{l_1}$ has slope -2, and the side $u_{l_1} a_i$ has slope -1, respectively, as illustrated in Fig. 10(b). By Lemma 6, one can easily show that $W(D_l) \leq 2n_l$ and $H(D_l) \leq 4n_l$, where n_l be the number of vertices of G_l.

4.3 Drawing of G_r

By using Lemma 4, one can easily show that G_r has a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to $v_1 = a_{i+1}$, $v_2 = u_{r_1}$ and $v_n = u_{r_2}$. We first obtain a triangular drawing of G_r by using the triangular drawing algorithm in Section 3 as $v_1 = a_{i+1}$, $v_2 = u_{r_1}$, $v_n = u_{r_2}$ and $v_l = v_r = v_n$, respectively, as illustrated in Fig. 10(c). We then rotate the drawing by 90° clockwise and obtain a drawing D_r of G_r whose outer polygon is a triangle with apices u_{r_1}, a_{i+1} and u_{r_2}, such that the side $u_{r_1} a_{i+1}$ has slope ∞, the side $a_{i+1} u_{r_2}$ has slope $+1$, and the side $u_{r_2} u_{r_1}$ has slope $+2$, as illustrated in Fig. 10(d). By Lemma 6, one can easily show that $W(D_r) \leq 2n_r$ and $H(D_r) \leq 4n_r$, where n_r be the number of vertices of G_r.

4.4 Drawing of G_c

In this section, we present a linear-time algorithm, called a *heptagonal drawing algorithm* to find a convex grid drawing of G_c with a heptagonal outer polygon. This algorithm finds a convex grid drawing of G_c whose outer polygon is a heptagon with apices $v_1, u_{l_1}, u_{l_2}, u_{r_1}, u_{r_2}, v_n$ and v_2, such that the side $v_1 u_{l_1}$ has slope -1, the side $u_{l_1} u_{l_2}$ has slope -2, the side $u_{l_2} u_{r_1}$ has slope ∞, the side $u_{r_1} u_{r_2}$ has slope $+2$, the side $u_{r_2} v_n$ has slope $+1$, the side $v_n v_2$ has slope -1 and the side $v_2 v_1$ has slope 0 respectively, as illustrated in Fig. 2(a). We modify the extended drawing algorithm in Section 3, as follows.

By using Lemma 4, one can easily show that G_c has a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to $v_1 = a_{i+4}$, $v_2 = a_{i+3}$ and $v_n = a_{i+2}$. Let $F_{\rm o}(G_c) = w_1, w_2, \cdots, w_{a-1}(= u_{l_1}), w_a(= u_{l_2}), w_{a+1}, \cdots, w_{b-1}(= u_{r_1}), w_b(= u_{r_2}), \cdots, w_t$, where $w_1 = v_1$,

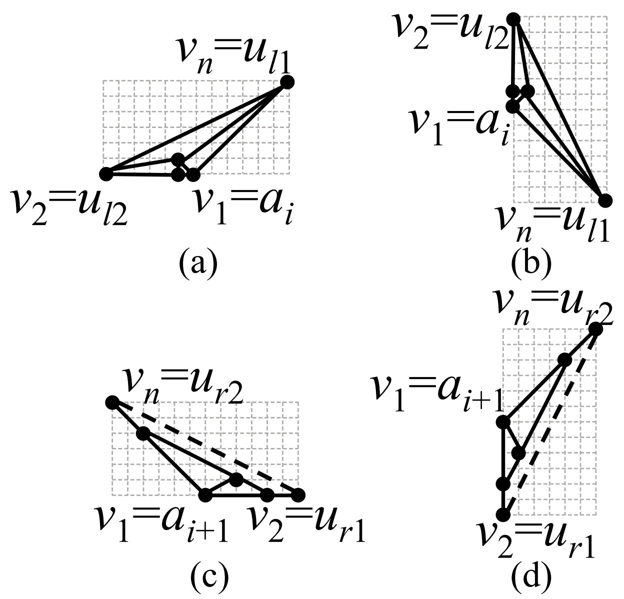

Figure 10: (a) A triangular drawing of G_l, (b) a drawing D_l of G_l, (c) a triangular drawing of G_r, and (d) a drawing D_r of G_r.

$w_t = v_2$. We assume without loss of generality that $w_a \neq w_{b-1}$. The case for $w_a = w_{b-1}$ is identical.

We use the extended triangular drawing algorithm to G_c as $v_1 = a_{i+4}$, $v_2 = a_{i+3}$, $v_n = a_{i+2}$, $v_l = u_{l_1}$ and $v_r = u_{r_1}$. Furthermore, we wish to put vertices w_a and w_b so that the edge $(w_{a-1}(= u_{l_1}), w_a(= u_{l_2}))$ has slope -2 and the edge $(w_{b-1}(= u_{r_1}), w_b(= u_{r_2}))$ has slope $+2$, as illustrated in Fig. 2(a). Actually, the method deciding the coordinates of each vertex other than the vertices $w_a = u_{l_2}$ and $w_b = u_{r_2}$ on $F_{\rm o}(G_c)$ is identical to the extended triangular drawing algorithm. Thus we will explain how to decide the coordinates of vertices $w_a = u_{l_2}$ and $w_b = u_{r_2}$ on $F_{\rm o}(G_c)$.

We first explain how to decide the coordinates of $w_a = u_{l_2}$. Let $u_{l_2} \in U_k$, then u_{l_2} should be the first vertex of U_k. Let $U_k = \{u_{l_2} = u_1, u_2, \cdots, u_{|U_k|}\}$, Let w_p be the leftmost neighbor of $u_{l_2} = u_1$, and let w_q be the rightmost neighbor of $u_{|U_k|}$. Then $w_p = u_{l_1}$, of course. Let $F_{\rm o}(G_{k-1}) = w_1, w_2, \cdots, w_p(= u_{l_1}), w_{p+1}, \cdots, w_q, w_{q+1}, \cdots, w_t$, where $w_1 = v_1$, $w_t = v_2$.

We first shift 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 $|U_k|$, as similarly in the extended triangular drawing algorithm

Let $D_{pq} = x(w_q) - x(w_p) + 2(y(w_q) - y(w_p))$. We wish to put vertices $U_k = \{u_{l_2} = u_1, u_2, \cdots, u_{|U_k|}\}$ so that the edge $(w_p, w_a) = (u_{l_1}, u_{l_2})$ has slope -2 and the edge $(u_{|U_k|}, w_q)$ has slope $-1/2$. Furthermore, if $|U_k| \geq 2$, then we wish to put vertices $u_2, u_3, \cdots, u_{|U_k|-1}$ so that, for each i, $1 \leq i \leq |U_k| - 1$, the edge (u_i, u_{i+1}) has slope 0 and the distance between two vertices u_i and u_{i+1} is equal to 1. Then $D_{pq} - (|U_k| - 1)$ should be a multiple of 3, that is, $D_{pq} - (|U_k| - 1)\ mod\ 3 = 0$ and hence we will do some additional shift operations if $D_{pq} - (|U_k| - 1)\ mod\ 3 \neq 0$. That is, we 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 $3 - (\{D_{pq} - (|U_k| - 1\}mod3)$.

Then, $D_{pq} - (|U_k| - 1)$ becomes a multiple of 3 and hence the straight line with slope -1 through $(x(w_p), y(w_p))$ and the straight line with slope $-1/2$ through $(x(w_q) - (|U_k| - 1), y(w_q))$ intersects at a grid point, which is denoted by P.

In Section 4.5, we will combine the drawing D_l of G_l and the drawing D_c of G_c so that the edge (u_{l_1}, u_{l_2})

of G_l overlaps the same one of G_c. If the length of the edge (u_{l_1}, u_{l_2}) of D_l is not equal to the length of the edge (u_{l_1}, u_{l_2}) of D_c, then we widen the narrow one by the shift operation so that both have the same length. There are the two cases to consider.

Case (i): $H(D_l) > y(P) - y(w_p)$. (See Fig. 11(a).)
In this case, we 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 $(H(D_l)-(y(P)-y(w_p))\times 3/2$. In Fig. 11(a), $(H(D_l)-(y(P)-y(w_p))$ is equal to 4 and hence we shift by distance 6, as illustrated in Fig. 11(b). Since $(H(D_l)-(y(P)-y(w_p))$ is multiple of 2, $(H(D_l)-(y(P)-y(w_p))\times 3/2$ is multiple of 3 and hence $D_{pq}-3(|U_k|-1)$ is still multiple of 3, as illustrated in Fig. 11(b).

Case (ii): $H(D_l) \leq y(P) - y(w_p)$. (See Fig. 11(c).)
In this case, we extend the drawing D_l so that the length of the edge (u_{l_1}, u_{l_2}) of D_l is equal to the length between $P(w_p)$ and P, as illustrated in Fig. 11(d).

We then put the vertices in U_k so that the edge $(w_p(= u_{l_1}), w_a(= u_{l_2}))$ has slope -2 and the edge $(u_{|U_k|}, w_q)$ has slope $-1/2$. Furthermore, if $|U_k| \geq 2$, then, for each i, $1 \leq i \leq |U_k|-1$, we put vertices $u_2, u_3, \cdots, u_{|U_k|-1}$ so that the edge (u_i, u_{i+1}) has slope 0 and the distance between two vertices u_i and u_{i+1} is equal to 2, as illustrated in Figs. 11(b) and (d).

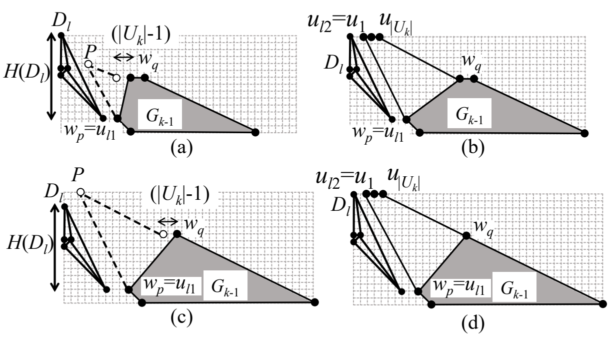

Figure 11: Illustrations for (a),(b) Case (i), and (c),(d) Case (ii).

We then explain how to decide the coordinates of $w_b = u_{r_2}$. Let $u_{r_2} \in U_{k'}$, then u_{r_2} should be the first vertex of $U_{k'}$. Let $U_{k'} = \{u_{r_2} = u'_1, u'_2, \cdots, u'_{|U_{k'}|}\}$, let w_p be the leftmost neighbor of $u_{r_2} = u'_1$, ($w_p = u_{r_1}$, of course,) and let w_q be the rightmost neighbor of $u'_{|U_{k'}|}$. Then we wish to put vertices $U_{k'} = \{u_{r_2} = u'_1, u'_2, \cdots, u'_{|U_{k'}|}\}$ so that the edge $(w_p, w_b) = (u_{r_1}, u_{r_2})$ has slope $+2$ and the edge $(u'_{|U_{k'}|}, w_q)$ has slope $-1/2$. Thus one can decide the coordinates of vertices in $U_{k'}$ similarly as above, in a sense that $D_{pq} - (|U_{k'}|-1)$ should be a multiple of 5.

We finally prove the correctness of the heptagonal drawing algorithm. One can prove similarly as in [8] that the drawing D_c is a convex grid drawing of G_c.

We then consider the width $W(D_c)$ and the height $H(D_c)$ of the drawing D_c of G_c. Let n_l be the number of vertices of G_l, let n_r be the number of vertices of G_r and let n_c be the number of vertices of G_c. Then $n_l+n_r+n_c = n+4$, of course. Since every vertex of a plane graph G has degree three or more, each components corresponding to l_i of $T(G)$, for each i, $1 \leq i \leq 5$ has four or more vertices and hence we have $n_l, n_r \geq 4$ and $n_c \geq 10$. One can easily observe that we may shift by distance $2+2+H(D_l)\times 3/2$ for the vertex $w_a = u_{l_2}$ and by distance $2+1+4+H(D_r)\times 5/2$ for the vertex $w_b = u_{r_2}$, respectively, and hence we have $W(D_c) \leq 6n_c + 6 + H(D_l)\times 3/2 + H(D_r)\times 5/2$. By Lemma 6 and the algorithms in Secs 4.2 and 4.3, we have $H(D_l) \leq 4n_l$ and $H(D_r) \leq 4n_r$ and hence $W(D_c) \leq 6n_c+6+6n_l+10n_r$. Since $n_l+n_r+n_c = n+4$ and $n_c \geq 10$, we have $W(D_c) \leq 10(n_l+n_r+n_c)+6-4n_l-4n_c \leq 10n$. Furthermore, one can prove similarly above, $H(D_c) \leq 5n$.

We thus have the following lemma.

Lemma 7 *For a plane graph G_c having a canonical decomposition $\Pi = (U_1, U_2, \cdots, U_m)$ with respect to v_1, v_2 and v_n, the heptagonal drawing algorithm yields a convex grid drawing of G_c on a $W \times H$ grid with $W \leq 10n$ and $H \leq 5n$ in linear time.*

4.5 Drawing of G

We first arrange D_c so that $x(a_{i+4}) = 0$ and $y(a_{i+4}) = 0$. We then arrange D_l so that the edge (u_{l_1}, u_{l_2}) of D_l overlap the same one of D_c. We also arrange D_r so that the edge (u_{r_1}, u_{r_2}) of D_r overlap the same one of D_c. We finally remove the edges (u_{l_1}, u_{l_2}) and (u_{r_1}, u_{r_2}) if they are not original edges of G, as illustrated in Fig. 2(b).

4.6 Validity of Drawing Algorithm

In this section, we show that the drawing D obtained above is a convex grid drawing of G. By Lemma 6, three drawings D_l,D_r and D_c are convex grid drawings. Therefore, one can easily show that D is a convex grid drawing of G with pentagonal contour. Clearly, the size of the grid of the drawing D of G is equal to the size of D_c of G_c and hence, by Lemma 7, we have $W(D) \leq 10n$ and $H(D) \leq 5n$.

We thus have the following 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 five leaves. Then our algorithm finds a convex grid drawing of G with a pentagonal outer polygon on a $10n \times 5n$ 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 five leaves has a convex grid drawing on a $10n\times 5n$ grid, and we present a linear-time algorithm to find such a drawing. The area bound $O(n^2)$ is optimal up to a constant factor since a plane graph of nested triangles needs an $\Omega(n^2)$ area. The remaining problem is to obtain an algorithm for an internally triconnected plane graph whose decomposition tree has six or more leaves.

Acknowledgement

This research has been supported by the Kayamori Foundation of Informational Science Advancement.

References

[1] 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.

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

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

[4] T. Hashimoto, K. Miura and T. Nishizeki, *Convex grid drawings of internally Triconnected plane graphs*, IEICE TRANSACTIONS on Information and Systems (In Japanese), J95-D, 3, pp. 356–365, 2012.

[5] K. Miura, *Convex Grid Drawings of Plane Graphs with Pentagonal Contours*, IEICE Trans. on Information and Systems, E97-D, 3, pp. 413–420, 2014.

[6] 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.

[7] 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.

[8] K. Miura, A. Kamada and T. Nishizeki, *Convex Grid Drawings of Plane Graphs with Rectangular Contours*, Journal of Graph Algorithms and Applications, 12, 2, pp. 197–224, 2008.

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

[10] K. Sato and K. Miura *Convex Grid Drawings of Plane Graphs with Pentagonal Contours on $O(n^2)$ grids*, IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences, Special Section on Discrete Mathematics and Its Applications, E104-A, 9, pp.1142–1149, 2021.

[11] R. Tamassia, Handbook of Graph Drawing and Visualization., *Chapman & Hall/CRC* Boca Raton, 2013.

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

[13] X. Zhou and T. Nishizeki, *Convex drawings of internally triconnected plane graphs on $O(n^2)$ grids*, Discrete Mathematics, Algorithms and Applications, 2, pp. 347–362, 2009.