Grid Drawings of Four-Connected Plane Graphs

Kazuyuki Miura[1], Shin-ichi Nakano[2], and Takao Nishizeki[1]

[1] Graduate School of Information Sciences
Tohoku University, Aoba-yama 05, Sendai 980-8579, Japan
miura@nishizeki.ecei.tohoku.ac.jp
nishi@ecei.tohoku.ac.jp

[2] Department of Computer Science, Faculty of Engineering, Gunma University;
Mailing address: 1-5-1 Tenjin-cho, Kiryu, Gunma, 376-8515 Japan
nakano@cs.gunma-u.ac.jp

Abstract. A grid drawing of a plane graph G is a drawing of G on the plane so that all vertices of G are put on plane grid points and all edges are drawn as straight line segments between their endpoints without any edge-intersection. In this paper we give a very simple algorithm to find a grid drawing of any given 4-connected plane graph G with four or more vertices on the outer face. The algorithm takes time $O(n)$ and needs a rectangular grid of width $\lceil n/2 \rceil - 1$ and height $\lceil n/2 \rceil$ if G has n vertices. The algorithm is best possible in the sense that there are an infinite number of 4-connected plane graphs any grid drawings of which need rectangular grids of width $\lceil n/2 \rceil - 1$ and height $\lceil n/2 \rceil$.

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 [1,2,3,4,5,6,7,8,9,10,11,13,15]. The most typical method is *the straight line drawing* in which all edges of a graph are drawn as straight line segments without any edge-intersection. Every plane graph has a straight line drawing [8,14,16]. However, not every straight line drawing is an aesthetic drawing since many vertices may be concentrated in a small area.

A straight line drawing of a plane graph G is called a *grid drawing* of G if the vertices of G are put on grid points of integer coordinates. Of course, the distance between any two vertices in the drawing is at least 1. *The integer grid* of size $W \times H$ consists of $W+1$ vertical segments and $H+1$ horizontal segments, and has a rectangular contour. W and H are called the *width* and *height* of the integer grid, respectively. It is known that every plane graph of $n \geq 3$ vertices has a grid drawing on an $(n-2) \times (n-2)$ grid, and that such a grid drawing can be found in linear time [3,6,9,13]. It is also shown that, for each $n \geq 3$, there exists a plane graph which needs a grid of size at least $\lfloor 2(n-1)/3 \rfloor \times \lfloor 2(n-1)/3 \rfloor$ for any grid drawing [4,9]. It has been conjectured that every plane graph has a grid drawing on a $\lceil 2n/3 \rceil \times \lceil 2n/3 \rceil$ grid, but it is still an open problem. On the other hand, a restricted class of graphs has a more compact grid drawing. For

J. Kratochvíl (Ed.): GD'99, LNCS 1731, pp. 145–154, 1999.

example, if G is a 4-connected plane graph and has at least four vertices on its outer face, then G has a grid drawing on a $W \times H$ grid such that $W + H \leq n$, $W \leq (n+3)/2$ and $H \leq 2(n-1)/3$, and one can find such a grid drawing in linear time [10]. However, the algorithm is rather complicated.

In this paper, we give a very simple algorithm which finds a grid drawing of any given 4-connected plane graph G on a $W \times H$ grid such that $W = \lceil n/2 \rceil - 1$ and $H = \lceil n/2 \rceil$ in linear time if G has four or more vertices on the outer face. Since $W = \lceil n/2 \rceil - 1$ and $H = \lceil n/2 \rceil$, $W + H \leq n$. Thus our bounds on W and H are better than He's bounds [10]. Our bounds are indeed best possible, because there exist an infinite number of 4-connected plane graphs, for example the nested quadrangles depicted in Fig. 1, which need grids of size at least $W = \lceil n/2 \rceil - 1$ and $H = \lceil n/2 \rceil$ for any grid drawing. An aspect ratio of a drawing obtained by the algorithm [10] may be $1 : 4/3$, while the ratio of our algorithm is always $1 : 1$. Both our algorithm and the proof of its correctness are very simple, and it is quite easy to understand them.

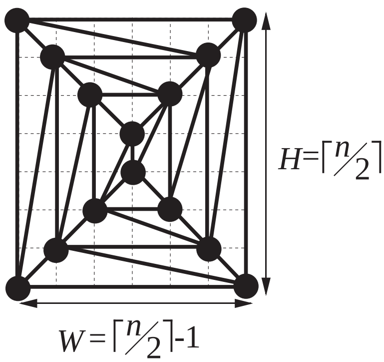


Fig. 1. Nested quadrangles attaining our bounds.

The outline of our algorithm is as follows. One can assume without loss of generality that a given graph G is internally triangulated as illustrated in Fig. 2(a). First, we find a "4-canonical ordering" of G [12]. Using the ordering, we then divide G into two graphs $G^{'}$ and $G^{''}$, each of which has about $n/2$ vertices as illustrated in Fig. 2(b) where $G^{'}$ and $G^{''}$ are shaded. Next, we draw the plane subgraph $G^{'}$ in an isosceles right-angled triangle whose base has length $W^{'} = n/2 - 1$ and whose height is $H^{'} = W^{'}/2$, as illustrated in Fig. 2(c). Similarly, we draw $G^{''}$ in the same triangle with its upside down. In Fig. 2(c) the two triangles are drawn by thick dotted lines. We place the two triangles so that their vertices opposite to their bases are separated by distance 1. Finally, we combine the drawings of $G^{'}$ and $G^{''}$ to obtain a grid drawing of G, as illustrated in Fig. 2(d). The drawing of G has sizes $W = W^{'} = n/2 - 1$ and $H = 2H^{'} + 1 = W^{'} + 1 = n/2$.

The remainder of the paper is organized as follows. In Section 2, we give some definitions and lemmas, and present our algorithm and a main theorem. In Section 3, we show how to draw $G^{'}$ and $G^{''}$.

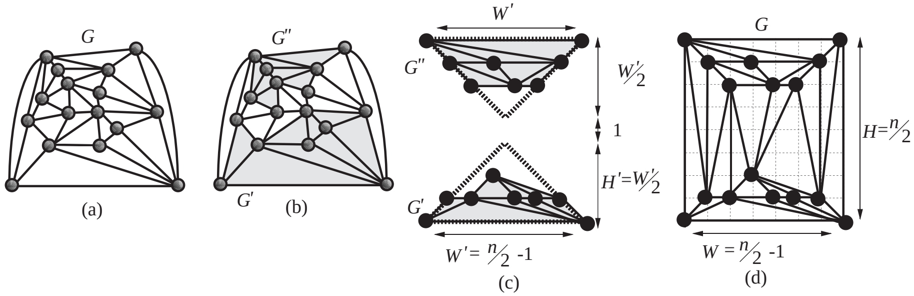


Fig. 2. Drawing process of our algorithm.

2 Main Theorem

In this section we first introduce some definitions and lemmas, and then present our algorithm and a main theorem.

Let $G = (V, E)$ be a simple connected graph having no multiple edge or loop. V is the vertex set and E is the edge set of G. Let n be the number of vertices of 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, and is denoted by $d(v, G)$. The *connectivity* $\kappa(G)$ of a graph G is the minimum number of vertices whose removal results in a disconnected graph or a single-vertex graph K_1. A graph G is *k-connected* if $\kappa(G) \geq k$. Let $x(v)$ and $y(v)$ be the x- and y-coordinates of vertex $v \in V$, respectively.

A graph is *planar* if it can be embedded in the plane so that no two edges intersect geometrically except at a vertex to which they are both incident. A *plane graph* is a planar graph with a fixed embedding. A plane graph divides the plane into connected regions called *faces*. We denote the boundary of a face by a clockwise sequence formed by the vertices and edges on the boundary. We call the boundary of the outer face of a plane graph G *the contour* of G, and denote it by $C_o(G)$. A plane graph G is *internally triangulated* if all inner faces of G are triangles. We can assume without loss of generality that a given graph G is internally triangulated. Otherwise, we internally triangulate G by adding some new edges to G, find a drawing of the resulting graph, and finally remove the added edges to obtain a drawing of G.

We then give a definition of a 4-canonical ordering of a plane graph G [12], on which both our algorithm and He's [10] are based. The 4-canonical ordering is a generalization of the "canonical ordering" [9] which is used to find a grid drawing of triangulated plane graph. Let $\Pi = (v_1, v_2, \cdots, v_n)$ be an ordering of set V. Fig. 3(a) illustrates an ordering of the graph G in Fig. 2(a). Let G_k, $1 \le k \le n$, be the plane subgraph of G induced by the vertices in $\{v_1, v_2, \cdots, v_k\}$, and let $\overline{G_k}$ be the plane subgraph of G induced by the vertices in $\{v_{k+1}, v_{k+2}, \cdots, v_n\}$. Thus $G = G_n = \overline{G_0}$. In Fig. 3(b), G_k is darkly shaded, while $\overline{G_{k-1}}$ is lightly shaded. We say that Π is a *4-canonical ordering* of G if the following two conditions are satisfied:

(co1) v_1 and v_2 are the ends of an edge on $C_o(G)$, and v_n and v_{n-1} are the ends of an edge on $C_o(G)$; and
(co2) for each k, $3 \le k \le n-2$, v_k is on $C_o(G_k)$, $d(v_k, G_k) \ge 2$, and $d(v_k, \overline{G_{k-1}}) \ge 2$.

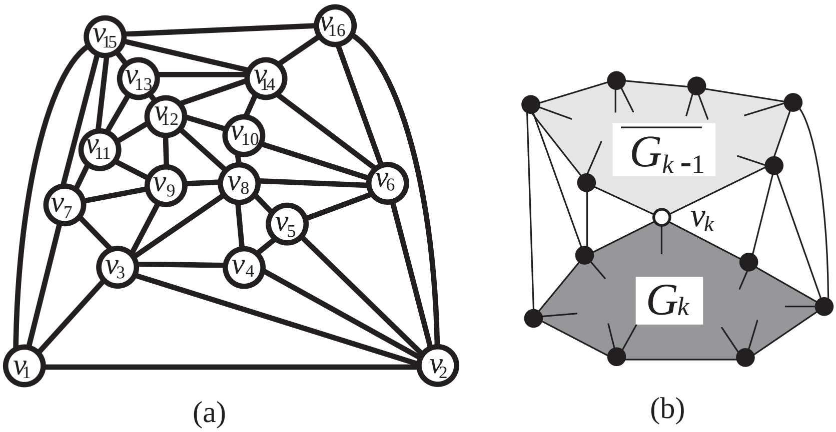


Fig. 3. (a) A 4-canonical ordering of a 4-connected plane graph of $n = 16$ vertices, and (b) an illustration for the condition (co2).

Although the definition of a 4-canonical ordering above is slightly different from that in [12], they are effectively equivalent each other. The following lemma is known.

Lemma 1. [12] Let G be a 4-connected plane graph having at least four vertices on $C_o(G)$. Then G has a 4-canonical ordering Π, and Π can be found in linear time.

We are now ready to present our algorithm *Draw*.

Procedure Draw(G)
begin
Find a 4-canonical ordering $\Pi = (v_1, v_2, \cdots, v_n)$ of a given 4-connected plane graph $G = (V, E)$;

2 Divide G into two subgraphs $G^{'}$ and $G^{''}$ where $n' = \lceil n/2 \rceil$, $G^{'} = G_{n'}$, and $G^{''} = \overline{G_{n'}}$;
3 Draw $G^{'}$ in an isosceles right-angled triangle whose base has length $W^{'} = \lceil n/2 \rceil - 1$ and whose height is $H^{'} = W^{'}/2$;
4 Draw $G^{''}$ in the same triangle with its upside down;
5 Place the two triangles so that their vertices opposite to their bases are separated by distance 1 and have the same x-coordinate;
6 Draw every edge of G joining a vertex in $G^{'}$ and a vertex in $G^{''}$ by a straight line segment;

end.

We say that a curve in the plane is *x-monotone* if the intersection of the curve and any vertical line is a single point when it is nonempty. We then have the following lemma for the drawing of $G^{'}$, the proof of which will be given later in Section 3.

Lemma 2. One can find in linear time a grid drawing of $G^{'}$ satisfying the following conditions (a), (b) and (c):
(a) the drawing is in an isosceles right-angled triangle whose base has length $W^{'} = \lceil n/2 \rceil - 1$ and whose height is $H^{'} = W^{'}/2$, and edge (v_1, v_2) is drawn as the base of the triangle;
(b) the absolute value of the slope of every edge on $C_o(G^{'})$ is at most 1; and
(c) the drawing of the path going clockwise on $C_o(G^{'})$ from v_1 to v_2 is x-monotone.

If $\Pi = (v_1, v_2, \cdots, v_n)$ is a 4-canonical ordering, then the reversed ordering $\Pi^{'} = (v_n, v_{n-1}, \cdots, v_1)$ is also a 4-canonical ordering. Therefore $G^{''}$ has a grid drawing in the same triangle. Hence we have the following main theorem.

Theorem 1. Algorithm *Draw* finds in linear time a grid drawing of a given 4-connected plane graph G on a $W \times H$ grid such that $W = \lceil n/2 \rceil - 1$ and $H = W + 1 = \lceil n/2 \rceil$ if there are four or more vertices on the contour $C_o(G)$.

Proof. If step 6 in *Draw* does not introduce any edge-intersection, then algorithm *Draw* correctly finds a grid drawing of G and clearly the size of a drawing of G satisfies $W = W^{'}$ and $H = H^{'} + H^{'} + 1$. (See Fig. 2.) By Lemma 2(a) $W^{'} = \lceil n/2 \rceil - 1$ and $H^{'} = W^{'}/2$. Therefore $W = \lceil n/2 \rceil - 1$ and $H = W + 1 = \lceil n/2 \rceil$. Thus we shall show that step 6 does not introduce any edge-intersection.

An oblique side of each isosceles right-angled triangle has slope $+1$ and the other oblique side has slope -1. The two vertices of the triangles opposite to the bases are separated by distance 1 and have the same x-coordinate. Therefore the absolute value of the slope of any straight line connecting a point in the triangle of $G^{'}$ and a point in the triangle of $G^{''}$ is greater than the slope $H/W = 1 + 1/W (> 1)$ of a diagonal of the $W \times H$ rectangle. Thus, the absolute value of the slope of any edge connecting a vertex on $C_o(G^{'})$ and a vertex on $C_o(G^{''})$ is greater than 1. On the other hand, by Lemma 2(b) the absolute value of the slope of every edge on $C_o(G^{'})$ or $C_o(G^{''})$ is less than or equal to 1. Furthermore, by Lemma 2(c) both the drawing of the path from v_1 to v_2 on $C_o(G^{'})$ and the drawing of the path from v_{n-1} to v_n on $C_o(G^{''})$ are x-monotone. Therefore, the straight line drawing of any edge of G connecting a vertex on $C_o(G^{'})$ and a vertex on $C_o(G^{''})$ does not intersect the drawings of $G^{'}$ and $G^{''}$. Furthermore the drawings of all these edges do not intersect each other since G is a plane graph and the drawings of the two paths above are x-monotone. Thus step 6 does not introduce any edge-intersection.

By Lemma 1 one can execute steps 1 and 2 of procedure *Draw* in linear time. By Lemma 2 one can execute steps 3 and 4 in linear time. Clearly one can execute steps 5 and 6 in linear time. Thus *Draw* runs in linear time. □

3 Drawing G'

In this section, we show how to find a drawing of $G^{'}$ satisfying the conditions (a), (b) and (c) in Lemma 2. It suffices to decide only the coordinates of all vertices of $G^{'}$, because one can immediately find a straight line drawing from the coordinates.

We first define some terms. Let $\Pi = (v_1, v_2, \cdots, v_n)$ be a 4-canonical ordering of G. For any two vertices $v_i, v_j \in V$, we write $v_i \prec v_j$ iff $1 \le i < j \le n$, and write $v_i \preceq v_j$ iff $1 \le i \le j \le n$. We will show later that the following lemma holds.

Lemma 3. If (u, v) is an edge in $G^{'}$ and $u \preceq v$, then the y-coordinates of vertices u and v decided by our algorithm satisfy $y(u) \le y(v)$.

We say that a vertex u in a graph G is *a smaller neighbor* of v if u is a neighbor of v and u is smaller than v, that is $u \prec v$. Similarly, we say that u is *a larger neighbor* of v if u is a neighbor of v and $u \succ v$. The smallest one among the neighbors of vertex v is called *the smallest neighbor* of v, and is denoted by $w_s(v)$. We often denote $w_s(v)$ simply by w_s. Let $3 \le k \le n$, and let $C_o(G_{k-1}) = w_1, w_2, \cdots, w_m$, where $w_1 = v_1$ and $w_m = v_2$. Since G is internally triangulated, all the smaller neighbors of v_k consecutively appear on $C_o(G_{k-1})$. Thus one may assume that they are $w_l, w_{l+1}, \cdots, w_r$ for some l and r, $1 \le l < r \le m$.

We now have the following lemma.

Lemma 4. Let $\Pi = (v_1, v_2, \cdots, v_n)$ be a 4-canonical ordering of G, and let $w_l, w_{l+1}, \cdots, w_r$ be the smaller neighbors of v_k, $3 \le k \le n$. Then the following (a) and (b) hold:

(a) there is no index t such that $l < t < r$ and $w_{t-1} \prec w_t \succ w_{t+1}$; and

(b) $w_l \succeq w_{l+1} \succeq \cdots \succeq w_s \preceq \cdots \preceq w_r$, and $y(w_l) \ge y(w_{l+1}) \ge \cdots \ge y(w_s) \le \cdots \le y(w_r)$ where $w_s = w_s(v_k)$.

Proof. (a) Assume for a contradiction that there is an index t such that $l < t < r$ and $w_{t-1} \prec w_t \succ w_{t+1}$. Let $w_t = v_i$, $1 \le i \le k-1$. Since v_k is adjacent to w_{t-1}, w_t and w_{t+1} in G_k, $w_t = v_i$ is neither on $C_o(G_k)$ nor on $C_o(G)$ and hence

$3 \le i \le n-2$. Therefore by the condition (co2) of the 4-canonical ordering, w_t has at least two larger neighbors. Let v_j be the largest one among the w_t's neighbors except v_k. Then $w_t = v_i \prec v_j \ne v_k$. Clearly vertex v_j is either in the triangular face v_k, w_t, w_{t-1} of graph G_k or in the triangular face v_k, w_{t+1}, w_t. Since $w_t \prec v_j$ and $w_{t-1} \prec w_t \succ w_{t+1}$, we have $v_j \ne w_{t-1}, w_{t+1}$. Therefore v_j must be in the proper inside of one of the two faces above. Since v_j is not on $C_o(G)$, we have $3 \le j \le n-2$. Since v_j is not in G_{k-1}, we have $v_{k-1} \prec v_j$ and hence $v_k \prec v_j$. Therefore v_k is contained in G_j, and hence v_j is not on $C_o(G_j)$, contrary to the condition (co2) of the 4-canonical ordering.

(b) Since $w_s \preceq w_r$, by (a) we have $w_s \preceq w_{s+1} \preceq \cdots \preceq w_r$. Therefore by Lemma 3 we have $y(w_s) \le y(w_{s+1}) \le \cdots \le y(w_r)$. Similarly we have $y(w_l) \ge y(w_{l+1}) \ge \cdots \ge y(w_s)$.

□

We are now ready to show how to find a drawing of G'. First, we put vertices v_1, v_2, v_3 on grid points $(0,0), (2,0)$ and $(1,1)$ so that G_3 is drawn as an isosceles right-angled triangle. Clearly the conditions (b) and (c) in Lemma 2 hold for G_3. Next, for each k, $4 \le k \le \lceil n/2 \rceil$, we decide the x-coordinate $x(v_k)$ and the y-coordinate $y(v_k)$ of v_k so that the conditions (b) and (c) in Lemma 2 hold for G_k. One may assume that the conditions hold for G_{k-1}. Let $C_o(G_{k-1}) = w_1, w_2, \cdots, w_m$, and let $w_l, w_{l+1}, \cdots, w_r$ be the smaller neighbors of v_k. Since the condition (c) of Lemma 2 holds for G_{k-1}, the drawing of the path $w_l, w_{l+1}, \cdots, w_r$ is x-monotone. Furthermore, by Lemma 4(b), we have $y(w_l) \ge y(w_{l+1}) \ge \cdots \ge y(w_s) \le \cdots \le y(w_r)$, as illustrated in Fig. 4.

We always shift a drawing of G_{k-1} to the x-direction before adding vertex v_k, as illustrated in Fig. 4. We have to determine which vertices of G_{k-1} must be shifted to the x-direction. Thus we will maintain a set $U(v_k)$ for each vertex v_k, $1 \le k \le \lceil n/2 \rceil$. This set will contain vertices located "under" v_k that need to be shifted whenever v_k is shifted. Initially, we set $U(v_k) = \{v_k\}$ for $k = 1, 2, 3$. For k, $4 \le k \le \lceil n/2 \rceil$, we set $U(v_k) = \{v_k\} \cup (\cup_{i=l+1}^{r-1} U(w_i))$. Thus all vertices in $U(v_k)$ except v_k are not on $C_o(G_k)$. The *shift operation* on a vertex w_j, denoted by $shift(w_j)$, is achieved by increasing the x-coordinate of each vertex $u \in \bigcup_{i=j}^{m} U(w_i)$ by 1 [3,4,6,9,10,11].

We then show how to decide $y(v_k)$ and $x(v_k)$. Let y_{max} be the maximum value of y-coordinates of $w_l, w_{l+1}, \cdots, w_r$, then either $y_{max} = y(w_l)$ or $y_{max} = y(w_r)$. There are the following six cases:

(i) $y(w_l) < y(w_r) = y_{max}$;
(ii) $y_{max} = y(w_l) > y(w_r)$;
(iii) $y(w_l) = y(w_r) = y_{max}$, $l < s < r$ and $y(w_{l+1}) \ne y_{max}$;
(iv) $y(w_l) = y(w_r) = y_{max}$, $l < s < r$ and $y(w_{l+1}) = y_{max}$;
(v) $y(w_l) = y(w_r) = y_{max}$ and $s = l$; and
(vi) $y(w_l) = y(w_r) = y_{max}$ and $s = r$.

We first consider the three cases (i), (iii) and (v). In these cases $y_{max} = y(w_r)$. We decide $y(v_k)$ and $x(v_k)$ as follows. We first execute $shift(w_{s+1})$, that is, we increase the x-coordinates of all vertices $w_{s+1}, w_{s+2}, \cdots, w_m$ and all vertices under them by 1, as illustrated in Figs. 4(a), (b) and (c). We then decide

$$y(v_k) = \begin{cases} y_{max} & \text{if } y(w_{r-1}) < y_{max}; \\ y_{max}+1 & \text{if } y(w_{r-1}) = y_{max}, \end{cases}$$

and

$$x(v_k) = x(w_s) + y(v_k) - y(w_s).$$

We denote the slope of a straight line segment uv by $slope(uv)$. Then clearly we have

$$slope(w_s v_k) = \frac{y(v_k) - y(w_s)}{x(v_k) - x(w_s)} = 1.$$

Since $y(v_k) \geq y(w_l) \geq y(w_s)$ and $x(w_l) \leq x(w_s) < x(v_k)$, we have

$$0 \leq slope(w_l v_k) = \frac{y(v_k) - y(w_l)}{x(v_k) - x(w_l)} \leq slope(w_s v_k) = 1$$

as illustrated in Figs. 4(a), (b) and (c).

If $y(w_{r-1}) < y_{max}$, then $y(v_k) = y_{max} = y(w_r)$ and hence $slope(v_k w_r) = 0$ as illustrated in Fig. 4(a). On the other hand, if $y(w_{r-1}) = y_{max}$, then $y(v_k) = y(w_r) + 1$, $x(v_k) \leq x(w_r) - 1$ and hence we have

$$-1 \leq slope(v_k w_r) = \frac{y(w_r) - y(v_k)}{x(w_r) - x(v_k)} < 0$$

as illustrated in Figs. 4(b) and (c).

The absolute slope of each straight line segment on $C_o(G_k)$ except $w_l v_k$ and $v_k w_r$ is equal to its absolute slope on $C_o(G_{k-1})$, and hence is at most 1.

Thus the condition (b) in Lemma 2 holds for G_k.

One can easily observe that the condition (c) in Lemma 2 holds for G_k.

We next consider the remaining three cases (ii), (iv) and (vi). In these cases we decide $y(v_k)$ and $x(v_k)$ in a mirror image way of the cases (i), (iii) and (v) above. That is, we execute $shift(w_s)$, and decide

$$y(v_k) = \begin{cases} y_{max} & \text{if } y(w_{l+1}) < y_{max}; \\ y_{max}+1 & \text{if } y(w_{l+1}) = y_{max}, \end{cases}$$

and

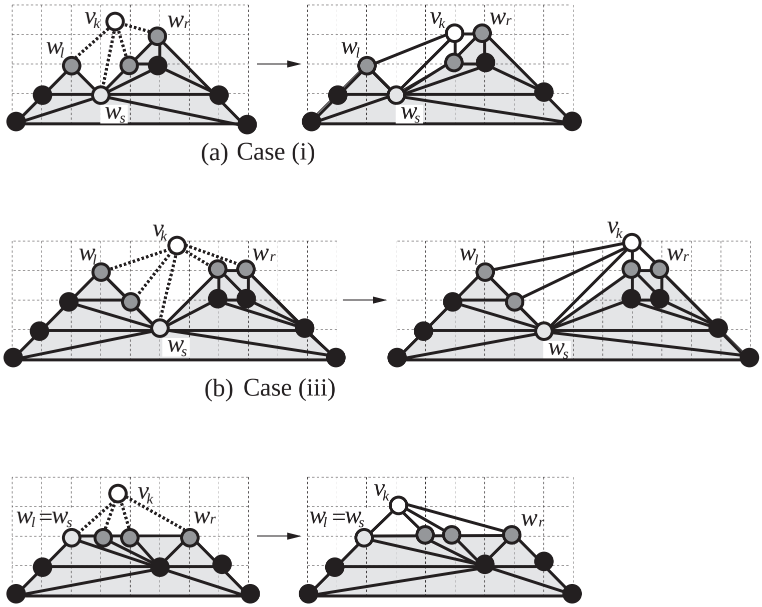
$$x(v_k) = x(w_s) - (y(v_k) - y(w_s)).$$

Then, similarly as in Case 1 above, the conditions (b) and (c) hold for G_k.

Since we decide the y-coordinate as above, Lemma 3 clearly holds.

We are now ready to prove Lemma 2.

Proof of Lemma 2 As shown above, the conditions (b) and (c) hold. Therefore the absolute value of the slope of every edge on $C_o(G')$ is at most 1, and the drawing of the path going clockwise on $C_o(G)$ from v_1 to v_2 is x-monotone.



(a) Case (i)

(b) Case (iii)

(c) Case (v)

Fig. 4. How to put v_k.

The drawing of G_3 has width 2. We execute the shift operation once when we add a vertex v_k, $4 \le k \le n' = \lceil n/2 \rceil$, to the drawing of G_{k-1}. Therefore the width W' of the drawing of G' is $W' = 2 + (n' - 3) = \lceil n/2 \rceil - 1$. Since the conditions (b) and (c) hold, the height is at most $W'/2$. Therefore G' is drawn in an isosceles right-angled triangle whose base has length $W' = \lceil n/2 \rceil - 1$ and whose height is $H' = W'/2$. Obviously (v_1, v_2) is drawn as the base of the triangle. Thus the condition (a) holds.

We then show that the drawing of G' obtained by our algorithm is a grid drawing. Our algorithm puts each v_k, $4 \le k \le \lceil n/2 \rceil$, on a grid point. Clearly each edge (v_k, w_j), $l \le j \le r$, does not intersect any edge of G_{k-1}. Furthermore, similarly to the proof of Lemma 2 in [6], one can easily prove by induction on k that any number of executions of the shift operation for G_{k-1} introduce no edge-intersection in G_{k-1}. Thus our algorithm obtains a grid drawing of G'.

All operations in our algorithm except the shift operation can be executed total in time $O(n)$. A simple implement of the shift operation takes time $O(n)$, and our algorithm executes the shift operation at most $\lceil n/2 \rceil$ times. Therefore a straightforward implementation would take time $O(n^2)$. However, using a data structure in [6] representing the sets $U(w_i)$, $1 \le i \le m$, one can implement the shift operation so that the total time required by the operation is $O(n)$.

Thus our algorithm finds a drawing of G' in time $O(n)$. $\square$

References

1. G. Di Battista, P. Eades, R. Tamassia and I. G. Tollis, *Automatic graph drawing: an annotated bibliography*, Computational Geometry: Theory and Applications, 4, 235-282 (1994).
2. G. Di Battista, P. Eades, R. Tamassia and I. G. Tollis, *Graph Drawing*, Prentice Hall, NJ (1998).
3. M. Chrobak and G. Kant, *Convex grid drawings of 3-connected planar graphs*, International Journal of Computational Geometry and Applications, 7, 211-223 (1997).
4. M. Chrobak and S. Nakano, *Minimum-width grid drawings of plane graphs*, Computational Geometry: Theory and Applications, 10, 29-54 (1998).
5. N. Chiba, K. Onoguchi and T. Nishizeki, *Drawing planar graphs nicely*, Acta Informatica, 22, 187-201 (1985).
6. M. Chrobak and T. Payne, *A linear-time algorithm for drawing planar graphs on a grid*, Information Processing Letters, 54, 241-246 (1995).
7. 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, 153-173 (1984).
8. I. Fáry, *On straight lines representation of plane graphs*, Acta. Sci. Math. Szeged, 11, 229-233 (1948).
9. H. de Fraysseix, J. Pach and R. Pollack, *How to draw a planar graph on a grid*, Combinatorica, 10, 41-51 (1990).
10. X. He, *Grid embedding of 4-connected plane graphs*, Discrete & Computational Geometry, 17, 339-358 (1997).
11. G. Kant, *Drawing planar graphs using the canonical ordering*, Algorithmica, 16, 4-32 (1996).
12. G. Kant and X. He, *Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems*, Theoretical Computer Science, 172, 175-193 (1997).
13. W. Schnyder, *Embedding planar graphs on the grid*, Proc. 1st Annual ACM-SIAM Symp. on Discrete Algorithms, San Francisco, 138-148 (1990).
14. S. K. Stein, *Convex maps*, Proc. Amer. Math. Soc., 2, 464-466 (1951).
15. W.T. Tutte, *How to draw a graph*, Proc. London Math. Soc., 13, 743-768 (1963).
16. K. Wagner, *Bemerkungen zum vierfarbenproblem*, Jahresbericht der Deutschen Mathematiker-Vereinigung, 46, 26-32 (1936).