アルゴリズム 48－6
（1995．11．17）

Rectangular Grid Drawings of Plane Graphs

Md. Saidur Rahman, Shin-ichi Nakano and Takao Nishizeki
Graduate School of Information Sciences
Tohoku University, Sendai 980-77, Japan.
E-mail: saidur@nishizeki.ecei.tohoku.ac.jp, nakano@ecei.tohoku.ac.jp, nishi@ecei.tohoku.ac.jp

The rectangular drawing of a plane graph G is a drawing of G such that each edge is drawn as a horizontal or vertical line segment and each face is drawn as a rectangle. To represent such a drawing on a screen having finite resolution needs integer-valued vertex coordinates, i.e., a grid drawing. In this paper we give a simple linear time algorithm to find a rectangular grid drawing of G if it exists. We also give upper bounds on the sum of required width and height: $W + H \leq \frac{n}{2}$, and on area: $W \times H \leq \frac{n^2}{16}$, where n is the number of vertices in G. These bounds are best possible, and hold for any area-efficient rectangular grid drawing.

平面グラフの格子矩形描画

モハマッド サイドゥル ラハマン　　中野眞一　　西関 隆夫
東北大学大学院情報科学研究科

平面グラフの矩形描画とは各辺を水平あるいは垂直線分で描き，辺の交差がなくしかも各面が長方形であるように描画することである．そのような描画を有限の解像度のスクリーン上に描くには各点の座標が整数値であることが望ましい．そのような描画を格子矩形描画という．本論文では与えられたグラフの格子矩形描画が存在する限りそれを見つける簡単な線形時間アルゴリズムを与える．さらに描画に必要な格子の高さ Hと幅 Wの和の上界 $W + H \leq n/2$ および格子の面積の上界 $W \times H \leq n^2/16$ を与える．ここで n はグラフの点の個数である．これらの上界は最良であり，すべてのつめた格子矩形描画において成立する．

1 Introduction

Recently automatic drawing of graphs has created intense interest due to its broad application to represent various concepts and objects in software, computer architecture, networks, VLSI circuits, etc [BETT94]. One important criteria of a graph drawing is that it should be easily understandable and good looking [CON85]. In this paper we consider *the rectangular drawing* of a plane graph, where each edge of the graph is drawn as a horizontal or vertical line segment with no bend and each face of the graph is drawn as a rectangle (See Fig. 1). Our input graph has four vertices of degree 2 on its outer face and all other vertices have degree 3. It is not always possible to draw such a graph in this way. C. Thomassen obtained a necessary and sufficient condition for such a plane graph to have a rectangular drawing [T84]. He used induction in the proof, which yields an algorithm for rectangular drawing based on divide and conquer strategy. In his method a lot of complex conditional checks are needed to find a "partitioning path" for dividing a graph G into two or more subgraphs. A straightforward implementation of his method leads to an $O(n^2)$ time algorithm using a suitable data structure.

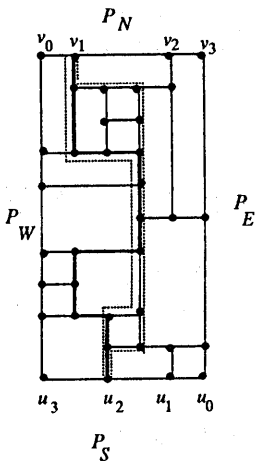

Figure 1: An example of a rectangular drawing of a plane graph.

On the other hand, K. Kozminski and E. Kinnen [KK84] established a necessary and sufficient condition for the existence of a rectangular dual of a plane graph, that is a rectangular drawing of the dual graph of a given plane graph, and gave an $O(n^2)$ algorithm to obtain it. Since a rectangular dual has beautiful applications on VLSI floorplan, much attention has been paid to [KK84]. Based on the characterization of [KK84], J. Bhasker and S. Sahni [BS88] and X. He [H93] respectively developed linear time algorithms to find a rectangular dual of a plane graph. Recently G. Kant and X. He [KH94] presented two more linear time algorithms. The algorithm in [BS88] is fairly complicated and consists of two steps: (1) constructing a "regular edge labeling" of G and (2) constructing the rectangular dual using that labeling. X. He [H93] simplified step (2), but used the complicated algorithm in [BS88] for step (1). Two methods for simplifying step (1) are given in [KH94]. The first one finds a regular edge labeling using "edge contraction and edge expansion techniques" and is indeed not a simple method. The second one finds a "canonical ordering" of the given graph in linear time and then obtains a regular edge labeling from the canonical ordering in linear time.

The two works, [T84] and [KK84], were completely independent. Since there exists a linear time algorithm to find the dual graph of a plane graph, algorithms in [BS88], [H93] and [KH94] can be used to find a rectangular drawing of a given plane graph and also Thomassen's

characterization can be applied to VLSI floorplan problems.

A rectangular drawing in which each vertex is located at a grid point is called a *rectangular grid drawing*. It is a very challenging problem to draw a plane graph on a grid of the minimum size. In recent years, several published works are devoted to this field [Sc90, CN94]; for example, any plane graph with n vertices has a straight line drawing on a grid of area $W \times H \leq (n-1)^2$, where W is the width and H is the height of the grid.

In this paper we simplify the Thomassen's conditions to check and also reduce the amount of such checks, and give a simple algorithm to find a rectangular drawing of a plane graph in linear time. Our algorithm is completely different from those of [BS88], [H93] and [KH94], and is simpler than them: we need not to find any regular edge labeling or a canonical ordering, and our algorithm finds a rectangular drawing directly using a simple depth-first search. Furthermore our algorithm finds a rectangular *grid* drawing. We also give upper bounds on the sizes of rectangular grid drawings: $W \times H \leq \frac{n^2}{16}$ and $W + H \leq \frac{n}{2}$. The bounds on grid sizes are interesting in a sense that they are best possible and hold for any area-efficient rectangular grid drawing of a plane graph.

This paper is organized as follows. Section 2 introduces some definitions and lemmas needed by our algorithm. A rectangular drawing algorithm is given in Section 3. Our bounds on grid sizes are described in Section 4. Finally we put our discussions and some open problems in Section 5.

2 Preliminaries

In this section we introduce some definitions and lemmas used in our algorithm.

Let $G = (V, E)$ be a graph with vertex set V and edge set E. $V(G)$ denotes the set of vertices of G, and $E(G)$ denotes the set of edges of G. The *union* $G = G' \cup G''$ of two graphs G' and G'' is a graph $G = (V(G') \cup V(G''), E(G') \cup E(G''))$. 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 regard the *contour* of a face as a *clockwise* cycle formed by the edges on the boundary of the face. We denote the contour of the outer face of graph G by $C_o(G)$ or simply C_o. In our input graph G all vertices have degree 3 except four vertices of degree 2 on $C_o(G)$. These four corner vertices divide C_o into four paths, which we call the *N-path* P_N, *E-path* P_E, *S-path* P_S, and *W-path* P_W. We will draw P_N and P_S on two horizontal straight line segments and P_E and P_W on two vertical line segments (See Fig. 1). For $E' \subseteq E(G)$, $G - E'$ denotes a graph obtained from G by deleting all the edges in E'. A connected component of $G - E(C_o)$ is called a *C_o-component*. We have the following Lemma.

Lemma 2.1 *Let $H_1, H_2, ..., H_p$ be the C_o-components of a plane graph G and $G_i = G - \bigcup_{j \neq i} E(H_j)$. Then G has a rectangular drawing with a fixed rectangular embedding of $C_o(G)$ if and only if G_i has a rectangular drawing with a fixed rectangular embedding of $C_o(G_i) = C_o(G)$ for each i, $1 \leq i \leq p$.* □

Note that the graph G_i mentioned in Lemma 2.1 may have some vertices of degree 2 in addition to the four vertices of degree 2 of G. In the remaining of this section, because of Lemma 2.1, we may assume that G has exactly one C_o-component as one in Fig. 1.

For a cycle C in G we denote by $G(C)$ the plane subgraph of G inside C (including C). An edge incident to a vertex on a cycle C and located outside C is called a *leg* of the cycle C. We have the following lemmas.

Lemma 2.2 *G has no rectangular drawing if the C_o-component has a cycle with less than four legs.* □

An inner face of G is called a *boundary face* if its contour contains an edge on C_o. The maximal (directed) path on the contour of a boundary face connecting two vertices on C_o without passing through any edge on C_o is called a *boundary path.* Note that the direction of a boundary path is the same as the contour of the face, and hence is clockwise. A boundary path starting at a vertex u and ending at a vertex v is called a *boundary N*-path* if $u \in V(P_N)$, a *boundary *N-path* if $v \in V(P_N)$, and a *boundary NN-path* if $u \in V(P_N)$ and $v \in V(P_N)$. Similarly we define *boundary E*-, *E-, NE-, WW-paths etc.*

Let $P = v_0v_1, v_1v_2, ..., v_{l-1}v_l$ be a (simple) path in G. A cycle C is *attached* to P if P does not contain any vertices in the proper inside of C and the intersection of C and P is a single subpath of P: $v_iv_{i+1}, v_{i+1}v_{i+2}, ..., v_{j-1}v_j$. We call v_i the *tail vertex* of C for P, and v_j the *head vertex.* Denote by $Q_c(C)$ the clockwise subpath on C from v_i to v_j and by $Q_{cc}(C)$ the counterclockwise one. A leg of C is called a *clockwise leg* for P if it is incident to a vertex in $V(Q_c(C)) - \{v_i, v_j\}$. Denote by $n_c(C)$ the number of clockwise legs of C for P. Similarly we define *counterclockwise leg* and denote the number of counterclockwise legs by $n_{cc}(C)$ of cycle C for P. Cycle C is called a *clockwise cycle* for P if $Q_c(C)$ is a subpath of P. A cycle C is called a *clockwise critical cycle* for P if C is a clockwise cycle and $n_{cc}(C) \leq 1$. A clockwise critical cycle C is *maximal* if $E(C)$ is not located inside any other clockwise critical cycle. Similarly we define the terms above for counterclockwise case. We have the following two lemmas.

Lemma 2.3 *G has no rectangular drawing if G has a critical cycle for path P_N, P_E, P_S or P_W.* □

Lemma 2.4 *G has no rectangular drawing if G has an attached cycle C with $n_{cc}(C) = 0$ for path $P = P_N + P_E, P_E + P_S, P_S + P_W$, or $P_W + P_N$.* □

We call the C_o-component as ones mentioned in Lemmas 2.2, 2.3 and 2.4 a *bad component.* In particular, we call the C_o-component of a graph mentioned in Lemma 2.4 a *bad corner.* We now have the following theorem on the necessary and sufficient condition for a graph to have a rectangular drawing.

Theorem 2.5 *Let G be a plane graph having four vertices of degree 2 on the contour C_o of the outer face and let all other vertices of G have degree 3. Let C_o be divided into four paths P_N, P_E, P_S and P_W by the four vertices of degree 2. Then G has a rectangular drawing for the fixed rectangular embedding of C_o by P_N, P_E, P_S and P_W if and only if G has no bad component.* □

The necessity of Theorem 2.5 is immediate from Lemmas 2.2, 2.3 and 2.4. By the end of this section we will be able to prove its sufficiency constructively. In the remaining of this section we assume that G has no bad component, that is, G satisfies the necessary and sufficient condition.

Let $P_N = v_0v_1, v_1v_2, ..., v_{p-1}v_p$ and $P_S = u_0u_1, u_1u_2, ..., u_{q-1}u_q$. We define that an *NS-path* is a path starting at v_i and ending at u_j without passing through any other vertex on C_o. An NS-path P divides graph G into two subgraphs G_W^P and G_E^P; G_W^P is the west part of G including P, and G_E^P is the east part of G. We fix the embedding of $C_o(G_W^P)$ as a rectangle with $P'_N(= v_0v_1, v_1v_2, ..., v_{i-1}v_i), P'_E(= P), P'_S(= u_ju_{j+1}, u_{j+1}u_{j+2}, ..., u_{q-1}u_q), P'_W(= P_W)$ by drawing P as a straight line segment. Similarly we fix the embedding of $C_o(G_E^P)$ as a rectangle. We say that P is a *good bipartition-path* if neither G_W^P nor G_E^P has a bad component. Similarly we define *good SN-, WE- and EW-bipartition-paths.* We have the following lemma.

Lemma 2.6 *Any boundary NS-,SN-,EW- or WE-path P of G is a good bipartition-path.* □

Thus we may assume that G has none of boundary NS-, EW-, SN- and WE-paths. Then the C_o-component has at least one vertex on each of paths P_N, P_E, P_S and P_W. We say that an NS-path P is *westmost* if (1) P starts at v_1, (2) P ends at u_{q-1}, and (3) the number of edges in G_W^P is minimum. The west most NS-path is drawn in thick lines in Fig. 1. We have the following two lemmas.

Lemma 2.7 *If G has none of boundary NS-, EW-, SN- and WE-paths, then G has the westmost NS-path.* □

Lemma 2.8 *Assume that a cycle C in the C_o-component has exactly four legs dividing C into four paths P'_N, P'_E, P'_S and P'_W. If G has no bad component, then the subgraph $G(C)$ of G inside C has no bad component for any fixed rectangular embedding of C by P'_N, P'_E, P'_S and P'_W.* □

Let P be an NS-path, and let $C_1, C_2, ..., C_k$ be cycles attached to P, each of which has exactly one clockwise leg and exactly one counterclockwise leg. We construct two paths P_c and P_{cc} from P and $C_1, C_2, ..., C_k$ as follows: $P_c = P_1 + Q_c(C_1) + P_2 + Q_c(C_2) + \cdots + Q_c(C_k) + P_{k+1}$ and $P_{cc} = P_1 + Q_{cc}(C_1) + P_2 + Q_{cc}(C_2) + \cdots + Q_{cc}(C_k) + P_{k+1}$ where (1) $P_1, P_2, \cdots, P_{k+1}$ are subpaths of P shared by P_c and P_{cc}; (2) P_1 starts at the vertex $v_s \in V(P_N) \cap V(P)$ and ends at the tail vertex of C_1; (3) for $2 \leq i \leq k$, P_i is a subpath of P starting at the head vertex of C_{i-1} and ending at the tail vertex of C_i; and (4) P_{k+1} starts at the head vertex of C_k and ends at the vertex $u_l \in V(P_S) \cap V(P)$.

By Lemma 2.8 we can assume that none of $G(C_1), G(C_2), ..., G(C_k)$ has a bad component for any fixed embedding of $C_1, C_2, ..., C_k$ as in Fig. 2. One can observe that if $G_W^{P_{cc}}$ has a rectangular drawing such that P_{cc} is drawn as a vertical line segment then it can be easily modified to fit in the area for $G_W^{P_{cc}}$ in Fig. 2 where P_{cc} is drawn as a zig-zag line. Similarly if $G_E^{P_c}$ has a rectangular drawing such that P_c is drawn as a vertical line segment, then it can be easily modified to fit in the area for $G_E^{P_c}$ in Fig. 2. Thus if we have rectangular drawings of graphs $G_W^{P_{cc}}, G_E^{P_c}, G(C_1), G(C_2), ..., G(C_k)$, then we can naturally patch them to get a rectangular drawing of G. We say that P_c and P_{cc} are *good NS-multipartition-paths* if neither $G_E^{P_c}$ nor $G_W^{P_{cc}}$ has a bad component. Note that for each C_i, $1 \leq i \leq k$, there are two alternative rectangular embeddings of C_i. We can arbitrarily choose one of the 2^k different embeddings of cycles $C_1, C_2, ..., C_k$. (See Fig. 2.) We have the following lemma.

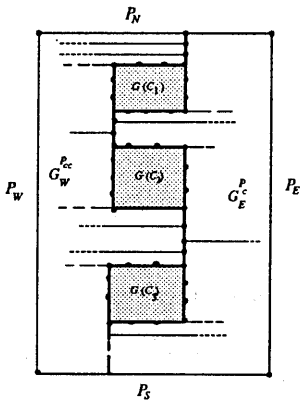

Figure 2: An example of the embedding of good multipartition-paths.

Lemma 2.9 *If G has no bad component and has none of boundary NS-, EW-, SN- and WE-paths, then one can find good NS-multipartition-paths P_c and P_{cc} from the westmost NS-path.* □

Using Lemmas 2.6, 2.7, 2.8 and 2.9, one can recursively find a rectangular drawing of a given plane graph G if it has no bad component. Thus we have proved the sufficiency of Theorem 2.5. One may also verify the correctness of our algorithm given in Section 3 using these lemmas.

3 Drawing Algorithm

The following algorithm DRAW-GRAPH finds a rectangular drawing of a plane graph G if it exists. We treat each C_o-component independently as in Lemma 2.1. If there exists a boundary NS-, SN-, WE-, or EW-path, we choose it as a good bipartition-path. Otherwise, we find the westmost NS-path and then find good NS-multipartition-paths P_c and P_{cc} from the westmost NS-path. The algorithm can be easily modified to check the existence of a rectangular drawing in a plane graph G.

Algorithm DRAW-GRAPH(G)
begin
1 draw the contour $C_o(G)$ of the outer face of G as a rectangle by two horizontal line segments P_N and P_S and two vertical line segments P_E and P_W;
{ the directions of edges on C_o are decided}
2 find all C_o-components $H_1, H_2, \cdots, H_p$;
3 **for** each C_o-component H_i **do**
begin
4 $G_i = C_o \cup H_i$; {G_i is the union of graphs C_o and H_i}
5 DRAW(H_i, G_i)
end
end.

Procedure DRAW(H, G)
{ H is the C_o-component of graph G}
begin
1 **if** H has a boundary NS-, EW-, SN-, or WE-path P
{P is a good bipartition-path}
then begin
assume without loss of generality that P is an NS-path;
2 draw all edges on P on a vertical line;
{the directions of edges are decided to be vertical}
3 **if** $|E(P)| \geq 2$ **then begin**
let $F_1, F_2, ..., F_q$ be the C_o-components of G_E^P for the fixed rectangular embedding of cycle $C_o(G_E^P)$; {G_W^P is a cycle}
4 **for** each F_i **do** DRAW($F_i, C_o(G_E^P) \cup F_i$)
end
end
5 **else begin** {there exists the westmost NS-path P }
6 find the westmost NS-path P;
7 find good multipartition-paths P_c and P_{cc} from P;
8 **if** $P_c = P_{cc}$ **then begin**
9 draw all edges on P_c on a vertical line segment;
let $G_1 = G_W^{P_c}$ and $G_2 = G_E^{P_c}$ be two resulting subgraphs with fixed rectangular embeddings of cycles $C_o(G_W^{P_c})$ and $C_o(G_E^{P_c})$;
10 **for** each G_i **do**
begin
let $F_1, F_2, ..., F_q$ be the C_o-components of G_i;
11 **for** each F_j **do** DRAW($F_j, C_o(G_i) \cup F_j$)
end
end
12 **else begin**
13 draw all edges on P_c and P_{cc} on zig-zag lines as in Fig. 2;
let $G_1 = G_W^{P_{cc}}, G_2 = G_E^{P_c}, G_3 = G(C_1), \cdots, G_{k+2} = G(C_k)$ be the

```
                resulting subgraphs with fixed rectangular embeddings of cycles
                C_o(G_W^{P_cc}), C_o(G_E^{P_c}), C_1, ..., C_k;
14              for each G_i do
                    begin
                      let F_1, F_2, ..., F_q be the C_o-components of G_i;
15                    for each F_j do DRAW(F_j, C_o(G_i) ∪ F_j)
                    end
            end
        end
    end
```

The algorithm DRAW-GRAPH(G) only finds the directions of all edges in G for rectangular drawing. From the directions one can determine the integer coordinates of vertices in G in linear time. Let the coordinates of the south-west corner be $(0,0)$, and let that of the north-east corner be (W,H). Then our grid drawing is "area-efficient" in a sense that there is at least one vertical line of x-coordinate i for each i, $0 \leq i \leq W$ and there is at least one horizontal line of y-coordinate j for each j, $0 \leq j \leq H$. We have a proof of the following Theorm on execution time of the algorithm DRAW-GRAPH.

Theorem 3.1 *The algorithm DRAW-GRAPH finds a rectangular drawing of a given plane graph in linear time if it exists.* □

4 Grid Area

In this section we prove the following theorem.

Theorem 4.1 *The sizes of any area-efficient rectangular grid drawing D of G satisfy $W+H \leq n/2$ and $W \times H \leq \frac{n^2}{16}$.* □

Before giving a proof of Theorem 4.1, we present a lemma.

Lemma 4.2 *Let G be a plane graph having four vertices of degree 2 on the contour C_o of outer face and let all other vertices have degree 3, and let f be the number of faces of G. Let D be any rectangular drawing of G, and l be the number of maximal horizontal and vertical line segments in D. Then $l = f + 2$.* □

We are now ready to prove Theorem 5.1.

Proof of Theorem 4.1. From Lemma 4.2 we have

$$l = f + 2. \tag{1}$$

As our input graph has exactly four vertices of degree 2, we have

$$\sum_{v \in V(G)} d(v) = 3n - 4 = 2m, \tag{2}$$

where $n = |V|$ and $m = |E|$. Using (2) and Euler's formula $n - m + f = 2$, we have

$$n = 2f. \tag{3}$$

By (1) and (3) we have

$$l = n/2 + 2. \tag{4}$$

Let l_h be the number of maximal horizontal line segments and l_v the number of maximal vertical line segments in an area-efficient rectangular drawing D. Then we have $H \leq l_h - 1$, and $W \leq l_v - 1$. Therefore,

$$W + H \leq l_v + l_h - 2 = l - 2 = n/2.$$

This relation immediately implies the bound on area: $W \times H \leq \frac{n^2}{16}$. This completes proof of Theorem 4.1. □

The above bounds are tight, because there are an infinite number of examples attaining the bounds.

5 Conclusion

In this paper we presented a simple linear time algorithm to find a rectangular grid drawing of a plane graph and also gave upper bounds on grid sizes. The bounds are tight and best possible. Our work raises several interesting open problems:

(1) What is the necessary and sufficient condition to have a rectangular drawing of a plane graph with vertices of degree less than or equal to 4?

(2) What is the complexity of an optimal parallel algorithm for rectangular grid drawings?

References

[BETT94] G. D. Battista, P. Eades, R. Tamassia and I. G. Tollis, *Algorithms for drawing graphs: an annotated bibliography*, Comp. Geom. Theory Appl., to appear.

[BS88] J. Bhasker and S. Sahni, *A linear algorithm to find a rectangular dual of a planar triangulated graph*, Algorithmica, 3 (1988), pp. 247-278.

[CON85] N. Chiba, K. Onoguchi, and T. Nishizeki, *Drawing planar graphs nicely*, Acta Informatica, 22 (1985), pp. 187-201.

[CN94] M. Chrobak and S. Nakano, *Minimum-width grid drawings of plane graphs*, Technical Report, UCR-CS-94-5, Department of Computer Science, University of California at Riverside, 1994.

[H93] X. He, *On finding the rectangular duals of planar triangulated graphs*, SIAM J. Comput., 22(6) (1993), pp. 1218-1226.

[KH94] G. Kant and X. He, *Two algorithms for finding rectangular duals of planar graphs*, Graph-Theoretic Concepts in Computer Science, 19th International Workshop, WG'93 Proceedings, (1994), pp. 396-410.

[KK84] K. Kozminski and E. Kinnen, *An algorithm for finding a rectangular dual of a planar graph for use in area planning for VLSI integrated circuits*, Proc. 21st DAC, Albuquerque, June (1984), pp. 655-656.

[Sc90] W. Schnyder, *Embedding planar graphs in the grid*, Proc. first ACM-SIAM Symp. on Discrete Algorithms, San Francisco, (1990), pp. 138-147.

[T84] C. Thomassen, *Plane representations of graphs*, (Eds.) J.A. Bondy and U.S.R. Murty, Progress in Graph Theory, Academic Press Canada, (1984), pp. 43-69.