10

Rectangular Drawing Algorithms

Takao Nishizeki
Kwansei Gakuin University, Japan

Md. Saidur Rahman
BUET, Bangladesh

10.1 Introduction

A *rectangular drawing* of a plane graph G, a planar graph G with a fixed embedding, is a drawing of G in which each vertex is drawn as a point, each edge is drawn as a horizontal or vertical line segment without edge-crossings, and each face is drawn as a rectangle. Figure 10.1(b) illustrates a rectangular drawing of the plane graph in Fig. 10.1(a).

Rectangular drawings have practical applications in VLSI floorplanning and architectural floorplanning [NR04]. In a VLSI floorplanning problem, an input is a plane graph F as illustrated in Fig. 10.2(a); F represents the functional entities of a chip, called *modules*, and interconnections among the modules; each vertex of F represents a module, and an edge between two vertices of F represents the interconnections between the two corresponding modules. An output of the problem for the input graph F is a partition of a rectangular chip area into smaller rectangles as illustrated in Fig. 10.2(d); each module is assigned to a smaller rectangle, and furthermore, if two modules have interconnections, then their corresponding rectangles must be adjacent, that is, must have a common boundary.

A similar problem arises in architectural floorplanning. When building a house, the owner may have some preference; for example, a bedroom should be adjacent to a reading room. The owner's choice of room adjacencies can be easily modeled by a plane graph F, as illustrated in Fig. 10.2(a); each vertex represents a room and an edge between two vertices represents the desired adjacency between the corresponding rooms. A rectangular drawing of a plane graph may provide a suitable solution of the floorplanning problem described above. First, obtain a plane graph F' by triangulating all inner faces of F as illustrated in Fig. 10.2(b), where dotted lines indicate new edges added to F. Then obtain a dual-like graph G of F' as illustrated in Fig. 10.2(c), where the four vertices of degree 2 drawn by white circles correspond to the four corners of the rectangular area. Finally, find a rectangular drawing of the plane graph G to obtain a possible floorplan for F as illustrated in Fig. 10.2(d).

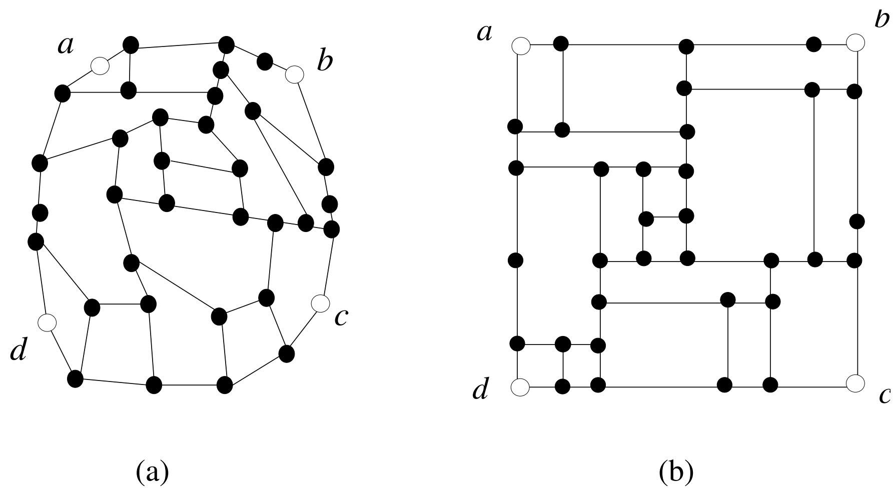


Figure 10.1 (a) Plane graph, and (b) its rectangular drawing for the designated corners a, b, c and d. (Figure taken from [NR04].)

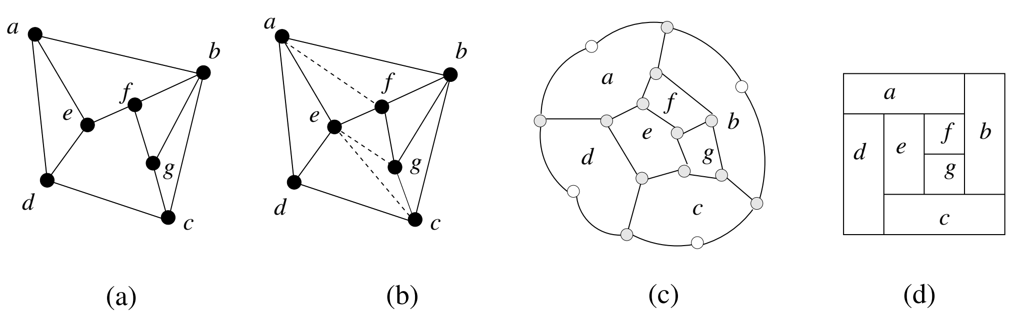


Figure 10.2 (a) Graph F, (b) triangulated graph F', (c) dual-like graph G, and (d) rectangular drawing of G. (Figure taken from [NR04].)

In a rectangular drawing of G, the outer cycle $C_o(G)$ is drawn as a rectangle and hence has four convex corners such as a, b, c and d drawn by white circles in Fig. 10.1. Such a convex corner is an outer vertex of degree two and is called a *corner of the rectangular drawing.* Not every plane graph G has a rectangular drawing. Of course, G must be 2-connected and the maximum degree Δ of G is at most four if G has a rectangular drawing. Miura et al. showed that a plane graph G with $\Delta \leq 4$ has rectangular drawing D if and only if a new bipartite graph constructed from G has a perfect matching, and D can be found in time $O(n^{1.5}/\log n)$ whenever G has D [MHN06]. In Section 10.2 we present their result on rectangular drawings of plane graphs with $\Delta \leq 4$.

Since a planar graph with $\Delta \leq 3$ often appears in many practical applications, much works are devoted to rectangular drawings of planar graphs with $\Delta \leq 3$ [BS88, Tho84, RNN98, RNN02]. In Section 10.3 we present a necessary and sufficient condition for a plane graph G with $\Delta \leq 3$ to have a rectangular drawing when four outer vertices of degree two are designated as the corners [Tho84], and also present a linear-time algorithm to obtain a rectangular drawing with the designated corners [RNN98]. The problem of examining whether a plane graph has a rectangular drawing becomes difficult when four outer vertices are not designated as the corners. We also present a necessary and sufficient condition for a plane graph with $\Delta \leq 3$ to have a rectangular drawing for some quadruplet of outer vertices appropriately chosen as the corners; the condition leads to a linear-time algorithm [RNN02, NR04].

In the floorplan described in Fig. 10.2(d), two rectangles are always adjacent if the modules corresponding to them have interconnections in F in Fig. 10.2(a). However, two rectangles may be adjacent even if the modules corresponding to them have no interconnections in F. For example, module e and module g have no interconnection in F, but their corresponding rectangles are adjacent in the floorplan in Fig. 10.2(d). Such unwanted adjacencies are not desirable in some other floorplanning problems.

In floorplanning of a MultiChip Module (MCM), two chips generating excessive heat should not be adjacent, or two chips operating on high frequency should not be adjacent to avoid malfunctioning due to their interference [She95]. Unwanted adjacencies may cause a dangerous situation in some architectural floorplanning, too [FW74]. For example, in a chemical industry, a processing unit that deals with poisonous chemicals should not be adjacent to a cafeteria. We can avoid the unwanted adjacencies if we obtain a floorplan for F by using a "box-rectangular drawing" instead of a rectangular drawing. A *box-rectangular drawing* of a plane graph G is a drawing of G such that each vertex is drawn as a rectangle, called a *box*, each edge is drawn as a straight line segment joining points on the two boxes corresponding to the ends, and the contour of each face is drawn as a rectangle, as illustrated in Fig. 10.3(c). A vertex may be drawn as a degenerate rectangle, that is, a point. A floorplan can be obtained by using a box-rectangular drawing as follows. First, without triangulating the inner faces of F, find a dual-like graph G of F as illustrated in Fig. 10.3(b). Then find a box-rectangular drawing of G to obtain a possible floorplan for F as illustrated in Fig. 10.3(c). In Fig. 10.3(c) rectangles e and g are not adjacent although there is a dead space corresponding to a vertex of G drawn by a rectangular box. Such a dead space to separate two rectangles in floorplanning is desirable for dissipating excessive heat in an MCM or for ensuring safety in a chemical industry. If G has multiple edges or a vertex of degree five or more, then G has no rectangular drawing but may have a box-rectangular drawing. However, not every plane graph has a box-rectangular drawing. Section 10.4 presents a necessary and sufficient condition for the existence of a box-rectangular drawing of a plane graph, and gives a linear algorithm to find a box-rectangular drawing if it exists.

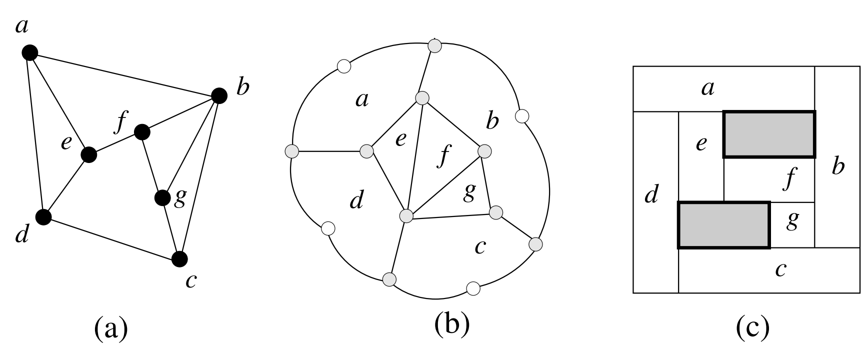


Figure 10.3 (a) F, (b) G, and (c) box-rectangular drawing of G. (Figure taken from [NR04].)

10.2 Rectangular Drawing and Matching

This section deals with rectangular drawings of plane graphs with $\Delta \leq 4$, and shows that a plane graph G with $\Delta \leq 4$ has rectangular drawing D if and only if a new bipartite graph $G_{\rm d}$ constructed from G has a perfect matching, and D can be found in time $O(n^{1.5}/\log n)$ if D exists [MHN06, NR04]. $G_{\rm d}$ is called a decision graph.

One may assume without loss of generality that G is 2-connected and $\Delta \leq 4$, and hence every vertex of G has degree two, three or four.

An angle formed by two edges e and e' incident to a vertex v in G is called an *angle of* v if e and e' appear consecutively around v. An angle of a vertex in G is called an *angle of* G. An angle formed by two consecutive edges on a boundary of a face F in G is called an *angle of* F. An angle of the outer face is called an *outer angle* of G, while an angle of an inner face is called an *inner angle*.

In any rectangular drawing, every inner angle is 90° or 180°, and every outer angle is 180° or 270°. Consider a labeling Θ which assigns a label 1, 2, or 3 to every angle of G, as illustrated in Fig. 10.4(b). Labels $1, 2$ and 3 correspond to angles 90°, 180° and 270°, respectively. Therefore each inner angle has label either 1 or 2, exactly four outer angles have label 3, and all other outer angles have label 2.

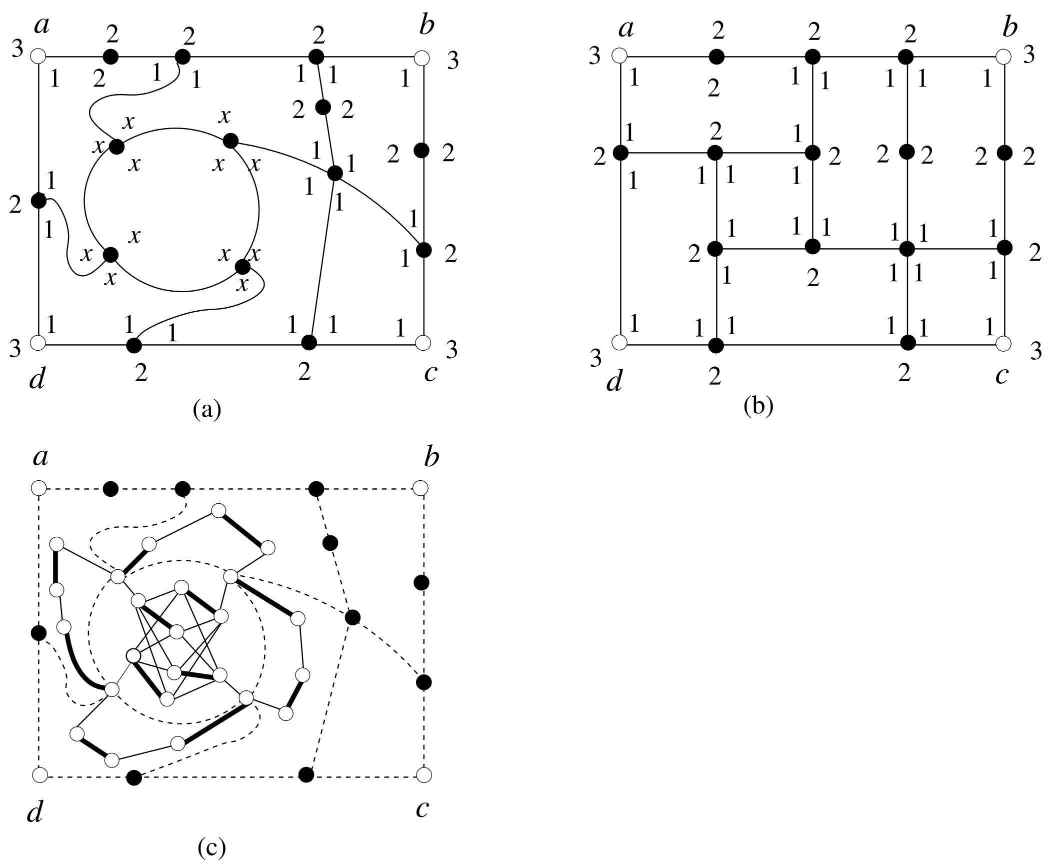


Figure 10.4 (a) Plane graph G, (b) rectangular drawing D and regular labeling Θ of G, and (c) decision graph $G_{\rm d}$. (Figure taken from [NR04].)

We call Θ a *regular labeling* of G if Θ satisfies the following three conditions (a)–(c):

(a) For each vertex v of G, the sum of the labels of all the angles of v is equal to 4;
(b) The label of any inner angle is 1 or 2, and every inner face has exactly four angles of label 1; and
(c) The label of any outer angle is 2 or 3, and the outer face has exactly four angles of label 3;

Figure 10.4(b) depicts a regular labeling Θ of the plane graph in Fig. 10.4(a) and a rectangular drawing D corresponding to Θ. A regular labeling is a special case of an orthogonal representation of an orthogonal drawing presented in [Tam87].

Conditions (a) and (b) imply the following (i)–(iii):

(i) If a non-corner vertex v has degree two, that is, $d(v) = 2$, then the two labels of v are 2 and 2.
(ii) If $d(v) = 3$, then exactly one of the three angles of v has label 2 and the other two have label 1.
(iii) If $d(v) = 4$, then all the four angles of v have label 1.

If G has a rectangular drawing, then clearly G has a regular labeling. Conversely, if G has a regular labeling, then G has a rectangular drawing, as can be proved by means of elementary geometric considerations. We thus have the following fact.

Fact 10.1 *A plane graph G has a rectangular drawing if and only if G has a regular labeling.*

Assume now that four outer vertices a, b, c and d of degree two are designated as corners. Then the outer angles of a, b, c and d must be labeled with 3, and all the other outer angles of G must be labeled with 2, as illustrated in Fig. 10.4(a). Some of the inner angles of G can be immediately determined, as illustrated in Fig. 10.4(a). If v is a non-corner outer vertex of degree two, then the inner angle of v must be labeled with 2. The two angles of any inner vertex of degree two must be labeled with 2. If v is an outer vertex of degree three, then the outer angle of v must be labeled with 2 and both of the inner angles of v must be labeled with 1. All the four angles of each vertex of degree four must be labeled with 1. On the other hand we label all the three angles of an inner vertex of degree three with x, because one cannot determine their labels although exactly one of them must be labeled with 2 and the others with 1. Label x means that x is either 1 or 2, and exactly one of the three labels x's attached to the same vertex must be 2 and the other two must be 1. (See Figs. 10.4(a) and (b).)

We now present how to construct a decision graph G_{d} of G. Let all vertices of G that have been attached label x be vertices of G_{d}. Thus all the inner vertices of degree three are vertices of G_{d}, and none of the other vertices of G is a vertex of G_{d}. We then add to G_{d} a complete bipartite graph inside each inner face F of G, as illustrated in Fig. 10.5 where G_{d} is drawn by solid lines and G by dotted lines. Let n_x be the number of angles of F labeled with x. For example, $n_x = 3$ for the face F in Fig. 10.5. Let n_1 be the number of angles of F which have been labeled with 1. Then n_1 is the number of vertices v on F such that one of the following (i)–(iii) holds:

(i) v is a corner vertex, that is, v is an outer vertex of degree 2 and the outer angle of v is labeled with 3;
(ii) v is an outer vertex of degree 3 and the outer angle of v is labeled with 2; and

(iii) $d(v) = 4$.

Thus $n_1 = 2$ for the example in Fig. 10.5. One may assume as a trivial necessary condition that $n_1 \leq 4$; otherwise, G has no rectangular drawing. Exactly $4 - n_1$ of the n_x angles of F labeled with x must be labeled with 1 by a regular labeling. Add a complete bipartite graph $K_{(4-n_1),n_x}$ in F, and join each of the n_x vertices in the second partite set with one of the n_x vertices on F whose angles are labeled with x. Repeat the operation above for each inner face F of G. The resulting graph is a *decision graph* G_d of G. The decision graph G_d of the plane graph G in Fig. 10.4(a) is drawn by solid lines in Fig. 10.4(c), where G is drawn by dotted lines.

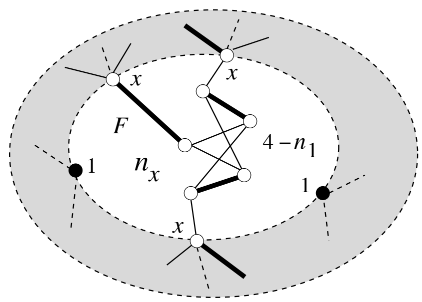


Figure 10.5 Construction of G_d for an inner face F of G. (Figure taken from [NR04].)

A *matching* of G_d is a set of pairwise non-adjacent edges in G_d. A *maximum matching* of G_d is a matching of the maximum cardinality. A matching M of G_d is called a *perfect matching* if an edge in M is incident to each vertex of G_d. A perfect matching of G_d is drawn by thick solid lines in Figs. 10.4(c) and 10.5.

Each edge e of G_d incident to a vertex v attached a label x corresponds to an angle α of v labeled with x. A fact that e is contained in a perfect matching M of G_d means that the label x of α is 2. Conversely, a fact that e is not contained in M means that the label x of α is 1. Then one can easily observe that G has a rectangular labeling if and only if G_d has a perfect matching.

Clearly, G_d is a bipartite graph, and $4-n_1 \leq 4$. Obviously, n_x is no more than the number of edges on face F. Let n be the number of vertices, and let m be the number of edges in G, then we have $2m \leq 4n$ since $\Delta \leq 4$. Therefore the sum $2m$ of the numbers of edges on all faces is at most $4n$. One can thus know that both the number n_d of vertices in G_d and the number m_d of edges in G_d are $O(n)$. Since G_d is a bipartite graph, a maximum matching of G_d can be found either in time $O(\sqrt{n_\mathrm{d}}m_\mathrm{d}) = O(n^{1.5})$ by an ordinary bipartite matching algorithm [HK73, MV80, PS82] or in time $O(n^{1.5}/\log n)$ by a pseudoflow-based bipartite matching algorithm using boolean word operations on $\log n$-bit words [Hoc04, HC04]. One can find a regular labeling Θ of G from a perfect matching of G_d in linear time. It is easy to find a rectangular drawing of G from Θ in linear time. Thus the following theorem holds [MHN06].

Theorem 10.1 *Let G be a plane graph with $\Delta \leq 4$ and four outer vertices a, b, c and d be designated as corners. Then G has a rectangular drawing D with the designated corners if and only if the decision graph G_d of G has a perfect matching. D can be found in time $O(n^{1.5}/\log n)$ whenever G has D.*

10.3 Linear Algorithms for Rectangular Drawing

This section presents Thomassen's theorem on a necessary and sufficient condition for a plane graph G with $\Delta \leq 3$ to have a rectangular drawing when four outer vertices of degree two are designated as the corners [Tho84], and gives a linear-time algorithm to find a rectangular drawing of G if it exists [RNN98].

10.3.1 Thomassen's Theorem

Before presenting Thomassen's theorem we recall some definitions. An edge of a plane graph G is called a *leg* of a cycle C if it is incident to exactly one vertex of C and located outside C. The vertex of C to which a leg is incident is called a *leg-vertex* of C. A cycle in G is called a *k-legged cycle* of G if C has exactly k legs in G and there is no edge which joins two vertices on C and is located outside C. Figure 10.6(a) illustrates 2-legged cycles C_1, C_2, C_3 and C_4, while Fig. 10.6(b) illustrates 3-legged cycles C_5, C_6, C_7 and C_8, where corners are drawn by white circles.

If a 2-legged cycle contains at most one corner like C_1, C_2 and C_3 in Fig. 10.6(a), then some inner face cannot be drawn as a rectangle and hence G has no rectangular drawing. Similarly, if a 3-legged cycle contains no corner like C_5 and C_8 in Fig. 10.6(b), then G has no rectangular drawing. One can thus observe the following fact.

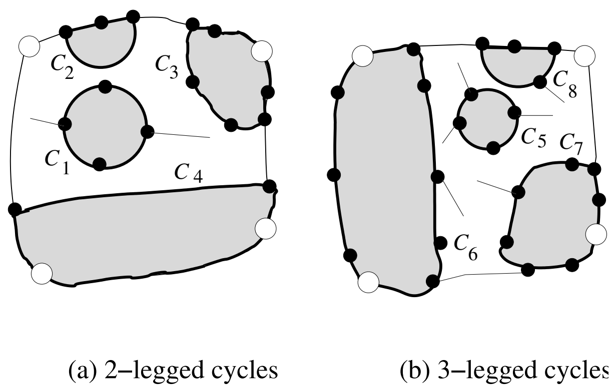


Figure 10.6 Good cycles C_4, C_6 and C_7, and bad cycles C_1, C_2, C_3, C_5 and C_8. (Figure taken from [NR04].)

Fact 10.2 *In any rectangular drawing D of G, every 2-legged cycle of G contains two or more corners, every 3-legged cycle of G contains one or more corner, and every cycle with four or more legs may contain no corner, as illustrated in Fig. 10.7.*

The necessity of the following Thomassen's theorem [Tho84] is immediate from Fact 10.2.

Theorem 10.2 *Assume that G is a 2-connected plane graph with $\Delta \leq 3$ and four outer vertices of degree two are designated as the corners a, b, c and d. Then G has a rectangular drawing if and only if*

(r1) *any 2-legged cycle contains two or more corners, and*

(r2) *any 3-legged cycle contains one or more corners.*

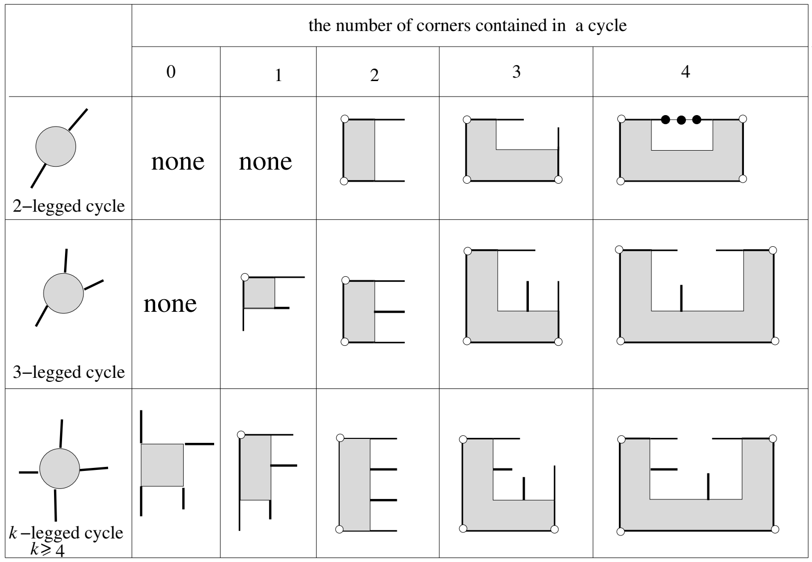


Figure 10.7 Numbers of corners in drawings of cycles. (Figure taken from [NR04].)

A cycle of type (r1) or (r2) is called *good*. Cycles C_4, C_6 and C_7 in Fig. 10.6 are good cycles; the 2-legged cycle C_4 contains two corners, and the 3-legged cycles C_6 and C_7 contain one or two corners. On the other hand, a 2-legged or 3-legged cycle is called *bad* if it is not good. Thus 2-legged cycles C_1, C_2 and C_3 and 3-legged cycles C_5 and C_8 are bad cycles. Thus Theorem 10.2 can be rephrased as follows: G has a rectangular drawing if and only if G has no bad cycle.

The rest of this section outlines a constructive proof of the sufficiency of Theorem 10.2 [RNN98].

The *union* $G = G' \cup G''$ of two graphs G' and G'' is a graph $G = (V(G') \cup V(G''), E(G') \cup E(G''))$.

In a given 2-connected plane graph G, four outer vertices of degree two are designated as the corners a, b, c and d. These four corners divide the outer cycle $C_o(G)$ of G into four paths, the north path P_N, the east path P_E, the south path P_S, and the west path P_W, as illustrated in Fig. 10.8(a). The north and south paths will be drawn as two horizontal straight line segments, and the east and west paths as two vertical line segments. Thus the embedding of $C_o(G)$ is fixed as a rectangle, which is called the *outer rectangle* of G.

A graph of a single edge, not in the outer cycle $C_o(G)$, joining two vertices in $C_o(G)$ is called a $C_o(G)$-*component* of G. A graph which consists of a connected component of $G - V(C_o(G))$ and all edges joining vertices in that component and vertices in $C_o(G)$ is also called a $C_o(G)$-*component.* The outer cycle $C_o(G)$ of the plane graph G in Fig. 10.8(a) is drawn by thick lines, and the $C_o(G)$-components J_1, J_2 and J_3 of G are depicted in Fig. 10.8(b). Clearly the following lemma holds.

LEMMA 10.1 Let $J_1, J_2, \cdots, J_p$ be the $C_o(G)$-components of a plane graph G, and let $G_i = C_o(G) \cup J_i$, $1 \le i \le p$, as illustrated in Fig. 10.9. Then G has a rectangular drawing with corners a, b, c and d if and only if, for each index i, $1 \le i \le p$, G_i has a rectangular drawing with corners a, b, c and d.

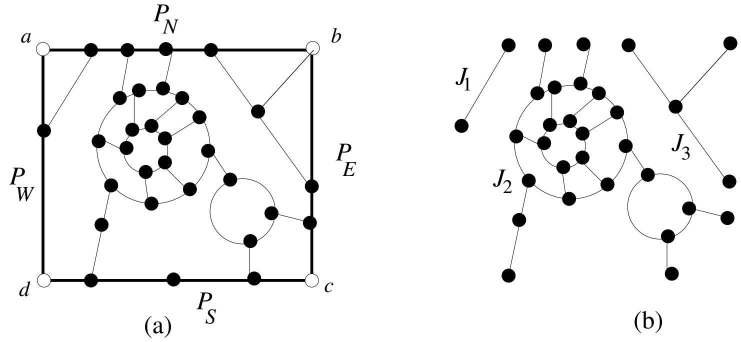


Figure 10.8 (a) Plane graph G, and (b) $C_o(G)$-components. (Figure taken from [NR04].)

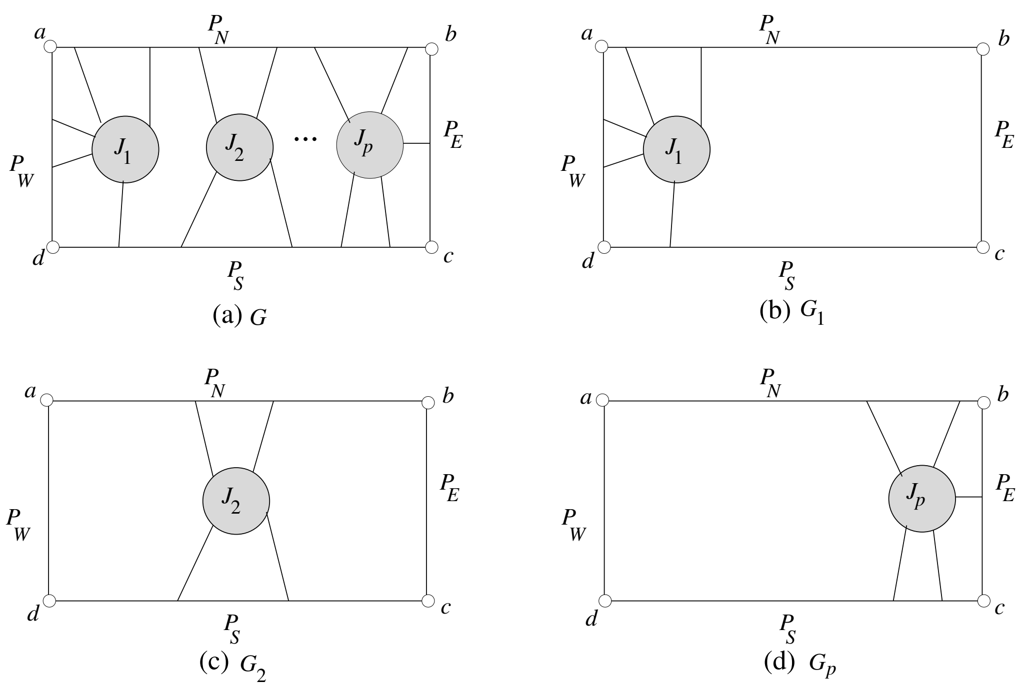


Figure 10.9 (a) G, (b) G_1, (c) G_2, and (d) G_p. (Figure taken from [NR04].)

In the remainder of this section, because of Lemma 10.1, one may assume that G has exactly one $C_o(G)$-component J.

The proof of the sufficiency of Theorem 10.2 is now outlined as follows. Assume that G has no bad cycle. We divide G into two subgraphs having no bad cycle by slicing G along one or two paths. For example, the graph G in Fig. 10.10(a) is divided into two subgraphs G_1 and G_2, each having no bad cycle, by slicing G along a path drawn by thick lines, as illustrated in Fig. 10.10(b). We then recursively find rectangular drawings of the two subgraphs as illustrated in Fig. 10.10(c), and obtain a rectangular drawing of G by patching them, as illustrated in Fig. 10.10(d). However, the problem is not so simple, because, for some graphs having no bad cycles like one in Fig. 10.11(a), there is no such path that the resulting two subgraphs have no bad cycle. For any path, one of the resulting two subgraphs has a bad 3-legged cycle C although C is not a bad cycle in G, as illustrated in Fig. 10.11(b) where a bad cycle C in a subgraph is indicated by dotted lines. For such a case, we split G into two or more subgraphs by slicing G along two paths P_c and P_{cc} having the same ends on P_N and P_S. For example, as illustrated in Fig. 10.11(c), the graph G in Fig. 10.11(a) is divided into three subgraphs G_1, G_2 and G_3, each having no bad cycle, by slicing G along path P_c indicated by dotted lines and path P_{cc} drawn by thick lines in Fig. 10.11(a). We then recursively find rectangular drawings of G_1, G_2 and G_3 as illustrated in Fig. 10.11(d), and slightly deform the drawings of G_1 and G_2, as illustrated in Fig. 10.11(e). We finally obtain a rectangular drawing of G by patching the drawings of the three subgraphs as illustrated in Fig. 10.11(f).

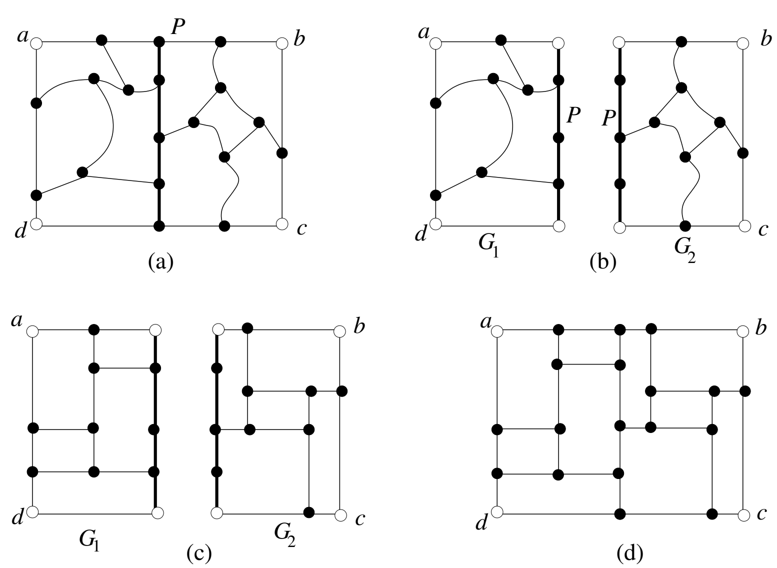


Figure 10.10 (a) G and P, (b) G_1 and G_2, (c) rectangular drawings of G_1 and G_2, and (d) rectangular drawing of G. (Figure taken from [NR04].)

We need some definitions before presenting the detail of a constructive proof. A cycle C in G like one in Fig. 10.11(a) is called "critical," because C is not a bad cycle in G but C would become a bad cycle in a subgraph obtained from G by splitting G along a path P. We now give a formal definition of a critical cycle. A cycle C in a plane graph G is *attached to a path* P if

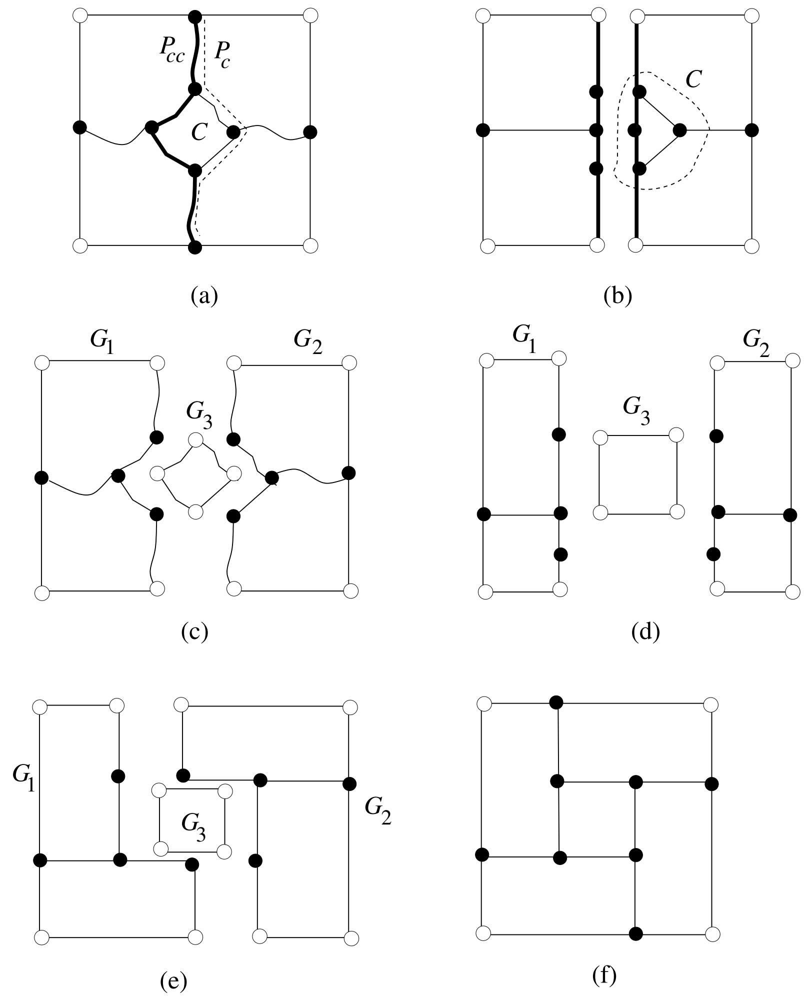


Figure 10.11 (a) G, (b) splitting G along a single path P_{cc}, (c) splitting G along two paths P_{cc} and P_c, (d) rectangular drawings of three subgraphs, (e) deformation, and (f) rectangular drawing of G. (Figure taken from [NR04].)

(i) P does not contain any vertex in the proper inside of C, and
(ii) the intersection of C and P is a single subpath of P,

as illustrated in Fig. 10.12. Let v_t be the starting vertex of the subpath, and let v_h be the ending vertex. We then call v_t the *tail vertex* of C for P, and v_h the *head vertex*. Denote by $Q_c(C)$ the path on C turning clockwise around C from v_t to v_h, and denote by $Q_{cc}(C)$ the path on C turning counterclockwise around C from v_t to v_h. A leg of C is called a *clockwise leg* for P if it is incident to a vertex in $V(Q_c(C)) - \{v_t, v_h\}$. Denote by $n_c(C)$ the number of clockwise legs of C for P. Similarly we define a *counterclockwise leg* and denote by $n_{cc}(C)$ the number of counterclockwise legs of C for P. A cycle C attached to P is called a *clockwise cycle* if $Q_{cc}(C)$ is a subpath of P, and is called a *counterclockwise cycle* if $Q_c(C)$ is a subpath of P. A cycle C is called a *critical cycle* if either C is a clockwise cycle and $n_c(C) \leq 1$ or C is a counterclockwise cycle and $n_{cc}(C) \leq 1$. Figure 10.12 illustrates a clockwise critical cycle with $n_c(C) = 1$.

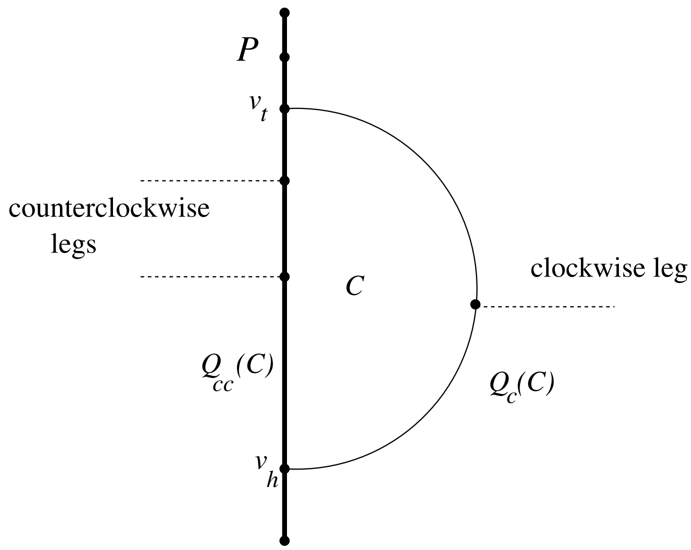


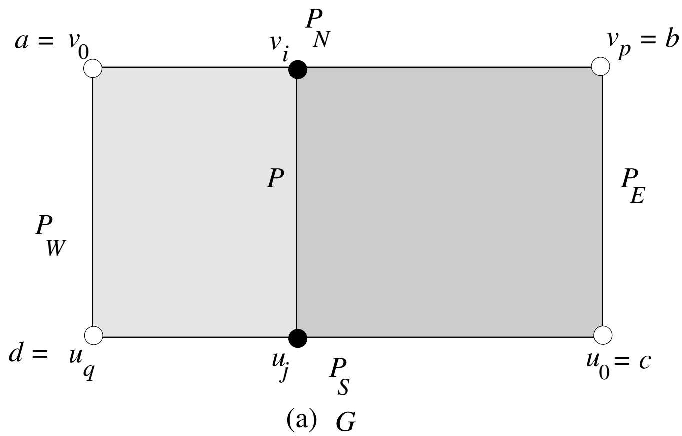
Figure 10.12 Clockwise critical cycle C attached to path P. (Figure taken from [NR04].)

We are now ready to give a constructive proof for the sufficiency of Theorem 10.2. Assume that G has no bad cycle. By Lemma 10.1, we further assume that G has exactly one $C_o(G)$-component. Let $P_N = v_0, v_1, \cdots, v_p$ where $v_0 = a$ and $v_p = b$, and let $P_S = u_0, u_1, \cdots, u_q$ where $u_0 = c$ and $u_q = d$, as illustrated in Fig. 10.13(a). An *NS-path* P is defined to be a path starting at a vertex v_i on P_N and ending at a vertex u_j on P_S without passing through any outer edge and any outer vertex other than the ends v_i and u_j. 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 has four corners a, v_i, u_j and d, and G_E^P is the east part of G including P and has four corners v_i, b, c and u_j. G_E^P and G_W^P are illustrated in Figs. 10.13(b) and (c), respectively. We say that P is an *NS-partitioning path* if neither G_W^P nor G_E^P has a bad cycle. Similarly we define a *WE-partitioning path.* If G has a partitioning path, say an NS-partitioning path P, then one can obtain a rectangular drawing of G by recursively finding rectangular drawings of G_W^P and G_E^P and patching them together along P, as illustrated in Fig. 10.10.

An inner face of G is called a *boundary face* if its contour contains at least one outer edge. A *boundary path* is a maximal path on the contour of a boundary face connecting two outer vertices without passing through any outer edge. Note that the direction of a boundary path is the same as the contour of the face, and hence is clockwise. For $X, Y \in \{N, E, S, W\}$, a *boundary XY-path* is a boundary path starting at a vertex on path P_X and ending at a vertex on path P_Y. One can easily verify the following lemma [RNN98].

LEMMA 10.2 If G has no bad cycle, then every boundary NS-, SN-, EW- or WE-path P of G is a partitioning path, that is, G can be split along P into two subgraphs, each having no bad cycle.

Thus one may assume that G has no boundary NS-, SN-, EW- or WE-paths. Then the $C_o(G)$-component J has at least one vertex on each of the paths P_N, P_E, P_S and P_W. In this case we find a pair of partitioning paths P_c and P_{cc}, and divide G into two or more subgraphs having no bad cycles by splitting G along P_c and P_{cc}. Both P_c and P_{cc} are NS-paths which have the same ends and do not cross each other in the plane although they may share several edges. Thus, if $P_c \neq P_{cc}$, then the edge set $E(P_c) \oplus E(P_{cc}) = E(P_c) \cup E(P_{cc}) - E(P_c) \cap E(P_{cc})$



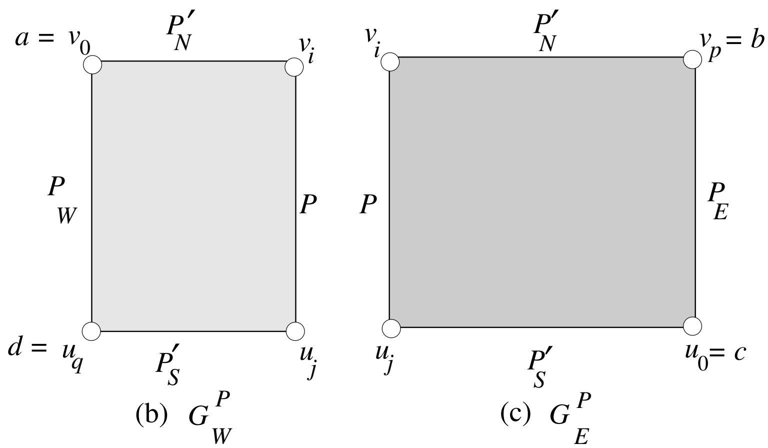


Figure 10.13 (a) Plane graph G and NS-path P, (b) G_W^P, and (c) G_E^P. (Figure taken from [NR04].)

induces vertex-disjoint cycles $C_1, C_2, \cdots, C_k$, $k \geq 1$, as illustrated in Figs. 10.14 and 10.15 where P_c and P_{cc} are indicated by dotted lines. Thus P_c and P_{cc} share $k+1$ maximal subpaths $P_1, P_2, \cdots, P_{k+1}$, as illustrated in Fig. 10.14(a). We assume that P_c turns around cycles $C_1, C_2, \cdots, C_k$ clockwise, and P_{cc} turns around them counterclockwise. We choose P_c and P_{cc} so that each cycle C_i has exactly four legs; assuming clockwise order, the first one is contained in P_i, $1 \leq i \leq k$, the second one is a clockwise leg, the third one is contained in P_{i+1} and the fourth one is a counterclockwise leg; and the four leg-vertices of C_i will be designated as the corners of the subgraph $G(C_i)$ of G inside C_i. Thus G is divided into subgraphs $G_W^{P_{cc}}$, $G_E^{P_c}$, $G(C_1), G(C_2), \cdots, G(C_k)$, as illustrated in Figs. 10.14(b) and 10.15(b). $G_W^{P_{cc}}$ has a, d and the two ends of P_{cc} as the corners, while $G_E^{P_c}$ has b, c and the two ends of P_c as the corners. $G_i(C_i)$, $1 \leq i \leq k$, has the four leg-vertices of C_i as the corners. Then the following lemma holds [RNN98].

LEMMA 10.3 Assume that a cycle C in the $C_o(G)$-component J has exactly four legs. Then the subgraph $G(C)$ of G inside C has no bad cycle when the four leg-vertices are designated as corners of $G(C)$.

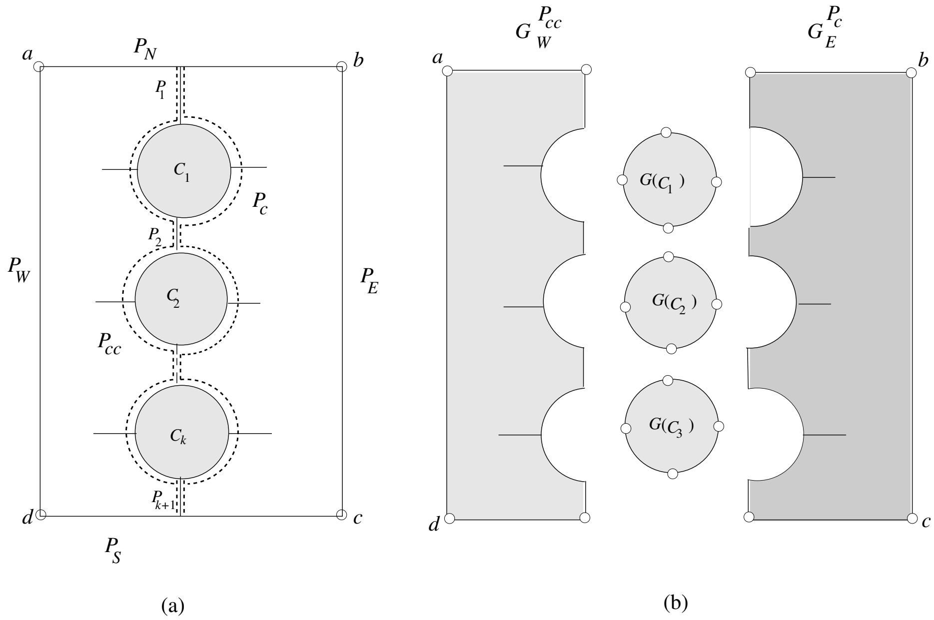


Figure 10.14 (a) G with partition-pair P_c and P_{cc}, and (b) splitting G along P_c and P_{cc}. (Figure taken from [NR04].)

By Lemma 10.3 one may assume that none of $G(C_1), G(C_2), \cdots, G(C_k)$ has a bad cycle. For each cycle C_i, $1 \leq i \leq k$, there are two alternative rectangular embeddings of C_i as illustrated in Fig. 10.16, where P'_N, P'_E, P'_S and P'_W are the four subpaths of C_i divided by the four leg-vertices. We arbitrarily choose one of them. Let G_1 be the graph obtained from $G_W^{P_{cc}}$ by contracting all edges of P_{cc} that are on the horizontal sides of rectangular embeddings of $C_1, C_2, \cdots, C_k$, as illustrated in Fig. 10.15(c). Note that every intermediate vertex on such a horizontal side has degree two in G_W^P. We denote by P'_{cc} the resulting path obtained from P_{cc} by the contraction above. Let G_1 have four corners a, d and the two ends of P_{cc}. Then one can observe that if G_1 has a rectangular drawing, in which the path P'_{cc} is drawn as a vertical straight line segment, then the rectangular drawing of G_1 can be easily modified to a drawing of $G_W^{P_{cc}}$ fitted in the area for $G_W^{P_{cc}}$ where P_{cc} is drawn as an alternating sequence of horizontal and vertical line segments, as illustrated in Figs. 10.15(b) and (c). Let G_2 be the graph obtained from $G_E^{P_c}$ by contracting all edges of P_c that are on the horizontal sides of rectangular embeddings of $C_1, C_2, \cdots, C_k$, and let P'_c be the resulting path obtained from P_c by the contraction, as illustrated in Fig. 10.15(d). Then, if G_2 has a rectangular drawing, then it can be easily modified to a drawing of $G_E^{P_c}$ fitted in the area for $G_E^{P_c}$ where P_c is drawn as an alternating sequence of horizontal and vertical line segments, as illustrated in Figs. 10.15(b) and (d). Thus if we have drawings of graphs $G_W^{P_{cc}}, G_E^{P_c}, G(C_1), G(C_2), \cdots, G(C_k)$, then we can immediately patch them to get a rectangular drawing of G. One can observe that G_1 and G_2 have no bad cycles if and only if $G_W^{P_{cc}}$ and $G_E^{P_c}$ have no bad cycles, respectively. We thus call P_c and P_{cc} a *pair of partitioning paths* or simply a *partition-pair* if neither $G_E^{P_c}$ nor $G_W^{P_{cc}}$ has a bad cycle. Especially when $P_c = P_{cc}$, it is a single partitioning path.

Thus the problem is how to prove that G has a partition-pair and to find a partition-pair efficiently. The following lemma was proved in [RNN98], and one can derive from the proof a linear algorithm to find a partition pair.

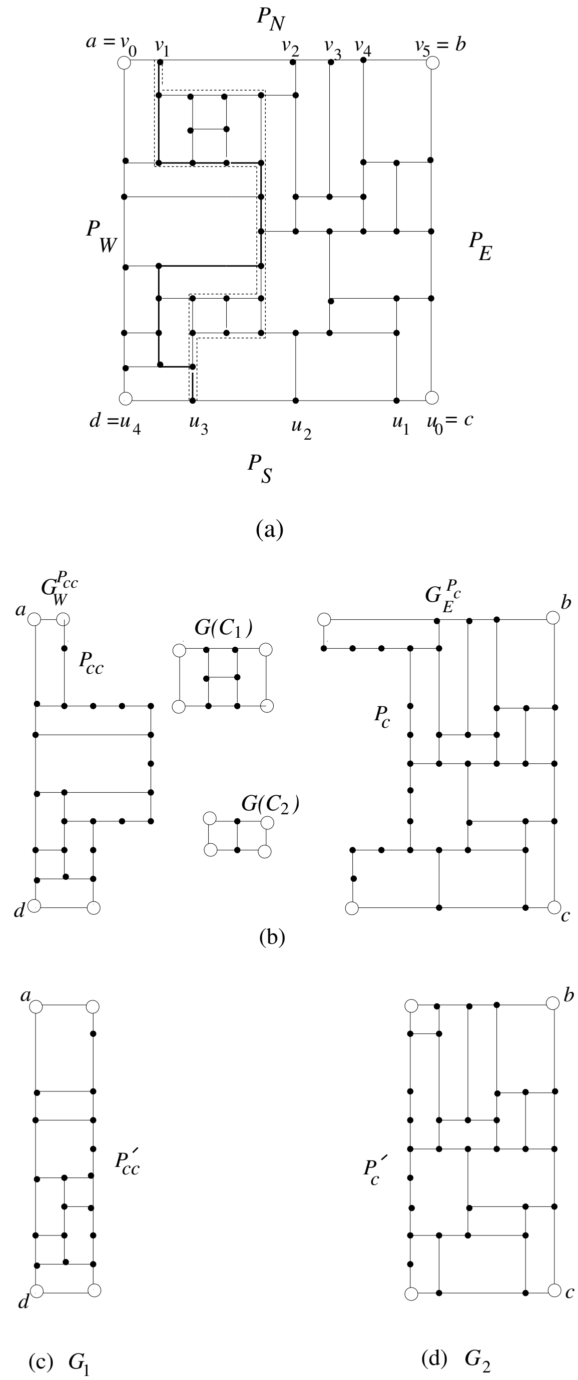


Figure 10.15 (a) G with P_c and P_{cc}, (b) splitting G along P_c and P_{cc}, (c) drawings of G_1, and (d) drawing of G_2. (Figure taken from [NR04].)

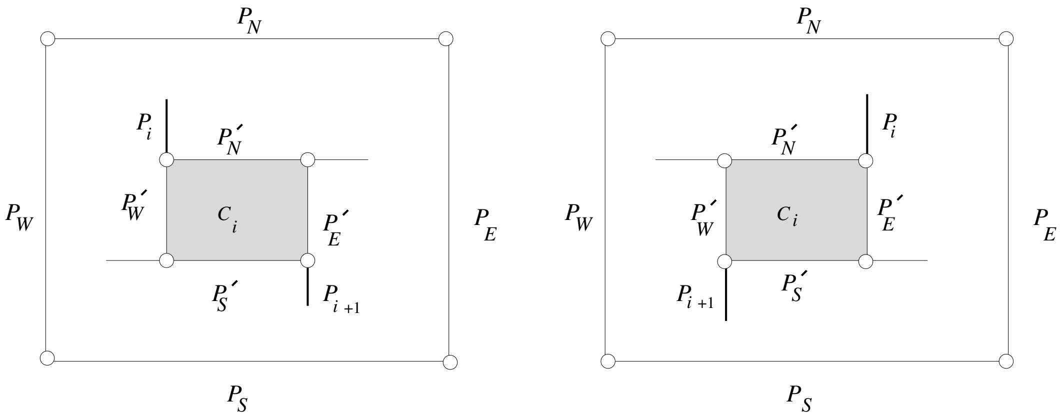


Figure 10.16 Two alternative rectangular embeddings of cycle C_i. (Figure taken from [NR04].)

LEMMA 10.4 If G has no bad cycle and has no boundary NS-, SN-, EW- or WE-path, then G has a partition-pair P_c and P_{cc}.

Using Lemmas 10.1, 10.2, 10.3 and 10.4, one can recursively find a rectangular drawing of a given plane graph G if G has no bad cycle. Thus the sufficiency of Theorem 10.2 can be constructively proved.

10.3.2 Drawing Algorithms

In this section, we assume that a given plane graph G has no bad cycle, and present an algorithm **Rectangular-Draw** to find a rectangular drawing of G. The algorithm outputs only the directions (vertical or horizontal) of edges of G. From the directions one can decide the integer coordinates of vertices as shown later in this section. It is easy to modify the algorithm so that it examines whether a given plane graph has a bad cycle or not.

We treat each $C_o(G)$-component independently as in Lemma 10.1. If there exists a boundary NS-, SN-, WE-, or EW-path, we choose it as a partitioning path. Otherwise, we find a partition-pair P_c and P_{cc} from the westmost NS-path, and then recurse to the subgraphs divided by P_c and P_{cc}.

Algorithm Rectangular-Draw(G)
begin
  Draw the outer cycle $C_o(G)$ 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 $C_o(G)$ are decided.}
  Find all $C_o(G)$-components $J_1, J_2, \cdots, J_p$; {See Fig. 10.9(a).}
  for each component J_i **do**
    begin
      $G_i = C_o(G) \cup J_i$;
      {G_i is the union of graphs $C_o(G)$ and J_i.}
      Draw(G_i, J_i)
    end
end.

Procedure Draw(G, J)

begin $\{G$ has exactly one $C_o(G)$-component $J.\}$

if G has a boundary NS-, SN-, EW-, or WE-path P

then $\{P$ is a partitioning path.$\}$

  begin {See Fig. 10.17.}

    Assume without loss of generality that P is a boundary NS-path;

    Draw all edges of P on a vertical line;

    if $|E(P)| \geq 2$ **then**

      begin

        Let $F_1, F_2, \cdots, F_q$ be the C_o-components of G_E^P;

        for each component F_i, $1 \leq i \leq q$, **do**

          Draw$(C_o(G_E^P) \cup F_i, F_i)$

      end

  end

else $\{G$ has no boundary NS-, SN-, EW-, or WE-path. $\}$

  begin

    Find a partition-pair P_c and P_{cc} as in the proof of Lemma 10.4 in [RNN98];

    if $P_c = P_{cc}$ **then** {See Fig. 10.10.}

      begin

        Draw all edges of P_c on a vertical line segment;

        Let $G_1 = G_W^{P_c}$ and $G_2 = G_E^{P_c}$ be the two resulting subgraphs;

        for each subgraph G_i, $i = 1, 2$, **do**

          begin

            Let $F_1, F_2, \cdots, F_q$ be the C_o-components of G_i;

            for each component F_j, $1 \leq j \leq q$, **do**

              Draw$(C_o(G_i) \cup F_j, F_j)$

          end

      end

    else $\{P_c \neq P_{cc}$. See Fig. 10.15.$\}$

      begin

        Draw all edges of P_c and P_{cc} on alternating sequences of horizontal and vertical line segments as in Fig. 10.15(b);

        Let G_1 be the graph obtained from $G_W^{P_{cc}}$ by contracting all edges of P_{cc} that are on horizontal sides of rectangular embeddings of $C_1, C_2, \cdots, C_k$;

        Let G_2 be the graph obtained from $G_E^{P_c}$ by contracting all edges of P_c that are on horizontal sides of rectangular embeddings of $C_1, C_2, \cdots, C_k$;

        Let $G_3 = G(C_1), G_4 = G(C_2), \cdots, G_{k+2} = G(C_k)$;

        for each graph G_i, $1 \leq i \leq k+2$, **do**

          begin

            Let $F_1, F_2, \cdots, F_q$ be the C_o-components of G_i;

            for each component F_j, $1 \leq j \leq q$, **do**

              DRAW$(C_o(G_i) \cup F_j, F_j)$

          end

      end

  end

end

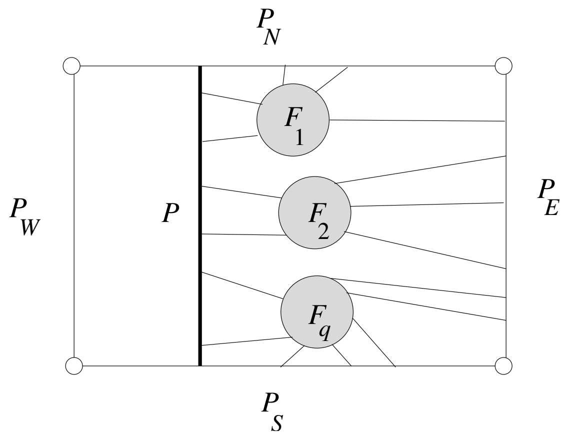


Figure 10.17 C_o-Components $F_1, F_2, \cdots, F_q$ of G_E^P. (Figure taken from [NR04].)

The algorithm **Rectangular-Draw**(G) finds only the directions of all edges in G. From the directions the integer coordinates of vertices in G can be determined in linear time as follows. From now on one may assume for simplicity that all vertices other than the four corners have degree three.

We now give a method of determining y-coordinates of the vertices in G; x-coordinates can be determined similarly. Consider a graph T_y obtained from G by deleting all upward vertical edges of three types drawn by dotted lines in Fig. 10.18. Thus any upward edge drawn by a thick line in Fig. 10.19 is not deleted. Clearly T_y is a spanning tree of G. (T_y for the graph G in Fig. 10.15(a) is drawn by thick lines in Fig. 10.20.) A rectangular drawing of G is composed of several maximal horizontal and vertical line segments. The drawing in Fig. 10.20 is composed of 16 maximal vertical line segments together with 15 maximal horizontal line segments. All these maximal horizontal line segments are contained in T_y, and every vertex of G is contained in one of them. For each maximal horizontal line segment L, we will assign an integer $y(L)$ as the y-coordinate of every vertex on L. P_S is the lowermost maximal horizontal line segment, while P_N is the topmost one. We first set $y(P_S) = 0$. We then compute $y(L)$ from bottom to top. For each vertex v in G we will assign an integer $temp(v)$ as a temporary value of the y-coordinate of v.

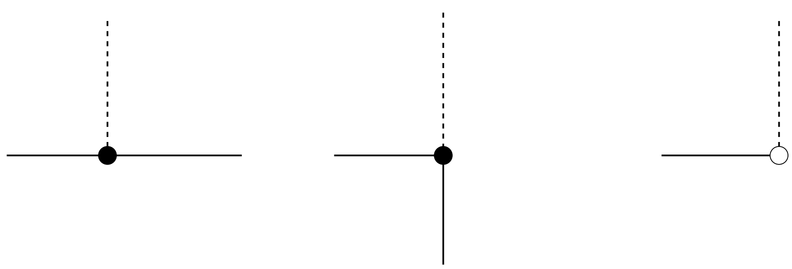


Figure 10.18 Deleted upward edges. (Figure taken from [NR04].)

For every vertex v on L there are two cases: either v has a neighbor u located below v or v has no neighbor u located below v. For the former case, we set $temp(v) = y(L') + 1$ where L' is the maximal horizontal line segment containing vertex u. For the latter case, we set $temp(v) = 0$. We then set $y(L) = \max_v\{temp(v)\}$ where the maximum is taken over

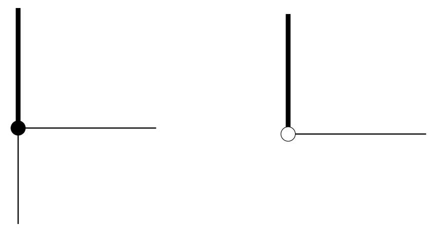


Figure 10.19 Non-deleted upward edges. (Figure taken from [NR04].)

all vertices v on L. One can easily compute $y(L)$ for all maximal horizontal line segments L from bottom to top using the counterclockwise depth-first search on T_y starting from the downward edge incident to the north-west corner a.

Thus the integer coordinates of all vertices in a rectangular grid drawing can be computed in linear time.

We now give upper bounds on the area and half perimeter of a grid for a rectangular grid drawing. Let the coordinate of the south-west corner d be $(0, 0)$, and let that of the north-east corner b be (W, H). Then the grid drawing is "compact" in a sense that there is at least one vertical line segment of x-coordinate i for each integer i, $0 \leq i \leq W$, and there is at least one horizontal line segment of y-coordinate j for each integer j, $0 \leq j \leq H$. The following theorem holds on the sizes of a compact rectangular grid drawing [RNN98].

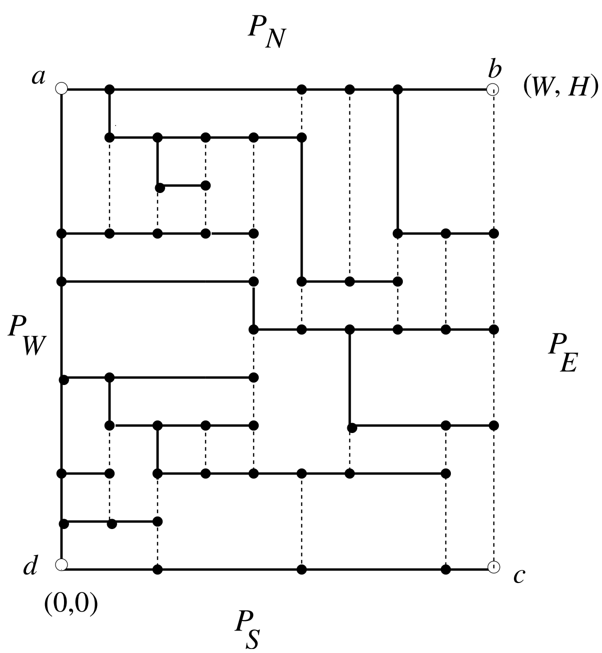


Figure 10.20 Illustration of T_y by thick lines. (Figure taken from [NR04].)

Theorem 10.3 *If all vertices of a plane graph G have degree three except the four corners, then the sizes of any compact rectangular grid drawing D of G satisfy $W + H \leq \frac{n}{2}$ and $W \cdot H \leq \frac{n^2}{16}$.*

The bounds in Theorem 10.3 are tight, because there are an infinite number of examples attaining the bounds, as one in Fig. 10.21.

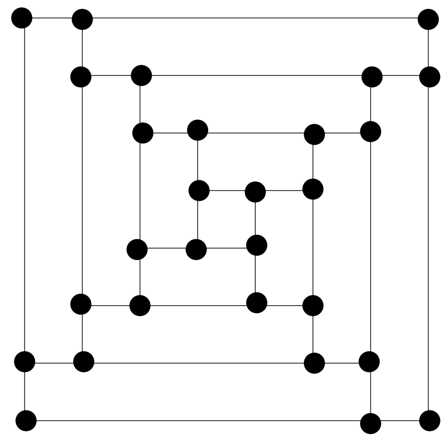


Figure 10.21 An example of a rectangular grid drawing attaining the upper bounds. (Figure taken from [NR04].)

10.3.3 Drawing without Designated Corners

In Sections 10.3.1– 10.3.2 we considered a rectangular drawing of a plane graph G with $\Delta \leq 3$ for the case where four outer vertices of degree two are designated as the corners. In this section we consider a general case where corners are not designated in advance. Then our problem is how to examine whether G has four outer vertices of degree two such that there is a rectangular drawing of G having them as the corners, and how to efficiently find them if there is.

For a cycle C in a plane graph G we denote by $G(C)$ the subgraph of G inside C. We say that cycles C and C' in a plane graph G are independent if $G(C)$ and $G(C')$ have no common vertex, and that a set $\mathcal{S}$ of cycles is independent if any pair of cycles in $\mathcal{S}$ are independent. Figure 10.22 illustrates a 2-connected plane graph G with $\Delta \leq 3$. Since every vertex of G has degree two or three, all 3-legged cycles are "laminar" in essence. Some of 2-legged and 3-legged cycles are indicated by dotted lines; C_1 and C_2 are 2-legged cycles, and C_3, C_4, C_5 and C_6 are 3-legged cycles. C_3 and C_4 are contained in $G(C_1)$, C_5 and C_6 are contained in $G(C_2)$, and C_5 is contained in $G(C_6)$. There are many independent sets of cycles. For example, $\mathcal{S}_1 = \{C_1, C_2\}$ and $\mathcal{S}_2 = \{C_2, C_3, C_4\}$ are independent sets of cycles.

We are now ready to present a necessary and sufficient condition for the existence of appropriate four outer vertices as in Theorem 10.4 [RNN02].

Theorem 10.4 *Assume that G is a 2-connected plane graph with $\Delta \leq 3$ and has four or more outer vertices of degree two. Then four of them can be designated as the corners so that G has a rectangular drawing with the designated corners if and only if G satisfies the following three conditions:*

(a) *every 2-legged cycle in G contains at least two outer vertices of degree two;*
(b) *every 3-legged cycle in G contains at least one outer vertex of degree two; and*

(c) $2c_2+c_3 \le 4$ *for every independent set* $\mathcal{S}$ *of cycles consisting of 2-legged cycles and 3-legged cycles, where* c_2 *and* c_3 *are the numbers of 2-legged cycles and 3-legged cycles in* $\mathcal{S}$, *respectively.*

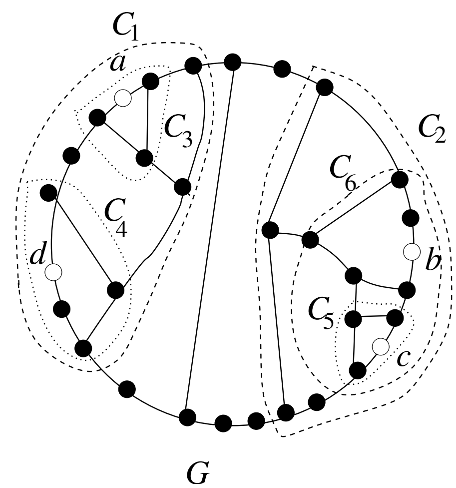


Figure 10.22 Plane graph. (Figure taken from [NR04].)

For the set $\mathcal{S}_1 = \{C_1, C_2\}$ above $c_2 = 2$, $c_3 = 0$ and hence $2c_2 + c_3 = 4$, while for $\mathcal{S}_2 = \{C_2, C_3, C_4\}$ $c_2 = 1$ and $c_3 = 2$ and hence $2c_2 + c_3 = 4$.

It is rather easy to prove the necessity of Theorem 10.4.

In order to prove the sufficiency of Theorem 10.4, it suffices to show that if the three conditions (a)–(c) in Theorem 10.4 hold then one can choose four outer vertices of degree two as the corners a, b, c and d so that the conditions (r1) and (r2) in Theorem 10.2 hold. See [RNN02] for the detail of a proof and a linear algorithm to find appropriate four outer vertices of degree two.

10.4 Box-Rectangular Drawing

A *box-rectangular drawing* of a plane graph G is a drawing of G such that each vertex is drawn as a rectangle, called a *box*, each edge is drawn as a straight line segment joining points on the two boxes corresponding to the ends, and the contour of each face is drawn as a rectangle, as illustrated in Fig. 10.24(b). A vertex may be drawn as a degenerate rectangle, that is, a point. We have seen in Section 10.1 that box-rectangular drawings have practical applications in floorplanning of MultiChip Modules (MCM) and in architectural floorplanning. If G has multiple edges or a vertex of degree five or more, then G has no rectangular drawing but may have a box-rectangular drawing. However, not every plane graph has a box-rectangular drawing. This section presents a necessary and sufficient condition for the existence of a box-rectangular drawing of a plane graph, and gives a linear algorithm to find a box-rectangular drawing if it exists [RNN00].

Before presenting the condition and the algorithm we need to present some definitions and preliminary observations regarding box-rectangular drawings.

Throughout this section we assume that a *graph* G is a so-called multigraph, which may have *multiple edges*, i.e., edges sharing both ends. If G has no multiple edges, then G is called a *simple* graph. For simplicity we assume that G has three or more vertices and is 2-connected.

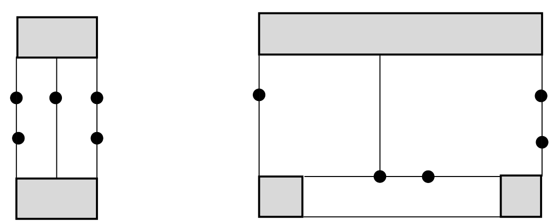
We call a box-rectangular drawing D of G a *box-rectangular grid drawing* if each edge as well as each side of a box is drawn along a grid line. A vertex may be drawn as a *degenerate box*, that is, a point, in a box-rectangular drawing D. We often call a degenerate box in D a *point* and call a non-degenerate box a *real box*. We call a rectangle corresponding to an outer cycle $C_o(G)$ the *outer rectangle*, which has exactly four corners. We call a corner of the outer rectangle simply a *corner*. A box in D is called a *corner box* if it contains at least one corner. A corner box may be degenerate.

We now have the following four facts and a lemma.

Fact 10.3 *Any box-rectangular drawing has either two, three, or four corner boxes.*

Fact 10.4 *Any corner box contains either one or two corners.*

Figure 10.23(a) illustrates a box-rectangular drawing having two corner boxes; each of them is a real box and contains two corners. Figure 10.23(b) illustrates a box-rectangular drawing having three corner boxes. Figure 10.24(b) illustrates a box-rectangular drawing having four corner boxes, one of which is degenerate.



(a) (b)

Figure 10.23 (a) Two corner boxes, and (b) three corner boxes. (Figure taken from [NR04].)

Fact 10.5 *In a box-rectangular drawing D of G, any vertex v of degree two or three satisfies one of the following* (i), (ii) *and* (iii).

(i) *Vertex v is drawn as a point containing no corner;*

(ii) *v is drawn as a corner box containing exactly one corner; and*

(iii) *v is drawn as a corner real box containing exactly two corners.*

Fact 10.6 *In any box-rectangular drawing D of G, every vertex of degree five or more is drawn as a real box.*

LEMMA 10.5 [RNN00] If G has a box-rectangular drawing, then G has a box-rectangular drawing in which every vertex of degree four or more is drawn as a real box.

The choice of vertices as corner boxes plays an important role in finding a box-rectangular drawing. For example, the graph in Fig. 10.24(a) has a box-rectangular drawing if we choose outer vertices a, b, c and d as corner boxes as illustrated in Fig. 10.24(b). However, the graph has no box-rectangular drawing if we choose outer vertices p, q, r and s as corner boxes. If all vertices corresponding to corner boxes are designated for a drawing, then it is rather easy to examine whether G has a box-rectangular drawing with the designated corner boxes. We thus first concentrate our attention to the case where all vertices of G corresponding to corner boxes are designated.

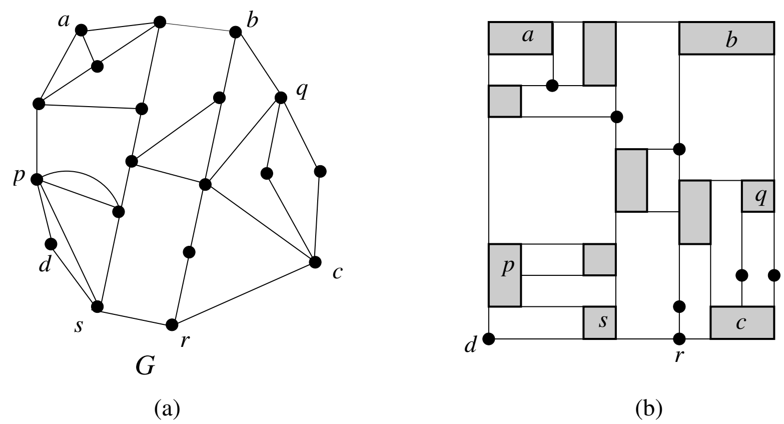


Figure 10.24 A graph G and its box-rectangular drawing with four corner boxes a, b, c and d. (Figure taken from [NR04].)

We now define two operations on a graph as follows. Let v be a vertex of degree two in G. Replace the two edges u_1v and u_2v incident to v with a single edge u_1u_2, and delete v. We call the operation above *the removal of a vertex of degree two* from G. Let v be a vertex of degree d in a plane graph, let $e_1 = vw_1, e_2 = vw_2, \cdots, e_d = vw_d$ be the edges incident to v, and assume that these edges $e_1, e_2, \cdots, e_d$ appear clockwise around v in this order as illustrated in Fig. 10.25(a). Replace v with a cycle $v_1, v_2, \cdots, v_d, v_1$, and replace edge vw_i with v_iw_i for $i = 1, 2, \cdots, d$, as illustrated in Fig. 10.25(b). We call the operation above the *replacement of a vertex by a cycle*. The cycle $v_1, v_2, \cdots, v_d, v_1$ in the resulting graph is called the *replaced cycle* corresponding to vertex v.

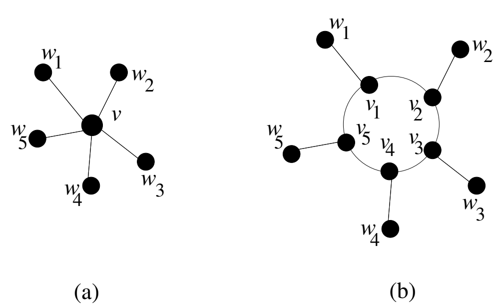


Figure 10.25 Replacement of a vertex by a cycle. (Figure taken from [NR04].)

By Fact 10.3, any box-rectangular drawing has either two, three or four corner boxes. However, we consider only box-rectangular drawings having four corner boxes for simplicity, and assume that exactly four outer vertices a, b, c and d in G are designated as the four corner boxes. We construct a new graph G'', called the *cycled graph*, from G through an intermediate graph G', and reduce the problem of finding a *box-rectangular drawing* of G with the four designated vertices to a problem of finding a *rectangular drawing* of the cycled graph G''.

We first construct G' from G as follows. If a vertex v of degree two in G, as vertex d in Fig. 10.26(a), is designated as a corner, then v must be drawn as a corner point in a box-rectangular drawing of G. On the other hand, if a vertex v of degree two is not designated as a corner, then the two edges incident to v must be drawn on a straight line segment. We thus remove all non-designated vertices of degree two one by one from G, as illustrated in Fig. 10.26(b). The resulting graph is G'. Thus all vertices of degree two in G' are designated vertices.

Clearly, G has a box-rectangular drawing with the four designated corner boxes if and only if G' has a box-rectangular drawing with the four designated corner boxes. Figure 10.26(f) illustrates a box-rectangular drawing D' of G' in Fig. 10.26(b), and Fig. 10.26(g) illustrates a box-rectangular drawing D of G in Fig. 10.26(a).

Since every vertex of degree two in G' is a designated vertex, it must be drawn as a corner point in any box-rectangular drawing of G'. Every designated vertex of degree three in G', as vertex a in Fig. 10.26(b), must be drawn as a real box since it is a corner. On the other hand, every non-designated vertex of degree three in G' must be drawn as a point. These facts together with Lemma 10.5 imply that if G' has a box-rectangular drawing then G' has a box-rectangular drawing D' in which all designated vertices of degree three and all vertices of degree four or more in G' are drawn as real boxes.

The cycled graph G'' is built from G' as follows. Replace by a cycle each of the designated vertices of degree three and the vertices of degree four or more, as illustrated in Fig. 10.26(c). The replaced cycle corresponding to a designated vertex x of degree three or more contains exactly one outer edge, say e_x, where $x = a, b, c$ or d. Put a dummy vertex x' of degree two on e_x, as shown in Fig. 10.26(d). The resulting graph is G''. We let $x' = x$ if a designated vertex x has degree two. The cycled graph G'' is a simple graph and has exactly four outer vertices a', b', c', and d' of degree two, and all the other vertices have degree three.

Then the following theorem holds.

Theorem 10.5 *Let G be a plane graph with four designated outer vertices a, b, c and d. Then G has a box-rectangular drawing with corner boxes a, b, c and d if and only if the cycled graph G'' has a rectangular drawing with designated corners a', b', c' and d'.*

Proof: The necessity is trivial, and hence it suffices to prove the sufficiency.

Assume that G'' has a rectangular drawing D'' as illustrated in Fig. 10.26(e). In D'', each replaced cycle is drawn as a rectangle, since it is a face in G''. Furthermore, the four outer vertices a', b', c' and d' of degree two in G'' are drawn as the corners of the rectangle corresponding to $C_o(G'')$. Therefore, D'' immediately gives a box-rectangular drawing D' of G' having the four vertices a, b, c and d as corner boxes, as illustrated in Fig. 10.26(f). Then, inserting non-designated vertices of degree two on horizontal or vertical line segments in D', one can immediately obtain from D' a box-rectangular drawing D of G having the designated vertices a, b, c and d as corner boxes, as illustrated in Fig. 10.26(g). □

Furthermore the following theorem holds [RNN00].

Theorem 10.6 *Given a plane graph G of m edges and four designated outer vertices a, b, c and d, one can examine in time $O(m)$ whether G has a box-rectangular drawing D with corner boxes a, b, c and d, and if G has D, then one can find D in time $O(m)$. The half perimeter of the box-rectangular grid drawing is bounded by $m + 2$.*

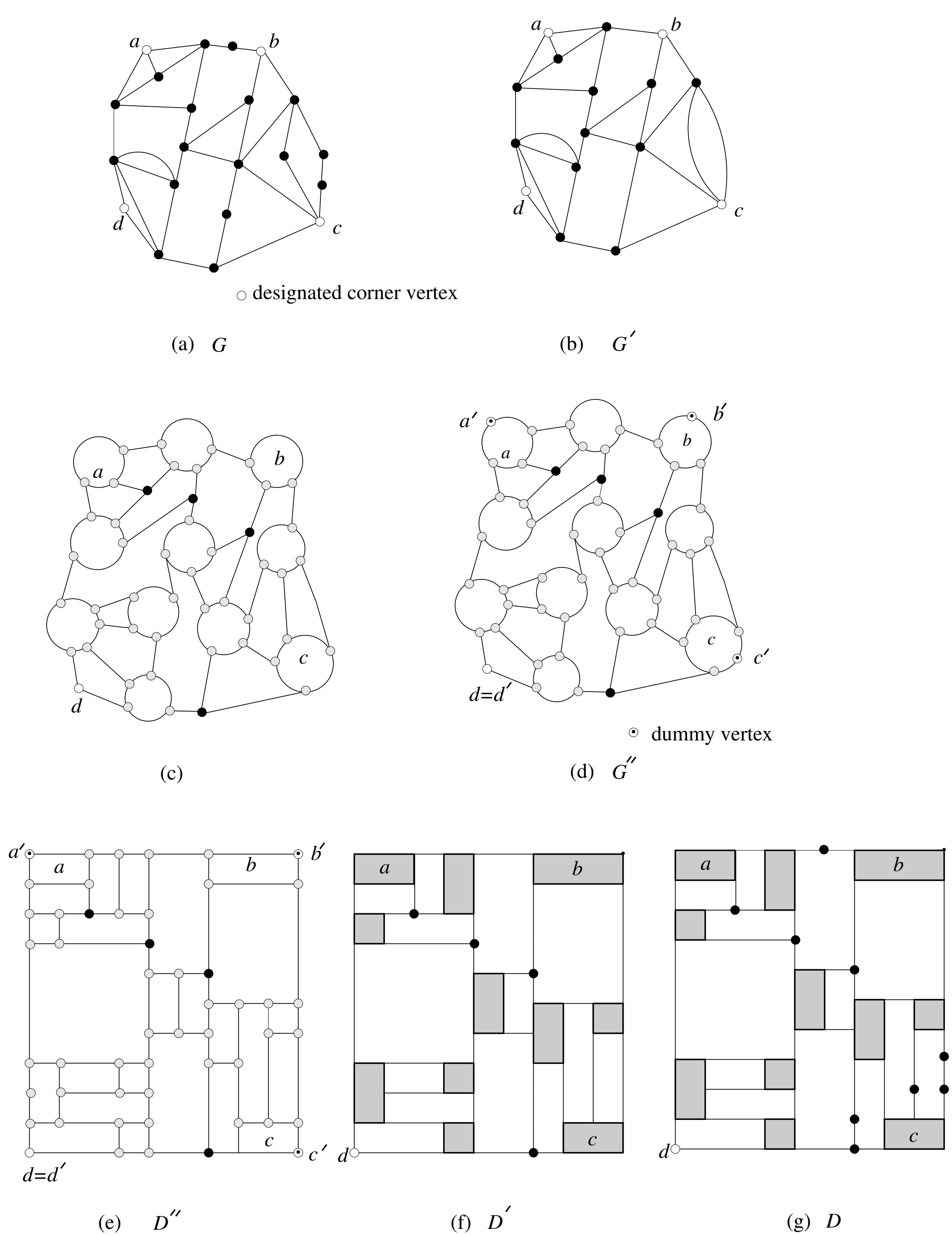


Figure 10.26 Illustration of G, G', G'', D'', D' and D. (Figure taken from [NR04].)

There are infinitely many cycles with four designated vertices for which the sum of the width and the height of any box-rectangular drawing of the cycles is $m-2$.

The rest of this section deals with a general case where no vertices are designated as corner boxes in advance. Then our problem is how to examine whether G has some set of outer vertices such that there is a box-rectangular drawing of G having them as the corner boxes, and how to find them if there are. We first present a necessary and sufficient condition for a plane graph G with $\Delta \leq 3$ to have a box-rectangular drawing D as in Theorem 10.7 [RNN00], and then give a linear-time algorithm to find D if it exists. We then reduce the box-rectangular drawing problem of a plane graph G with $\Delta \geq 4$ to that of a new plane graph J with $\Delta \leq 3$ as in Theorem 10.9.

Theorem 10.7 *A plane graph G with $\Delta \leq 3$ has a box-rectangular drawing if and only if G satisfies the following four conditions:*

(c1) *every 2-legged or 3-legged cycle in G has an outer edge;*

(c2) *at most two 2-legged cycles of G are independent of each other;*

(c3) *at most four 3-legged cycles of G are independent of each other; and*

(c4) *if G has a pair of independent 2-legged cycles C_1 and C_2, then $\{C_1, C_2, C_3\}$ is not independent for any 3-legged cycle C_3 in G, and neither $G(C_1)$ nor $G(C_2)$ has more than two independent 3-legged cycles of G.*

Then the following theorem holds.

Theorem 10.8 *Given a plane graph with $\Delta \leq 3$, one can examine in time $O(m)$ whether G has a box-rectangular drawing D or not, and if G has D, one can find D in time $O(m)$, where m is the number of edges in G.*

Proof: One can find all 2-legged and 3-legged cycles in G, as follows. We first traverse the contour of each inner face of G containing an outer edge as illustrated in Fig. 10.27, where the traversed contours of faces are indicated by dotted lines. Clearly each outer edge is traversed exactly once, and each inner edge is traversed at most twice. The inner edges traversed exactly once form cycles, called *singly traced cycles*, the insides of which have not been traversed. In Fig. 10.27 C_4, C_8 and C_9 are singly traced cycles, the insides of which are shaded. During this traversal one can easily find all 2-legged and all 3-legged cycles that contain outer edges; C_1, C_2 and C_3 drawn by thick lines in Fig. 10.27 are some of

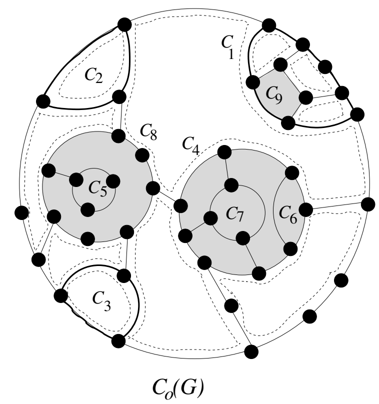


Figure 10.27 Finding all 2-legged and 3-legged cycles. (Figure taken from [NR04].)

these cycles. (Note that a 3-legged cycle containing outer edges has two legs on $C_o(G)$ and the other leg is an inner edge which is traversed twice; if an end of a doubly traversed inner edge is an inner vertex, then it is a leg-vertex of such a 3-legged cycle.) Any of the remaining 2-legged and 3-legged cycles either is a singly traced cycle or is located inside a singly traced cycle. One can find all 2-legged and 3-legged cycles inside a singly traced cycle by recursively applying the method to the singly traced cycle. The method traverses the contour of each face by a constant number of times. Hence one can examine in time $O(m)$ whether G satisfies Condition (c1) in Theorem 10.7 or not.

One can examine Condition (c2) in Theorem 10.7 as follows. Assume that G satisfies Condition (c1). Then each 2-legged cycle must have an outer edge, and hence has the two leg-vertices on $C_o(G)$. By traversing the faces of G containing an outer edge, one can detect the leg-vertices of all 2-legged cycles of G on $C_o(G)$. While detecting the leg-vertices of 2-legged cycles, we give labels to the two leg-vertices of each 2-legged cycle; the labels indicate the name of the cycle. In Fig. 10.28, the leg-vertices of 2-legged cycles are drawn by white circles, and their labels are written next to them. It is clear that if G has k 2-legged cycles which are independent of each other then G has k minimal 2-legged cycles which are independent of each other. A 2-legged cycle C is minimal if and only if no intermediate vertex of the maximal subpath of C on $C_o(G)$ is a leg-vertex of any other 2-legged cycle. Therefore, traversing the outer vertices and checking the labels of leg-vertices, one can find all minimal 2-legged cycles, and one can also know whether two 2-legged cycles are independent or not. In Fig. 10.28 C_1, C_2 and C_3 are minimal 2-legged cycles, and they are independent. Thus one can examine Condition (c2) by traversing the edges on the contours of faces containing an outer edge by a constant number of times, and hence one can examine Condition (c2) in linear time.

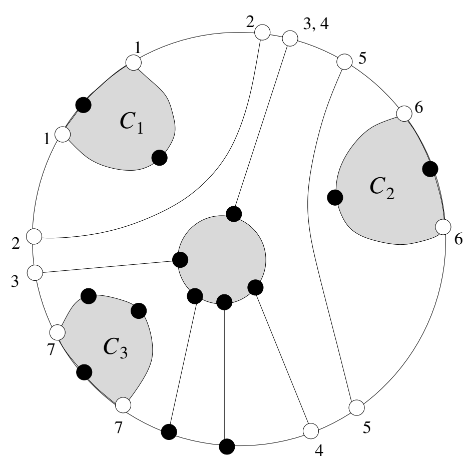


Figure 10.28 Illustration for minimal 2-legged cycles. (Figure taken from [NR04].)

One can examine Condition (c3) in linear time using a similar technique used to examine Condition (c2). One can easily examine Condition (c4) by checking the labels of the leg-vertices of minimal 2-legged cycles and minimal 3-legged cycles.

If G satisfies the conditions in Theorem 10.7, then a box-rectangular drawing of G can be found by choosing appropriate four corner boxes [RNN00, NR04]. One can find all minimal 2-legged cycles and all minimal 3-legged cycles in linear time by the technique used to examine Conditions (c2) and (c3), and hence one can choose the four designated vertices in linear time. Thus one can find a box-rectangular drawing of G in linear time. □

We now reduce the box-rectangular drawing problem (without given corners) of a plane graph G with $\Delta \geq 4$ to that of a new plane graph J with $\Delta \leq 3$. Let G be a plane graph with $\Delta \geq 4$. We construct a new plane graph J from G by replacing each vertex v of degree four or more in G by a cycle. Figures 10.29(a) and (b) illustrate G and J, respectively. A replaced cycle corresponds to a real box in a box-rectangular drawing of G. We do not replace a vertex of degree two or three by a cycle since such a vertex may be drawn as a point by Fact 10.5. Thus $\Delta(J) \leq 3$. Then the following theorem holds.

Theorem 10.9 *Let G be a plane graph with $\Delta \geq 4$, and let J be the graph transformed from G as above. Then G has a box-rectangular drawing if and only if J has a box-rectangular drawing.*

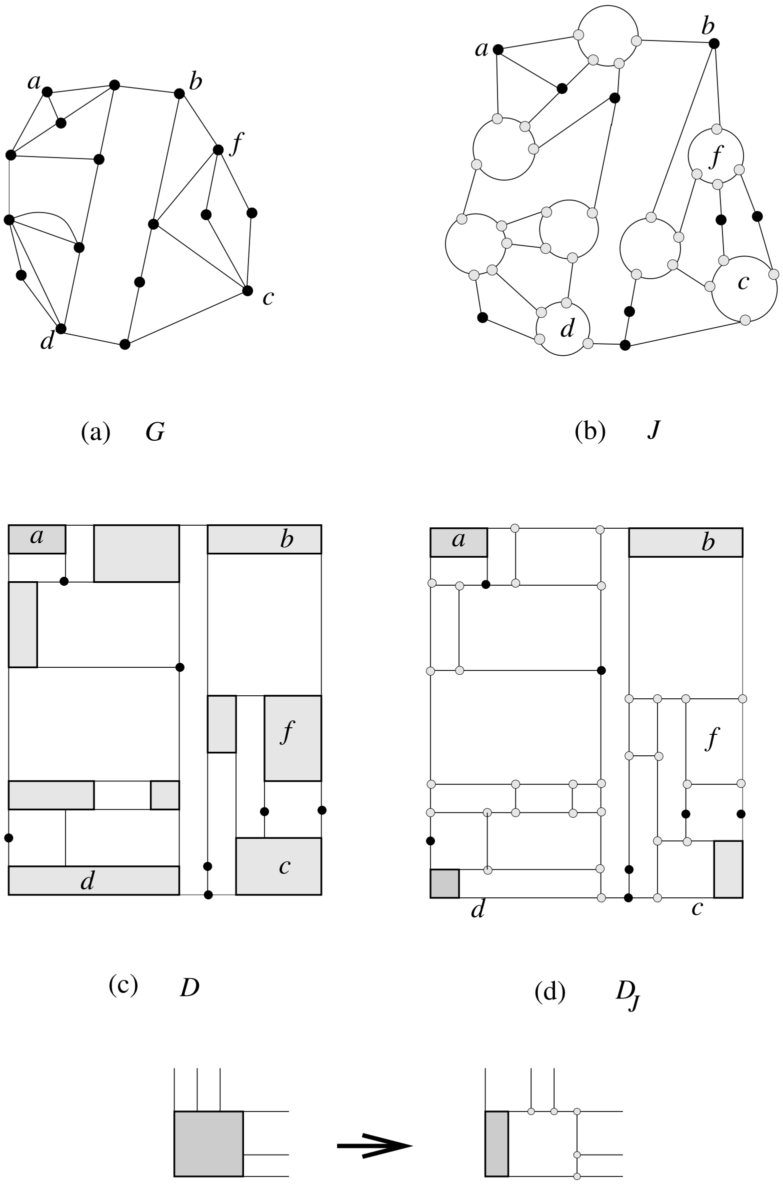
Figures 10.29(c) and (d) illustrate D and D_J, respectively. Box f in D is a non-corner real box, and it is regarded as a face in D_J. Corner boxes a and b in D are vertices of degree three in G, and they remain as boxes in D_J. Corner boxes c and d in D are vertices of degree four or more in G, and are transformed to a drawing of a replaced cycle with one real box in D_J as illustrated in Fig. 10.29(e). We omit the proof of Theorem 10.9, which can be found in [RNN00].

10.5 Conclusions

The outer face boundary must be rectangular in a rectangular drawing, as illustrated in Fig. 10.1(b). However, the outer boundary of a VLSI chip or an architectural floor plan is not always rectangular, but is often a rectilinear polygon of L-shape, T-shape, Z-shape etc., as illustrated in Figs. 10.30(a)–(c). Such a drawing of a plane graph G is called an *inner rectangular drawing* if every inner face of G is a rectangle although the outer face boundary is not always a rectangle. Miura et al. [MHN06] reduced the problem of finding an inner rectangular drawing of a plane graph G with $\Delta \leq 4$ to a problem of finding a perfect matching of a new bipartite graph constructed from G. It immediately yields the result presented in Section 10.2 on an ordinary rectangular drawing of plane graphs with $\Delta \leq 4$.

Kozminski and Kinnen [KK84] established a necessary and sufficient condition for the existence of a "rectangular dual" of an inner triangulated plane graph, that is, a rectangular drawing of the dual graph of an inner triangulated plane graph, and gave an $O(n^2)$ algorithm to obtain it. Based on the characterization of [KK84], Bhasker and Sahni [BS88] and Xin He [He93] developed linear-time algorithms to find a rectangular dual. Kant and Xin He [KH97] presented two more linear-time algorithms. Xin He [He95] presented a parallel algorithm for finding a rectangular dual. Lai and Leinwand [LL90] reduced the problem of finding a rectangular dual of an inner triangulated plane graph G to a problem of finding a perfect matching of a new bipartite graph constructed from G. Their construction is different from that in Section 10.2, their bipartite graph has an $O(n^2)$ number of edges, and hence their method takes time $O(n^{2.5})$ to find a rectangular dual or a rectangular drawing of a plane graph with $\Delta \leq 3$.

A planar graph may have many embeddings. We say that *a planar graph G has a rectangular drawing* if at least one of the plane embeddings of G has a rectangular drawing. Since a planar graph may have an exponential number of embeddings, it is not a trivial problem to examine whether a planar graph has a rectangular drawing.



(e)

Figure 10.29 Illustration of G, J, D_J, D and a transformation. (Figure taken from [NR04].)

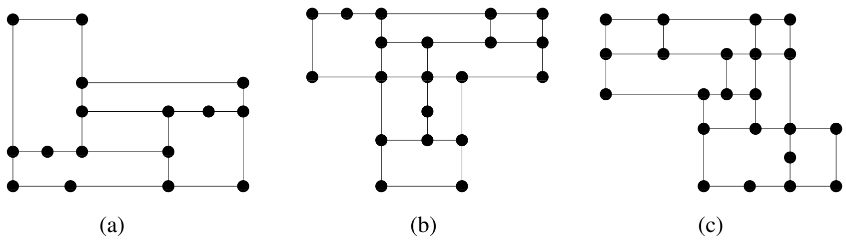


Figure 10.30 Inner rectangular drawings of (a) L-shape, (b) T-shape, (c) Z-shape.

Rahman et al. gave a linear-time algorithm to examine whether a planar graph G with $\Delta \leq 3$ has a rectangular drawing or not, and find a rectangular drawing of G if it exists [RNG04].

A similar concept of a box-rectangular drawing, called a strict 2-box drawing, is presented by Thomassen in [Tho86]. A polynomial-time algorithm can be designed for finding a strict 2-box drawing of a graph by following his method.

A box-rectangular drawing of G is called a *proper box-rectangular drawing* if every vertex of G is drawn as a real box, i.e., no vertex of G is drawn as a degenerate box. Xin He [He01] presents a necessary and sufficient condition for a plane graph G to have a proper box-rectangular drawing and gives a linear algorithm for finding a proper box-rectangular drawing of G if it exists.

In a VLSI floorplanning problem each module needs some physical area and hence each face in the drawing should satisfy some area requirements. However, when the area of each face of G is prescribed, there may not exist a rectangular drawing of G. In such a case it is desirable that each inner face of G is drawn as a rectilinear polygon of a simple shape. Recently several results have been published on rectilinear drawings of plane graphs with prescribed face areas [KN07, KN09, RMN09].

References

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

[DETT99] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph Drawing*. Prentice Hall, Upper Saddle River, NJ, 1999.

[FW74] R. L. Francis and J. A. White, *Facility Layout and Location*, Prentice-Hall, New Jersey, 1974.

[He01] X. He, A simple linear time algorithm for proper box rectangular drawings of plane graphs, *Journal of Algorithms*, 40(1), pp. 82-101, 2001.

[He93] X. He, On finding the rectangular duals of planar triangular graphs, *SIAM J. Comput.*, 22(6), pp. 1218-1226, 1993.

[He95] X. He, An efficient parallel algorithm for finding rectangular duals of plane triangulated graphs, *Algorithmica*, 13, pp. 553-572, 1995.

[HK73] J. E. Hopcroft and R. M. Karp, An $n^{5/2}$ algorithm for maximum matching in bipartite graphs, *SIAM J. Comput.*, 2, pp. 225-231, 1973.

[Hoc04] D. S. Hochbaum, Faster pseudoflow-based algorithms for the bipartite matching and the closure problems, Abstract, *CORS/SCRO-INFORMS Joint Int. Meeting*, Banff, Canada, p. 46, 2004.

[HC04] D. S. Hochbaum and B. G. Chandran, Further below the flow decomposition barrier of maximum flow for bipartite matching and maximum closure, Working paper, 2004.

[KH97] G. Kant and X. He, Regular edge labeling of 4-connected plane graphs and its applications in graph drawing problems, *Theoretical Computer Science*, 172, pp. 175-193, 1997.

[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. Design Automation Conference*, Albuquerque, pp. 655-656, 1984.

[KN07] A. Kawaguchi and H. Nagamochi, Orthogonal drawings of plane graphs with prescribed face areas, *Proc. of 4th International Conference on Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, 4484, Springer, pp. 584-594, 2007.

[KN09] A. Kawaguchi and H. Nagamochi, Drawing slicing graphs with face areas, *Theoretical Computer Science*, 410, pp. 1061-1072, 2009.

[LL90] Y.-T. Lai and S. M. Leinwand, A theory of rectangular dual graphs, *Algorithmica*, 5, pp. 467-483, 1990.

[MHN06] K. Miura, H. Haga and T. Nishizeki, Inner rectangular drawings of plane graphs, *Int. J. Computational Geometry & Applications*, 16,2 & 3, pp. 247-270, 2006.

[MV80] S. Micali and V. V. Vazirani, An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs, *Proc. Symposium on Foundations of Computer Science*, pp. 17-27, 1980.

[NR04] T. Nishizeki and M. S. Rahman, *Planar Graph Drawing*, World Scientific, Singapore, 2004.

[PS82] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization*, Prentice Hall, Englewood Cliffs, New Jersey, 1982.

[RMN09] M. S. Rahman, K. Miura and T. Nishizeki, Octagonal drawings of plane graphs with prescribed face areas, *Comp. Geom. Theo. and Appl.*, 42, pp. 214-230, 2009.

[RNG04] M. S. Rahman, T. Nishizeki, and S. Ghosh Rectangular drawings of planar graphs, *Journal of Algorithms*, 50, pp. 62-78, 2004.

[RNN98] M. S. Rahman, S. Nakano and T. Nishizeki, Rectangular grid drawings of plane graphs, *Comp. Geom. Theo. Appl.*, 10(3), pp. 203-220, 1998.

[RNN00] M. S. Rahman, S. Nakano and T. Nishizeki, Box-rectangular drawings of plane graphs, *Journal of Algorithms*, 37, pp. 363-398, 2000.

[RNN02] M. S. Rahman, S. Nakano and T. Nishizeki, Rectangular drawings of plane graphs without designated corners, *Comp. Geom. Theo. and Appl.*, 21(3), pp. 121-138, 2002.

[She95] N. Sherwani, *Algorithms for VLSI Physical Design Automation*, 2nd edition, Kluwer Academic Publishers, Boston, 1995.

[Tam87] R. Tamassia, On embedding a graph in the grid with the minimum number of bends, *SIAM J. Computing*, 16(3), pp. 421-444, 1987.

[Tho84] C. Thomassen, Plane representations of graphs, (Eds.) J. A. Bondy and U. S. R. Murty, Progress in Graph Theory, Academic Press Canada, pp. 43-69, 1984.

[Tho86] C. Thomassen, Interval representations of planar graphs, *J. of Combinat. Theory, Series B*, 40, pp. 9-20, 1986.