Journal of Graph Algorithms and Applications
http://jgaa.info/
vol. 7, no. 4, pp. 335–362 (2003)

Orthogonal Drawings of Plane Graphs Without Bends

Md. Saidur Rahman *Takao Nishizeki*
Graduate School of Information Sciences
Tohoku University
Aoba-yama 05, Sendai 980-8579, Japan
http://www.nishizeki.ecei.tohoku.ac.jp/
saidur@nishizeki.ecei.tohoku.ac.jp, nishi@ecei.tohoku.ac.jp

Mahmuda Naznin
Department of Computer Science
North Dakota State University
Fargo, ND 58105-5164, USA
Mahmuda.Naznin@ndsu.nodak.edu

Abstract

In an orthogonal drawing of a plane graph each vertex is drawn as a point and each edge is drawn as a sequence of vertical and horizontal line segments. A bend is a point at which the drawing of an edge changes its direction. Every plane graph of the maximum degree at most four has an orthogonal drawing, but may need bends. A simple necessary and sufficient condition has not been known for a plane graph to have an orthogonal drawing without bends. In this paper we obtain a necessary and sufficient condition for a plane graph G of the maximum degree three to have an orthogonal drawing without bends. We also give a linear-time algorithm to find such a drawing of G if it exists.

Communicated by: P. Mutzel and M. Jünger;
submitted May 2002; revised November 2002.

Part of this work was done while the first and the third authors were in Bangladesh University of Engineering and Technology (BUET). This work is supported by the grants of Japan Society for the Promotion of Science (JSPS).

1 Introduction

Automatic graph drawings have numerous applications in VLSI circuit layout, networks, computer architecture, circuit schematics etc. For the last few years many researchers have concentrated their attention on graph drawings and introduced a number of drawing styles. Among the styles, "orthogonal drawings" have attracted much attention due to their various applications, specially in circuit schematics, entity relationship diagrams, data flow diagrams etc. [1]. An *orthogonal drawing* of a plane graph G is a drawing of G with the given embedding in which each vertex is mapped to a point, each edge is drawn as a sequence of alternate horizontal and vertical line segments, and any two edges do not cross except at their common end. A *bend* is a point where an edge changes its direction in a drawing. Every plane graph of the maximum degree four has an orthogonal drawing, but may need bends. For the cubic plane graph in Fig. 1(a) each vertex of which has degree 3, two orthogonal drawings are shown in Figs. 1(b) and (c) with 6 and 5 bends respectively. Minimization of the number of bends in an orthogonal drawing is a challenging problem. Several works have been done on this issue [2, 3, 8, 13]. In particular, Garg and Tamassia [3] presented an algorithm to find an orthogonal drawing of a given plane graph G with the minimum number of bends in time $O(n^{7/4}\sqrt{\log n})$, where n is the number of vertices in G. Rahman *et al.* gave an algorithm to find an orthogonal drawing of a given triconnected cubic plane graph with the minimum number of bends in linear time [8].

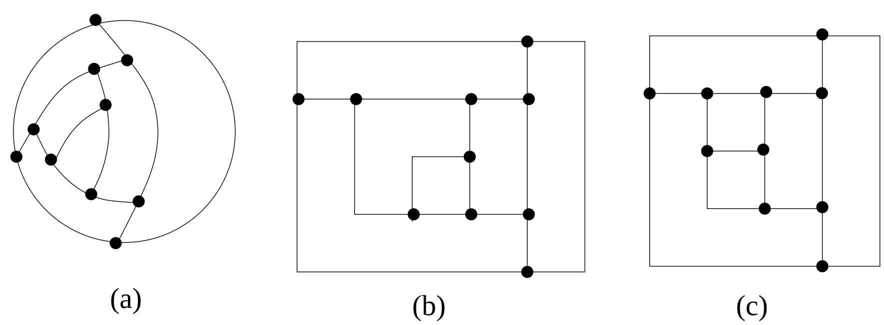

Figure 1: (a) A plane graph G, (b) an orthogonal drawing of G with 6 bends, and (c) an orthogonal drawing of G with 5 bends.

In a VLSI floorplanning problem, an input is often a plane graph of the maximum degree 3 [4, 9, 10]. Such a plane graph G may have an orthogonal drawing without bends. The graph in Fig. 2(a) has an orthogonal drawing without bends as shown in Fig. 2(b). However, not every plane graph of the maximum degree 3 has an orthogonal drawing without bends. For example, the cubic plane graph in Fig. 1(a) has no orthogonal drawing without bends, since any orthogonal drawing of an outer cycle have at least four convex corners which must be bends in a cubic graph. One may thus assume that there are four or more vertices of degree two on the outer cycle of G. It is interesting to know which classes of such plane graphs have orthogonal drawings without bends. However, no simple necessary and sufficient condition has been known for a plane graph to have an orthogonal drawing without bends, although one can know in time $O(n^{7/4}\sqrt{\log n})$ by the algorithm [3] whether a given plane graph has an orthogonal drawing without bends.

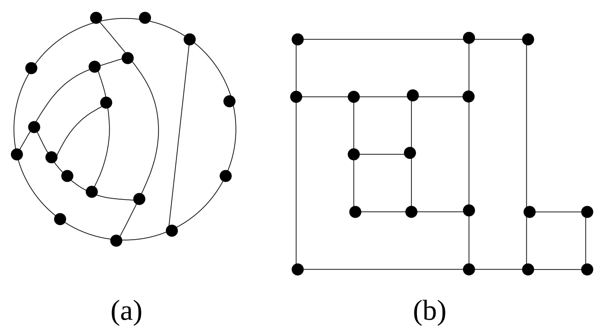

Figure 2: (a) A plane graph G and (b) an orthogonal drawing of G without bends.

In this paper we obtain a simple necessary and sufficient condition for a plane graph G of the maximum degree 3 to have an orthogonal drawing without bends. The condition is a generalization of Thomassen's condition for the existence of "rectangular drawings" [12]. Our condition leads to a linear-time algorithm to find an orthogonal drawing of G without bends if it exists.

The rest of paper is organized as follows. Section 2 describes some definitions and presents known results. Section 3 presents our results on orthogonal drawings of biconnected plane graphs without bends. Section 4 deals with orthogonal drawings of arbitrary (not always biconnected) plane graphs without bends. Finally Section 5 gives the conclusion. A preliminary version of this paper is presented in [11].

2 Preliminaries

In this section we give some definitions and preliminary known results.

Let G be a connected simple graph with n vertices and m edges. The *degree* of a vertex v is the number of neighbors of v in G. A vertex of degree 2 in G is called a *2-vertex* of G. We denote the maximum degree of graph G by $\Delta(G)$ or simply by Δ. The *connectivity* $\kappa(G)$ of a graph G is the minimum number of vertices whose removal results in a disconnected graph or a single vertex graph. We say that G is *k-connected* if $\kappa(G) \geq k$. We call a vertex of G a *cut vertex* if its removal results in a disconnected graph.

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* G is a planar graph with a fixed planar embedding. A plane graph G divides the plane into connected regions called *faces.* We refer the *contour* of a face as a cycle formed by the edges on the boundary of the face. We denote the contour of the outer face of G by $C_o(G)$.

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.

An *orthogonal drawing* of a plane graph G is a drawing of G with the given embedding in which each vertex is mapped to a point, each edge is drawn as a sequence of alternate horizontal and vertical line segments, and any two edges do not cross except at their common end. A *bend* is a point where an edge changes its direction in a drawing. Any cycle C in G is drawn as a rectilinear polygon in an orthogonal drawing $D(G)$ of G. The polygon is denoted by $D(C)$. A (polygonal) vertex of the rectilinear polygon is called a *corner of the drawing* $D(C)$. A corner has an interior angle 90° or 270°. A corner of an interior angle 90° is called a *convex corner* of $D(C)$, while a corner of an interior angle 270° is called a *concave corner.* A vertex v on C is called a *non-corner* of $D(C)$ if v is not a corner of $D(C)$. Thus any vertex on C is a convex corner, a concave corner, or a non-corner of $D(C)$.

A *rectangular drawing* of a plane biconnected graph G is a drawing of G such that each edge is drawn as a horizontal or a vertical line segment, and each face is drawn as a rectangle. (See Fig. 9.) Thus a rectangular drawing is an orthogonal drawing in which there is no bends and each face is drawn as a rectangle. The rectangular drawing of $C_o(G)$ is called the *outer rectangle.* The following result is known on rectangular drawings.

Lemma 1 *Assume that G is a plane biconnected graph with $\Delta \leq 3$, and that four 2-vertices on $C_o(G)$ are designated as the four (convex) corners of the outer rectangle. Then G has a rectangular drawing if and only if G satisfies the following two conditions [12]:*

(r1) every 2-legged cycle contains at least two designated vertices, and

(r2) every 3-legged cycle contains at least one designated vertex.

Furthermore one can examine in linear time whether G satisfies the condition above, and if G does then one can find a rectangular drawing in linear time [7].

Consider two examples in Fig. 3, where the four designated corner vertices are drawn by white circles in each graph. Cycles C_1, C_2 and C_3 are 2-legged, and C_4, C_5 and C_6 are 3-legged. C_3, C_5 and C_6 do not violate the conditions in Lemma 1. On the other hand, cycles C_1, C_2 and C_4 violate the conditions.

A cycle in G violating (r1) or (r2) is called a *bad cycle*: a 2-legged cycle is *bad* if it contains at most one designated vertex; a 3-legged cycle is *bad* if it contains no designated vertex.

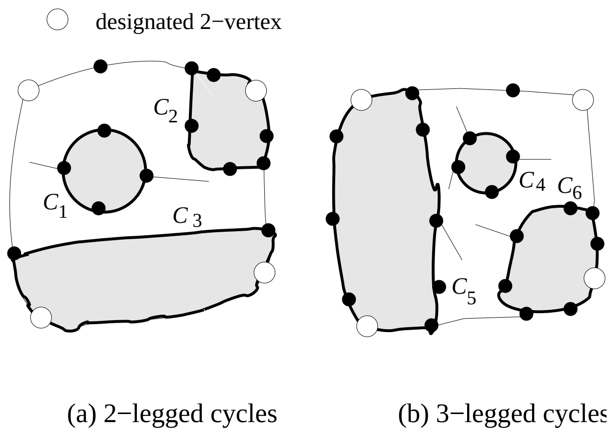

Figure 3: (a) 2-legged cycles C_1, C_2 and C_3, and (b) 3-legged cycles C_4, C_5 and C_6.

Rahman *et al.* [7] have obtained a linear-time algorithm to find a rectangular drawing of a plane graph G if G satisfies the conditions in Lemma 1 for four designated corner vertices on $C_o(G)$. We call it Algorithm **Rectangular-Draw** and use it in our orthogonal drawing algorithm of this paper.

For a cycle C in a plane graph G, we denote by $G(C)$ the plane subgraph of G inside C (including C). A bad cycle C in G is called a *maximal bad cycle* if $G(C)$ is not contained in $G(C')$ for any other bad cycle C' of G. In Fig. 4 C_1, C_3, C_4, C_5 and C_6 are bad cycles, but C_2 is not a bad cycle, where C_2 and C_4 are drawn by thick lines. C_1, C_4, C_5 and C_6 are the maximal bad cycles. C_3 is not a maximal bad cycle because $G(C_3)$ is contained in $G(C_4)$ for a bad cycle C_4. We say that cycles C and C' in a plane graph G are *independent* of each other if $G(C)$ and $G(C')$ have no common vertex. We now have the following lemma.

Lemma 2 *If G is a biconnected plane graph of $\Delta \leq 3$ and four 2-vertices on $C_o(G)$ are designated as corners, then the maximal bad cycles in G are independent of each other.*

Proof: Assume for a contradiction that a pair of maximal bad cycles C_1 and C_2 in G are not independent. Then the subgraphs $G(C_1)$ and $G(C_2)$ have a common vertex. In particular, the cycles C_1 and C_2 have a common vertex, because C_1 and C_2 are maximal bad cycles. Since $\Delta \leq 3$, C_1 and C_2 share a common edge; C_1 contains two legs of C_2, and C_2 contains two legs of C_1. There are two cases to consider.

Case 1: C_1 and C_2 have a common vertex not on $C_o(G)$.

There are three cases; (i) both C_1 and C_2 are 2-legged cycles, (ii) one of C_1 and C_2 is a 2-legged cycle and the other is a 3-legged cycle, and (iii) both

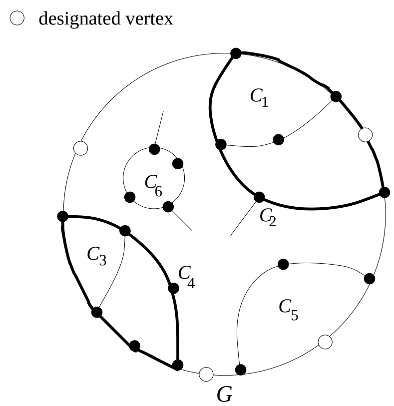

Figure 4: Maximal bad cycles C_1, C_4, C_5 and C_6.

C_1 and C_2 are 3-legged cycles. If both C_1 and C_2 are 2-legged cycles, then G would be a disconnected graph as illustrated in Fig. 5(a), a contradiction to the assumption that G is biconnected. If one of C_1 and C_2 is a 2-legged cycle and the other is a 3-legged cycle, then G would have a cut-vertex v as illustrated in Fig. 5(b), a contradiction to the assumption that G is biconnected. If both C_1 and C_2 are 3-legged cycles, then there would exist a 2-legged bad cycle C^* in G such that $G(C^*)$ contains both C_1 and C_2, a contradiction to the assumption that C_1 and C_2 are maximal bad cycles. In Fig. 5(c) C^* is drawn by thick lines.
Case 2: C_1 and C_2 have a common vertex on $C_o(G)$.

If both C_1 and C_2 are 2-legged cycles, then one of $G(C_1)$ and $G(C_2)$ would be contained in the other as illustrated in Fig. 5(d), a contradiction to the assumption that both C_1 and C_2 are maximal bad cycles. If one of C_1 and C_2 is a 2-legged cycle and the other is a 3-legged cycle, then one of $G(C_1)$ and $G(C_2)$ would be contained in the other, as illustrated in Fig. 5(e) and Fig. 5(f), contrary to the assumption. If both C_1 and C_2 are 3-legged cycles, then they have no designated vertex and there would exist a bad 2-legged cycle C^* such that $G(C^*)$ contains both of C_1 and C_2, a contradiction to the assumption. In Fig. 5(g) C^* is drawn by thick lines. □

3 Orthogonal Drawings of Biconnected Plane Graphs

In this section we present our results on biconnected plane graphs. From now on we assume that G is a biconnected plane graph with $\Delta \leq 3$ and there are four or more 2-vertices on $C_o(G)$. The following theorem is the main result of this section.

Theorem 1 *Assume that G is a plane biconnected graph with $\Delta \leq 3$ and there are four or more 2-vertices of G on $C_o(G)$. Then G has an orthogonal drawing*

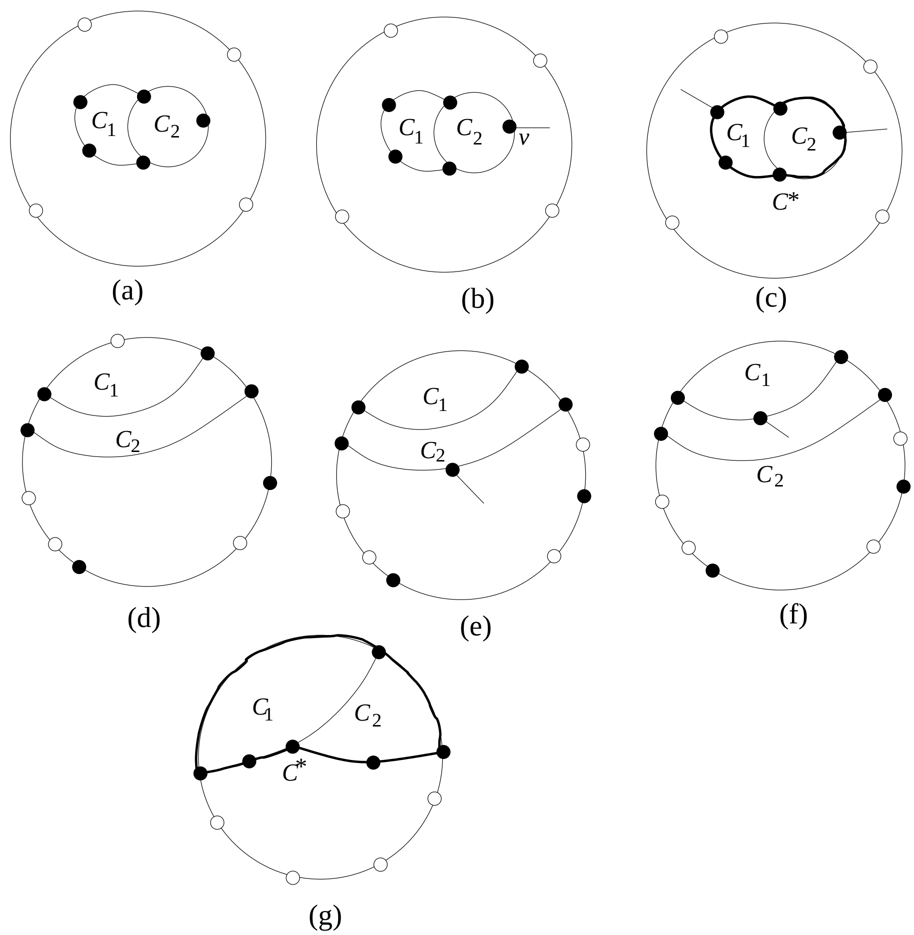

Figure 5: Illustration for the proof of Lemma 2.

without bends if and only if every 2-legged cycle contains at least two 2-vertices of G and every 3-legged cycle contains at least one 2-vertex of G.

Note that Theorem 1 is a generalization of Thomassen's condition for rectangular drawings in Lemma 1; applying Theorem 1 to a plane biconnected graph G in which all vertices have degree 3 except the four 2-vertices on $C_o(G)$, one can derive the condition.

It is easy to prove the necessity of Theorem 1, as follows.

Necessity of Theorem 1 Assume that a plane biconnected graph G has an orthogonal drawing D without bends.

Let C be any 2-legged cycle. Then the rectilinear polygon $D(C)$ in D has at least four convex corners. These convex corners must be vertices since D has no bends. The two leg-vertices of C may serve as two of the convex corners. However, each of the other convex corners must be a 2-vertex of G. Thus C must contain at least two 2-vertices of G.

One can similarly show that any 3-legged cycle C in G contains at least one 2-vertex of G. □

In the rest of this section we give a constructive proof for the sufficiency of Theorem 1 and show that the proof leads to a linear-time algorithm to find an orthogonal drawing without bends if it exists.

Assume that G satisfies the condition in Theorem 1. We now need some definitions. Let C be a 2-legged cycle in G, and let x and y be the two leg-vertices of C. We say that an orthogonal drawing $D(G(C))$ of the subgraph $G(C)$ is *feasible* if $D(G(C))$ has no bend and satisfies the following condition (f1) or (f2).

(f1) The drawing $D(G(C))$ intersects neither the first quadrant with the origin at x nor the third quadrant with the origin at y after rotating the drawing and renaming the leg-vertices if necessary, as illustrated in Fig. 6. Note that C is not always drawn by a rectangle.

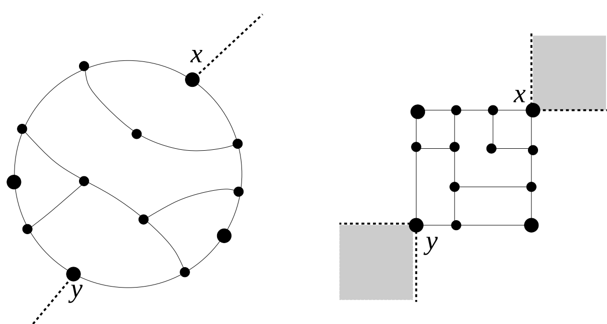

Figure 6: Illustration of (f1) for a 2-legged cycle.

(f2) The drawing $D(G(C))$ intersects neither the first quadrant with the origin at x nor the fourth quadrant with the origin at y after rotating the drawing and renaming the leg-vertices if necessary, as illustrated in Fig. 7.

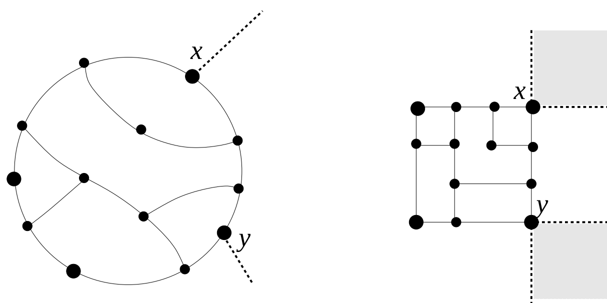

Figure 7: Illustration of (f2) for a 2-legged cycle.

Let C be a 3-legged cycle in G, and let x, y and z be the three leg-vertices. One may assume that x, y and z appear clockwise on C in this order. We say that an orthogonal drawing $D(G(C))$ is *feasible* if $D(G(C))$ has no bend and $D(G(C))$ satisfies the following condition (f3).

(f3) The drawing $D(G(C))$ intersects none of the following three quadrants: the first quadrant with origin at x, the fourth quadrant with origin at y, and the third quadrant with origin at z after rotating the drawing and renaming the leg-vertices if necessary, as illustrated in Fig. 8.

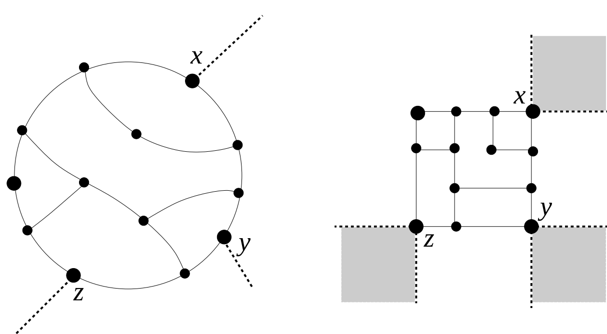

Figure 8: Illustration of (f3) for a 3-legged cycle.

Each of Conditions (f1), (f2) and (f3) implies that, in the drawing of $G(C)$, any vertex of $G(C)$ except the leg-vertices is located in none of the shaded quadrants in Figs. 6, 7 and 8, and hence a leg incident to x, y or z can be drawn by a horizontal or vertical line segment without edge-crossing as indicated by dotted lines in Figs. 6, 7 and 8.

We now have the following lemma.

Lemma 3 *If G satisfies the condition in Theorem 1, that is, every 2-legged cycle in G contains at least two 2-vertices of G and every 3-legged cycle in G contains at least one 2-vertex of G, then $G(C)$ has a feasible orthogonal drawing for any 2- or 3-legged cycle C in G.*

Proof: We give a recursive algorithm to find a feasible orthogonal drawing of $G(C)$. There are two cases to be considered.

Case 1: C is a 2-legged cycle.

Let x and y be the two leg-vertices of C, and let e_x and e_y be the legs incident to x and y, respectively. Since C satisfies the condition in Theorem 1, C has at least two 2-vertices of G. Let a and b be any two 2-vertices of G on C. We now regard the four vertices x, y, a and b as the four designated corner vertices of C.

We first consider the case where $G(C)$ has no bad cycle with respect to the four designated vertices. In this case, by Lemma 1 $G(C)$ has a rectangular drawing D with the four designated corner vertices, as illustrated in Fig. 9. Such a rectangular drawing D of $G(C)$ can be found by the algorithm **Rectangular-Draw** in [7]. The outer cycle C of $G(C)$ is drawn as a rectangle in D, and x, y, a and b are the convex corners of the rectangle. Hence D satisfies Condition (f1) or (f2). Since D is a rectangular drawing, D has no bend. Thus D is a feasible orthogonal drawing of $G(C)$.

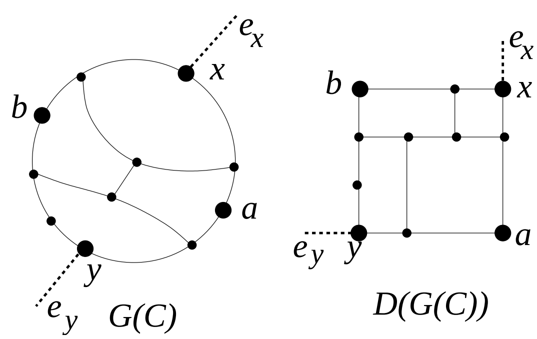

Figure 9: Subgraph $G(C)$ and its rectangular drawing $D(G(C))$.

We then consider the case where $G(C)$ has a bad cycle. Let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles of $G(C)$. By Lemma 2 $C_1, C_2, \cdots, C_l$ are independent of each other. Construct a plane graph H from $G(C)$ by contracting each subgraph $G(C_i), 1 \leq i \leq l$, to a single vertex v_i, as illustrated in Figs. 10(a) and (b). Clearly H is a plane biconnected graph with $\Delta \leq 3$. Every bad cycle C_i in $G(C)$ contains at most one designated vertex. If C_i contains a designated vertex, then we newly designate v_i as a corner vertex of H in place of the designated vertex. Thus H has exactly four designated vertices. (In Fig. 10 H has four designated vertices a, b, x, and v_2 since the bad cycle C_2 contains y.) Since all maximal bad cycles are contracted to single vertices in H, H has no bad cycle with respect to the four designated vertices, and hence by Lemma 1 H has a rectangular drawing $D(H)$, as illustrated in Fig. 10(c). Such a drawing $D(H)$ can be found by Algorithm **Rectangular-Draw**. Clearly there is no bend on $D(H)$. The shrunken outer cycle of $G(C)$ is drawn as a rectangle in $D(H)$, and hence $D(H)$ satisfies Conditions (f1) or (f2). If C_i is a 2-legged cycle, then the two legs e_{x_i}, e_{y_i} and vertex v_i are embedded in $D(H)$ as illustrated in Figs. 11(b) and 12(b) or as in their rotated ones, and the two legs e_{x_i}, e_{y_i} and C_i can be drawn as illustrated in Figs. 11(c) and 12(c) or as in their rotated ones for a feasible orthogonal drawing $D(G(C_i))$ of $G(C_i)$. If C_i is a 3-legged cycle, then v_i and the three legs e_{x_i}, e_{y_i} and e_{z_i} are embedded in $D(H)$ as illustrated in Fig. 13(b) or as in their rotated ones, and C_i and three legs e_{x_i}, e_{y_i} and e_{z_i} can be drawn as illustrated in Fig. 13(c) or as in their rotated ones for a feasible orthogonal drawing $D(G(C_i))$ of $G(C_i)$. One can obtain a drawing $D(G(C))$ of $G(C)$ from $D(H)$ and $D(G(C_i))$ $1 \leq i \leq l$, as follows. Replace each v_i, $1 \leq i \leq l$, in $D(H)$ with one of the feasible drawings of $G(C_i)$ in Fig. 11(c), Fig. 12(c) and Fig. 13(c) and their rotated ones that corresponds to the embedding of v_i and the legs of C_i in $D(H)$, and draw each leg of C_i in $D(G(C))$ by a straight line segment having the same direction as the leg in $D(H)$, as illustrated in Fig. 10(d). We call this operation a *patching operation*.

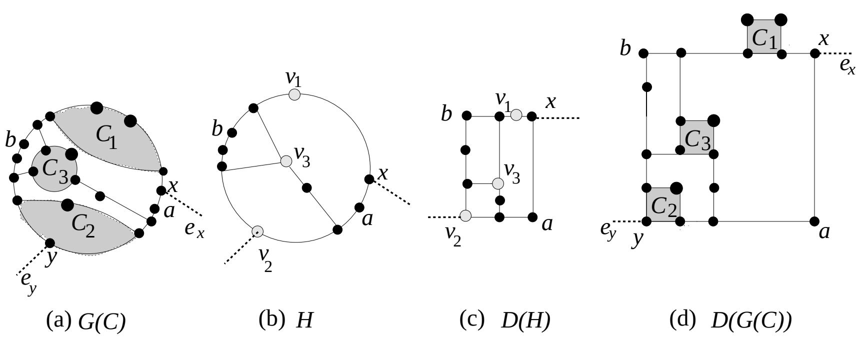

Figure 10: Illustration for Case 1 where C has the maximal bad cycles C_1, C_2 and C_3.

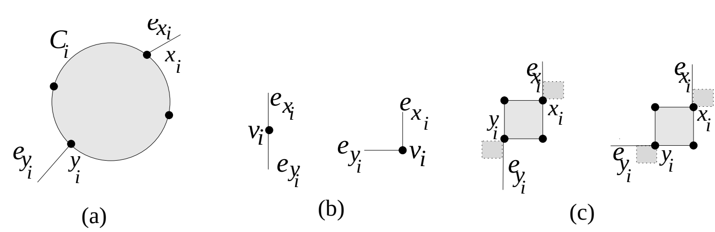

Figure 11: (a) A 2-legged cycle C_i having a feasible orthogonal drawing satisfying (f1), (b) embeddings of a vertex v_i and two legs e_{x_i} and e_{y_i} incident to v_i, and (c) feasible orthogonal drawings of $G(C_i)$ with two legs.

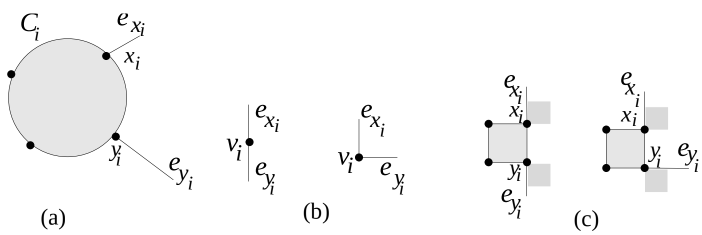

Figure 12: (a) A 2-legged cycle C_i having a feasible orthogonal drawing satisfying (f2), (b) embeddings of a vertex v_i and two legs e_{x_i} and e_{y_i} incident to v_i, and (c) feasible orthogonal drawings of $G(C_i)$ with two legs.

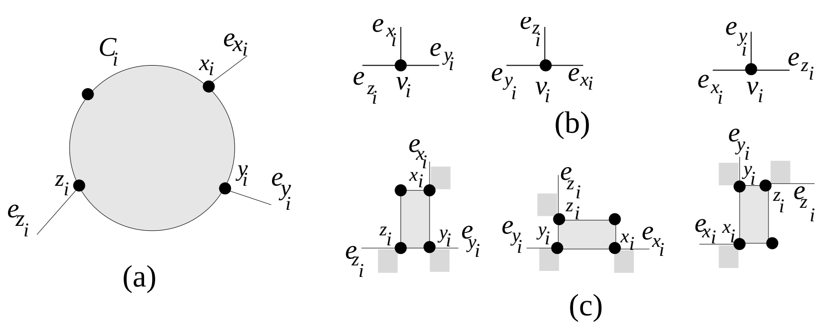

Figure 13: (a) A 3-legged cycle C_i having feasible orthogonal drawings satisfying (f3), (b) embeddings of a vertex v_i and three legs e_{x_i}, e_{y_i} and e_{z_i} incident to v_i, and (c) feasible orthogonal drawings of $G(C_i)$ with three legs.

We find a feasible orthogonal drawing $D(G(C_i))$ of $G(C_i), 1 \le i \le l$, in a recursive manner. We then patch the drawings $D(G(C_1)), D(G(C_2)), \cdots, D(G(C_l))$ into $D(H)$ by patching operation. Since there is no bend in any of $D(G(C_1))$, $D(G(C_2)), \cdots, D(G(C_l))$, there is no bend in the resulting drawing $D(G(C))$. Since the outer cycle of $D(H)$ is a rectangle and the resulting drawing $D(G(C))$ always expands outwards, $D(C)$ is not always a rectangle but $D(G(C))$ satisfies (f1) or (f2). Hence $D(G(C))$ is a feasible orthogonal drawing.

Case 2: C is a 3-legged cycle.

Let x, y and z be the three leg-vertices of C, and let e_x, e_y and e_z be the legs incident to x, y and z, respectively. Since C satisfies the condition in Theorem 1, C has at least one 2-vertex of G. Let a be any 2-vertex of G on C. We now regard the four vertices x, y, z and a as designated corner vertices.

We first consider the case where $G(C)$ has no bad cycle with respect to the four designated vertices. In this case by Lemma 1 $G(C)$ has a rectangular drawing D with the four designated vertices as illustrated in Fig. 14. Since the outer cycle C of $G(C)$ is drawn as a rectangle in D, D satisfies Condition (f3). Since D is a rectangular drawing, D has no bend. Thus D is a feasible orthogonal drawing of $G(C)$.

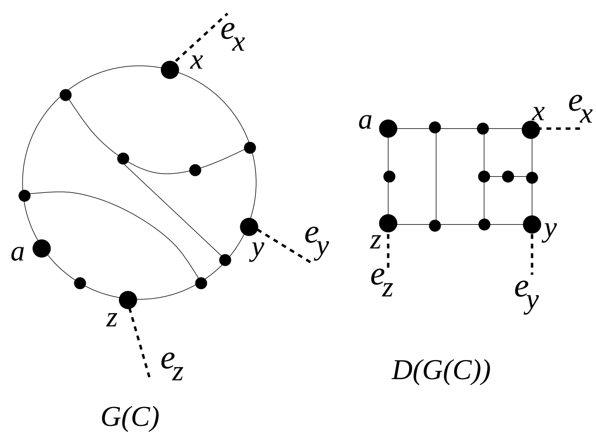

Figure 14: Illustration for Case 2 where C has no bad cycle.

We then consider the case where $G(C)$ has a bad cycle. Let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles of $G(C)$. By Lemma 2 $C_1, C_2, \cdots, C_l$ are independent of each other. Construct a plane graph H from $G(C)$ by contracting each subgraph $G(C_i), 1 \le i \le l$, to a single vertex v_i, as illustrated in Figs. 15(a) and (b). Clearly H is a plane biconnected graph with $\Delta \le 3$, H has no bad cycle with respect to the four designated vertices, and hence H has a rectangular drawing $D(H)$ as illustrated in Fig. 15(c). Clearly there is no bend on $D(H)$. Since the outer cycle of H is drawn as a rectangle in $D(H)$, $D(H)$ satisfies Condition (f3).

We then find a feasible orthogonal drawing $D(G(C_i))$ of $G(C_i), 1 \le i \le l$, in a recursive manner, and patch the drawings $D(G(C_1)), D(G(C_2)), \cdots, D(G(C_l))$

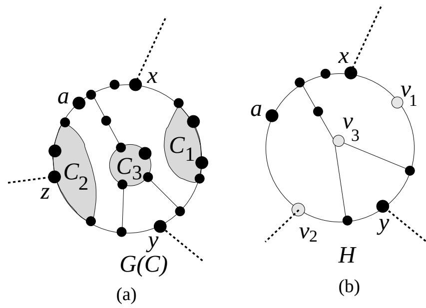

Figure 15: Illustration for Case 2 where C has bad cycles C_1, C_2 and C_3.

into $D(H)$ as illustrated in Fig. 15(d). Since there is no bend in any of $D(G(C_1))$, $D(G(C_2)), \cdots, D(G(C_l))$, there is no bend in the resulting drawing $D(G(C))$. Since the outer boundary of $D(H)$ is a rectangle and $D(G(C))$ expands outwards, $D(G(C))$ satisfies (f3). Thus $D(G(C))$ is a feasible orthogonal drawing of $G(C)$. □

We call the algorithm for obtaining a feasible orthogonal drawing of $G(C)$ as described in the proof of Lemma 3 Algorithm **Feasible-Draw.** We now have the following lemma.

Lemma 4 *Algorithm* **Feasible-Draw** *finds a feasible orthogonal drawing of $G(C)$ in time $O(n(G(C))$, where $n(G(C))$ is the number of vertices in $G(C)$.*

Proof: Let $T_R(G)$ be the computation time of **Rectangular-Draw** for graph G. Then $T_R(G) = O(n)$ by Lemma 1, and hence there is a positive constant c such that

$$T_R(G) \quad \leq \quad c \cdot m(G) \tag{1}$$

for any plane graph G, where $m(G)$ is the number of edges in G.

We first consider the computation time needed for contraction and patching operations in Algorithm **Feasible-Draw.** During the traversal of all inner faces of $G(C)$ we can find the leg-vertices for each bad cycle [8]. Given the leg-vertices of a bad cycle, we can contract the bad cycle to a single vertex in constant time. Therefore the contraction operations in **Feasible-Draw** take $O(n(G(C)))$ time in total. Similarly the patching operations in **Feasible-Draw** take $O(n(G(C)))$ time in total.

We then consider the time needed for operations in **Feasible-Draw** other than the contractions and patchings. Let $T(G(C))$ be the computation time of **Feasible-Draw** for finding a feasible orthogonal drawing of $G(C)$ excluding the time for the contractions and patchings. We claim that $T(G(C)) = O(n(G(C)))$. Since G is a plane graph, $m(G(C)) \leq 3n(G(C))$, where $m(G(C))$ denotes the number of edges in $G(C)$. Therefore it is sufficient to show that

$$T(G(C)) \quad \leq \quad c \cdot m(G(C)). \tag{2}$$

We prove Eq. (2) by induction.

We first consider the case where $G(C)$ has no bad cycle. In this case Algorithm **Feasible-Draw** finds a rectangular drawing of $G(C)$ by **Rectangular-Draw.** Hence, by Eq. (1) we have

$$T(G(C)) = T_R(G(C)) \leq c \cdot m(G(C)).$$

We next consider the case where $G(C)$ has the maximal bad cycles $C_1, C_2, \cdots, C_l$ where $l \geq 1$. Suppose inductively that Eq. (2) holds for each $C_i, 1 \leq i \leq l$, that is,

$$T(G(C_i)) \leq c \cdot m(G(C_i)) \tag{3}$$

for $1 \leq i \leq l$. Algorithm **Feasible-Draw** constructs a plane graph H from $G(C)$ by contracting $G(C_i), 1 \leq i \leq l$, to a single vertex. H has no bad cycles, and the rectangular drawing $D(H)$ can be found by **Rectangular-Draw**. Therefore, by Eq. (1)

$$T_R(H) \leq c \cdot m(H). \tag{4}$$

Algorithm **Feasible-Draw** recursively finds drawings of $G(C_i), 1 \leq i \leq l$, and patches them into the rectangular drawing $D(H)$. Therefore,

$$T(G(C)) = T_R(H) + \sum_{i=1}^{l} T(G(C_i)). \tag{5}$$

One can observe that

$$m(H) + \sum_{i=1}^{l} m(G(C_i)) \quad = \quad m(G(C)). \tag{6}$$

Using Eqs. (3), (4), (5), and (6), we have

$$\begin{aligned} T(G(C)) \quad &\leq \quad c \cdot m(H) + \sum_{i=1}^{l} c \cdot m(G(C_i)) \\ &= \quad c \cdot m(G(C)). \end{aligned}$$

□

We are now ready to prove the sufficiency of Theorem 1; we actually prove the following lemma.

Lemma 5 *If G satisfies the condition in Theorem 1, then G has an orthogonal drawing without bends.*

Proof: Since there are four or more 2-vertices on $C_o(G)$, we designate any four of them as (convex) corners.

Consider first the case where G does not have any bad cycle with respect to the four designated (convex) corners. Then by Lemma 1 there is a rectangular drawing of G. Since the rectangular drawing of G has no bends, it is an orthogonal drawing $D(G)$ of G without bends.

Consider next the case where G has bad cycles. Let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles in G. By Lemma 2 $C_1, C_2, \cdots, C_l$ are independent of each other. We contract each $G(C_i)$, $1 \leq i \leq l$, to a single vertex v_i. Let G^* be the resulting graph. Clearly, G^* has no bad cycle with respect to the four designated vertices, some of which may be vertices resulted from the contraction of bad cycles. By Lemma 1 G^* has a rectangular drawing $D(G^*)$, which can be found by the algorithm **Rectangular-Draw.** We recursively find a feasible orthogonal drawing of each $G(C_i)$, $1 \leq i \leq l$, by **Feasible-Draw.** Patch the feasible orthogonal drawings of $G(C_1), G(C_2), \cdots, G(C_l)$ into $D(G^*)$ by patching operations. The resulting drawing is an orthogonal drawing D of G. Note that $D(G^*)$ and $D(G(C_i))$, $1 \leq i \leq l$, have no bend. Furthermore, patching operation introduces no new bend. Thus D has no bend. □

We now formally describe our algorithm as follows.

Algorithm Bi-Orthogonal-Draw(G)
begin
Select any four 2-vertices on $C_o(G)$ as designated corners;
Find the maximal bad cycles $C_1, C_2, \cdots, C_l$ in G;
For each i, $1 \leq i \leq l$, contract cycle C_i to a single vertex v_i;
Let G^* be the resulting graph;
Find a rectangular drawing of G^* by **Rectangular-Draw**;
Find a feasible orthogonal drawing of each $C_1, C_2, \cdots, C_l$ by **Feasible-Draw**;
Patch the drawings $D(G(C_1)), D(G(C_2)), \cdots, D(G(C_l))$ into $D(G^*)$;
The resulting drawing $D(G)$ is an orthogonal drawing of G without bends.
end.

We then have the following theorem.

Theorem 2 *If G satisfies the condition in Theorem 1, then Algorithm* **Bi-Orthogonal-Draw** *finds an orthogonal drawing of G without bends in linear time.*

Proof: Using a method similar to one in [7, 8, 9], one can find all the maximal bad cycles in G in linear-time. Algorithms **Rectangular-Draw** and **Feasible-Draw** take linear-time. Patching operations take linear time. Therefore the overall time complexity of the algorithm **Bi-Orthogonal-Draw** is linear. □

Using an algorithm similar to Algorithm **Bi-Orthogonal-Draw**, one can find a particular orthogonal drawing without bends, which we call a "four-corner drawing" and define as follows. Let $C_o(G)$ contain four or more 2-vertices of G, and let x, y, z and w be any four of them. Then an orthogonal drawing $D(G)$ of G is called a *four-corner orthogonal drawing* for x, y, z and w if the drawing intersects none of the four quadrants, the first quadrant with the origin at x, the fourth quadrant with the origin at y, the third quadrant with the origin at z, and the second quadrant with the origin at w, after rotating the drawing and renaming vertices x, y, z and w if necessary. In Fig. 16 the four quadrants are shaded. Vertices x, y, z, and w must be convex corners of the drawing $D(C_o(G))$ of the outer cycle $C_o(G)$, and $D(G)$ should intersect neither the horizontal open halfline with left end at x nor the vertical halfline with the lower end at x, and so on for y, z, and w. Clearly, a rectangular drawing with four designated corners is a four-corner drawing for the designated corners. A four-corner drawing has applications in finding an orthogonal drawing with the minimum number of bends [5].

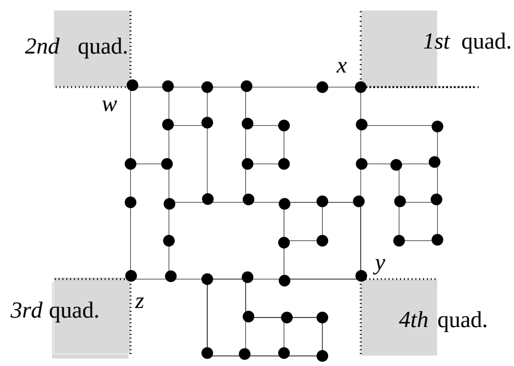

Figure 16: Illustration for a four-corner orthogonal drawing.

We now have the following corollary.

Corollary 6 *Assume that G is a plane biconnected graph with $\Delta \leq 3$ and there are four or more 2-vertices on $C_o(G)$. If every 2-legged cycle in G contains at least two 2-vertices of G and every 3-legged cycle in G contains at least one 2-vertex of G, then one can find a four-corner orthogonal drawing $D(G)$ without bends for any four 2-vertices x, y, z and w on $C_o(G)$ in linear time.*

Proof: One can find a four-corner orthogonal drawing $D(G)$ without bends for any four 2-vertices x, y, z and w on $C_o(G)$ by using an algorithm similar to Algorithm **Bi-Orthogonal-Draw**. The algorithm selects x, y, z and w as designated corners in Step 1 and expands the drawing outwards, if necessary, after each patching operation in Step 7. Other steps of Algorithm **Bi-Orthogonal-Draw** remain unchanged in the algorithm. Clearly the algorithm takes linear time, since Algorithm **Bi-Orthogonal-Draw** takes linear time. □

4 Orthogonal Drawings of Arbitrary Plane Graphs

In this section we extend our result on biconnected plane graphs in Theorem 1 to arbitrary (not always biconnected) plane graphs with $\Delta \leq 3$ as in the following theorem.

Theorem 3 *Let G be a plane graph with $\Delta \leq 3$. Then G has an orthogonal drawing without bends if and only if every k-legged cycle C in G contains at least $4-k$ 2-vertices of G for any k, $0 \leq k \leq 3$.*

Theorem 3 is a generalization of both Theorem 1 and Thomassen's condition [12].

The proof for the necessity of Theorem 3 is similar to the proof for the necessity of Theorem 1. In the rest of this section we give a constructive proof for the sufficiency of Theorem 3. We need some definitions.

We may assume that G is connected. We call a subgraph B of G a *biconnected component* of G if B is a maximal biconnected subgraph of G. We call an edge (u, v) a *bridge* of G if the deletion of (u, v) results in a disconnected graph. Any graph can be decomposed to biconnected components and bridges. The graph G in Fig. 17(a) has three biconnected components B_1, B_2 and B_3 depicted in Fig. 17(b) and six bridges (v_3, v_{24}), (v_{24}, v_{25}), (v_{24}, v_{26}), (v_4, v_{16}), (v_{10}, v_{11}) and (v_7, v_{27}).

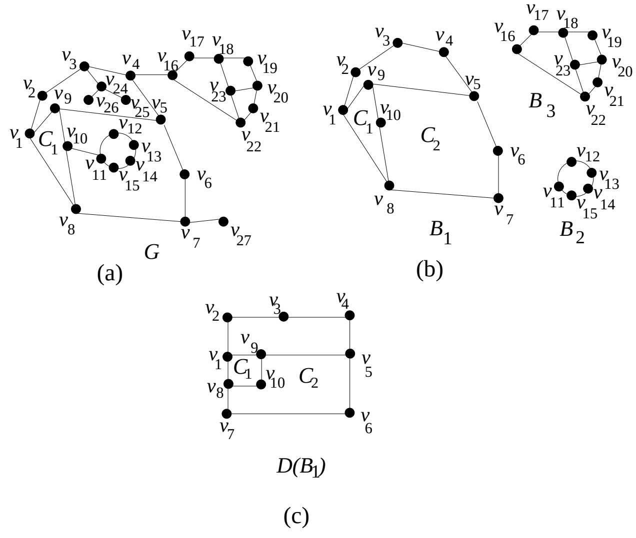

Figure 17: (a) A connected plane graph G, (b) three biconnected components B_1, B_2 and B_3 of G, and (c) a feasible orthogonal drawing of B_1.

Let C be a cycle in G, and let v be a cut vertex of G on C. We call v an *outcut vertex* for C if v is a leg-vertex of C in G, otherwise we call v an *incut* *vertex* for C. (See Fig. 18.) Any outcut vertex for C is either a convex corner or a non-corner of $D(C)$ in any orthogonal drawing $D(G)$ of G, because if it were a concave corner then the leg of C could not be drawn as a horizontal or vertical line segment without edge-crossing. Similarly, any incut vertex for C is either a concave corner or a non-corner of $D(C)$. Thus any orthogonal drawing of G must satisfy the following condition (f4).

(f4) If v is an outcut vertex for a cycle C in G then v is either a convex corner or a non-corner of $D(C)$, and if v is an incut vertex for a cycle C then v is either a concave corner or a non-corner of $D(C)$.

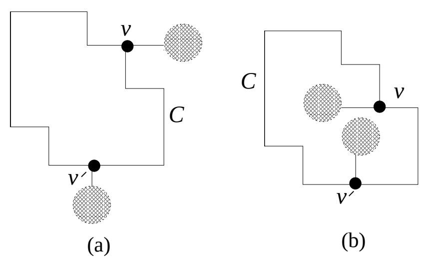

Figure 18: (a) Outcut vertices v at a convex corner and v' at a non-corner, and (b) incut vertices v at a concave corner and v' at a non-corner.

Of course, for any subgraph B of G, the drawing $D(B)$ in an orthogonal drawing $D(G)$ of G satisfies Condition (f4). In the plane graph G in Fig. 17(a), vertices v_4 and v_7 are outcut vertices for the cycle $C_o(B_1)$, and v_3 is an incut vertex for the cycle $C_o(B_1)$. Vertex v_{10} is an outcut vertex for the cycle $C_1 = v_1, v_9, v_{10}, v_8$, but is an incut vertex for the cycle $C_2 = v_5, v_6, v_7, v_8, v_{10}, v_9$. The orthogonal drawing $D(B_1)$ of the biconnected component B_1 in Fig. 17(c) satisfies (f4), because the outcut vertices v_4, v_7 for $C_o(B_1)$ and v_{10} for C_1 are convex corners while the incut vertices v_3 for $C_o(B_1)$ is a non-corner and v_{10} for C_2 is a concave corner. We call an orthogonal drawing $D(B)$ of a subgraph B of G a *mergeable* orthogonal drawing if $D(B)$ satisfies (f4) and has no bends.

Let B be a biconnected subgraph of G, let C be a cycle in B, and let v be a 2-vertex of B on C. We say that vertex v is *good* for C if v is not an incut vertex for C in G. For B_1 in Fig. 17(b), v_2, v_4, v_6 and v_7 are the good vertices for $C_o(B_1)$ while v_3 is not a good vertex. Only a good vertex for C can be drawn as a convex corner of the rectilinear polygon $D(C)$ in a mergeable orthogonal drawing $D(B)$.

We now have the following lemmas.

Lemma 7 *If G satisfies the condition in Theorem 3 and B is a biconnected component of G, then*

(a) there are at least four good vertices for $C_o(B)$,

(b) there are at least two good vertices for every 2-legged cycle C in B, and

(c) there are at least one good vertex for every 3-legged cycle C in B.

Proof: (a) Assume that $C_o(B)$ is a k-legged cycle in G for some $k \geq 0$. The k leg-vertices are outcut vertices, are not incut vertices, have degree 2 in B, and hence are good vertices for $C_o(B)$. Thus, if $k \geq 4$, then there are at least four good vertices for $C_o(B)$.

If $k = 1$, then the condition in Theorem 3 implies that $C_o(B)$ contains at least three 2-vertices of G. Since these vertices have degree 2 in the biconnected component B, they are not incut vertices for $C_o(B)$ and hence are good vertices for $C_o(B)$. The leg-vertex of $C_o(B)$ is a good vertex, too. Thus there are at least four good vertices for $C_o(B)$.

Similarly we can prove the claim when $k = 0$, 2 or 3.

(b) Let C be a 2-legged cycle in B. The two leg-vertices have degree 3 in B, and hence they are not good for C. Let C be a k-legged cycle in G for some $k \geq 2$. If $k = 2$, then the condition in Theorem 3 implies that C contains at least two 2-vertices of G, which are good for C. If $k \geq 4$, then C contains at least two outcut vertices which are 2-vertices of B and hence are good for C. Thus one may assume that $k = 3$. Then C contains an outcut vertex which is a 2-vertex of B and hence is good for C. Furthermore, the condition in Theorem 3 implies that C contains at least one 2-vertex of G, which is good for C. Thus C contains at least two good vertices.

(c) Similar to (b). □

We now have the following lemmas.

Lemma 8 *Let G be a connected plane graph of $\Delta \leq 3$ satisfying the condition in Theorem 3, let B be a biconnected component of G, and let C be a 2- or 3-legged cycle in B. Then the plane subgraph $B(C)$ of B inside C has a mergeable feasible orthogonal drawing $D(B(C))$.*

Proof: We can recursively find a mergeable feasible orthogonal drawing $D(B(C))$ by an algorithm similar to Algorithm **Feasible-Draw** for finding a feasible orthogonal drawing. However, in each recursive step, we have to choose the four designated corner vertices carefully in a way that none of the incut vertices for an outer cycle is chosen as a designated (convex) corner. This can be done, because by Lemma 7(b) every 2-legged cycle in B contains at least two good vertices, by Lemma 7(c) every 3-legged cycle in B contains at least one good vertex, and hence one can choose leg-vertices and good vertices as the four designated corner vertices. These vertices are convex corners in the drawing of the cycle, while any cut vertex which is not chosen as a designated corner is drawn as a non-corner. Hence $D(B(C))$ satisfies (f4), and $D(B(C))$ is a mergeable feasible orthogonal drawing. □

Lemma 9 *If G is a connected plane graph of $\Delta \leq 3$ and satisfies the condition in Theorem 3, then every biconnected component B of G has a mergeable orthogonal drawing.*

Proof: One can find a mergeable orthogonal drawing of B as follows.

By Lemma 7(a) one can select four good vertices on $C_o(B)$ as designated corners.

Consider first the case where B has no bad cycle with respect to the four designated corners. Then by Lemma 1 there is a rectangular drawing $D(B)$ of B. Of course, $D(B)$ has no bend. Any vertex of B which is an outcut or incut vertex of G has degree 2 in B. In $D(B)$, the four designated vertices on $C_o(B)$ are convex corners, and every 2-vertex of B except the four designated vertices is a non-corner. Hence the rectangular drawing $D(B)$ satisfies Condition (f4). Therefore $D(B)$ is a mergeable orthogonal drawing.

Consider next the case where B has bad cycles. Let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles in B. Then by Lemma 2 $C_1, C_2, \cdots, C_l$ are independent of each other. We contract each $B(C_i)$, $1 \leq i \leq l$, to a single vertex v_i. Let B^* be the resulting graph. Clearly, B^* has no bad cycle with respect to the four designated vertices, some of which may be vertices resulted from the contraction of bad cycles. By Lemma 1 B^* has a rectangular drawing $D(B^*)$. We recursively find a mergeable feasible orthogonal drawing $D(B(C_i))$ of each $B(C_i)$, $1 \leq i \leq l$, by the method described in the proof of Lemma 8. Patch the mergeable feasible orthogonal drawings of $D(B(C_i))$, $1 \leq i \leq l$, into $D(B^*)$ by patching operations. Clearly the resulting drawing is an orthogonal drawing $D(B)$ of B. $D(B^*)$ is a mergeable drawing and and $D(B(C_i))$, $1 \leq i \leq l$, are mergeable feasible drawings. Furthermore, the patching operation introduces neither a convex corner at any incut vertex nor a concave corner at any outcut vertex in the drawing of a cycle in B. Hence $D(B)$ is a mergeable orthogonal drawing. □

The algorithm described in the proof of Lemma 9 takes linear time similarly as Algorithm **Bi-Orthogonal-Draw** in Section 3.

Based on the algorithm described in the proof of Lemma 9, we now present a result on four-corner orthogonal drawing in Lemma 10. The result described in Lemma 10 is used to obtain a bend-minimum orthogonal drawing of a plane graph with $\Delta \leq 3$ in [5].

Lemma 10 *Let G be a connected plane graph with $\Delta \leq 3$, let B be a biconnected subgraph of G, and let x, y, z and w be any four 2-vertices of B on $C_o(B)$. Then B has a mergeable four-corner orthogonal drawing $D(B)$ for x, y, z and w if and only if the following (a), (b) and (c) hold:*

(a) all the vertices x, y, z and w are good for the cycle $C_o(B)$ in G;

(b) there are at least two good vertices for every 2-legged cycle C in B; and

(c) there are at least one good vertex for every 3-legged cycle C in B.

Furthermore the drawing above can be found in linear time.

Proof: Necessity: Suppose that B has a mergeable four-corner orthogonal drawing $D(B)$. Then $D(B)$ has no bends and satisfies Condition (f4).

(a) All the vertices x, y, z and w are convex corners of $D(C_o(B))$, and hence by Condition (f4) none of x, y, z and w is an incut vertex for $C_o(B)$. Therefore x, y, z and w are good vertices for $C_o(B)$.

(b) The rectilinear polygonal drawing $D(C)$ of a 2-legged cycle C in $D(B)$ has at least four convex corners. Since $D(B)$ has no bend, every convex corner of $D(C)$ is either a 2-vertex of B or a leg-vertex of C in B. The two leg-vertices of C may serve as two convex corners. Any other convex corner of $D(C)$ is a 2-vertex of B and is not an incut vertex for C in G by Condition (f4). Hence there are at least two good vertices for C.

(c) Similar to (b).

Sufficiency: Assume that B satisfies Conditions (a)–(c) in Lemma 10. Then we can obtain a mergeable orthogonal drawing $D(B)$ of B in linear time by the algorithm described in the proof of Lemma 9. To ensure that $D(B)$ is a four-corner orthogonal drawing for x, y, z and w, the algorithm must select x, y, z and w as the four designated corners, and the drawing may be needed to expand outwards after each patching operation. □

A *block* of a connected graph G is either a biconnected component or a bridge of G. The graph in Fig. 19(a) has the blocks $B_1, B_2, \cdots, B_9$ depicted in Fig. 19(b). The blocks and cut vertices in G can be represented by a tree T,

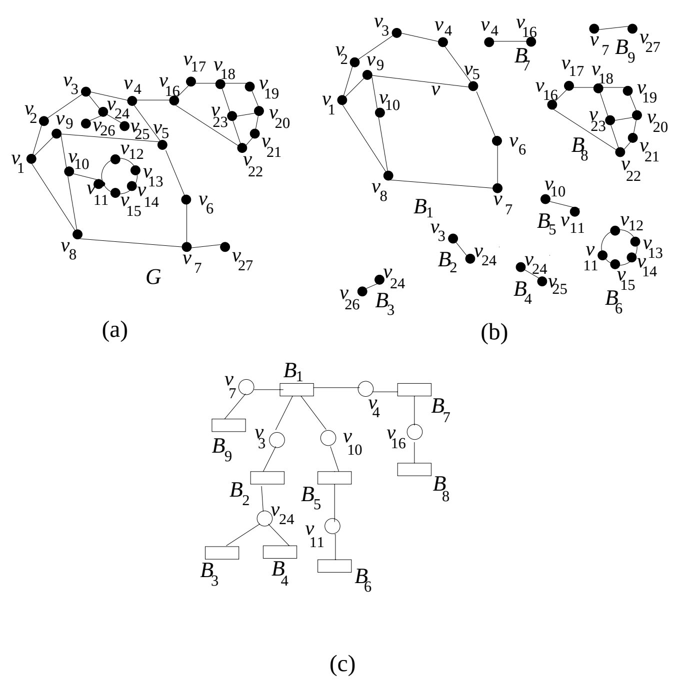

Figure 19: (a) G, (b) blocks, and (c) BC-tree T.

called the BC-*tree* of G. In T each block is represented by a B-node and each cut vertex of G is represented by a C-node. The BC-tree T of the plane graph G in Fig. 19(a) is depicted in Fig. 19(c), where each B-node is represented by a rectangle and each C-node is represented by a circle.

We call a cycle C in a plane graph G a *maximal cycle* of G if $G(C)$ is not contained in $G(C')$ for any other cycle C' in G. Thus a maximal cycle is an outer cycle of a biconnected component of G. The graph G in Fig. 20(a) has two maximal cycles C_1 and C_2 drawn by thick lines. $G(C)$ is called a *maximal closed subgraph* of G if C is a maximal cycle of G. We now have the following lemma.

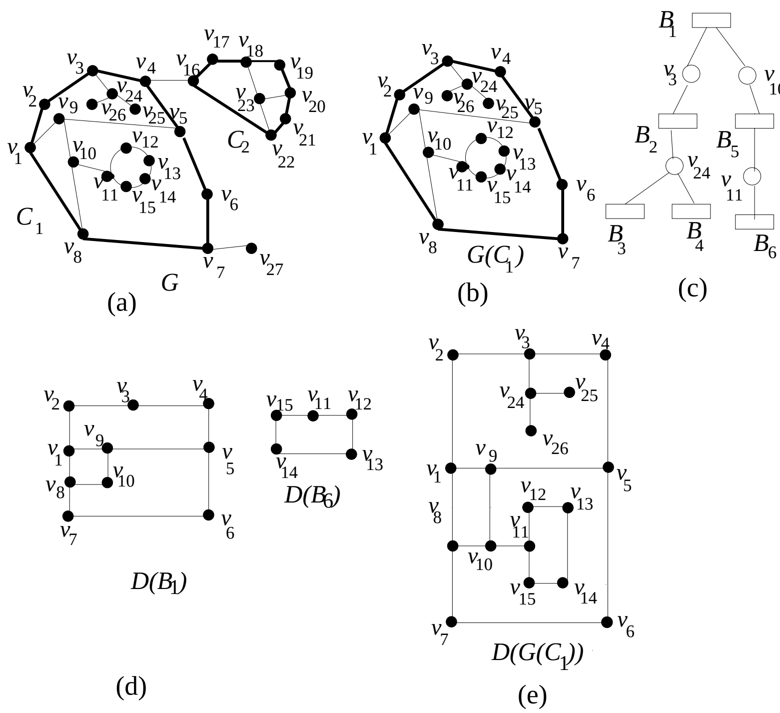

Figure 20: (a) A plane graph G with two maximal cycles C_1 and C_2, (b) $G(C_1)$, (c) BC-tree of $G(C_1)$, (d) drawings of the two biconnected components B_1 and B_6 of $G(C_1)$, and (e) the final drawing of $G(C_1)$.

Lemma 11 *If G is a connected plane graph of $\Delta \leq 3$ and satisfies the condition in Theorem 3, then $G(C)$ has a mergeable orthogonal drawing for any maximal cycle C in G.*

Proof: We give an algorithm for finding a mergeable orthogonal drawing of $G(C)$, that is, an orthogonal drawing of $G(C)$ which has no bends and satisfies (f4).

If $G(C)$ is a biconnected component of G, then by Lemma 9 $G(C)$ has a mergeable orthogonal drawing. One may thus assume that $G(C)$ is not a biconnected component of G. Then $G(C)$ has some biconnected components and bridges. Clearly the biconnected components of $G(C)$ are vertex-disjoint with each other. For each biconnected component we can find a mergeable orthogonal drawing by the algorithm described in the proof of Lemma 9, while we draw each bridge by a horizontal or vertical line segment. We then merge the drawings of biconnected components and bridges without introducing new bends and edge crossings as follows.

We construct a BC-tree T of $G(C)$. Let B_1 be the node in the BC-tree corresponding to the biconnected component of $G(C)$ whose outer cycle is C. We consider T as a rooted tree with root B_1. Starting from the root B_1 we visit the tree by depth-first search and merge the orthogonal drawings of the blocks in the depth first-search order.

Let $B_1, B_2, \cdots, B_b$ be the ordering of the blocks following a depth-first search order starting from B_1. $G(C_1)$ for the graph G in Fig. 20(a) is depicted in Fig. 20(b) and the BC-tree of $G(C_1)$ is depicted in Fig. 20(c), where B_1 is the root of the tree and the other B-nodes are numbered according to a depth-first search order starting from B_1.

We assume that we have obtained a mergeable orthogonal drawing D_i by merging the orthogonal drawings of the blocks $B_1, B_2, \cdots, B_i$, and that we are now going to obtain a mergeable orthogonal drawing D_{i+1} by merging D_i with an orthogonal drawing of the block B_{i+1}. Let v_t be the cut vertex corresponding to the C-node which is the parent of B_{i+1} in T. Let B_x be the parent of v_t in T. Then both B_x and B_{i+1} contain v_t, and D_i contains the drawing of B_x. We have the following three cases to consider.

Case 1: B_x is a biconnected component and B_{i+1} is a bridge.

In this case B_{i+1} is an edge and will be drawn inside an inner face of the drawing D_i. Let C_f be the facial cycle of B_x corresponding to the inner face. Then v_t is an incut vertex for C_f. Since we have obtained a mergeable orthogonal drawing $D(B_x)$ of B_x, v_t is a concave corner or a non-corner of the drawing of C_f in $D(B_x)$, and hence the two edges incident to v_t are drawn in D_i as in Fig. 21 or as a rotated one. We can draw the bridge B_{i+1} as a horizontal or a vertical line segment started from v_t as illustrated by dotted lines in Fig. 21. We thus obtain a drawing D_{i+1}. Clearly no new bend is introduced in D_{i+1} and D_i may be expanded outwards to avoid edge crossings. In Fig. 20(e) the bridge $B_2 = (v_3, v_{24})$ is merged with a biconnected component B_1 at vertex v_3.

Case 2: Both B_x and B_{i+1} are bridges.

In this case v_t is drawn in an inner face of D_i and has degree 1 or 2 in D_i. (See Fig. 22.)

We first consider the case where v_t has degree 1. We then draw B_{i+1} as indicated by the dotted line in Fig. 22(a).

We next consider the case where v_t has degree 2 in D_i. Then v_t has degree 3 in $G(C)$, and let x, y, and z be the three neighbors of v_t in G. We may assume without loss of generality that edges (v_t, x) and (v_t, y) are bridges and are already drawn in D_i and that B_x is either (v_t, x) or (v_t, y). We now merge the drawing of bridge $B_{i+1} = (v_t, z)$ to D_i. It is evident from the drawing described above that bridges (v_t, x) and (v_t, y) are drawn on a (horizontal or

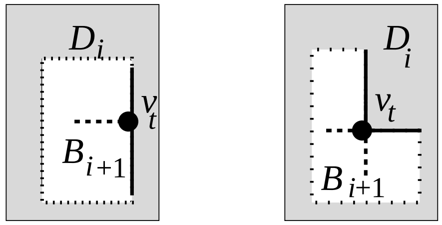

Figure 21: Drawing of edges incident to v_t in D_i when B_x is a biconnected component and B_{i+1} is a bridge.

vertical) straight line segment. We draw B_{i+1} as indicated by a dotted line in Fig. 22(b).

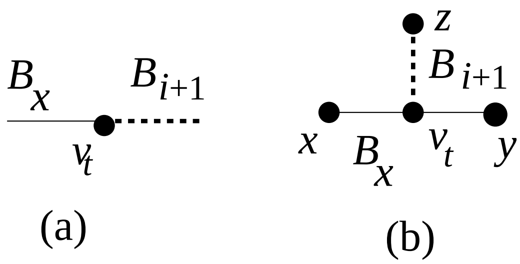

Figure 22: Drawings of B_i when both B_x and B_{i+1} are bridges

.

Case 3: B_x is a bridge and B_{i+1} is a biconnected component.

In this case v_t is drawn in D_i as an end of a horizontal or vertical line segment inside an inner face of D_i. Vertex v_t has degree 2 in B_{i+1} and is an outcut vertex for $C_o(B_{i+1})$. By Lemma 9 $D(B_{i+1})$ is a mergeable orthogonal drawing, and hence v_t is a convex corner or a non-corner of the drawing of $C_o(B_{i+1})$ in $D(B_{i+1})$. Therefore $D(B_{i+1})$ can be easily merged with D_i by rotating $D(G(B_{i+1}))$ by 90° or 180° or 270° and expanding the drawing D_i if necessary. In Fig. 20(e) the orthogonal drawing of B_6 is merged to D_5 at vertex v_{11} where $D(B_6)$ in Fig. 20(d) has been rotated by 90° and the drawing D_5 is expanded outwards. □

We call the algorithm described in the proof of Lemma 11 Algorithm **Maximal-Orthogonal-Draw**. Clearly Algorithm **Maximal-Orthogonal-Draw** takes linear time.

We are now ready to give a proof for sufficiency of Theorem 3.

Proof for sufficiency of Theorem 3

We decompose G into maximal closed subgraphs and bridges. We find an orthogonal drawing of each maximal closed subgraph by Algorithm **Maximal-**

Orthogonal-Draw. Each of the bridges can be drawn by a horizontal or a vertical line segment. Using a technique similar to one in the proof of Lemma 11, we merge the drawings of the maximal closed subgraphs and bridges. The resulting drawing is an orthogonal drawing of G without bends. □

We call the algorithm described in the proof for sufficiency of Theorem 3 Algorithm **No-bend-Orthogonal-Draw**. An execution of the algorithm **No-bend-Orthogonal-Draw** is illustrated in Fig. 23. We now have the following theorem.

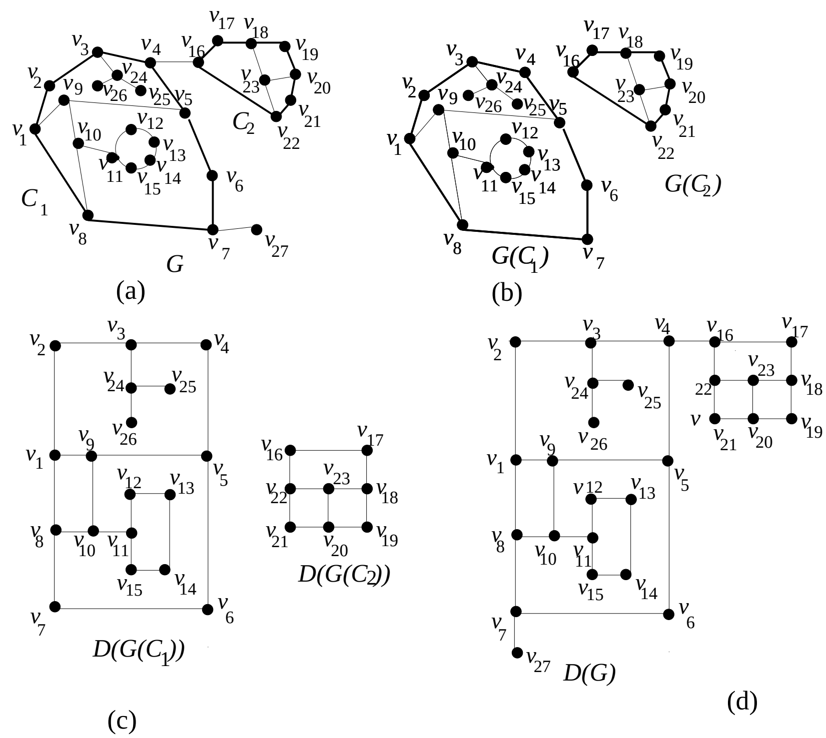

Figure 23: (a) A plane graph G, (b) two maximal closed subgraphs $G(C_1)$ and $G(C_2)$ of G, (c) orthogonal drawings of $G(C_1)$ and $G(C_2)$ without bends, and (d) orthogonal drawings of G without bends.

Theorem 4 *If G is a plane connected graph of $\Delta \leq 3$ and satisfies the condition in Theorem 3, then Algorithm* **No-bend-Orthogonal-Draw** *finds an orthogonal drawing of G without bends in linear time.* □

5 Conclusions

In this paper we established a necessary and sufficient condition for a plane graph G of $\Delta \leq 3$ to have an orthogonal drawing without bends, and gave a linear-time algorithm to examine whether G has an orthogonal drawing without bends and find such a drawing of G if it exists. The condition is a generalization of Thomassen's condition for rectangular drawings [12]. The algorithm presented in this paper has applications in finding an orthogonal drawing of a plane graph of $\Delta \leq 3$ with the minimum number of bends in linear time [5]. It is remained as a future work to establish a necessary and sufficient condition for a plane graph of $\Delta \leq 4$ to have an orthogonal drawing without bends.

An orthogonal drawing of a plane graph G without bends is called a *rectangular drawing* of G if each face of G including the outer face is drawn as a rectangle. A planar graph is said to have a rectangular drawing if at least one of its plane embeddings has a rectangular drawing. Recently Rahman *et al.* [6] gave a necessary and sufficient condition for a planar graph of $\Delta \leq 3$ to have a rectangular drawing which leads to a linear time algorithm to find a rectangular drawing of a planar graph, if it exists. It is thus an interesting future work to generalize the condition of Rahman *et al.* [6] for orthogonal drawings of planar graphs of $\Delta \leq 3$ without bends.

References

[1] G. Di Battista, P. Eades, R. Tamassia, I. G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice-Hall Inc., Upper Saddle River, New Jersey, 1999.

[2] A. Garg and R. Tamassia, *On the computaional complexity of upward and rectilinear planarity testing*, SIAM J. Comput., 31(2), pp. 601-625, 2001.

[3] A. Garg and R. Tamassia, *A new minimum cost flow algorithm with applications to graph drawing*, Proc. of Graph Drawing'96, Lect. Notes in Computer Science, 1190, pp. 201-206, 1997.

[4] T. Lengauer, *Combinatorial Algorihms for Integrated Circuit Layout*, Wiley, Chichester, 1990.

[5] M. S. Rahman and T. Nishizeki, *Bend-minimum orthogonal drawings of plane 3-graphs*, Proc. of WG'02, Lect. Notes in Computer Science, 2573, pp. 365-376, 2002.

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

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

[8] M. S. Rahman, S. Nakano and T. Nishizeki, *A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs*, Journal of Graph Alg. and Appl., http://jgaa.info, 3(4), pp. 31-62, 1999.

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

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

[11] M. S. Rahman, M. Naznin and T. Nishizeki, *Orthogonal drawings of plane graphs without bends*, Proc. of Graph Drawing'01, Lect. Notes in Computer Science, 2265, pp. 392-406, 2002.

[12] 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.

[13] R. Tamassia, *On embedding a graph in the grid with the minimum number of bends*, SIAM J. Comput., 16, pp. 421-444, 1987.