Journal of Graph Algorithms and Applications
http://www.cs.brown.edu/publications/jgaa/
vol. 3, no. 4, pp. 31–62 (1999)

A Linear Algorithm for Bend-Optimal Orthogonal Drawings of Triconnected Cubic Plane Graphs

Md. Saidur Rahman, Shin-ichi Nakano and 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, nakano@ecei.tohoku.ac.jp,
nishi@ecei.tohoku.ac.jp

Abstract

An orthogonal drawing of a plane graph G is a drawing of G in which each edge is drawn as a sequence of alternate horizontal and vertical line segments. In this paper we give a linear-time algorithm to find an orthogonal drawing of a given 3-connected cubic plane graph with the minimum number of bends. The best previously known algorithm takes time $O(n^{7/4}\sqrt{\log n})$ for any plane graph with n vertices.

Communicated by Giuseppe Di Battista and Petra Mutzel.
Submitted: March 1998. Revised: November 1998 and March 1999.

1 Introduction

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. Orthogonal drawings have attracted much attention due to their numerous practical applications in circuit schematics, data flow diagrams, entity relationship diagrams, etc. [1, 5, 9, 10, 11]. In particular, we wish to find an orthogonal drawing with the minimum number of bends. For the plane graph in Fig. 1(a), the orthogonal drawing in Fig. 1(b) has the minimum number of bends, that is, seven bends.

For a given planar graph G with n vertices, if one is allowed to choose its planar embedding, then finding an orthogonal drawing of G with the minimum number of bends is NP-complete [3]. However, Tamassia [10] and Garg and Tamassia [4] presented algorithms that compute an orthogonal drawing of a given plane graph G with the minimum number of bends in $O(n^2 \log n)$ and $O(n^{7/4}\sqrt{\log n})$ time respectively, where a *plane graph* is a planar graph with a fixed planar embedding. They reduce the minimum-bend orthogonal drawing problem to a minimum cost flow problem. On the other hand, several linear-time algorithms are known for finding orthogonal drawings of plane graphs with a presumably small number of bends although they do not always find orthogonal drawings with the minimum number of bends [1, 5].

In this paper we give a linear-time algorithm to find an orthogonal drawing of a 3-connected cubic plane graph with the minimum number of bends. To the best of our knowledge, our algorithm is the first linear-time algorithm to find an orthogonal drawing with the minimum number of bends for a fairly large class of graphs.

An orthogonal drawing in which there is no bend and each face is drawn as a rectangle is called a *rectangular drawing*. Linear-time algorithms have been known to find a rectangular drawing of a plane graph in which every vertex has degree three except four vertices of degree two on the outer boundary, whenever such a graph has a rectangular drawing [6, 8]. The key idea behind our algorithm is to reduce the orthogonal drawing problem to the rectangular drawing problem.

An outline of our algorithm is illustrated in Fig. 1. Given a plane graph as shown in Fig. 1(a), we first put four dummy vertices a, b, c and d of degree two on the outer boundary of G, and let G' be the resulting graph. Fig. 1(c) illustrates G', where the four dummy vertices are drawn by white circles. We then contract each of some cycles $C_1, C_2, \cdots$ and their interiors (shaded in Fig. 1(c)) into a single vertex as shown in Fig. 1(d) so that the resulting graph G'' has a rectangular drawing as shown in Fig. 1(e). We also find orthogonal drawings of those cycles $C_1, C_2, \cdots$ and their interiors recursively (see Figs. 1(d) and (e)). Patching the obtained drawings, we get an orthogonal drawing of G' as shown in Fig. 1(f). Replacing the dummy vertices a, b, c and d in the drawing of G'

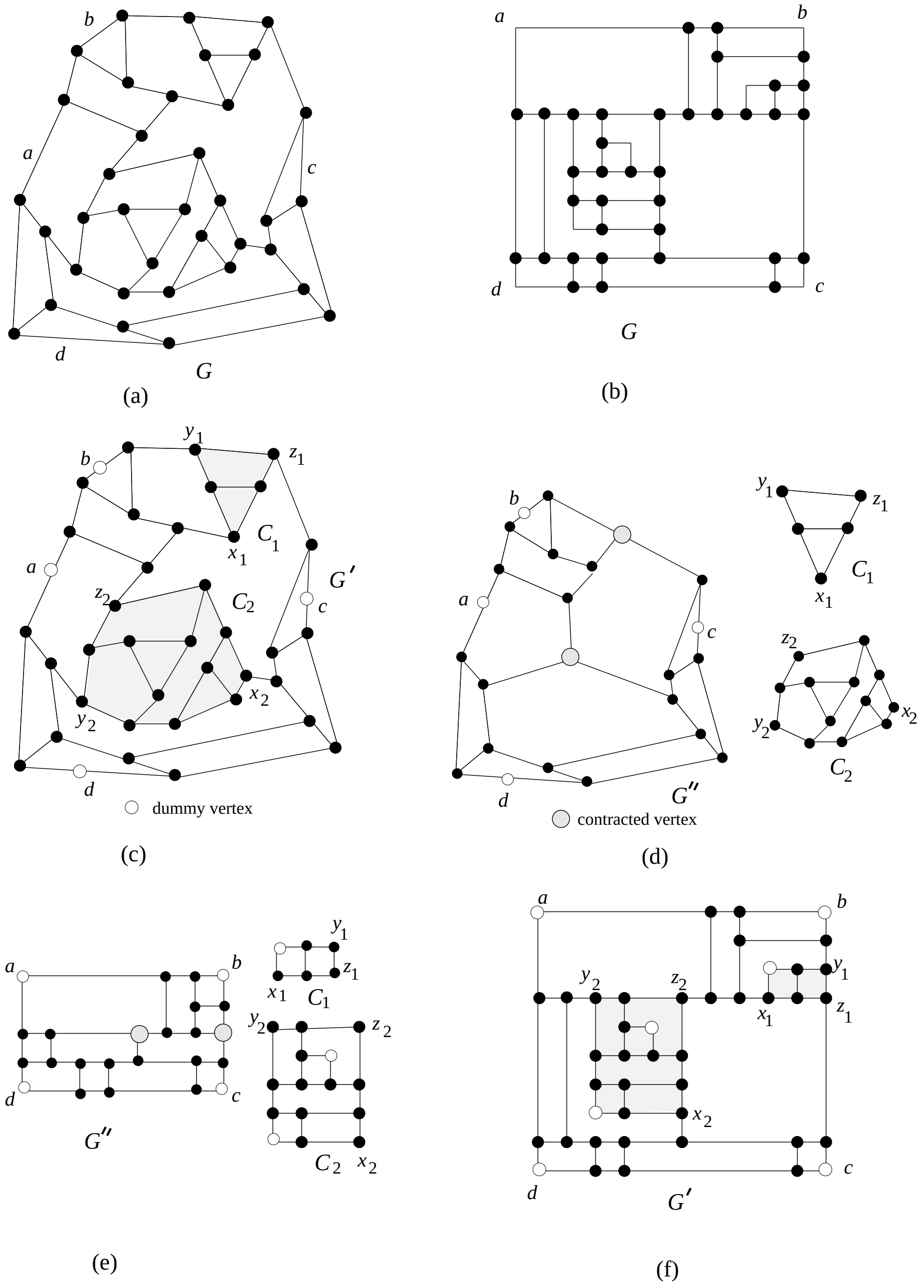


Figure 1: Illustration of our algorithm.

with bends, we finally obtain an orthogonal drawing of G as shown in Fig. 1(b).

The rest of the paper is organized as follows. Section 2 gives some definitions and presents preliminary results. Section 3 presents an algorithm to find an orthogonal drawing in which the number of bends may not be the minimum but does not exceed the minimum number by more than four. Section 4 presents an algorithm to find an orthogonal drawing with the minimum number of bends, modifying the algorithm in Section 3. Section 5 presents bounds on the grid size of our orthogonal drawing on the plane grid. Finally Section 6 concludes the paper. A preliminary version of the paper was presented in [7].

2 Preliminaries

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

Let G be a connected simple graph with n vertices and m edges. We denote the set of vertices of G by $V(G)$, and the set of edges of G by $E(G)$. We also denote by $n(G)$ the number of vertices in G and by $m(G)$ the number of edges in G. Thus $n(G) = |V(G)|$ and $m(G) = |E(G)|$. The *degree* of a vertex v is the number of neighbors of v in G. If every vertex of G has degree three, then G is called a *cubic graph.* The *connectivity* $\kappa(G)$ of a graph G is the minimum number of vertices whose removal results in a disconnected graph or a single-vertex graph K_1. We say that G is *k-connected* if $\kappa(G) \geq k$.

A graph is *planar* if it can be embedded in the plane so that no two edges intersect geometrically except at a vertex to which they are both incident. A *plane* graph is a planar graph with a fixed embedding. A plane graph divides the plane into connected regions called *faces.* We regard the *contour* of a face as a clockwise cycle formed by the edges on the boundary of the face. We denote the contour of the outer face of graph G by $C_o(G)$.

For a simple cycle C in a plane graph G, we denote by $G(C)$ the plane subgraph of G inside C (including 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. An edge e of $G(C)$ is called an *outer edge* of $G(C)$ if e is located on C; otherwise, e is called an *inner edge* of $G(C)$. An edge of G which is incident to exactly one vertex of a simple cycle C and located outside C is called a *leg* of the cycle C. The vertex of C to which a leg is incident is called a *leg-vertex* of C. A simple cycle C in G is called a *k-legged cycle* of G if C has exactly k legs in G. The cycle C indicated by a dotted line in Fig. 2(a) is a 3-legged cycle. In Fig. 2(a) the three legs of C are drawn by thin lines and the three leg-vertices by black big circles.

Let C be a 3-legged cycle in a 3-connected cubic plane graph G. Then the three leg-vertices of C are distinct with each other since G is cubic. We denote by $\mathcal{C}_C$ the set of all 3-legged cycles of G in $G(C)$ including C itself. For the cycle C in Fig. 2(a) $\mathcal{C}_C = \{C, C_1, C_2, \cdots, C_7\}$, where all cycles in $\mathcal{C}_C$ are drawn by thick lines. For any two 3-legged cycles C' and C'' in $\mathcal{C}_C$, we say that C''

is a *descendant cycle* of C' and C' is an *ancestor cycle* of C'' if C'' is contained in $G(C')$. We also say that a descendant cycle C'' of C' is a *child-cycle* of C' if C'' is not a descendant cycle of any other descendant cycle of C'. In Fig. 2(a) cycles $C_1, C_2, \cdots, C_7$ are the descendant cycles of C, cycles $C_1, C_2, \cdots, C_5$ are the child-cycles of C, and cycles C_6 and C_7 are the child-cycles of C_4. We now have the following lemma.

Lemma 1 *Let C be a 3-legged cycle in a 3-connected cubic plane graph G. Then the child-cycles of C are independent of each other.*

Proof: Suppose for a contradiction that a pair of distinct child-cycles C_1 and C_2 of C are not independent. Then C_1 and C_2 have a common vertex. However, either cannot be a descendant cycle of the other since both are child-cycles of C. Therefore C_2 has a vertex in $G(C_1)$ and a vertex not in $G(C_1)$. Thus C_2 must pass through two of the three legs of C_1. Let v be the leg-vertex of the other leg of C_1. Similarly, C_1 must pass through two of the three legs of C_2. Let w be the leg-vertex of the other leg of C_2. Then removal of v and w disconnects G, contrary to the 3-connectivity of G. □

Lemma 1 implies that the containment relation among cycles in $\mathcal{C}_C$ is represented by a tree as illustrated in Fig. 2(b); the tree is called the *genealogical tree* of C and denoted by T_C.

We have the following two lemmas.

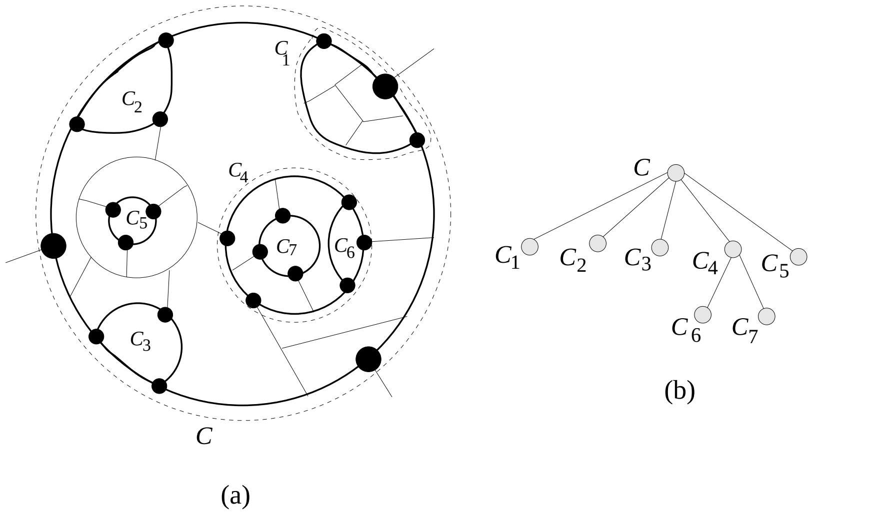


Figure 2: (a) Cycles in $\mathcal{C}_C$ and (b) genealogical tree T_C.

Lemma 2 *Let C be a 3-legged cycle in a 3-connected cubic plane graph G. Then*

$$|\mathcal{C}_C| \leq n(G(C))/2.$$

Proof: It suffices to show that one can assign two vertices of $G(C)$ to each cycle in $\mathcal{C}_C$ without duplication; thus each vertex of $G(C)$ is assigned to at most one cycle in $\mathcal{C}_C$. We decide the assignment in the top-down order on the tree T_C as follows.

We first assign any two leg-vertices of C to C. For each child-cycle C_i of C we next assign two of C_i's three leg-vertices to C_i. Since the child-cycles of C are independent of each other by Lemma 1, no two child-cycles of C share any vertex. Cycles C and C_i share at most one common leg-vertex; otherwise, C_i would have at least four legs. The common leg-vertex may have been assigned to C. However, since C_i has three distinct leg-vertices, C_i has at least two leg vertices which have not been assigned yet. Thus we can assign these two leg-vertices to C_i. In a similar fashion, for each child-cycle C_j of a child-cycle of C, we can assign two of C_j's leg-vertices to C_j, and so on. Clearly the assignment above can be done without duplication. □

Lemma 3 *Let C be a 3-legged cycle in a 3-connected cubic plane graph G. Then the genealogical tree T_C can be found in linear time.*

Proof: We outline an algorithm to find all 3-legged cycles in $\mathcal{C}_C$ and construct T_C in linear time. We first traverse the contour of each inner face of $G(C)$

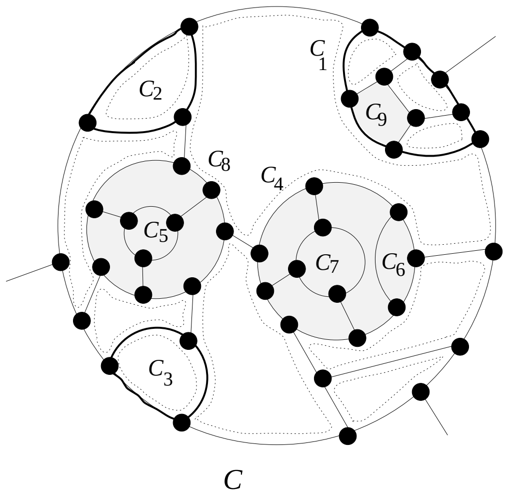


Figure 3: Illustration for the proof of Lemma 3.

containing an outer edge of $G(C)$ as illustrated in Fig. 3, where the traversed contours of faces are indicated by dotted lines. Clearly each outer edge of $G(C)$ is traversed exactly once, and each inner edge of $G(C)$ is traversed at most twice. The inner edges of $G(C)$ traversed exactly once form cycles, called *singly traced* *cycles*, the insides of which have not been traversed. In Fig. 3 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 3-legged cycles in $\mathcal{C}_C$ that share edges with C; C_1, C_2 and C_3 drawn in thick lines in Fig. 3 are these cycles. (Note that a 3-legged cycle in $\mathcal{C}_C$ sharing edges with C has two legs on C and the other leg is either an inner edge which is traversed twice or a leg of C. Using edge-labellings similar to one in [8, pp. 215-216], one can find such a 3-legged cycle.) Any of the remaining 3-legged cycles in $\mathcal{C}_C$ either is a singly traced cycle or is located inside a singly traced cycle. One can find all 3-legged cycles inside a singly traced cycle by recursively applying the method to the singly traced cycle. In Fig. 3 cycle $C_4 \in \mathcal{C}_C$ is a singly traced 3-legged cycle, cycles $C_6, C_7 \in \mathcal{C}_C$ are inside C_4, cycle $C_5 \in \mathcal{C}_C$ is inside C_8, and there is no 3-legged cycle inside C_9. One can also determine the containment relation of the cycles in $\mathcal{C}_C$ while finding them. Since the algorithm traverses the contour of each inner face of $G(C)$ exactly once, each edge of $G(C)$ is traversed at most twice. Thus the algorithm takes linear time. □

An *orthogonal drawing* of a plane graph G is a drawing of G in which 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 defined to be a point where an edge changes its direction in a drawing. We denote by $b(G)$ the minimum number of bends needed for an orthogonal drawing of G.

A *rectangular drawing* of a plane graph G is a drawing of G such that each edge is drawn as a horizontal or vertical line segment, and each face is drawn as a rectangle. Thus a rectangular drawing is an orthogonal drawing in which there is no bend and each face is drawn as a rectangle. The drawing of G'' in Fig. 1(e) is a rectangular drawing. The drawing of G' in Fig. 1(f) is not a rectangular drawing, but is an orthogonal drawing. The following result on rectangular drawings is known.

Lemma 4 *Let G be a connected plane graph such that all vertices have degree three except four vertices of degree two on $C_o(G)$. Then G has a rectangular drawing if and only if G has none of the following three types of simple cycles [12]:*

(r1) 1-legged cycles;

(r2) 2-legged cycles which contain at most one vertex of degree two; and

(r3) 3-legged cycles which contain no vertex of degree two.

Furthermore one can check in linear time whether G satisfies the condition above, and if G does then one can find a rectangular drawing of G in linear time [8].

In a rectangular drawing of G, the four vertices of degree two are the four corners of the rectangle corresponding to $C_o(G)$. A cycle of type (r1), (r2) or

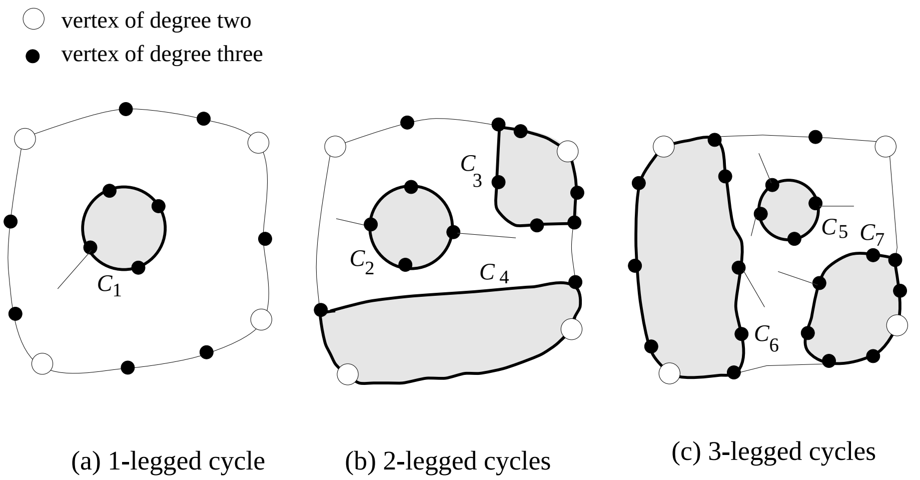


Figure 4: Bad cycles C_1, C_2, C_3 and C_5, and non-bad cycles C_4, C_6 and C_7.

(r3) is called a *bad cycle*. Figs. 4(a), (b) and (c) illustrate 1-legged, 2-legged and 3-legged cycles, respectively. Cycles C_1, C_2, C_3 and C_5 are bad cycles. On the other hand, cycles C_4, C_6 and C_7 are not bad cycles; C_4 is a 2-legged cycle but contains two vertices of degree two, and C_6 and C_7 are 3-legged cycles but contain one or two vertices of degree two.

Linear-time algorithms to find a rectangular drawing of a plane graph satisfying the condition in Lemma 4 have been obtained [6, 8]. Our orthogonal drawing algorithm uses the algorithm in [8], which we call **Rectangular-Draw** in this paper.

3 Orthogonal Drawing

In this section we give a linear-time algorithm to find an orthogonal drawing of a 3-connected cubic plane graph G with at most $b(G) + 4$ bends. Thus there are at most four extra bends in a drawing produced by the algorithm.

Let G be a 3-connected cubic plane graph. Since G is 3-connected, G has no 1- or 2-legged cycle. Every cycle C of G has at least four convex corners, i.e., polygonal vertices of inner angle 90°, in any orthogonal drawing of G. Since G is cubic, such a corner must be a bend if it is not a leg-vertex of C. Thus we have the following facts for any orthogonal drawing of G.

Fact 5 *At least four bends must appear on $C_o(G)$.*

Fact 6 *At least one bend must appear on each 3-legged cycle in G.*

An outline of our algorithm is as follows.

Let G' be a graph obtained from G by adding four dummy vertices a, b, c and d of degree two on $C_o(G)$ as follows. If there are four or more edges on $C_o(G)$, then add four dummy vertices on any four distinct edges on $C_o(G)$, one for each. If there are exactly three edges on $C_o(G)$, then add two dummy vertices on any two distinct edges on C_o and two dummy vertices on the remaining edge.

If the resulting graph G' has no bad cycle, then by Lemma 4 G' has a rectangular drawing, in which the four dummy vertices become the corners of the rectangle corresponding to $C_o(G')$. From the rectangular drawing of G' one can immediately obtain an orthogonal drawing of G with exactly four bends by replacing the four dummy vertices with bends at the corners. By Fact 5 the orthogonal drawing of G has the minimum number of bends.

Thus we may assume that G' has a bad cycle. Since G has no 1- or 2-legged cycle, every bad cycle in G' is a 3-legged cycle containing no dummy vertex of degree two like C_5 in Fig. 4(c). A bad cycle C in G' is defined to be *maximal* if C is not contained in $G'(C')$ for any other bad cycle C' in G'. In Fig. 5(a) $C_1, C_2, \cdots, C_6$ are the bad cycles, $C_1, C_2, \cdots, C_4$ are the maximal bad cycles in G', and C_5 and C_6 are not maximal bad cycles since they are contained in $G'(C_4)$. The 3-legged cycle C_7 indicated by a dotted line in Fig. 5(a) is not a bad cycle in G' since it contains a dummy vertex a. Since G is a 3-connected cubic plane graph, all maximal bad cycles in G' are independent of each other similarly as in Lemma 1. Let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles in G'. (In Fig. 1(c) $l = 2$, and in Fig. 5(a) $l = 4$.) Let G'' be the graph obtained from G' by contracting $G'(C_i)$ into a single vertex v_i for each maximal bad cycle C_i, $1 \leq i \leq l$, as illustrated in Figs. 1(d) and 5(b). Clearly G'' has no bad cycle. We find a rectangular drawing of G'', and recursively find a "suitable" orthogonal drawing of $G'(C_i), 1 \leq i \leq l$, with the minimum number of bends, defined later and called a *feasible drawing*, and finally patch them to get an orthogonal drawing of G. (See Figs. 1, 5 and 12.)

We define the following terms for each 3-legged cycle C in G in a recursive manner based on its genealogical tree T_C. Each 3-legged cycle C in G is divided into three paths P_1, P_2 and P_3 by the three leg-vertices x, y and z of C as illustrated in Fig. 6. These three paths P_1, P_2 and P_3 are called the *contour paths* of C. Each contour path of C is classified as either a *green path* or a *red path*. In addition, we assign an integer $bc(C)$, called the *bend-count* of C, to each 3-legged cycle C in G. We will show later that $G(C)$ has an orthogonal drawing with $bc(C)$ bends and has no orthogonal drawing with fewer than $bc(C)$ bends, that is $b(G(C)) = bc(C)$. Furthermore we will show that, for any green path of C, $G(C)$ has an orthogonal drawing with $bc(C)$ bends including a bend on the green path. On the other hand, for any red path of C, $G(C)$ does not have any orthogonal drawing with $bc(C)$ bends including a bend on the red path. We define the $bc(C)$, red paths and green paths in a recursive manner on the genealogical tree T_C, as follows.

Let C be a 3-legged cycle in G, and let $C_1, C_2, \cdots, C_l$ in $\mathcal{C}_C$ be the child-

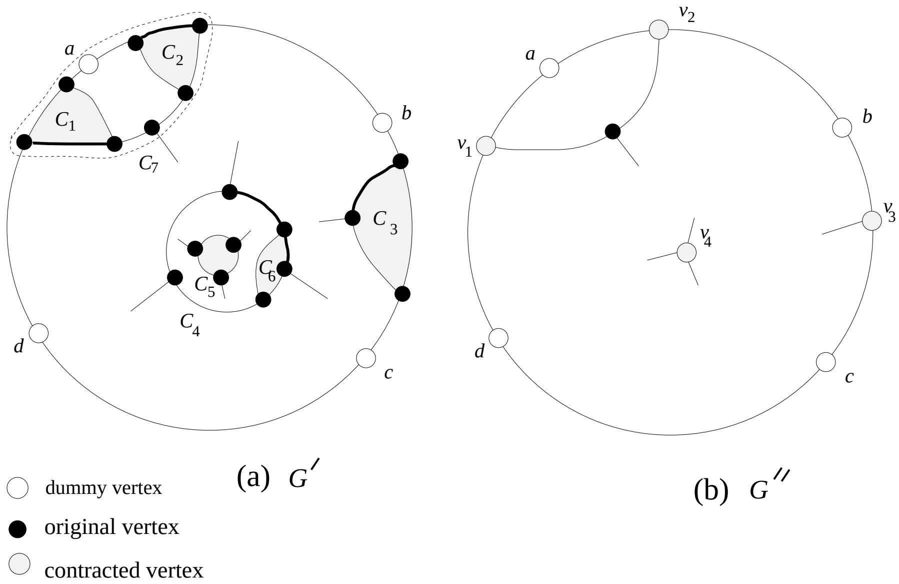


Figure 5: G' and G''.

cycles of C. Assume that we have already defined the classification and the assignment for all child-cycles of C and are going to define them for C. There are three cases.

Case 1: C has no child-cycle, that is, $l = 0$, and hence T_C consists of a single vertex (see Fig. 6(a)).

In this case, we classify all the three contour paths of C as green paths, and set

$$bc(C) = 1. \tag{1}$$

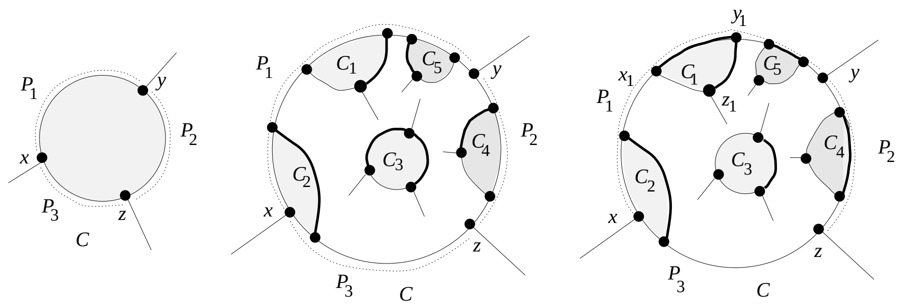
(By Fact 6 we need at least one bend on C. In Fig. 6(a) green paths of C are indicated by dotted lines.)

Case 2: None of the child-cycles of C has a green path on C.

In this case, we classify all the three contour paths of C as green paths, and set

$$bc(C) = 1 + \sum_{i=1}^{l} bc(C_i). \tag{2}$$

(In Fig. 6(b) the child-cycles of C are $C_1, C_2, \cdots, C_5$, and all green paths of them, drawn by thick lines, do not lie on C. Since none of $C_1, C_2, \cdots, C_l$ and their descendant 3-legged cycles has a green path on C as known later, the orthogonal drawings of $G(C_1), G(C_2), \cdots, G(C_l)$ with the minimum number of



(a) Case 1 (b) Case 2 (c) Case 3

Figure 6: Green paths.

bends have no bend on C and hence we need to introduce a new bend on C in an orthogonal drawing of $G(C)$. In Fig. 6(b) the three green paths of C are indicated by dotted lines.)

Case 3: Otherwise (see Fig. 6(c)).

In this case at least one of the child-cycles $C_1, C_2, \cdots, C_l$, for example C_1, C_4 and C_5 in Fig. 6(c), has a green path on C. Classify a contour path $P_i, 1 \leq i \leq 3$, of C as a green path if a child-cycle of C has its green path on P_i. Otherwise, classify P_i as a red path. Thus at least one of P_1, P_2 and P_3 is a green path. We set

$$bc(C) = \sum_{i=1}^{l} bc(C_i). \tag{3}$$

(In Fig. 6(c) P_1 and P_2 are green paths but P_3 is a red path. For a child-cycle C_j having a green path on C, $G(C_j)$ has an orthogonal drawing with $bc(C_j)$ bends including a bend on the green path, and hence we need not to introduce any new bend on C.)

We have the following lemmas.

Lemma 7 *At least one of the three contour paths of every 3-legged cycle in G is a green path under the classification above.*

Proof: Immediate. □

Lemma 8 *Let C be a 3-legged cycle in G. Then the classification and assignment for all descendant cycles of C can be done in linear time, that is, in time $O(n(G(C)))$, where $n(G(C))$ is the number of vertices in $G(C)$.*

Proof: By Lemma 3 T_C can be found in linear time. Using T_C, the classification and assignment for all descendant cycles of C can be done in linear time. □

Lemma 9 *Let C be a 3-legged cycle in G, then $G(C)$ has at least $bc(C)$ vertex-disjoint 3-legged cycles of G which do not contain any edge on red paths of C.*

Proof: We will prove the claim by induction based on T_C.

We first assume that C has no child-cycle. According to the classification and assignment, all the three contour paths of C are green paths, and $bc(C) = 1$. Clearly $G(C)$ has a 3-legged cycle C of G which does not contain any edge on red paths of C. Thus the claim holds for C.

We next assume that C has at least one child-cycle, and suppose inductively that the claim holds for any descendant 3-legged cycle of C. Let $C_1, C_2, \cdots, C_l$ be the child-cycles of C, then the hypothesis implies that, for each C_i, $1 \le i \le l$, $G(C_i)$ has at least $bc(C_i)$ vertex-disjoint 3-legged cycles of G which do not contain any edge on red paths of C_i. There are the following two cases to consider.

Case 1: None of the child-cycles of C has a green path on C (see Fig. 6(b)).

In this case, all the three contour paths of C are green, and $bc(C) = 1 + \sum_{i=1}^{l} bc(C_i)$ by (2). For each i, $1 \le i \le l$, a child-cycle C_i of C has no green path on C, and hence all C_i's contour paths on C are red. By the induction hypothesis $G(C_i)$ has $bc(C_i)$ vertex-disjoint 3-legged cycles which do not contain any edge on red paths of C_i. Therefore, these $bc(C_i)$ cycles do not contain any edge on C. Furthermore by Lemma 1 the child-cycles $C_1, C_2, \cdots, C_l$ of C are independent of each other. Therefore $G(C)$ has $\sum_{i=1}^{l} bc(C_i)$ vertex-disjoint 3-legged cycles of G which do not contain any edge on C. Since G is cubic, C and these $\sum_{i=1}^{l} bc(C_i)$ 3-legged cycles are vertex-disjoint. Trivially C does not contain any edge on red paths of C since all the contour paths of C are green. Thus $G(C)$ has at least $bc(C) = 1 + \sum_{i=1}^{l} bc(C_i)$ vertex-disjoint 3-legged cycles of G which do not contain any edge on red paths of C.

Case 2: At least one of the child-cycles of C has a green path on C (see Fig. 6(c)).

In this case, $bc(C) = \sum_{i=1}^{l} bc(C_i)$ by (3). By the induction hypothesis each cycle C_i, $1 \le i \le l$, has $bc(C_i)$ vertex-disjoint 3-legged cycles which do not contain any edge on red paths of C_i. Furthermore by Lemma 1 the child-cycles C_i, $1 \le i \le l$, are independent of each other. Therefore $G(C)$ has $\sum_{i=1}^{l} bc(C_i)$ vertex-disjoint 3-legged cycles which do not contain any edge on red paths of any child-cycle C_i. These $\sum_{i=1}^{l} bc(C_i)$ cycles may contain edges on green paths of C_i, but any green path of C_i is not contained in a red path of C by the classification of contour paths. Therefore, $G(C)$ has at least $bc(C) = \sum_{i=1}^{l} bc(C_i)$ vertex-disjoint 3-legged cycles which do not contain any edge on red paths of C. $\square$

Lemma 10 *Every 3-legged cycle C of G satisfies $b(G(C)) \ge bc(C)$.*

Proof: By Fact 6 at least one bend must appear on each of the 3-legged cycles. By Lemma 9 $G(C)$ has at least $bc(C)$ vertex-disjoint 3-legged cycles. Therefore any orthogonal drawing of $G(C)$ has at least $bc(C)$ bends, that is, $b(G(C)) \ge bc(C)$. $\square$

Conversely proving $b(G(C)) \leq bc(C)$, we have $b(G(C)) = bc(C)$ for any 3-legged cycle C in G. Indeed we will prove a stronger claim later in Lemmas 11 and 12 after introducing the following definition.

Let x, y and z be the three leg-vertices of C in G. One may assume that x, y and z appear on C in clockwise order. For a green path P with ends x and y on C, an orthogonal drawing of $G(C)$ is defined to be *feasible for* P if the drawing satisfies the following properties (p1)–(p3):

(p1) The drawing of $G(C)$ has exactly $bc(C)$ bends.

(p2) At least one bend appears on the green path P.

(p3) The drawing of $G(C)$ intersects none of the the following six open halflines.

- the vertical open halfline with the upper end at x.
- the horizontal open halfline with the right end at x.
- the vertical open halfline with the lower end at y.
- the horizontal open halfline with the left end at y.
- the vertical open halfline with the upper end at z.
- the horizontal open halfline with the left end at z.

The property (p3) implies that, in the drawing of $G(C)$, any vertex of $G(C)$ except x, y and z is located in none of the following three areas (shaded in Fig. 7): the third quadrant with the origin x, the first quadrant with the origin y, and the fourth quadrant with the origin z. It should be noted that each leg of C must start with a line segment on one of the six open halflines above if an orthogonal drawing of G is extended from an orthogonal drawing of $G(C)$ feasible for P. Fig. 7 illustrates an orthogonal drawing feasible for a green path P.

We will often call an orthogonal drawing of $G(C)$ feasible for a green path of C simply a *feasible orthogonal drawing* of $G(C)$.

Lemma 11 *For any 3-legged cycle C of G and any green path P of C, $G(C)$ has an orthogonal drawing feasible for P.*

Proof: We give a recursive algorithm to find an orthogonal drawing of $G(C)$ feasible for P, as follows. There are three cases to consider.
Case 1: C has no child-cycle (see Fig. 6(a)).

In this case $bc(C) = 1$ by (1). We insert, as a bend, a dummy vertex t of degree two on an arbitrary edge on the green path P in graph $G(C)$, and let F be the resulting graph. Then every vertex of F has degree three except four vertices of degree two: the three leg-vertices x, y and z, and the dummy vertex t. Since C has no child-cycle, trivially F has no bad cycle. Therefore by Algorithm **Rectangular-Draw** in [8] one can find a rectangular drawing

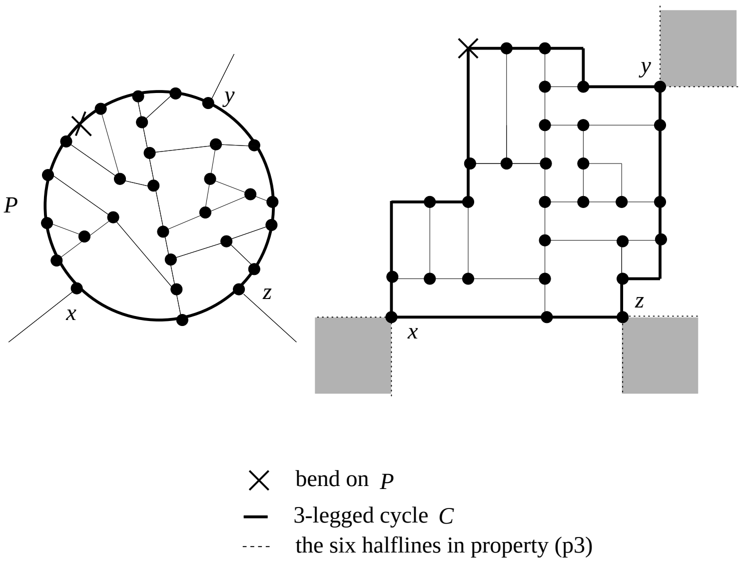


Figure 7: An example of a feasible drawing.

of F with four corners on x, y, z and t. The drawing of F immediately yields an orthogonal drawing of $G(C)$ having exactly one bend at t, in which C is a rectangle. Thus the drawing satisfies (p1)–(p3), and hence is feasible for P.

Case 2: None of the child-cycles of C has a green path on C (see Fig. 6(b)).

Let $C_1, C_2, \cdots, C_l$ be the child-cycles of C, where $l \geq 1$. First, for each i, $1 \leq i \leq l$, we choose an arbitrary green path of C_i, and find an orthogonal drawing $D(G(C_i))$ of $G(C_i)$ feasible for the green path in a recursive manner.

Next, we construct a plane graph F from $G(C)$ by contracting each $G(C_i)$, $1 \leq i \leq l$, to a single vertex v_i. Fig. 8(a) illustrates F for the graph $G(C)$ in Fig. 6(b) where the green path P is assumed to be P_1. One or more edges on P are contained in none of C_i, $1 \leq i \leq l$, and hence these edges remain in F. Add a dummy vertex t on any of these edges of P as shown in Fig. 8(b), and let H be the resulting plane graph. All vertices of H have degree three except the four vertices x, y, z and t on $C_o(H)$ of degree two, and H has no bad cycle. Therefore, by **Rectangular-Draw**, we can find a rectangular drawing $D(H)$ of H with four corners on x, y, z and t. $D(H)$ immediately yields an orthogonal drawing of F with exactly one bend at t. Fig. 8(c) illustrates a rectangular drawing of H for C and $P = P_1$ in Fig. 6(b).

Finally, as explained below, patching the drawings $D(G(C_1)), D(G(C_2))$, $\cdots, D(G(C_l))$ into $D(H)$, we can construct an orthogonal drawing of $G(C)$ with $bc(C) = 1 + \sum_{i=1}^{l} bc(C_i)$ bends (see Fig. 8). As illustrated in Fig. 9(b), there are twelve distinct embeddings of a contracted vertex v_i and the three legs incident

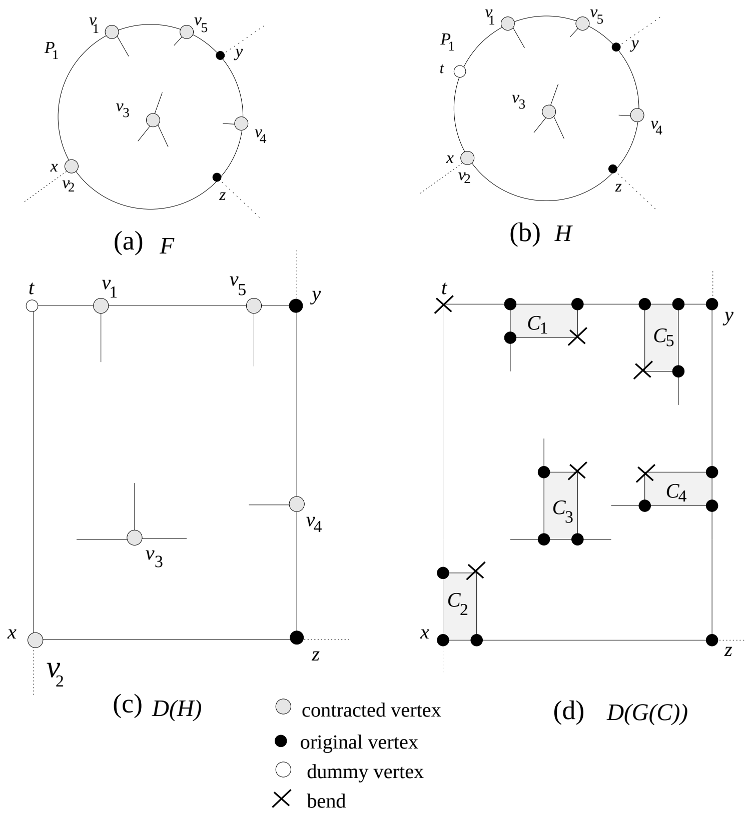


Figure 8: F, H, $D(H)$ and $D(G(C))$ for Case 2.

to v_i, depending on both the directions of the three legs and the chosen green path of C_i, where the ends of the path are denoted by x and y. For each of the twelve cases, we can replace a contracted vertex v_i with an orthogonal drawing of $G(C_i)$ feasible for the green path or a rotated one shown in Fig. 9(c), where the drawing of $G(C_i)$ is depicted as a rectangle for simplicity. For example, the embedding of the contracted vertex v_1 with three legs in Fig. 8(c) is the same as the middle one of the leftmost column in Fig. 9(b) (notice the green path of C_1 drawn in a thick line in Fig. 6(b)); and hence v_1 in $D(H)$ is replaced by $D(G(C_1))$, the middle one of the leftmost column in Fig. 9(c). Clearly t is a bend on P, and C is a rectangle in the drawing of $G(C)$. Thus the drawing is feasible for P. We call the replacement above a *patching operation.*[1]

Case 3: Otherwise (see Fig. 6(c)).

Let $C_1, C_2, \cdots, C_l$ be the child-cycles of C, where $l \geq 1$. In this case, for

[1]A replacement operation similar to our patching operation is used in [5].

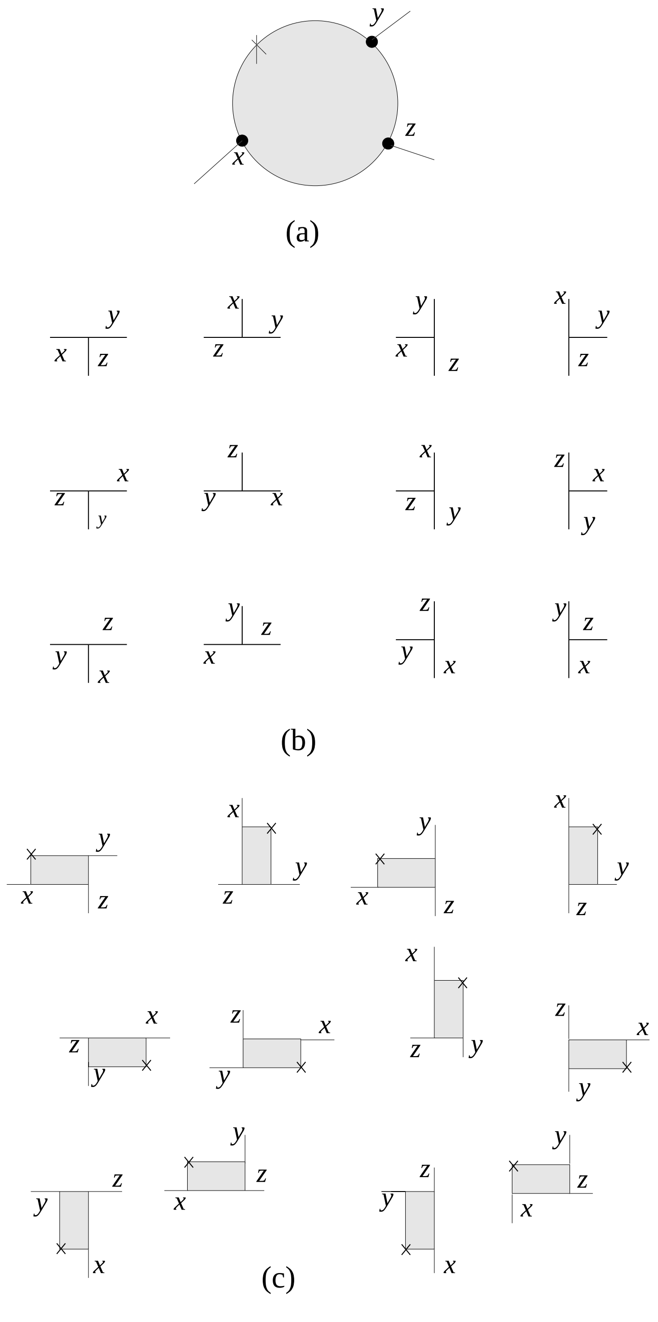


Figure 9: (a) A 3-legged cycle, (b) twelve embeddings of a vertex v_i and three legs incident to v_i, and (c) twelve feasible orthogonal drawings of $G(C_i)$ and rotated ones.

any green path P on C, at least one of $C_1, C_2, \cdots, C_l$ has a green path on P. One may assume without loss of generality that C_1 has a green path Q on the green path P of C, that the three leg-vertices x_1, y_1 and z_1 of C_1 appear on C_1 clockwise in this order, and that x_1 and y_1 are the ends of Q as illustrated in Fig. 6(c).

We first construct a plane graph F from $G(C)$ by contracting each $G(C_i), 1 \leq i \leq l$, to a single vertex v_i. Fig. 10(a) illustrates F for $G(C)$ in Fig. 6(c). Replace v_1 in F with a quadrangle $x_1ty_1z_1$ as shown in Fig. 10(b) where t is a dummy vertex of degree two, and let H be the resulting plane graph. Thus all vertices of H have degree three except four vertices on $C_o(H)$ of degree two: the dummy vertex t and the three leg-vertices x, y and z of C. Furthermore H has no bad cycle. Therefore, by **Rectangular-Draw**, we can find a rectangular drawing $D(H)$ of H with four corners on t, x, y and z, in which the contour $x_1ty_1z_1$ of a face is drawn as a rectangle. Fig. 10(c) illustrates a rectangular drawing of H for $G(C)$ in Fig. 6(c).

We next find feasible orthogonal drawings $D(G(C_1))$, $D(G(C_2))$, $\cdots$, $D(G(C_l))$ in a recursive manner; $D(G(C_1))$ is feasible for the green path Q, and $D(G(C_i))$ is feasible for an arbitrary green path of C_i for each $i, 2 \leq i \leq l$.

Finally, patching the drawings $D(G(C_1)), D(G(C_2)), \cdots,$ $D(G(C_l))$ into $D(H)$, we can construct an orthogonal drawing $D(G(C))$ of $G(C)$ feasible for P; we replace the rectangle $x_1ty_1z_1$ of $D(H)$ by $D(G(C_1))$, and replace each vertex v_i, $2 \leq i \leq l$, by $D(G(C_i))$. In this case C is not always a rectangle in $D(G(C))$. One can observe with the help of Fig. 9 that each of the replacement above can be done without introducing any new bend or edge-crossing and without any conflict of coordinates of vertices as illustrated in Fig. 10. Note that the resulting drawing always expands outwards, satisfying the property (p3). Since we replace the rectangle $x_1ty_1z_1$ in $D(H)$ by $D(G(C_1))$ and we have already counted the bend corresponding to t for C_1, we need not count it for C. Thus one can observe that the drawing $D(G(C))$ has exactly $bc(C) = \sum_{i=1}^{l} bc(C_i)$ bends. Since a bend of $D(G(C_1))$ appears on Q, the bend appears on the green path P of C in $D(G(C))$. Hence $D(G(C))$ is an orthogonal drawing feasible for P. $\Box$

The definition of a feasible orthogonal drawing and Lemmas 10 and 11 immediately imply the following Lemma 12.

Lemma 12 *For any 3-legged cycle C in G, $b(G(C)) = bc(C)$, and a feasible orthogonal drawing of $G(C)$ has the minimum number $b(G(C))$ of bends.*

The algorithm for finding a feasible orthogonal drawing of $G(C)$ described in the proof of Lemma 11 above is hereafter called **Feasible-Draw**. We have the following lemma on **Feasible-Draw**.

Lemma 13 *Algorithm* **Feasible-Draw** *finds a feasible orthogonal drawing of $G(C)$ for a 3-legged cycle C in linear time, that is, in time $O(n(G(C)))$.*

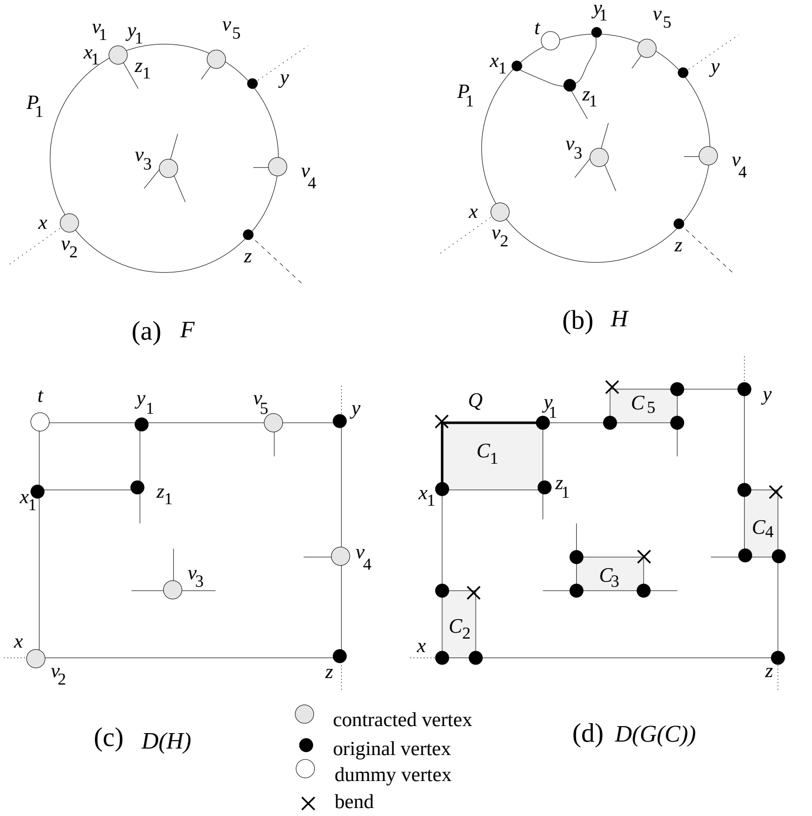


Figure 10: F, H, $D(H)$ and $D(G(C))$ for Case 3.

Proof: Denote by $T_{RG}(G)$ the computation time of **Rectangular-Draw(G)**. Since $T_{RG}(G) = O(n)$ [8], there is a constant c such that

$$T_{RG}(G) \quad \leq \quad c \cdot m(G) \tag{4}$$

for any connected plane graph G. By Lemma 3 one can find the genealogical tree T_C of C in linear time. By Lemma 8 one can classify the three contour paths as green or red paths for all cycles in $\mathcal{C}_C$ in linear time.

We first consider the time needed for contraction and patching operations. During the traversal of all inner faces of $G(C)$ for constructing T_C, we can find the three leg-vertices for each cycle in $\mathcal{C}_C$. Given the three leg-vertices of a 3-legged cycle, we can contract the 3-legged cycle to a vertex in constant time. Since $|\mathcal{C}_C| \leq n(G(C))/2$ by Lemma 2, 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 other than the contractions and patchings. Denote by $T(G(C))$ the time needed for **Feasible-Draw($G(C)$)** excluding the time for the contractions and patchings. We claim that $T(G(C)) = O(n(G(C)))$. The number $m(G(C))$ of edges in a plane graph $G(C)$ satisfies $m(G(C)) \leq 3n(G(C))$. Furthermore $|\mathcal{C}_C| \leq n(G(C))/2$ by Lemma 2. Therefore it suffices to prove that

$$T(G(C)) \quad \leq \quad c \cdot m(G(C)) + 4 \cdot c \cdot |\mathcal{C}_C|. \tag{5}$$

We prove (5) by induction based on T_C.

First consider the case where C has no child-cycle. Then $|\mathcal{C}_C| = 1$. In this case **Feasible-Draw** adds a dummy vertex on C to obtain a graph F from $G(C)$. Therefore $m(F) = m(G(C)) + 1$. **Feasible-Draw** finds a rectangular drawing of F by **Rectangular-Draw**. Hence, by (4) we have $T(G(C)) = T_{RG}(F) \leq c \cdot m(F)$. Thus $T(G(C)) \leq c \cdot m(G(C)) + 4 \cdot c \cdot |\mathcal{C}_C|$, as desired.

Next consider the case where C has child-cycles $C_1, C_2, \cdots, C_l$ where $l \geq 1$. Suppose inductively that (5) holds for each C_i, $1 \leq i \leq l$:

$$T(G(C_i)) \quad \leq \quad c \cdot m(G(C_i)) + 4 \cdot c \cdot |\mathcal{C}_{C_i}|. \tag{6}$$

Algorithm **Feasible-Draw** constructs a plane graph F from $G(C)$ by contracting each $G(C_i), 1 \leq i \leq l$, to a single vertex, and then constructs a graph H from F by either adding a dummy vertex on $C_o(F)$ or replacing exactly one contracted vertex on $C_o(F)$ by a quadrangle as illustrated in Figs. 8 and 10. Therefore one can observe that

$$m(H) + \sum_{i=1}^{l} m(G(C_i)) \quad \leq \quad m(G(C)) + 4. \tag{7}$$

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

$$T(G(C)) \quad = \quad T_{RG}(H) + \sum_{i=1}^{l} T(G(C_i)). \tag{8}$$

By (4) we have

$$T_{RG}(H) \quad \leq \quad c \cdot m(H). \tag{9}$$

Using (6), (7), (8) and (9), we have

$$T(G(C)) \quad \leq \quad c \cdot m(H) + \sum_{i=1}^{l} (c \cdot m(G(C_i)) + 4 \cdot c \cdot |\mathcal{C}_{C_i}|)$$

$$\begin{aligned} &= c \cdot (m(H) + \sum_{i=1}^{l} m(G(C_i))) + 4 \cdot c \cdot \sum_{i=1}^{l} |\mathcal{C}_{C_i}| \\ &\leq c \cdot (m(G(C)) + 4) + 4 \cdot c \cdot \sum_{i=1}^{l} |\mathcal{C}_{C_i}|. \qquad (10) \end{aligned}$$

Since $\mathcal{C}_C = \{C\} \cup (\bigcup_{i=1}^{l} \mathcal{C}_{C_i})$, we have

$$|\mathcal{C}_C| = 1 + \sum_{i=1}^{l} |\mathcal{C}_{C_i}|. \qquad (11)$$

By using (10) and (11), we have

$$T(G(C)) \leq c \cdot (m(G(C)) + 4) + 4 \cdot c \cdot (|\mathcal{C}_C| - 1) = c \cdot m(G(C)) + 4 \cdot c \cdot |\mathcal{C}_C|.$$

□

We are now ready to present our algorithm for orthogonal drawings of G, which is shown in Fig. 11.

Algorithm Orthogonal-Draw(G)
begin
1 add four dummy vertices of degree two on $C_o(G)$;
{if $C_o(G)$ has four or more edges, then add four dummy vertices on four distinct edges, otherwise, add two dummy vertices on two distinct edges and two dummy vertices on the remaining edge.}
2 let G' be the resulting graph;
3 let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles in G';
4 for each i, $1 \leq i \leq l$, construct genealogical trees T_{C_i} and determine green paths and red paths for every cycle in T_{C_i};
5 for each i, $1 \leq i \leq l$, find an orthogonal drawing $D(G(C_i))$ of $G(C_i)$ feasible for an arbitrary green path of C_i by **Feasible-Draw**;
6 let G'' be a plane graph derived from G' by contracting each $G(C_i)$, $1 \leq i \leq l$, to a single vertex v_i; {G'' has no bad cycle.}
7 find a rectangular drawing $D(G'')$ of G'' by **Rectangular-Draw**;
8 patch the drawings $D(G(C_1)), D(G(C_2)), \cdots, D(G(C_l))$ into $D(G'')$ to get an orthogonal drawing of G
end.

Figure 11: Algorithm **Orthogonal-Draw**.

Fig. 12(a) illustrates a rectangular drawing of G'' in Fig. 5(b). The specified green path of each of the maximal bad cycles C_1, C_2, C_3 and C_4 of G' is drawn by a thick line in Fig. 5(a). Fig. 12(b) illustrates a final orthogonal drawing of G' in Fig. 5(a).

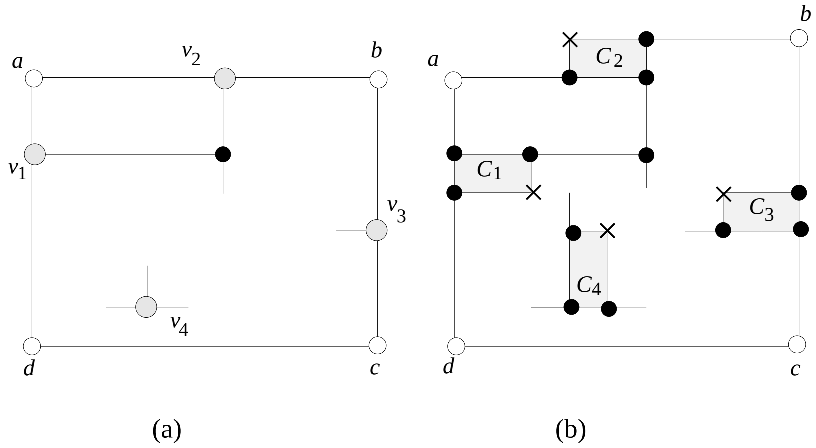


Figure 12: (a) A rectangular drawing of G'' and (b) an orthogonal drawing of G'.

We now have the following theorem.

Theorem 1 *Let G be a 3-connected cubic plane graph, let G' be the graph obtained from G by adding four dummy vertices in Algorithm* **Orthogonal-Draw**, *and let C_1, C_2, $\cdots$, C_l be the maximal bad cycles in G'. Then* **Orthogonal-Draw** *finds an orthogonal drawing of G with exactly $4+\sum_{i=1}^{l} bc(C_i)$ bends in linear time. Furthermore, we have $4+\sum_{i=1}^{l} bc(C_i) \leq 4 + b(G)$.*

Proof: (a) *Number of bends.*

There are two cases.

Case 1: G' has no bad cycle.

In this case we have a drawing with exactly four bends. By Fact 5 it is a drawing with the minimum number of bends.

Case 2: Otherwise.

Let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles in G'. For each i, $1 \leq i \leq l$, an orthogonal drawing $D(G(C_i))$ feasible for an arbitrary green path of C_i has exactly $bc(C_i)$ bends. Furthermore the rectangular drawing $D(G'')$ has exactly four bends corresponding to the four dummy vertices. Algorithm **Orthogonal-Drawing** patches the drawings $D(G(C_1)), D(G(C_2)), \cdots, D(G(C_l))$ into $D(G'')$ to get an orthogonal drawing of G. Therefore we have an orthogonal drawing of G with exactly $4+\sum_{i=1}^{l} bc(C_i)$ bends. Since the 3-legged cycles $C_1, C_2, \cdots, C_l$ are independent of each other, by Lemma 9 G has at least $\sum_{i=1}^{l} bc(C_i)$ vertex-disjoint 3-legged cycles. Therefore Fact 6 implies that $\sum_{i=1}^{l} bc(C_i) \leq b(G)$. Thus $4 + \sum_{i=1}^{l} bc(C_i) \leq 4 + b(G)$.

(b) *Time complexity.*

By a method similar to one in the proof of Lemma 3 we can find all maximal bad cycles in G' in linear time. **Orthogonal-Draw** calls **Rectangular-Draw** for G'' and **Feasible-Draw** for $G(C_i), 1 \leq i \leq l$. Both **Rectangular-Draw** and **Feasible-Draw** run in linear time. Since cycles $C_i, 1 \leq i \leq l$, are independent of each other, $\sum_{i=1}^{l} n(G(C_i)) \leq n$. Therefore the total time needed by **Feasible-Draw** is $O(n)$. Furthermore all contraction operations and all patching operations can be done in time $O(n)$ in total. Therefore **Orthogonal-Draw** runs in linear time. □

4 Bend Minimization

Algorithm **Orthogonal-Draw** in the preceding section finds an orthogonal drawing of a 3-connected cubic plane graph G with at most $b(G) + 4$ bends. In this section, by modifying **Orthogonal-Draw**, we obtain a linear-time algorithm **Minimum-Bend** to find an orthogonal drawing of G with the minimum number $b(G)$ of bends. Our idea behind **Minimum-Bend** is as follows.

If a 3-legged cycle in G has a green path on $C_o(G)$, then we can save one of the four bends mentioned in Fact 5, because a bend on the green path can be a bend on $C_o(G)$ and a bend on the 3-legged cycle at the same time; hence one of the four bends mentioned in Fact 5 has been accounted for by the bends necessitated by Fact 6. We therefore want to find as many such 3-legged cycles as possible, up to a total number of four. We had better to find a 3-legged cycle which has a green path on $C_o(G)$ but none of whose child-cycles has a green path on $C_o(G)$, because a bend on such a cycle can play also a role of a bend on its ancestor cycle. We call such a cycle a "corner cycle", that is, a *corner cycle* is a 3-legged cycle C in G such that C has a green path on $C_o(G)$ but no child-cycle of C has a green path on $C_o(G)$. (In Fig. 14(a) C'_1 and C'_2 drawn in thick lines are corner cycles. On the other hand, the two 3-legged cycles indicated by dotted lines are not corner cycles since C'_1 is their descendant cycle.) If G has $k(\leq 4)$ independent corner cycles $C'_1, C'_2, \cdots, C'_k$, then we can save k bends. By a method similar to one given in the proof of Lemma 3 one can find independent corner cycles of G as many as possible in linear time.

We are now ready to give the algorithm **Minimum-Bend** to find an orthogonal drawing with the minimum number of bends, which is shown in Fig. 13.

We have the following lemma.

Lemma 14 *Let C'_i be a corner cycle of a 3-connected cubic plane graph G. Then none of the child-cycles of C'_i has a green path on C'_i, and all contour paths of C'_i are green. (See Fig. 15 where C'_i is indicated by a dotted line.)*

Algorithm Minimum-Bend(G)
begin

1 find as many independent corner cycles $C'_1, C'_2, \cdots, C'_k$ of G as possible, up to a total number of four; $\{k \leq 4$. In Fig. 14(a) $k = 2.\}$

2 let $P'_{i1}, 1 \leq i \leq k$, be the green path of C'_i on $C_o(G)$;

3 let x'_i, y'_i and z'_i be the leg-vertices of C'_i, and let x'_i and y'_i be the ends of P'_{i1};

4 replace each subgraph $G(C'_i), 1 \leq i \leq k$, in G with a quadrangle $x'_i t'_i y'_i z'_i$ where t'_i is a dummy vertex of degree two, and let G^* be the resulting graph; {See Figs. 14(a) and (b).}

5 add $4 - k$ dummy vertices $t_1, t_2, \cdots, t_{4-k}$ on edges of $C_o(G^*)$ so that these vertices are adjacent to none of $t'_1, t'_2, \cdots, t'_k$ as in step 1 of **Orthogonal-Draw**, and let G' be the resulting graph; {See Fig. 14(c).}

6 let $C_1, C_2, \cdots, C_l$ be the maximal bad cycles in G' with respect to the four dummy vertices $t'_1, t'_2 \cdots, t'_k$ and $t_1, t_2 \cdots, t_{4-k}$ of degree two; {In Fig. 14(c) $l = 2$, and the insides of the two maximal bad cycles C_1 and C_2 are shaded.}

7 let G'' be a plane graph derived from G' by contracting each $G(C_i)$, $1 \leq i \leq l$, to a single vertex v_i; $\{G''$ has no bad cycle. See Fig. 14(d).}

8 find a rectangular drawing $D(G'')$ of G'' by **Rectangular-Draw**; {The drawing of $C_o(G'')$ in $D(G'')$ has exactly four corners $t'_1, t'_2 \cdots, t'_k$ and $t_1, t_2 \cdots, t_{4-k}$, and the quadrangle $x'_i t'_i y'_i z'_i$ is drawn as a rectangle for each i, $1 \leq i \leq k$, in $D(G'')$. See Fig. 14(e).}

9 find an orthogonal drawing $D(G(C'_i))$ of $G(C'_i)$ feasible for P'_{i1} for each i, $1 \leq i \leq k$, and find an orthogonal drawing $D(G(C_i))$ of $G(C_i)$ feasible for an arbitrary green path of C_i for each i, $1 \leq i \leq l$, by **Feasible-Draw**; { See Fig. 14(f).}

10 patch the drawings $D(G(C'_1)), D(G(C'_2)), \cdots, D(G(C'_k))$ and $D(G(C_1))$, $D(G(C_2))$, $\cdots$, $D(G(C_l))$ into $D(G'')$ to get an orthogonal drawing $D(G)$ of G { See Fig. 14(g).}

end.

Figure 13: Algorithm **Minimum-Bend**.

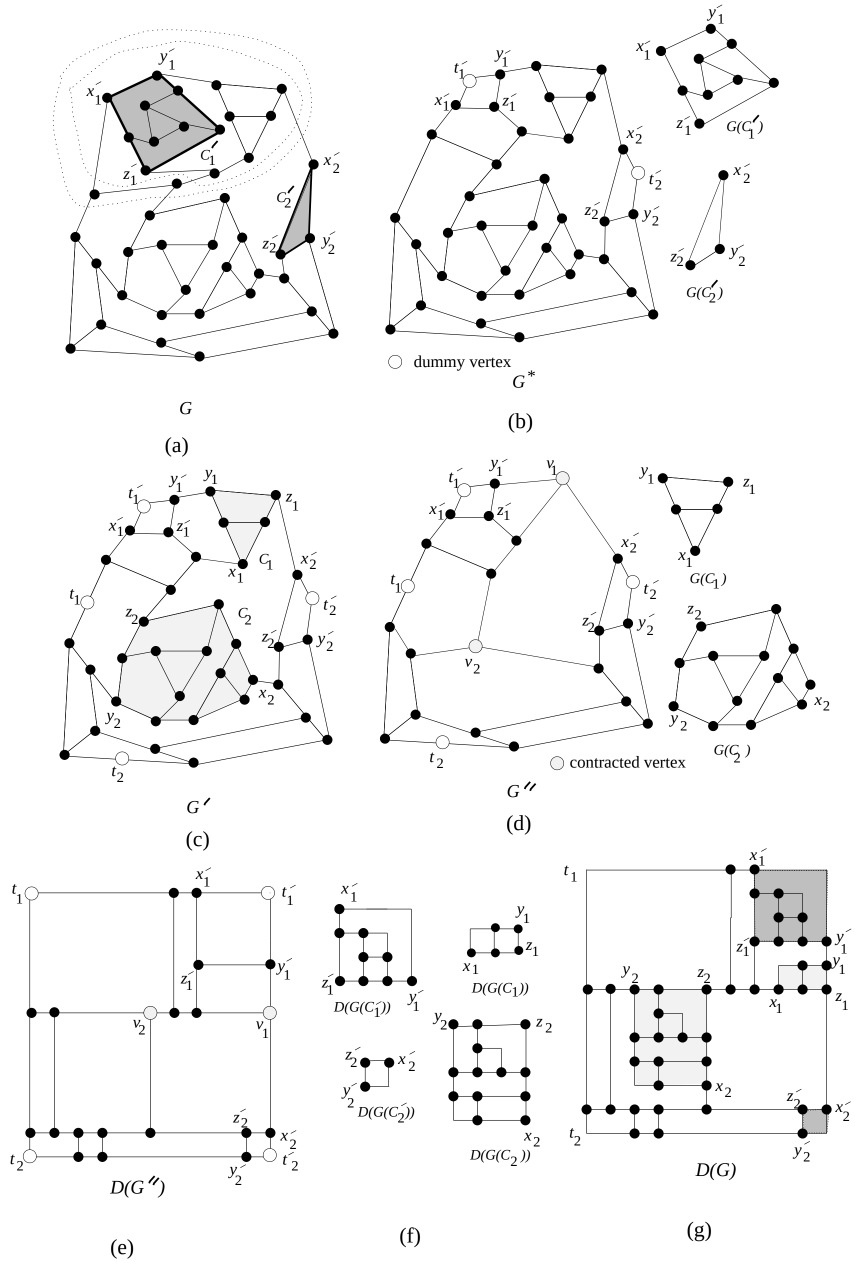


Figure 14: Illustration for Algorithm **Minimum-Bend**.

Proof: Let P'_{i1}, P'_{i2} and P'_{i3} be the contour paths of C'_i. According to the definition of a corner cycle, one of them, say P'_{i1}, is a green path on $C_o(G)$, but none of the child-cycles of C'_i has a green path on $C_o(G)$. (In Fig. 15 all green paths of the child-cycles of C'_i are drawn by thick lines.) Since P'_{i1} is on $C_o(G)$, none of the child-cycles of C'_i has a green path on P'_{i1}.

Furthermore none of the child-cycles of C'_i has a green path on P'_{i2} or P'_{i3}. Otherwise, according to Case 3 of the classification of contour paths, P'_{i1} would be a red path, a contradiction.

Thus none of the child-cycles of C'_i has a green path on C'_i. Therefore, according to Case 1 or 2 of the classification of contour paths, all contour paths P'_{i1}, P'_{i2} and P'_{i3} of C'_i are green. □

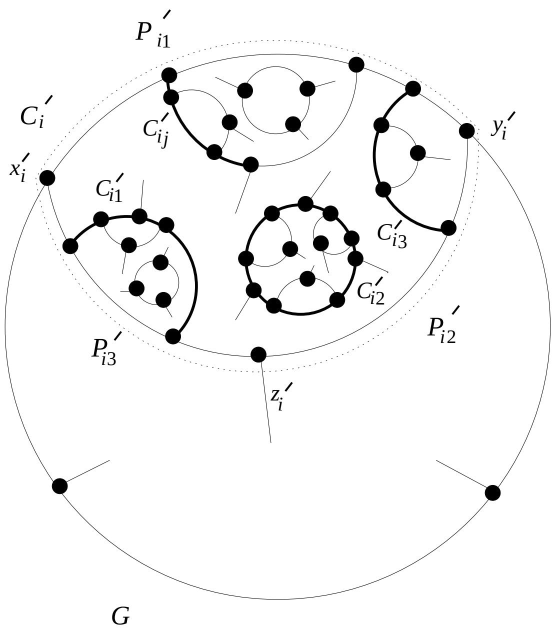


Figure 15: Corner cycle C'_i, its child-cycles, and their child-cycles.

We now have the following theorem.

Theorem 2 *Algorithm* **Minimum-Bend** *produces an orthogonal drawing of a 3-connected cubic plane graph G with the minimum number $b(G)$ of bends in linear time. Furthermore, we have*

$$b(G) = \sum_{i=1}^{k} bc(C'_i) + \sum_{i=1}^{l} bc(C_i) + 4 - k, \qquad (12)$$

where k, $C'_1, C'_2, \cdots, C'_k$ *and* $C_1, C_2, \cdots, C_l$ *are defined as in algorithm* **Minimum-Bend**.

Proof: (a) *Number of bends.*

We first show that **Minimum-Bend**(G) produces an orthogonal drawing of G with exactly $\sum_{i=1}^{k} bc(C'_i) + \sum_{i=1}^{l} bc(C_i) + 4 - k$ bends. For each i, $1 \le i \le k$, an orthogonal drawing $D(G(C'_i))$ feasible for P'_i has exactly $bc(C'_i)$ bends. Also, for each i, $1 \le i \le l$, an orthogonal drawing $D(G(C_i))$ feasible for an arbitrary green path of C_i has exactly $bc(C_i)$ bends. The rectangular drawing $D(G'')$ has exactly four dummy vertices $t'_1, t'_2 \cdots, t'_k$ and $t_1, t_2 \cdots, t_{4-k}$ of degree two, as illustrated in Fig. 14(e). Algorithm **Minimum-Bend** patches the drawings $D(G(C'_1)), D(G(C'_2)), \cdots, D(G(C'_k))$ and $D(G(C_1)), D(G(C_2)), \cdots, D(G(C_l))$ into $D(G'')$ to get an orthogonal drawing of G. The rectangle $x'_i t'_i y'_i z'_i$ in $D(G'')$ is replaced by $D(G(C'_i))$ for each i, $1 \le i \le k$ as illustrated in Fig. 14(g), and the bend corresponding to t'_i in the final drawing has been counted by $bc(C'_i)$. Therefore only $t_1, t_2, \cdots, t_{4-k}$ among the four dummy vertices in G'' should be counted as bends with $C_o(G)$ in the final drawing. Thus the final orthogonal drawing of G has exactly $\sum_{i=1}^{k} bc(C'_i) + \sum_{i=1}^{l} bc(C_i) + 4 - k$ bends.

Thus it suffices to show that $b(G) \ge \sum_{i=1}^{k} bc(C'_i) + \sum_{i=1}^{l} bc(C_i) + 4 - k$. There are two cases.

Case 1: $k = 4$.

Since cycles $C'_1, C'_2, \cdots, C'_k$ and $C_1, C_2, \cdots, C_l$ are independent of each other in G, by Lemma 9 G has at least $\sum_{i=1}^{k} bc(C'_i) + \sum_{i=1}^{l} bc(C_i)$ vertex-disjoint 3-legged cycles. Thus by Fact 6 $b(G) \ge \sum_{i=1}^{k} bc(C'_i) + \sum_{i=1}^{l} bc(C_i) = \sum_{i=1}^{k} bc(C'_i) + \sum_{i=1}^{l} bc(C_i) + 4 - k$.

Case 2: $k \le 3$.

For a corner cycle C'_i, $1 \le i \le k$, let $C'_{i1}, C'_{i2}, \cdots, C'_{il_i}$ be the child-cycles of C'_i in $\mathcal{C}_{C_i}$ where $l_i \ge 0$. By Lemma 14 all three contour paths P'_{i1}, P'_{i2} and P'_{i3} of C'_i are green, and none of $C'_{i1}, C'_{i2}, \cdots, C'_{il_i}$ has a green path on C'_i. (In Fig. 15 C'_i is indicated by a dotted line, and all green paths of $C'_{i1}, C'_{i2}, \cdots, C'_{il_i}$ are drawn by thick lines.) Therefore, by (1) or (2) we have

$$bc(C'_i) = 1 + \sum_{j=1}^{l_i} bc(C'_{ij}). \tag{13}$$

By Lemma 9 $G(C'_{ij})$, $1 \le j \le l_i$, has $bc(C'_{ij})$ vertex-disjoint 3-legged cycles which do not contain any edge on red paths of C'_{ij}. Therefore, if such a cycle contains an edge on C'_{ij}, then the edge is necessarily on a green path of C'_{ij}, which is not on C'_i. Thus none of these cycles contains any edge on C'_i, and hence contains any edge on $C_o(G)$. Therefore, by (13), $G(C'_i)$ has $\sum_{j=1}^{l_i} bc(C'_{ij}) = bc(C'_i) - 1$ vertex-disjoint 3-legged cycles which do not contain any edge on $C_o(G)$.

Since $k \le 3$, none of the maximal bad cycles C_i, $1 \le i \le l$, of G' has a green path on $C_o(G)$; otherwise, such a cycle C_i or its descendant cycle would be a corner cycle of G and hence G would have $k+1$ (≤ 4) independent corner cycles, a contradiction. Therefore only a red path of C_i can be on $C_o(G)$. However, by Lemma 9, $G(C_i)$ has $bc(C_i)$ vertex-disjoint 3-legged cycles of G which do not contain any edge on red paths of C_i. Hence these $bc(C_i)$ cycles in $G(C_i)$ do not contain any edge on $C_o(G)$.

Thus G has $\sum_{i=1}^{k}(bc(C'_i) - 1) + \sum_{i=1}^{l} bc(C_i)$ vertex-disjoint 3-legged cycles which do not contain any edge on $C_o(G)$ since cycles $C'_1, C'_2, \cdots, C'_k$ and $C_1, C_2, \cdots, C_l$ are independent of each other. Therefore by Fact 6 at least $\sum_{i=1}^{k}(bc(C'_i) - 1) + \sum_{i=1}^{l} bc(C_i)$ bends must appear in the proper inside of $C_o(G)$. By Fact 5 at least four bends must appear on $C_o(G)$. Thus we have $b(G) \geq \sum_{i=1}^{k}(bc(C'_i)-1)+\sum_{i=1}^{l} bc(C_i)+4 = \sum_{i=1}^{k} bc(C'_i)+\sum_{i=1}^{l} bc(C_i)+4-k$.

This completes a proof of (12).

(b) *Time complexity.*

Similar to (b) in the proof of Theorem 1. □

5 Grid Drawing

In this section we give our bounds on the grid size for an orthogonal grid drawing corresponding to an orthogonal drawing obtained by the algorithm **Minimum-Bend**.

An orthogonal drawing is called an *orthogonal grid drawing* if all vertices and bends are located on integer grid points. Given an orthogonal drawing, one can transform it to an orthogonal grid drawing in linear time [10, 2 (pp. 157–161)]. Let W be the *width* of a grid, that is the number of vertical lines in the grid minus one, and let H be the *height* of a grid. Let n be the number of vertices, and let m be the number of edges in a given graph. It is known that any orthogonal drawing using b bends has a grid drawing on a grid such that $W+H \leq b+2n-m-2$ [1]. It is also known that any 3-connected cubic plane graph has an orthogonal grid drawing using at most $\frac{n}{3}+3$ bends on a grid such that $W \leq \frac{n}{2}$ and $H \leq \frac{n}{2}$ [1, 5].

Given a 3-connected cubic plane graph G, one can find in linear time an orthogonal drawing of G with the minimum number $b(G)$ of bends using our algorithm **Minimum-Bend**, then one can also transform it in linear time to an orthogonal grid drawing with the same number of bends using the algorithm in [10, 2]. The grid size of a produced drawing satisfies $W+H \leq b(G)+2n-m-2 = b(G) + \frac{1}{2}n - 2$ [1].

In the rest of this section we will prove that any orthogonal drawing produced by our algorithm **Minimum-Bend** can be transformed to an orthogonal grid drawing on a grid such that $W \leq \frac{n}{2}$ and $H \leq \frac{n}{2}$. We have the following known result on the grid size of a rectangular grid drawing [8].

Lemma 15 *Any rectangular drawing of a plane graph G produced by Algorithm* **Rectangular-Draw** *can be transformed to a rectangular grid drawing on a grid* *such that* $W + H \leq \frac{n}{2}$.

We now show that the following lemma holds for an orthogonal grid drawing of $G(C)$ for a 3-legged cycle C in G.

Lemma 16 *Let C be a 3-legged cycle in a 3-connected cubic plane graph G. Then an orthogonal drawing of $G(C)$ produced by Algorithm* **Feasible-Draw** *can be transformed to an orthogonal grid drawing on a grid such that* $W \leq \frac{n(G(C))-1}{2}$ *and* $H \leq \frac{n(G(C))-1}{2}$.

Proof: We only give a proof for the bound on W since the proof for the bound on H is similar. We prove the bound on W by induction based on T_C.

First consider the case where C has no child-cycle. In this case **Feasible-Draw** adds a dummy vertex on C to obtain a graph F from $G(C)$. Therefore $n(F) = n(G(C)) + 1$. **Feasible-Draw** finds a rectangular drawing of F by **Rectangular-Draw**. By Lemma 15 the rectangular drawing of F can be transformed to a rectangular grid drawing on a grid such that $W + H \leq \frac{n(F)}{2} = \frac{n(G(C))+1}{2}$. The rectangular grid drawing of F immediately gives an orthogonal grid drawing of $G(C)$ on the same grid regarding the dummy vertex as a bend. Therefore the width W and the height H of the grid required for the orthogonal grid drawing of $G(C)$ satisfies $W + H \leq \frac{n(G(C))+1}{2}$. One can easily observe that $H \geq 1$ for any orthogonal grid drawing of $G(C)$. Therefore, for a grid required for the orthogonal grid drawing of $G(C)$ corresponding to the orthogonal drawing of $G(C)$ obtained by **Feasible-Draw**, $W \leq \frac{n(G(C))+1}{2} - 1 = \frac{n(G(C))-1}{2}$.

Next consider the case where C has child-cycles $C_1, C_2, \cdots, C_l$ where $l \geq 1$. Suppose inductively that the following bound on the width W_i of a grid required for the orthogonal grid drawing of each $G(C_i)$, $1 \leq i \leq l$ holds:

$$W_i \leq \frac{n(G(C_i)) - 1}{2}. \tag{14}$$

Algorithm **Feasible-Draw** constructs a plane graph F from $G(C)$ by contracting each $G(C_i), 1 \leq i \leq l$, to a single vertex, and then constructs a graph F' from F by either adding a dummy vertex on $C_o(F)$ or replacing exactly one contracted vertex on $C_o(F)$ by a quadrangle as illustrated in Figs. 8 and 10 where $F' = H$.

Consider the case where F' is constructed from F by adding a dummy vertex. In this case $n(F') = n(F) + 1$. Algorithm **Feasible-Draw** patches orthogonal drawings of $G(C_i), 1 \leq i \leq l$, into a rectangular drawing $D(F')$ of F' found by **Rectangular-Draw**. Therefore

$$W \leq W_{F'} + \sum_{i=1}^{l} W_i, \tag{15}$$

where $W_{F'}$ is the width of the grid required for the rectangular grid drawing of F'. By Lemma 15 $W_{F'} + H_{F'} \le \frac{n(F')}{2} = \frac{n(F)+1}{2}$, where $H_{F'}$ is the height of the grid required for the rectangular grid drawing of F'. Since F' has at least four vertices, $H_{F'} \ge 1$. Hence

$$\begin{aligned} W_{F'} &\le \frac{n(F)+1}{2} - 1 \\ &= \frac{n(F)-1}{2}. \end{aligned} \tag{16}$$

From (14), (15) and (16) we have

$$\begin{aligned} W &\le \frac{n(F)-1}{2} + \sum_{i=1}^{l} \frac{n(G(C_i))-1}{2} \\ &= \frac{n(F) + \sum_{i=1}^{l}(n(G(C_i))-1) - 1}{2}. \end{aligned} \tag{17}$$

During the patching operation exactly one contracted vertex is replaced by the orthogonal drawing of each $G(C_i)$, and hence

$$n(F) + \sum_{i=1}^{l}(n(G(C_i)) - 1) = n(G(C)). \tag{18}$$

From (17) and (18) we have $W \le \frac{n(G(C))-1}{2}$.

We now consider the case where F' is constructed from F by replacing exactly one contracted vertex on $C_o(F)$ by a quadrangle. In this case $n(F') = n(F)+3$. As in the former case, Algorithm **Feasible-Draw** patches orthogonal drawings of $G(C_i), 1 \le i \le l$, into a rectangular drawing $D(F')$ of F' found by **Rectangular-Draw**. During the patching operation one of $G(C_i)$, $1 \le i \le l$, say $G(C_1)$, replaces the quadrangle, and each $G(C_i)$, $2 \le i \le l$ replaces exactly one contracted vertex in F'. Furthermore, any drawing of a quadrangle on a grid has width at least one. Therefore the following equation holds:

$$W \le W_{F'} + (W_1 - 1) + \sum_{i=2}^{l} W_i, \tag{19}$$

where $W_{F'}$ is the width of the grid required for the rectangular grid drawing of F'. By Lemma 15 $W_{F'} + H_{F'} \le \frac{n(F')}{2} = \frac{n(F)+3}{2}$, where $H_{F'}$ is the height of the grid required for the rectangular grid drawing of F'. Since $H_{F'} \ge 1$,

$$W_{F'} \le \frac{n(F)+1}{2}. \tag{20}$$

From (14), (19) and (20) we have,

$$\begin{aligned} W &\leq \frac{n(F)+1}{2} + \frac{n(G(C_1))-1}{2} - 1 + \sum_{i=2}^{l} \frac{n(G(C_i))-1}{2} \\ &= \frac{n(F) + \sum_{i=1}^{l}(n(G(C_i))-1) - 1}{2}. \end{aligned} \tag{21}$$

Clearly

$$n(F) + \sum_{i=1}^{l}(n(G(C_i))-1) = n(G(C)). \tag{22}$$

From (21) and (22) we have $W \leq \frac{n(G(C))-1}{2}$. □

We are now ready to show $W \leq \frac{n}{2}$ and $H \leq \frac{n}{2}$ for a grid required for an orthogonal grid drawing corresponding to an orthogonal drawing of G produced by **Minimum-Bend**. We here use the same notations used in **Minimum-Bend**. By Lemma 15, the rectangular drawing of G'' has a corresponding rectangular grid drawing on a grid such that $W_{G''} + H_{G''} \leq \frac{n(G'')}{2}$, where $W_{G''}$ and $W_{G''}$ respectively are the width and the height of the grid. Since G is a 3-connected plane graph, $H_{G''} \geq 2$. Hence

$$W_{G''} \leq \frac{n(G'')}{2} - 2. \tag{23}$$

Algorithm **Minimum-Bend** patches the orthogonal drawings of $G(C'_1)$, $G(C'_2)$, $\cdots, G(C'_k)$ and $G(C_1), G(C_2), \cdots, G(C_l)$ into the rectangular drawing of G'' and get an orthogonal drawing of G. During the patching operation, the drawing of each $G(C'_i)$, $1 \leq i \leq k$, replaces the drawing of a quadrangle, and the drawing of each $G(C_i)$, $1 \leq i \leq l$, replaces a contracted vertex in G''. The width of a quadrangle on a grid is at least one. Thus one can observe that the width W of a grid required for an orthogonal grid drawing of G obtained by algorithm **Minimum-Bend** satisfies the following relation.

$$W \leq W_{G''} + \sum_{i=1}^{k}(W'_i - 1) + \sum_{i=1}^{l} W_i, \tag{24}$$

where W'_i is the width of the grid required for an orthogonal grid drawing of $G(C'_i)$ for $1 \leq i \leq k$ and W_i is the width of the grid required for an orthogonal grid drawing of $G(C_i)$ for $1 \leq i \leq l$. Then by Lemma 16, Eqs. (23) and (24) we have

$$\begin{aligned} W &\leq \frac{n(G'')}{2} - 2 + \sum_{i=1}^{k}(\frac{n(G(C'_i))-1}{2} - 1) + \sum_{i=1}^{l} \frac{n(G(C_i))-1}{2} \\ &= \frac{n(G'') + \sum_{i=1}^{k}(n(G(C'_i))-3) + \sum_{i=1}^{l}(n(G(C_i))-1) - 4}{2}. \end{aligned} \tag{25}$$

One can observe that

$$n(G'') + \sum_{i=1}^{k} (n(G(C'_i)) - 3) + \sum_{i=1}^{l} (n(G(C_i)) - 1) - 4 = n. \qquad (26)$$

From (25) and (26) we have $W \leq \frac{n}{2}$. Similarly we can prove $H \leq \frac{n}{2}$. Thus we have the following theorem.

Theorem 3 *Let G be a 3-connected cubic plane graph with n vertices. Any orthogonal drawing of G with the minimum number $b(G)$ of bends produced by Algorithm* **Minimum-Bend** *has a corresponding orthogonal grid drawing on a grid with width H and height H such that $W + H \leq b(G) + \frac{1}{2}n - 2$, $W \leq \frac{n}{2}$ and $H \leq \frac{n}{2}$.*

6 Conclusions

In this paper we have presented a linear-time algorithm to find an orthogonal drawing of a 3-connected cubic plane graph with the minimum number of bends. It is left as future work to find a linear-time algorithm for a larger class of graphs.

Acknowledgement: We wish to thank the three anonymous referees for their valuable comments and suggestions for improving the presentation of the paper.

References

[1] T. C. Biedl. Optimal orthogonal drawings of triconnected plane graphs. *Proc. of SWAT'96, Lect. Notes in Computer Science*, 1097, pp. 333-344, 1996.

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

[3] A. Garg and R. Tamassia. On the computational complexity of upward and rectilinear planarity testing. *Proc. of Graph Drawing'94, Lect. Notes in Computer Science*, 894, pp. 286-297, 1995.

[4] 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. 201-226, 1997.

[5] G. Kant. Drawing planar graphs using the canonical ordering., *Algorithmica*, 16, pp. 4-32, 1996.

[6] 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.

[7] M. S. Rahman, S. Nakano and T. Nishizeki. A linear algorithm for optimal orthogonal drawings of triconnected cubic plane graphs. *Proc. of Graph Drawing'97, Lect. Notes in Computer Science*, 1353, pp. 99-110, 1998.

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

[9] J. Storer. On minimum node-cost planar embeddings. *Networks*, 14, pp. 181-212, 1984.

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

[11] R. Tamassia, I. G. Tollis and J. S. Vitter. Lower bounds for planar orthogonal drawings of graphs. *Inf. Proc. Letters*, 39, pp. 35-40, 1991.

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