数理解析研究所講究録
1185 巻 2001 年 160-168

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

(Extended Abstract)

Shin-ichi NAKANO and Makiko YOSHIKAWA

Gunma University
Kiryu 376-8515, Japan
nakano@cs.gunma-u.ac.jp

Abstract: An orthogonal drawing of a plane graph G is a drawing of G with the given planar 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. Observe that only a planar graph with the maximum degree four or less has an orthogonal drawing. The best known algorithm to find an orthogonal drawing runs in time $O(n^{7/4}\sqrt{\log n})$ for any plane graph with n vertices. In this paper we give a linear-time algorithm to find an orthogonal drawing of a given biconnected cubic plane graph with the minimum number of bends.

Keywords: Graph Drawings, Algorithms, Orthogonal Drawings.

1 Introduction

An *orthogonal drawing* of a plane graph G is a drawing of G with the given planar 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 its numerous practical applications in circuit schematics, etc. [BLV93, K96, T87]. 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, eleven bends.

For a given planar graph G, if it is allowed to choose its planar embedding, then finding an orthogonal drawing of G with the minimum number of bends is NP-complete[GT94]. However, Tamassia[T87] and Garg and Tamassia [GT96] presented algorithms which find 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 unless it is allowed to choose its planar embedding, where n is the number of vertices in G. 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 an orthogonal drawing of a plane graph with a presumably small number of bends[K96], and for 3-connected cubic plane graphs a linear-time algorithm is known for finding an orthogonal drawing with the minimum number of bends[RNN99]. Observe that only a planar graph with the maximum degree four or less has an orthogonal drawing.

In this paper, generalizing the result in [RNN99], we give a linear-time algorithm to find an orthogonal drawing of a biconnected cubic plane graph with the minimum number of bends.

An orthogonal drawing in which there is no bend and each face is drawn as a rectangle is called a *rectangular drawing*. Given a plane graph G such that every vertex has degree either two or three, in linear-time we can find a rectangular drawing of G whenever such a graph has a rectangular drawing [KH94, RNN96, RNN00]. The key idea of our algorithm is to reduce the orthog-

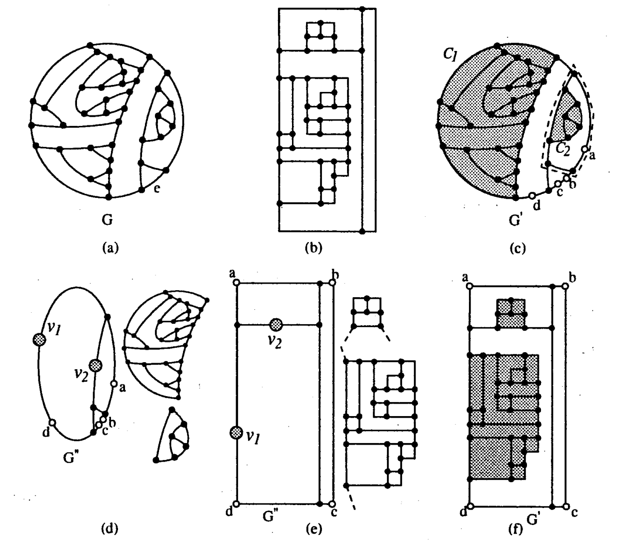

Figure 1: A plane graph and its orthogonal drawing.

onal drawing problem to the rectangular drawing problem.

An outline of our algorithm is illustrated in Fig. 1. Given a plane graph G as shown in Fig. 1(a), we first find a tree structure among some cycles in G, then by analyzing the tree structure we 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. The four dummy vertices are drawn by white circles in Fig. 1(c). We then contract each of some cycles $C_1, C_2, \cdots$ and their insides (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 insides 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' 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 a known result. Section 3 shows a tree structure among some cycles in G. Section 4 presents an algorithm to find an orthogonal drawing with the minimum number of bends. Finally Section 5 is a conclusion.

2 Preliminaries

In this section we give some definitions and present a known result.

Let G be a connected graph with n vertices. An edge connecting vertices x and y is denoted by (x, y). 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 planar 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_1 and C_2 in a plane graph G are *independent* if $G(C_1)$ and $G(C_2)$ have no common vertex. Cycles C_1 and C_2 are *vertex-disjoint* if C_1 and C_2 have no common vertex. An edge which is incident to exactly one vertex of a simple cycle C and located outside of C is called a *leg* of the cycle C, and the vertex on C to which the leg is incident is called a *leg-vertex* of C. A simple cycle with exactly k legs is called a *k-legged cycle.* For k-legged cycle C the k subpaths of C dividing C at the k leg-vertices are called *the contour paths* of C.

An *orthogonal drawing* of a plane graph G is a drawing of G with the given planar 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 point where an edge changes its direction in a drawing is called a *bend.* We denote by $b(G)$ the minimum number of bends for orthogonal drawings of G. An orthogonal drawing of G with exactly $b(G)$ bends is *bend-optimal.*

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. In any rectangular drawing D of G, the four corners of the rectangle corresponding to $C_o(G)$ are vertices of degree two on $C_o(G)$. We call these four vertices *the corner vertices* of D. The following result on rectangular drawings is known.

Lemma 2.1 *Let G be a connected plane graph such that every vertex has degree either two or three, and let a, b, c, d be four designated vertices of degree two on $C_o(G)$. Then G has a rectangular drawing with the corner vertices a, b, c, d if and only if G has none of the following three types of simple cycles [T84]:*

(r1) 1-legged cycles,
(r2) 2-legged cycles which contain at most one designated vertex of degree two, and
(r3) 3-legged cycles which contain no designated 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 [RNN96, RNN00].

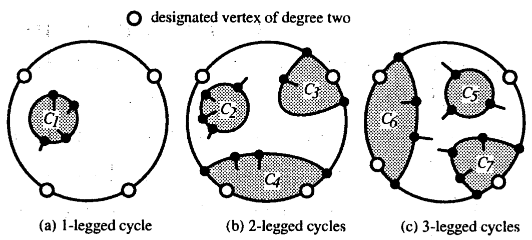

Figure 2: Bad cycles C_1, C_2, C_3 and C_5, and non-bad cycles C_4, C_6 and C_7.

A cycle of type (r1), (r2) or (r3) is called a *bad cycle*. Figs. 2(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 designated vertices of degree two, and C_6 and C_7 are 3-legged cycles but contain one or two designated vertices of degree two.

Some linear-time algorithms to find a rectangular drawing of a plane graph satisfying the condition in Lemma 2.1 have been obtained [KH94, RNN96, RNN00].

3 Genealogical Tree

In this section we first show a tree structure among some cycles in a biconnected cubic plane graph G.

Let G be a biconnected cubic plane graph. For a pair of distinct cycles C_a and C_d in G, C_d is called a *descendant-cycle* of C_a if (i) C_d is either 2- or 3-legged cycle, and (ii) $G(C_d)$ is a proper subgraph of $G(C_a)$. Note that since G is biconnected there is neither 0- nor 1-legged cycle except the only 0-legged cycle $C_o(G)$. Now we choose an edge $e = (x, y)$ on $C_o(G)$, and replace e with two edges (x, z) and (z, y). Let G' be the resulting plane graph. (Note that, for $G - e$, that is a plane subgraph of G obtained from G by deleting e, $C_o(G - e)$ is a 2-legged cycle of G', however, $C_o(G - e)$ is not a 2-legged cycle of G.) Let $D_e(C_o) = \{C | C$ is a descendant cycle of $C_o(G')$ not containing $z\}$. A cycle C_c in $D_e(C_o)$ is called a *child-cycle* of $C_o(G')$ (with respect to edge e) if C_c is not located inside of any other cycle in $D_e(C_o)$. Since G is a biconnected cubic plane graph, $C_o(G')$ has exactly one child-cycle $C_o(G - e)$ (with respect to edge e). (See Fig 3.) Then, recursively, for each child-cycle C_c we define its child-cycle as follows. We have the following two cases.

Case 1: C_c is a 2-legged cycle.

Choose a leg-vertex of C_c as z. Let $D_z(C_c) = \{C | C$ is a descendant cycle of C_c not containing z $\}$. A cycle C_{cc} in $D_z(C_c)$ is called a *child-cycle* of C_c (with respect to z) if C_{cc} is not located inside of any other cycle in $D_z(C_c)$. Since G is a biconnected cubic plane graph, C_c has at most one 3-legged child-cycle. (C_c has no 3-legged child-cycle if $G(C)$ has an inner face F containing the two leg-vertices, and C_c has exactly one 3-legged child-cycle otherwise.)

Case 2: Otherwise, C_c is a 3-legged cycle.

Let $D(C_c)$ be the set of all descendant cycles of C_c. A cycle C_{cc} in $D(C_c)$ is called a *child-cycle* of C_c if C_{cc} is not located inside of any other cycle in $D(C_c)$.

In both cases above all child-cycles of C_c are independent each other.

By the definition above we can find child-cycles of each child-cycle recursively, and eventually we get a (hierarchical) tree structure of cycles in G represented by a "genealogical tree" T_g, as shown in Fig 3. Because of the choices for e and z, T_g may have some variations. We choose an arbitrary (but fixed) one as T_g.

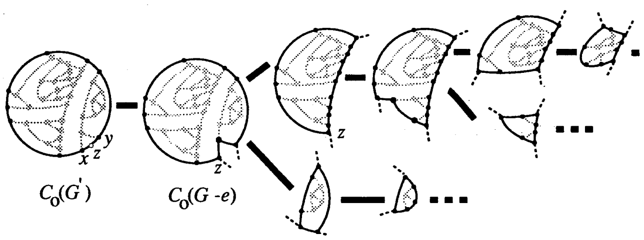

Figure 3: cycles in G' and a genealogical tree T_g.

Using a method similar to one in [RNN96, RNN99, RNN00], in linear time one can find such a tree structure T_g among cycles by traversing the contour of each face a constant number of times.

Now we observe the following. In any orthogonal drawing of G, every cycle C in G has at least four convex corners, i.e., polygonal vertices of inner angle 90°. 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 1 At least four bends must appear on $C_o(G)$.

Fact 2 At least two bend must appear on each 2-legged cycle in G.

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

4 Orthogonal Drawing

In this section we give a linear-time algorithm to find a bend-optimal orthogonal drawing of a biconnected cubic plane graph. Assume that we have a genealogical tree T_g of a biconnected cubic plane graph G. We need some definitions.

We define "feasible drawings" as follows. Note that rotated cases are omitted.

Let C be a 2-legged cycle with the two leg-vertices x and y, and P_1 and P_2 be the clockwise contour paths from x to y and from y to x, respectively. A bend-optimal orthogonal drawing D of $G(C)$ is *feasible* for (P_1, P_1) if none of the following four open halflines intersects D. (See Fig. 4(a). Intuitively D needs two convex bends on P_1.)

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

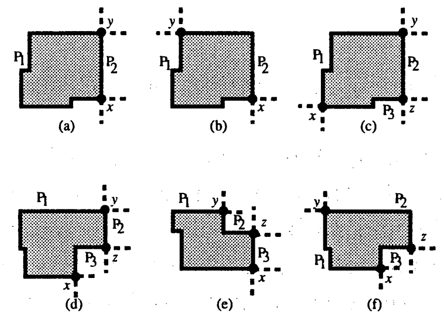

Figure 4: Illustration for feasible drawings.

Also, a bend-optimal orthogonal drawing D of $G(C)$ is *feasible* for (P_1, P_2) if none of the four open halflines depicted in dashed lines in Fig. 4(b) intersects D.

Let C be a 3-legged cycle with the three leg-vertices x, y and z appearing clockwise in this order, and P_1, P_2 and P_3 be the clockwise contour path from x to y, from y to z, and from z to x, respectively. A bend-optimal orthogonal drawing D of $G(C)$ is *feasible* for (P_1) if none of the six open halflines depicted in dashed lines in Fig. 4(c) intersects D. Similarly, we define *feasible* orthogonal drawings for $(P_1, P_1, -P_3)$, $(P_1, P_1, -P_2)$ and $(P_1, P_2, -P_3)$.(See Fig. 4(d)–(f).)

Now, for each cycle $C \neq C_o(G)$ corresponding to a vertex in T_g, we determine whether $G(C)$ has each type of feasible drawings by a bottom-up computation on T_g. For the bottom-up computation we also compute a set S_C of vertex-disjoint cycles in $G(C)$ consisting of ℓ_2 2-legged cycles and ℓ_3 3-legged cycles for some ℓ_2 and ℓ_3. Thus $b(G(C)) \geq 2 \cdot \ell_2 + \ell_3$ by Facts 3.2 and 3.3. We then show that $G(C)$ always has at least one feasible drawing using $2 \cdot \ell_2 + \ell_3$ bends. Thus $b(G(C)) = 2 \cdot \ell_2 + \ell_3$ holds.

In the bottom-up computation we classify each contour path of each cycle as either *0-, 1-, or 2-corner path.* Intuitively k-corner path has a chance to have k convex bends. And we define

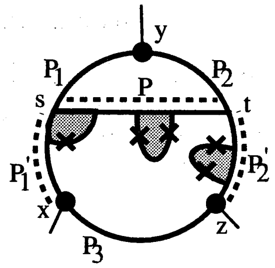

Figure 5: Illustration for P_1P_2-strain.

P_1P_2-strain by those corner paths as follows. Let x, y, z be the three leg-vertices of a 3-legged cycle C, P_1 and P_2 be the clockwise contour paths from x to y and y to z, respectively. Assume that s and t are vertices on P_1 and P_2, respectively, and let P_1' be the subpath of P_1 from x to s, and P_2' be the subpath of P_2 from t to z. If (i) there is a path P from s to t such that the left side of P is an inner face of $G(C)$, and (ii) $G(C)$ has no child cycle having 1- or 2-corner path on P, P_1' or P_2', then the path consisting of P_1', P, P_2' are called *P_1P_2-strain.* An example is illustrated in Fig. 5. Intutively, we have only two chance to turn right at s and t on P_1P_2-strain from x to z.

In the bottom-up computation we show that the following conditions (c1) – (c9) hold.

- (c1) Any cycle C has at least one 1- or 2-corner path.
- (c2) No cycle in S_C contains any edge on any 0-corner path of C.
- (c3) For any 2-legged cycle C if C has a 1-corner path P_1, then $G(C)$ has a set S_C' of vertex-disjoint cycles containing no edge on P_1 and consisting of ℓ_2' 2-legged cycles and ℓ_3' 3-legged cycles such that $2 \cdot \ell_2' + \ell_3' = b(G(C)) - 1$.
- (c4) For any 2-legged cycle C if C has a 0-corner path P_1, then the other contour path P_2 is a 2-corner path, and $G(C)$ has an orthogonal drawing feasible for (P_2, P_2).
- (c5) For any 3-legged cycle C if C has a 1-corner path P_1, then $G(C)$ has a set S_C' of vertex-disjoint cycles containing no edge on P_1, and consisting of ℓ_2' 2-legged cycles and ℓ_3' 3-legged cycles such that $2 \cdot \ell_2' + \ell_3' = b(G(C)) - 1$.
- (c6) For any 3-legged cycle C if C has a 1- or 2-corner path P_1, then $G(C)$ has an orthogonal drawing feasible for (P_1).
- (c7) For any 3-legged cycle C if C has a 2-corner path P_1 and no P_1P_2-strain , then $G(C)$ has an orthogonal drawing feasible for $(P_1, P_1, -P_3)$,
- (c8) For any 3-legged cycle C if C has a 2-corner path P_1 and no P_3P_1-strain, then $G(C)$ has an orthogonal drawing feasible for $(P_1, P_1, -P_2)$,
- (c9) For any 3-legged cycle C if C has 1-corner paths P_1 and P_2, and no P_1P_2-strain, then $G(C)$ has an orthogonal drawing feasible for $(P_1, P_2, -P_3)$.

Now we explain the bottom-up computation in the following four cases.

Case 1: C is a 2-legged cycle having no child-cycle.

Let x, y be the two leg-vertices of C, let P_1 and P_2 be the clockwise contour paths from x to y and from y to x, respectively. Now $G(C) = C$, since for any 2-legged cycle C if $G(C)$ has an edge in proper inside of C then C always has a child-cycle.

Computation for S_C: Set $S_C = \{C\}$. By Fact 3.2 any orthogonal drawing of $G(C)$ has at least two bends.

Feasible drawings: By introducing two bends on P_1, we can easily construct an orthogonal drawing of $G(C)$ feasible for (P_1, P_1). Similarly we can construct orthogonal drawings of $G(C)$ feasible for (P_2, P_2) and (P_1, P_2), respectively. Thus $G(C)$ has each type of feasible orthogonal drawings.

Classification and proof for (c1)–(c9): In this case every contour path of C is classified as a 2-corner paths. Conditions (c1)–(c4) hold since every contour path of C is 2-corner, and (c5)–(c9) hold since C is not a 3-legged cycle.

Case 2: C is a 3-legged cycle having no child-cycle.

Let x, y, z be the three leg-vertices of C, let P_1, P_2, P_3 be the clockwise contour path from x to y, from y to z, and from z to x, respectively. Now if we remove all edges on C from $G(C)$, then either $G(C) = C$ or the remaining edges induce a connected graph containing at least one vertex on each P_1, P_2, P_3, since otherwise C has a child-cycle, a contradiction.

Computation for S_C: Set $S_C = \{C\}$. By Fact 3.3 any orthogonal drawing of $G(C)$ has at least one bend.

Feasible drawings: Construct a new graph G' from $G(C)$ by adding one dummy vertices v on P_1. Now the resulting graph G' has no bad cycle (since G has no child-cycle) with respect to corner vertices x, v, y, z, and then G' has a rectangular drawing with the corner vertices x, v, y, z. The rectangular drawing is also an orthogonal drawing of $G(C)$ feasible for (P_1) using exactly one bend (corresponding to v). Similarly we can easily construct orthogonal drawings of $G(C)$ feasible for (P_2) and (P_3).

Now $G(C)$ has no orthogonal drawing feasible for $(P_1, P_1, -P_2)$, since it needs at least two bends only on P_1. Similarly $G(C)$ has no orthogonal drawing feasible for $(P_i, P_j, -P_k)$ for any $i, j, k \in \{1, 2, 3\}$.

Classification and proof for (c1)–(c9): In this case every contour path of C is classified as a 1-corner path. Conditions (c1),(c2) hold since every contour path of C is 1-corner, (c3),(c4) hold since C is not a 2-legged cycle, (c5) holds by choosing $S'_C = \phi$, (c6) holds since $G(C)$ has orthogonal drawings feasible for (P_1), (P_2), (P_3), respectively, as mentioned above, and (c7)–(c9) hold since $G(C)$ has no 2-corner path.

Case 3: C is a 2-legged cycle having one or more child-cycles.

Let x, y be the two leg-vertices of C, and let P_1 and P_2 be the clockwise contour paths from x to y and from y to x, respectively. If $G(C)$ has an inner face containing x and y, then C has no 3-legged child-cycle, otherwise, C has exactly one 3-legged child-cycle, which contains exactly one leg-vertices of C. Thus C has at most one 3-legged child-cycle.

Let $C_1, C_2, \cdots, C_\ell$ be the child-cycle of C. Assume that for C_i, $1 \le i \le l$, we already have S_{C_i}, we know whether $G(C_i)$ has each type of feasible drawings, and conditions (c1)–(c9) holds. We have the following four subcases. Proofs for (c1)–(c9) are omitted.

Case 3(a): C has no child-cycle having a 1- or 2- corner path on C.

Computation for S_C: Condition (c2) means that no cycle in $S_{C_1}, S_{C_2}, \cdots, S_{C_\ell}$ contains any edge on C. Also since G is cubic, C is vertex-disjoint to any cycle in $S_{C_1}, S_{C_2}, \cdots, S_{C_\ell}$. Set $S_C = \{C\} \cup S_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$. Thus we need to introduce two new bends.

Feasible drawings: We first consider whether $G(C)$ has an orthogonal drawing feasible for (P_1, P_1). Construct a new graph from $G(C)$ by adding two dummy vertices v, w on P_1 but not on any child cycle of C. Then contract each $G(C_1), G(C_2), \cdots, G(C_\ell)$ to vertices $v_1, v_2, \cdots, v_\ell$, respectively. See Figs. 6(a) and (b). Now the resulting graph is a cycle and has a rectangular drawing D with the corner vertices x, v, w, y. See Fig. 6(c). Next, if C has a 3-legged child-cycle, say C', then find an orthogonal drawing of $G(C')$ feasible for (P') where P' is the contour path of C' not on C, in a recursive manner. By conditions (c1) and (c6) $G(C')$ always has such a drawing. Next, find an orthogonal drawing of each 2-legged child-cycle $G(C_i)$ feasible for (P''_i, P''_i) where P''_i is the contour path of C_i not on C, in a recursive manner. By condition (c4) $G(C)$ always has such a drawing. Finally patch the drawings of $G(C_1), G(C_2), \cdots, G(C_\ell)$ into D. See Fig. 6(d). The patching for 2- and 3-legged child-cycles always works correctly as shown in Fig. 7 and Fig. 8. Thus we can construct an orthogonal drawing of $G(C)$ feasible for (P_1, P_1). Similarly we can construct orthogonal drawings feasible for (P_2, P_2) and (P_1, P_2), respectively.

Classification: In this case every contour path of C is classified as a 2-corner path.

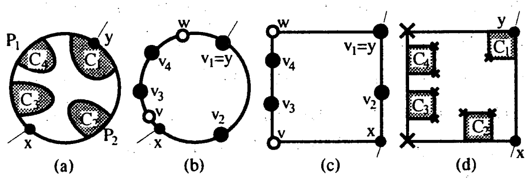

Figure 6: Illustration for Case 3(a).

Case 3(b): C has exactly one child-cycles having a 1- or 2- corner path on C, and the child-cycle is a 2-legged cycle.

Computation for S_C: Let C_1 be the 2-legged child-cycle having a corner path on C. We consider two cases. If C_1 has a 2-corner path on C, then set $S_C = S_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$. In this case we do not need to introduce any new

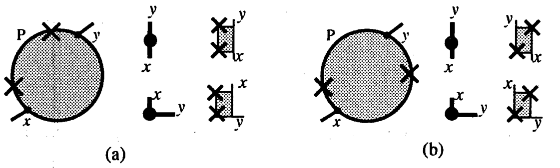

Figure 7: Illustration for patchings.

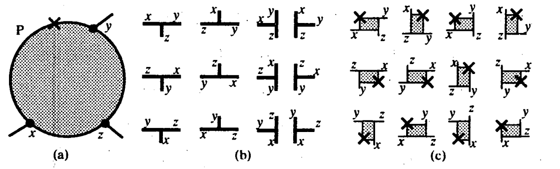

Figure 8: Illustration for patchings. (Rotated cases are omitted.)

bends. If C_1 has a 1-corner path on C, then, by (c3), $G(C_1)$ has a set S'_{C_1} of vertex-disjoint cycles containing no edge on C, and consisting of ℓ'_2 2-legged cycles and ℓ'_3 3-legged cycles such that $2 \cdot \ell'_2 + \ell'_3 = b(G(C_1)) - 1$. Condition (c2) means that no cycle in $S_{C_2}, S_{C_3}, \cdots, S_{C_\ell}$ contains any edge on C. Set $S_C = \{C\} \cup S'_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$. In this case we need to introduce one new bend.
Feasible drawings: Omitted. Similar to the previous case.
Classification: If C_1 has a 2-corner path on P_1, then P_1 is a 2-corner path and P_2 is a 0-corner path. If C_1 has a 2-corner path on P_2, then P_1 is a 0-corner path and P_2 is a 2-corner path. If C_1 has a 1-corner path on P_1, then P_1 is a 2-corner path and P_2 is a 1-corner path. (In this case we can add one new bend either on P_1 or P_2.) If C_2 has a 1-corner path on P_2, then P_1 is a 1-corner path and P_2 is a 2-corner path.
Case 3(c): C has exactly one child-cycles having a 1- or 2-corner path on C, and the child-cycle is a 3-legged cycle.

Let C_1 be the 3-legged child-cycle having a 1- or 2-corner path on C. Assume that C_1 shares y with C as a leg-vertex. Let P_{11} be the contour path of C_1 on P_1 and P_{12} be the contour path of C_1 on P_2.
Computation for S_C: We consider three cases.

If C_1 has a $P_{11}P_{12}$-strain, then set $S_C = \{C_S\} \cup S_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$, where C_S is the 3-legged cycle consisting of the $P_{11}P_{12}$-strain and the edges on P_1 and P_2 not contained in C_1. By the definition of strain and (c2), C_S is vertex-disjoint to any cycle in S_{C_1}. In this casewe need to introduce one new bend for C_S. (See Figs. 9(a)–(d).)

Otherwise, if C_1 has no $P_{11}P_{12}$-strain and either (i)P_{11} is a 2-corner path, (ii)P_{12} is a 2-corner path or (iii) P_{11} is a 1-corner path and P_{12} is a 1-corner path, then set $S_C = S_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$. In this case we do not need to introduce any new bends. (See Figs. 9(e)–(g).)

Otherwise, C_1 has no $P_{11}P_{12}$-strain, and either (i) P_{11} is a 1-corner path and P_{12} is a 0-corner path, or (ii) P_{11} is a 0-corner path and P_{12} is a 1-corner path. By (c5) $G(C_1)$ has a set S'_{C_1} of vertex-disjoint cycles containing no edge on C, and consisting of ℓ'_2 2-legged cycles and ℓ'_3 3-legged cycles such that $2 \cdot \ell'_2 + \ell'_3 = b(G(C_1)) - 1$. Set $S_C = \{C\} \cup S'_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$. Thus in this case we need to introduce one new bend. (See Figs. 9(a)–(d).)
Feasible drawings: Omitted. Similar to the previous case.

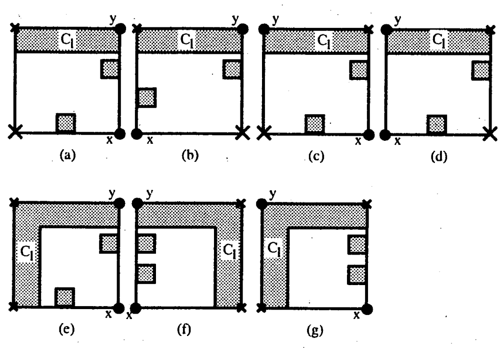

Figure 9: Illustration for Case 3(c).

Classification: If either (i) P_{11} is a 2-corner path and C_1 has no $P_{11}P_{12}$-strain, (ii) P_{11} is a 1- or 2-corner path and C_1 has a $P_{11}P_{12}$-strain, or (iii) P_{11} is a 1-corner path, P_{12} is a 0-corner path and C_1 has no $P_{11}P_{12}$-strain, then P_1 is a 2-corner path. (See Figs. 9(e),(a),(a), respectively.) Otherwise if (i) P_{11} is a 1-corner path, P_{12} is a 1-corner path and C_1 has no $P_{11}P_{12}$-strain, (ii) P_{11} is a 0-corner path, P_{12} is a 1- or 2-corner path and C_1 has a $P_{11}P_{12}$-strain, or (iii) P_{11} is a 0-corner path, P_{12} is a 1-corner path and C_1 has no $P_{11}P_{12}$-strain, then P_1 is a 1-corner path. (See Figs. 9(g),(c),(c), respectively.) Otherwise, P_{11} is a 0-corner path, P_{12} is a 2-corner path and C_1 has no $P_{11}P_{12}$-strain, then P_1 is a 0-corner path. (See Fig. 9(f).) Classify P_2 similarly.

Case 3(d): C has two or more child-cycles having a 1- or 2- corner path on C.

Omitted

Case 4: C is a 3-legged cycle having one or more child-cycles.

Let x, y, z be the three leg-vertices of C, and let P_1, P_2, P_3 be the clockwise contour path from x to y, from y to z, and from z to x, respectively.

Computation for S_C: If C has no child-cycle having a 1- or 2-corner path on C then set $S_C = \{C\} \cup S_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$. In this case we need to introduce one new bend. Otherwise set $S_C = S_{C_1} \cup S_{C_2} \cup \cdots \cup S_{C_\ell}$. In this case we do not need to introduce any new bend.

Feasible drawings: If $G(C)$ has no child-cycle having a 1- or 2-corner path on C then $G(C)$ has orthogonal drawings feasible for (P_1), (P_2), (P_3), respectively. (In this case we need to introduce one new bend.)

Otherwise, $G(C)$ has an orthogonal drawing feasible for (P_1) if and only if $G(C)$ has a child-cycle having a 1- or 2-corner path on P_1. Similarly we can determine whether $G(C)$ has orthogonal drawings feasible for (P_2) and (P_3).

If C has no child-cycle having a 1- or 2-corner path on C then $G(C)$ has no orthogonal drawing feasible for $(P_1, P_1, -P_3)$, since we have no chance to have two bend on P_1 even if we introduce one new bend on P_1.

$G(C)$ has an orthogonal drawing feasible for $(P_1, P_1, -P_3)$ if and only if (i) C has two child-cycle having a 1- or 2-corner path on P_1, or C has a child-cycle having a 2-corner path on P_1, and (ii) C has no P_1P_2-strain. (Construction is omitted. See Figs. 10 and 11.)

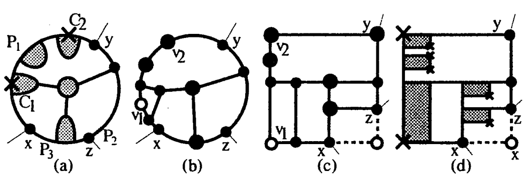

Figure 10: Illustration for Case 4.

Classification: If C has no child-cycle having a 1- or 2-corner path on C, then P_1, P_2 and P_3 are

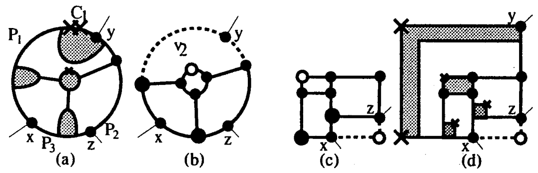

Figure 11: Illustration for Case 4.

1-corner paths. Otherwise, if either (i) C has two or more child-cycles having a 1- or 2-corner path on P_1, or C has a child-cycle having a 2-corner path on P_1, then P_1 is classified as a 2-corner path. Otherwise if C has exactly one child-cycle having 1-corner path on P_1, then P_1 is classified as a 1-corner path. Otherwise P_1 is classified as a 0-corner path. We classify P_2 similarly.

Now we give our algorithm to find a bend-optimal orthogonal drawing. Using a method similar to one in [RNN96, RNN99, RNN00] the algorithm above runs in linear time.

Algorithm Orthogonal-Draw(G)
begin

1 Choose an edge e on $C_o(G)$; Find a genealogical tree T_g;
2 Do the bottom-up computation;
3 Find minimal cycles having 1- or 2-corner path on $C_o(G')$ as many as possible;
4 Do the following until G_0 has exactly four vertices of degree two.
    For each minimal 2-legged cycle C having 2-corner path on G_0 replace $G(C)$ with a quadrangle containing two vertices of degree two on G_0.
    For each minimal 2-legged cycle C having 1-corner path on G_0 replace $G(C)$ with a vertex of degree two.
    For each minimal 3-legged cycle C having 1-corner path on G_0 replace $G(C)$ with a quadrangle containing one vertex of degree two on G_0.
    Put vertices of degree two on the edge e.
5 Find maximal bad cycles $C_1, C_2, \cdots, C_\ell$;
6 Let G'' be the graph derived from G' by contracting each $G(C_i)$, $i = 1, 2, \cdots, \ell$ into a vertex v_i;
7 Find a rectangular drawing $D(G'')$ of G'';
8 For each $i = 1, 2, \cdots, \ell$, find a feasible orthogonal drawing $D(G(C_i))$ of $G(C_i)$;
9 Patch the drawings $D(G(C_i))$, $i = 1, 2, \cdots, \ell$, into $D(G'')$ to get an orthogonal drawing of G; (See Figs. 1(e) and (f).)

end.

Theorem 4.1 *The algorithm above find a bend-optimal orthogonal drawing of a biconnected cubic plane graph in linear time.*

5 Conclusion

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

References

[BLV93] G. Di Battista, G. Liotta and F. Vargiu, *Spirality of orthogonal representations and optimal drawings of series-parallel graphs and 3-planar graphs*, Proc. of Workshop on Algorithms and Data structures, LNCS 709, Springer (1993) 151-162.

[GT94] A. Garg and R. Tamassia, *On the computational complexity of upward and rectilinear planarity testing*, Proc. of Graph Drawing'94, LNCS 894, Springer (1995) 286-297.

[GT96] A. Garg and R. Tamassia, *A new minimum cost flow algorithm with applications to graph drawing*, Proc. of Graph Drawing'96, LNCS 1190, Springer (1997) 201-226.

[K96] G. Kant, *Drawing planar graphs using the canonical ordering*, Algorithmica, 16 (1996) 4-32.

[KH94] G. Kant and X. He, *Two algorithms for finding rectangular duals of planar graphs*, Proc. of WG'93, LNCS 790, Springer (1994) 396-410.

[RNN96] M. S. Rahman, S. Nakano and T. Nishizeki, *Rectangular grid drawings of plane graphs*, Proc. of COCOON'96, LNCS 1090, Springer (1996) 92-105. Also, Computational Geometry: Theory and Applications, 10 (1998) 203-220.

[RNN99] M. S. Rahman, S. Nakano and T. Nishizeki, *A linear algorithm for bend-optimal orthogonal drawings of triconnected cubic plane graphs*, Journal of Graph Algorithms and Applications, 3 (1999) 31-62.

[RNN00] M. S. Rahman, S. Nakano and T. Nishizeki, *Rectangular Drawings of Plane Graphs without Designated Corners*, Proc. of COCOON'00, LNCS 1858, Springer (2000) 85-94.

[T87] R. Tamassia, *On embedding a graph in the grid with the minimum number of bends*, SIAM J. Comput., 16 (1987) 421-444.

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