Almost Bend-Optimal Planar Orthogonal Drawings of Biconnected Degree-3 Planar Graphs in Quadratic Time*

Ashim Garg¹ and Giuseppe Liotta²

¹ Department of Computer Science and Engineering
State University of New York at Buffalo, Buffalo, New York 14221, USA
agarg@cse.buffalo.edu
² Dipartimento di Ingegneria Elettronica e dell’Informazione
Università’ Di Perugia, Perugia, Italy
liotta@diei.unipg.it

Abstract. Let $G$ be a degree-3 planar biconnected graph with $n$ vertices. Let $Opt(G)$ be the minimum number of bends in any orthogonal planar drawing of $G$. We show that $G$ admits a planar orthogonal drawing $D$ with at most $Opt(G) + 3$ bends that can constructed in $O(n^2)$ time. The fastest known algorithm for constructing a bend-minimum drawing of $G$ has time-complexity $O(n^3 \log n)$ and therefore, we present a significantly faster algorithm that constructs almost bend-optimal drawings.

1 Introduction

An orthogonal drawing of a graph maps its vertices to points in the plane and its edges to a sequence of alternating horizontal and vertical line segments. A planar drawing is one with no edge-crossings. Orthogonal drawings have applications in a variety of fields such as Databases, Software Engineering, and VLSI design.

Bend-minimization is an important aesthetic criteria for orthogonal drawings. Several heuristics for bend-minimization are available. (see for example [7,4]). Garg and Tamassia [2] have shown that the bend-minimization for general planar graphs is NP-hard. Tamassia [6] has given an $O(n^2 \log n)$ time bend-minimization algorithm for embedded planar graphs. Later, Garg and Tamassia [3] improved the time-complexity of the algorithm to $O(n^{1.75} \log n)$. Di Battista, Liotta, and Vargiu [1] have given an $O(n^3)$ algorithm for constructing bend-minimum planar orthogonal drawings of series-parallel graphs.

In this paper, we study the problem of constructing bend-minimum planar orthogonal drawings of degree-3 planar graphs. Previously, Rahman, Nakano, and Nishizeki [5] have given a linear time algorithm for constructing a bend-minimum drawing of a triconnected cubic (each vertex has degree exactly 3)

* Research supported in part by a Reifler award, by the CNR Project “Geometria Computazionale Robusta con Applicazioni alla Grafica ed al CAD’ and by the ESPRIT LTR Project 20244 (ALCOM-IT)

plane graph. Di Battista et. al. [1] have given an $O(n^5 \log n)$ time algorithm for constructing bend minimum planar drawing of a degree-3 planar graph.

Our main result is as follows: Let $G$ be a degree-3 planar biconnected graph with $n$ vertices. Let $\text{Opt}(G)$ be the minimum number of bends in any orthogonal planar drawing of $G$. We show that $G$ admits a planar orthogonal drawing $D$ with at most $\text{Opt}(G)+3$ bends that can constructed in $O(n^2)$ time. The fastest known algorithm (of [1]) for constructing a bend-minimum drawing of $G$ has time-complexity $O(n^5 \log n)$ and therefore, we present a significantly faster algorithm that constructs almost bend-optimal drawings.

Let $G$ be a planar connected graph. The degree of a vertex of $G$ is equal to the number of edges incident on it. The degree of $G$ is equal to the maximum degree of a vertex of $G$. A split pair of $G$ is either a pair of adjacent vertices of $G$ or a pair of vertices whose removal divides $G$ into two or more connected graphs.

2 Spirality and Polar Drawings

Let $G$ be a degree-3 planar connected graph. A drawing $D$ of $G$ is a mapping of its vertices to points in the plane and its edges to a set of alternating horizontal and vertical line-segments connecting its end vertices such that no two edges intersect each other. A bend $B$ of an edge $e$ of $G$ in $D$ is the common end point of two consecutive line-segments of $e$ such that the angle between the line segments is not equal to 180°.

Let $D$ be a drawing of $G$. Let $f$ be a face of $D$. Let $l_1$ and $l_2$ be two line-segments that appear consecutively in a clockwise traversal of the line-segments of $f$. Let $v$ be a vertex or bend of $f$ such that $l_1$ and $l_2$ are incident on $v$. Let $\theta$ be the counterclockwise angle between $l_1$ and $l_2$. We say that $v$ makes an angle $\theta$ in $f$. Let $\text{deg}(f, \theta)$ denote the total number of vertices and bends of $f$ (those vertices that make angle $\theta$ more than once, for different pairs of consecutive line segments incident on them, will also get counted more than once). The following lemma can be derived easily from the results of [6]:

**Lemma 1.** [6] Let $D$ be a drawing of a planar connected graph $G$. Let $f$ be an internal face of $D$. Then, $\text{deg}(f, 270°, D) - \text{deg}(f, 90°, D) + 2\text{deg}(f, 360°) = -4$.

Let $G$ be a connected planar graph with two distinguished vertices $u$, and $v$ with degree at most 2, called its poles. A polar drawing of $G$ is one in which both $u$ and $v$ are on its external face (see Figure [1]). Let $D$ be a polar drawing of $G$ with external face $f$. $f$ consists of two subpaths $p_1$ and $p_2$ connecting $u$ and $v$, which are called its contour paths. Let $\text{deg}(p_1, \theta)$ denotes the total number of vertices and bends of $p_1$, except $u$ and $v$, that make an angle $\theta$ in $f$. The spirality of $p_1$ in $D$ is equal to $\text{deg}(p_1, 270°) - \text{deg}(p_1, 90°)$. We likewise define the spirality of $p_2$ in $D$. The spirality of $D$ is defined as equal to the minimum of the spiralities of $p_1$ and $p_2$.

1 our definition of spirality is slightly different from that of [1].
$D$ is a diagonal drawing of $G$ if either $G$ consists only of one vertex (in which case $u = v$), or each pole of $G$ with degree 2 makes a $270^\circ$ angle in $f$, and the spirality of both the contour paths of $f$ is 1. (see Figure 1(a)).

$D$ is a side-on drawing of $G$ if $G$ consists of at least two vertices, each pole of $G$ with degree 2 makes a $270^\circ$ angle in $f$, the spirality of one contour path of $f$ is 0 and the spirality of the other contour path of $f$ is either 0, 1, or 2 (see Figure 1(b)). A bend-minimum side-on drawing of $G$ is one with the fewest bends amongst all the side-on drawings of $G$. We define the bend-minimum diagonal and bend-minimum polar drawings likewise. Notice that a side-on drawing has spirality 0 and a diagonal drawing has spirality 1.

Fig. 1. Polar Drawings of a graph $G$ with poles $u$ and $v$: (a) A diagonal drawing $D_1$ of $G$; (b) A side-on drawing $D_2$ of $G$. $p_1$ and $p_2$ are the contour paths of the external face of $D_1$ and $D_2$. (c) A schematic representation of a diagonal drawing with poles $u$ and $v$. (d) A schematic representation of a side-on drawing with poles $u$ and $v$

**Lemma 2.** Let $G$ be a degree-3 connected planar graph with poles $u$ and $v$. Then,

- $G$ admits either a diagonal or a side-on drawing that has the minimum number of bends among all the polar drawings of $G$, and
- if $G$ does not admit a side-on drawing that has the minimum number of bends among all the polar drawings of $G$, then $G$ also does not admit any polar drawing with spirality less than 0 that has the minimum number of bends among all the polar drawings of $G$.

### 3 Drawing Triconnected Cubic Plane Graphs with Minimum Bends

Rahman, Nakano, and Nishizeki [5] have given a linear time algorithm for constructing a bend-minimum drawing of a triconnected cubic plane graph. As we will see later, we use this algorithm for drawing an $R$ node.

A cubic graph is one where each vertex has degree exactly 3. A plane graph is one with a fixed embedding and a fixed external face. A drawing of a plane graph is one that preserves the embedding and external face of the graph. Let $G$ be a cubic triconnected plane graph. Let $C_3(G)$ denote the external face of $G$. Let $G'$ be a plane graph obtained from $G$ by inserting four dummy vertices,
called its corner vertices, in \( C_0(G) \). We define a \( k \)-legged cycle, descendent cycle, child cycle, corner cycle, and genealogical tree of \( G \), and \( G' \) as in \([5]\). Similarly, we define a leg-vertex, the contour paths, and red and green paths of a 3-legged cycles as in \([5]\). \( C_0(G') \) consists of four contour paths, each connecting two corner vertices that are adjacent in a counterclockwise traversal of \( C_0(G') \). Let \( C \) be a 3-legged cycle of \( G \). \( C \) consists of 3 contour paths that connect two leg-vertices that are adjacent in a counterclockwise traversal of \( C \). Let \( G(C) \) denote the plane subgraph of \( G \) inside (and including) \( C \). (See \([5]\) for details). A feasible drawing \( D \) of \( G' \) is one such that \( D \) has the minimum number of bends among all the drawings of \( G' \), each corner-vertex of \( G' \) makes \( 270^\circ \) angle in \( C_0(G') \), and the spirality of each contour path of \( C_0(G') \) is 0.

Let \( C \) be a 3-legged descendent cycle of \( G' \). Let \( p \) be a contour path of \( C \). A feasible drawing \( D \) of \( G(C) \) with respect to \( p \) is one such that \( D \) has the minimum number of bends among all the drawings of \( G(C) \), each leg-vertex of \( C \) makes \( 270^\circ \) angle in the external face of \( D \), the spirality of the two contour paths of \( C \) other than \( p \) is 0, and the spirality of \( p \) is 1.

Notice that \([5]\) gives stronger definitions of feasible drawings, but for our purposes, the above definitions are sufficient. A rectangular drawing \( D \) is one in which each edge is drawn as a single line-segment and each face is drawn as a rectangle. \( D \) has exactly four vertices, called its corners, that make \( 270^\circ \) angles in its external face.

We describe below a small variation of the algorithm of \([5]\), which we call Algorithm LinearDraw(\( G \)), that constructs a bend-minimum drawing \( D \) of a cubic triconnected plane graph \( G \) in linear time:

**Algorithm LinearDraw(\( G \))**:

1. Find as many as and up to 4 independent corner cycles \( L_1, L_2, \ldots, L_k \) (where \( k \leq 4 \)) of \( G \). For each \( L_i \), insert one dummy vertex \( l_i \) in an edge common to \( C_0(G) \) and a green path of \( C_i \). Let \( G^* \) be the plane graph thus obtained. If \( k \) is less than 4, then insert \( 4 - k \) more dummy vertices \( l'_1, l'_2, \ldots, l'_{4-k} \) in to the edges of \( C_0(G^*) \) such that overall, at most two dummy vertices get inserted in to the same edge of \( C_0(G) \). Let \( G' \) be the plane graph thus obtained. \( G' \) has four corner vertices \( l_1, \ldots, l_k, l'_1, \ldots, l'_{4-k} \).

2. Let \( C_1, C_2, \ldots, C_m \) be the child cycles of \( G' \). Collapse each \( C_i \) into a super node \( S(C_i) \). Let \( G'' \) be the plane graph with no 3-legged cycles thus obtained.

3. Construct a rectangular drawing \( D(G'') \) of \( G'' \) with the corner vertices of \( G'' \) (i.e., vertices \( l_1, \ldots, l_k, l'_1, \ldots, l'_{4-k} \)) as its corners.

4. For each \( C_i \), invoke Algorithm FeasibleDraw(\( C_i, p \)), where \( p \) is a green path of \( C_i \), to construct a feasible drawing \( D(C_i) \) with respect to \( p \) of \( G(C_i) \).

5. Patch each \( D(C_i) \) into drawing \( D(G'') \) at \( S(C_i) \) without introducing any additional bends to get a drawing \( D(G') \) of \( G' \).

6. In \( D(G') \), replace vertices \( l_1, \ldots, l_k, l'_1, \ldots, l'_{4-k} \) by bends to obtain a bend-minimum drawing \( D \) of \( G \).

Given a 3-legged cycle \( C \) with leg-vertices \( a, b, \) and \( c \), and a green path \( p \) of \( C \), Algorithm FeasibleDraw(\( C, p \)) constructs a feasible drawing of \( C \) with
respect to \( p \). It is implemented similar to Algorithm LinearDraw, except that in the Step 1 we insert exactly one dummy vertex \( d \), and designate \( a, b, c \) and \( d \) as the corner vertices of the graph \( G'(C) \) thus obtained. The vertex \( d \) is inserted in an edge of \( p \).

**Lemma 3.** Let \( H \) be a cubic triconnected plane graph. Let \( e = (u, v) \) be a distinguished edge of \( G \) on its external face, called its reference edge. Let \( G \) be a plane graph with poles \( u \) and \( v \), and external face \( C_0(G) \), obtained from \( H \) by removing \( e \) and inserting one or more degree-2 vertices in some edges of \( H \). Suppose \( n \) is the number of vertices in \( G \). Then, we can construct in \( O(n) \) time, a bend-minimum diagonal or side-on drawing \( D \) of \( G \), where \( D \) is a diagonal drawing if and only if \( G \) does not admit a side-on drawing with less than or equal number of bends than \( D \).

**Sketch of Proof.** We construct \( D \) using Algorithm LinearDraw(\( G \)) after making two modifications in it:

1. If a 3-legged cycle \( C \) contains a degree-2 vertex \( z \), then \( z \) can be designated a corner vertex of \( C \). This obviates the need to insert a new dummy vertex in an edge of \( C \) in Step 1 of Algorithm FeasibleDraw. We, therefore, change the definition of a green path of a 3-legged cycle \( C \) as follows:
   - If \( C \) does not contain any child cycle, or no child cycle of \( C \) has a green path that has an edge in common with \( C \), then
     - if none of the three contour paths contain a degree-2 vertex, then all the three contour paths are categorized as green paths, otherwise
       - a contour path of \( C \) is categorized as a green path if and only if it contains a degree-2 vertex,
   - if \( C \) contains at least one child cycle that has a green path that has an edge in common with \( C \), then a contour path of \( C \) is categorized as a green path if and only if either it has an edge in common with a green path of a child cycle, or it contains a degree-2 vertex.

Correspondingly, in Step 1 of Algorithm FeasibleDraw (\( C, p \)) if \( p \) contains a degree-2 vertex \( z \), then we designate \( z \) as a corner vertex instead of inserting a new dummy vertex \( d \) in an edge of \( p \). Since \( d \), if inserted, would have appeared as a bend in the final drawing, designating \( z \) as a corner vertex instead of inserting \( d \) saves us one bend. Using a similar proof of [5], it can be shown that Algorithm FeasibleDraw (\( C, p \)) will construct a feasible drawing of \( C \) with respect to \( p \) in linear time.

2. We change Step 1 of Algorithm LinearDraw as follows: We can designate \( u \) and \( v \) as two corner vertices of \( G' \). We still need to designate two more corner vertices for \( G' \). Let us denote these corner vertices by \( a_1 \) and \( a_2 \). Let \( p_1 \) and \( p_2 \) be the two subpaths of \( C_0(G) \) with end points \( u \) and \( v \). We say that the corner vertex \( a_j \), where \( 1 \leq j \leq 2 \), gets assigned to path \( p_i \), where \( 1 \leq i \leq 2 \), if we designate as \( a_j \), either a degree-2 vertex of \( p_i \), or a dummy vertex that we insert into an edge of \( p_i \) during the execution of Step 1. In a side-on drawing of \( G \) with poles \( u \) and \( v \), both \( a_1 \) and \( a_2 \) get assigned to the same path, either \( p_1 \) or \( p_2 \), and in a diagonal drawing, one each is assigned
such that each $Q$ nodes, i.e., correspond to single edges of vertex with $C$ X of the children of $C$ can be easily derived from the fact that each vertex of $G$ is a non-$Q$ node. In other words, for each non-$Q$ child of $X$, the children of $X$ before and after it in the canonical ordering are $Q$ nodes, i.e., correspond to single edges of $G$. More over, if the parent of $X$ is not the root of $\tau$, then both $C_1$ and $C_k$ are $Q$ nodes, i.e., correspond to single edges of $G$. Also, we can always construct a $\tau$ such that each $C_i$ is either a $P$ node, $Q$ node, or an $R$ node.

4 SPQR Tree

Let $G$ be a planar biconnected graph. Let $e = (s, t)$ be an edge of $G$. An SPQR decomposition of $G$ with reference edge $e$ is its recursive decomposition into components of four types: $S$, $P$, $Q$, and $R$, where the initial decomposition divides $G$ at the split pair $\{s, t\}$. Each SPQR decomposition corresponds to an SPQR decomposition tree $\tau$. $\tau$ consists of four types of nodes: $S$, $P$, $Q$, and $R$, which correspond to the $S$, $P$, $Q$ and $R$ components, respectively, of the decomposition. We can always orient the edges of $\tau$ such that the root of $\tau$ is a $Q$ node that corresponds to the edge $e$. Each node of $\tau$ corresponds to a subgraph of $G$, called its pertinent graph. In particular, the pertinent graph of a $Q$ node consists of a single edge of $G$. The pertinent graph of each node $X$ of $\tau$ has two distinguished vertices called its poles. Notice that the poles of both the root of $\tau$ and its child are $s$ and $t$. Also associated with each node $X$ of $\tau$ is a graph, called its skeleton, and denoted as $\text{skel}(X)$. The skeleton of an $R$ node is a triconnected graph, of a $P$ node is a bundle of parallel edges, of an $S$ node is a chain of edges, and of a $Q$ node is a single edge. See [1] for details.

Let $X$ be an $S$ node of $\tau$. Let $u$ and $v$ be the parent split vertices of $X$. We can order the children of $X$ as $C_1, C_2, \ldots, C_{k-1}, C_k$, where the child component $C_1$ is incident on $u$, child component $C_2$ shares a vertex with $C_1$, $C_3$ shares a vertex with $C_4$, and so on, and finally, $C_k$ shares a vertex with $C_{k-1}$ and also, $C_k$ is incident on $v$. We call $C_1$ and $C_k$ the extreme children of $X$, and the ordering $C_1, C_2, \ldots, C_{k-1}, C_k$ a canonical ordering of the children of $X$.

Let $G$ be a biconnected degree 3 planar graph. Let $\tau$ be an SPQR decomposition tree of $G$ with reference edge $e$. Let $X$ be a node of $\tau$. The following facts can be easily derived from the fact that each vertex of $G$ has degree at most 3:

**Fact 1** If $X$ be an $S$ node of $\tau$. Let $C_1, C_2, \ldots, C_{k-1}, C_k$ is a canonical ordering of the children of $X$. We have that, if $C_i$ is a non-$Q$ node, then $C_{i-1}$ and $C_{i+1}$ are $Q$ nodes. In other words, for each non-$Q$ child of $X$, the children of $X$ before and after it in the canonical ordering are $Q$ nodes, i.e., correspond to single edges of $G$. More over, if the parent of $X$ is not the root of $\tau$, then both $C_1$ and $C_k$ are $Q$ nodes, i.e., correspond to single edges of $G$. Also, we can always construct a $\tau$ such that each $C_i$ is either a $P$ node, $Q$ node, or an $R$ node.
**Fact 2** Each $P$-node of $\tau$ has exactly two children which are either $S$ or $Q$ nodes.

**Fact 3** Each child of an $R$-node of $\tau$ is either an $S$ node or a $Q$ node.

We define the core graph and pole of a node $X$ of $\tau$ as follows: If $X$ is a $P$ node, $Q$ node, $R$ node, or an $S$ node whose parent is the root of $\tau$, then its core graph is the same as its pertinent graph, and the pole of its core graph is the same as the pole of its pertinent graph.

If $X$ is an $S$ node whose parent is not the root of $\tau$, then let $C_1, C_2, \ldots, C_{k-1}, C_k$ be a canonical ordering of the children of $X$. Let $u$ and $v$ be the poles of the pertinent graph of $X$. From Fact 1, $C_1$ and $C_k$ are $Q$-nodes, i.e., they correspond to single edges of $G$. Suppose $C_1$ and $C_k$ correspond to edges $(u, a)$ and $(b, v)$, respectively, of $G$. Let $H(C_i)$ denote the core graph of $C_i$. Since each $C_i$ is a $P$, $Q$, or $R$ node, $H(C_i)$ is the same as the pertinent graph of $C_i$. The core graph $H$ of $X$ is defined as the subgraph of the pertinent graph of $H$ that consists of the graphs $H(C_2), H(C_3), \ldots, H(C_{k-1})$, i.e., $H = H(C_2) \cup H(C_3) \cup H(C_{k-1})$.

In other words, the core graph of $X$ is the graph obtained by removing edges $(u, a)$ and $(b, v)$ from the pertinent graph of $X$. Vertices $a$ and $b$ are designated the poles of $H$.

The poles of $X$ are the same as the poles of its core graph.

**Lemma 4.** Let $G$ be a degree-3 planar graph. Let $\tau$ be an SPQR tree of $G$ that corresponds to an SPQR decomposition of $G$ with a reference edge $e$. Let $X$ be a non-root node of $\tau$, whose pertinent graph has $n$ vertices. Then, we can construct in $O(n)$ time, a bend-minimum polar drawing $D$ of the core graph $H$ of $X$ such that:

1. if $X$ is an $S$ node whose core graph consists of at least two vertices, or is a $Q$ node, then $D$ is a side-on drawing,
2. if $X$ is an $S$ node whose core graph consists of exactly one vertex, then $D$ is a diagonal drawing,
3. if $X$ is a $P$, or an $R$ node, then $D$ is either a diagonal or a side-on drawing, and
4. $D$ is a diagonal drawing if and only if $H$ does not admit a side-on drawing with less than or equal number of bends than $D$.

**Sketch of Proof.** Our proof is constructive. Starting with the leaves of $\tau$, for each node $X$ of $\tau$, we construct a side-on or diagonal drawing of the core graph of $X$ with properties as given in the statement of the lemma.

Let $X$ be a non-root node of $\tau$. Let $H$ be the core graph of $X$. Let $u$ and $v$ be the poles of $X$. Let $D(X)$ denotes the drawing—side-on or diagonal—of $H$ constructed by our proof. We consider the following cases:

- $X$ is a $Q$ node: Then, $H$ is a single edge $t = (u, v)$. We draw $t$ as a single line-segment.
- $X$ is an $S$ node: We have two subcases:
  - $H$ consists of a single vertex $a = u = v$: Then, we draw $a$ as a point, which by the definition of a diagonal drawing, is a diagonal drawing.
• $H$ consists of more than one vertex: Let $C_1, C_2, \ldots, C_{k-1}, C_k$ be a canonical ordering of the children of $X$. Recall that if the parent of $S$ is not the root of $\tau$, then $H = H(C_2) \cup H(C_3) \cup H(C_{k-1})$, where $H(C_1)$ is the core graph of $C_1$. We have three cases, depending on whether both $D(C_2)$ and $D(C_{k-1})$ are side-on drawings, or are both diagonal drawings, or one is a side-on and the other is a diagonal drawing. As shown in Figure 2, in all three cases, by simply stacking the drawings $D(C_2), D(C_3), \ldots, D(C_{k-1})$ one above the other in that order, we can construct a side-on drawing $D(X)$. Since each $D(C_i)$ is a bend-minimum polar drawing of $H(C_i)$, and we do not insert any new bends while stacking them, it follows that $D(X)$ is a bend-minimum polar drawing of $H$. If the parent of $S$ is the root of $\tau$, then we can construct $D(X)$ in a similar fashion by vertically stacking drawings $D(C_1), D(C_2), \ldots, D(C_k)$.

Fig. 2. Constructing $D(X)$ when $X$ is an $S$ node such that the parent of $X$ is not the root node: (a) An $S$ node $X$ with 8 children $C_1, C_2, \ldots, C_8$ of which only $C_2, C_4$ and $C_7$ are non-$Q$ nodes; (b,c,d) Constructing $D(X)$ from $D(C_2), D(C_3), \ldots, D(C_7)$: (b) When $D(C_2)$ is a side-on and $D(C_7)$ is a diagonal drawing; (c) When both $D(C_2)$ and $D(C_7)$ are diagonal drawings; (d) When both $D(C_2)$ and $D(C_7)$ are side-on drawings. Vertices $u$ and $v$ are the poles of $X$.

- $X$ is a $P$ node: From Fact 2, $X$ has two children $C_1$ and $C_2$, and they are either $S$ or $Q$ nodes. We have two subcases:

  • $C_2$ is a $Q$ node: Let $H(C_1)$ be the core graph of $C_1$. Suppose $C_2$ corresponds to a single edge $t = (u, v)$ of $G$. If $D(C_1)$ is a side-on drawing, then we can construct $D(X)$ from $D(C_1)$ and $D(C_2)$ as shown in Figure 3(a) without adding any new bends. Since, $D(C_1)$ is a bend-minimum polar drawing of $H(C_1)$, it follows that $D(X)$ is also bend-minimum polar drawing of the core graph of $X$. If $D(C_2)$ is a diagonal-drawing, then we can construct $D(X)$ from $D(C_1)$ and $D(C_2)$ as shown in Figure 3(b) by adding one more bend. To show that $D(X)$ is also a bend-minimum polar drawing of $H$, consider a bend-minimum polar drawing $D$ of $H$ (also see Figure 3(c)). $D$ contains a polar subdrawing $D'$ of the core graph of $C_1$. Since $D(C_1)$ is a bend-minimum polar drawing, $D'$ has at
least as many bends as $D(C_1)$. Hence, if any of the edges $(a, u)$, $(b, v)$ and $t$ have a bend in $D$, then we are done, otherwise, consider the face $f$ of $D$ that contains the edges $(a, u)$, $(b, v)$, and $t$, where $a$ and $b$ are the poles of $H(C_1)$ (see Figure 3(c)). $f$ also contains a contour path $p_1$ of the external face of $D'$. From Lemma 1, it follows that if face $f$ does not have any bend in edges $(a, u)$, $(b, v)$, or $t$, then $p_1$ must have spirality at most 0, and hence $D'$ must have spirality at most 0. However, then, since Lemma 4 holds for $D(C_1)$, it follows from Lemma 2 that $D'$ has at least one bend more than $D(C_1)$. That is, $D$ has at least as many bends as $D(X)$, and therefore $D(X)$ is also a bend-minimum polar drawing.

![Figure 3](image)

**Fig. 3.** Constructing $D(X)$ for a $P$ node $X$ with children $C_1$ and $C_2$, where $C_2$ corresponds to a single edge $t = (u, v)$: (a) When $D(C_1)$ is a side-on drawing, (b) When $D(C_1)$ is a diagonal drawing. In both cases, $D(X)$ is a side-on drawing; (c) Proof of the bend-optimality of $D(X)$. $a$ and $b$ are the poles of $C_1$, and $a$ and $v$ are poles of $X$.

- $C_2$ is not a $Q$ node: If both $D(C_1)$ and $D(C_2)$ are side-on drawings, than we construct a bend-minimum side-on drawing $D(X)$ as shown in Figure 4(a). If at least one of $D(C_1)$ and $D(C_2)$ is a diagonal drawing, then we can construct a diagonal drawing $D(X)$ as shown in Figure 4(b,c). In both the cases, we do not add any bends, and hence, $D(X)$ is bend-minimum side-on or diagonal drawing. Using a reasoning similar to one for the previous case, where $C_2$ is a $Q$ node, we can show that when $D(X)$ is a diagonal drawing, then $X$ does not admit any side-on drawing with less than or equal number of bends than $D(X)$.

- $X$ is a $R$ node: Let $r$ be the reference edge of $skel(X)$. From Fact 3 each child of $X$ is either an $S$ node or a $Q$ node. We first remove $r$ from $skel(X)$. Next, for each edge $k$ of $skel(X)$ that corresponds to an $S$ child of $X$ whose core graph consists of at least two vertices, we insert two dummy vertices in $k$. Also, for each edge $k$ of $skel(X)$ that corresponds to an $S$ child of $X$ whose core graph consists of exactly one vertex, we insert one dummy vertex in $k$. Let $L$ be the graph thus obtained. We designate $u$ and $v$ as the poles of $L$. We construct a bend-minimum side-on or diagonal drawing $D$ of $L$ using Lemma 3. From $D$, we construct $D(X)$ without adding any new bends as follows: Let $k$ be an edge of $skel(X)$ that corresponds to an $S$ child $C$ of $X$. Let $H(C)$ be the core graph of $C$. We have two cases:
Fig. 4. Constructing $D(X)$ for a $P$ node with children $C_1$ and $C_2$, neither of which is a $Q$ node: (a) When both $D(C_1)$ and $D(C_2)$ are side-on drawings; (b) When both $D(C_1)$ and $D(C_2)$ are diagonal drawings; (c) When $D(C_1)$ is a diagonal drawing and $D(C_2)$ is a side-on drawing. $a$ and $b$ are the poles of $C_1$, $a'$ and $b'$ are the poles of $C_2$, and $u$ and $v$ are the poles of $X$.

- $H(C)$ consists of a single vertex $z$: Let $a$ be the dummy vertex introduced in $k$ to obtain $L$ from $skel(X)$. We simply replace $a$ by $z$.
- $H(C)$ consists of at least two vertices: Let $a$ and $b$ be the dummy vertices introduced in $k$ to obtain $L$ from $skel(X)$. Since $D(C)$ is a side-on drawing (because $C$ is an $S$ node) irrespective of the angles made by $a$ and $b$ in their incident faces in $D$, we can replace $a$ and $b$ and the edge $(a, b)$ by $D(C)$ without adding any new bends.

Lemma 5. Let $G$ be a biconnected degree-3 planar graph with $n$ vertices. Let $e = (u, v)$ be an edge of $G$. Let $M$ be the minimum number of bends in any drawing of $G$ with $e$ on the external face. Then, we can construct in $O(n)$ time, a drawing $D$ of $G$ with $e$ on the external face such that $D$ has at most $M + 3$ bends.

Sketch of Proof:. Let $\tau$ be an SPQR decomposition tree of $G$ with reference edge $e$. As mentioned earlier in Section 4, the root $X$ of $\tau$ is a $Q$ node that corresponds to $e$. Let $C$ be the child of $X$ with core graph $H(C)$. $u$ and $v$ are the poles of $H(C)$. Let $D(C)$ be the bend-minimum polar drawing of $H(C)$ constructed using Lemma 4. As shown in Figure 5, from $D(C)$, we can construct a drawing $D$ of $G$ with $e$ on external face by adding at most 3 more bends irrespective of whether $D(C)$ is side-on or diagonal. Since a bend-minimum drawing of $G$ with $e$ on external face contains a polar drawing of $H(C)$ as a subdrawing, and $D(C)$ is the bend-minimum polar drawing of $H(C)$, it follows that $D$ has at most $M + 3$ bends. Since, from Lemma 4, $D(C)$ can be constructed in $O(n)$ time, it follows that $D$ can also be constructed in $O(n)$ time.

5 Main Theorem

Theorem 1. Let $G$ be a biconnected degree-3 planar graph with $n$ vertices. Let $Opt(G)$ be the minimum number of bends in any drawing of $G$. Then, we can construct in $O(n^2)$ time, a drawing $D$ of $G$ such that $D$ has at most $Opt(G) + 3$ bends.
Fig. 5. Proof of Lemma 5: Constructing a drawing for the root node $X$ from $D(C)$, where $C$ is the child of $X$: (a) when $C$ is a side-on drawing, and (b) when $C$ is a diagonal drawing. $e$ is the reference edge.

**Sketch of Proof.** For each edge $e$ of $G$, we construct a drawing using Lemma 5 that has $e$ on the external face. The drawing with minimum number of bends among these drawings will have at most $\text{Opt}(G) + 3$ bends. Since $G$ has $O(n)$ edges, and constructing each drawing takes $0(n)$, the total running time is $O(n^2)$.

**References**