

# Improved Algorithms and Bounds for Orthogonal Drawings

Achilleas Papakostas and Ioannis G. Tollis

Dept. of Computer Science  
The University of Texas at Dallas  
Richardson, TX 75083-0688

email: papakost@utdallas.edu, tollis@utdallas.edu

**Abstract.** An orthogonal drawing of a graph is a drawing such that nodes are placed on grid points and edges are drawn as sequences of vertical and horizontal segments. In this paper we present linear time algorithms that produce orthogonal drawings of graphs with  $n$  nodes. If the maximum degree is four, then the drawing produced by our algorithm needs area no greater than  $0.8n^2$  and no more than  $1.9n$  bends. Notice that our upper bound on the bends is below the lower bound for *planar* orthogonal drawings of *planar* graphs. To the best of our knowledge, this is the first algorithm for orthogonal drawings that needs area less than  $n^2$ . If the maximum degree is three, then the drawing produced by our algorithm needs  $(\frac{n}{2} + 1) \times \frac{n}{2}$  area and at most  $\frac{n}{2} + 3$  bends. These upper bounds match the upper bounds known for triconnected planar graphs of degree 3.

## 1 Introduction

Graphs have been extensively used in recent years to represent various important concepts or objects not only in computer science but in other sciences as well. Examples of such objects in computer science include parallel computer architectures, networks, state graphs, entity-relationship diagrams, subroutine call graphs, automata, data-flow graphs, Petri nets, VLSI circuits, etc. In all of these cases, we require that the graph be represented (or drawn) in the plane so that we can understand and study its structure and properties. It is for that reason that, typically, drawing of a graph is accompanied by optimizing some cost function such as area, number of bends, number of edge crossings, uniformity in the placement of vertices, minimum angle, etc. As a result, research on algorithms for drawing graphs has received increasing attention recently. For a survey of graph drawing algorithms and other related results see the annotated bibliography of Di Battista, Eades, Tamassia and Tollis [3].

In this paper we focus on the problem of *orthogonal* drawings of graphs, that is drawings in which each edge of the graph is a polygonal chain consisting of horizontal and vertical segments. A graph admits such a drawing if it has maximum degree 4. Orthogonal drawings are typically used in circuit schematics. Our goal is to minimize the area (this is of paramount importance in circuit

layout), and to obtain a drawing with a small number of bends (this is also important because it is related to the minimization of vias).

Several results have appeared in the literature regarding planar orthogonal drawings of graphs. In [12] and [14] it is shown that every biconnected planar graph of maximum degree 4 can be embedded in a  $n \times n$  grid with  $2n + 4$  bends. If the graph is not biconnected then the total number of bends rises to  $2.4n + 2$ . In all cases, no more than 4 bends per edge are required. The algorithms of [14] take linear time and produce drawings, such that at most one edge may have 4 bends. Kant [8] shows that if the graph is triconnected of maximum degree 4, then it can be drawn on an  $n \times n$  grid with at most 3 bends per edge. The total number of bends is no more than  $\lceil \frac{3}{2}n \rceil + 3$ . For triconnected graphs of maximum degree 3 it is shown in the same paper that a gridsizes of  $(\frac{n}{2} - 1) \times \frac{n}{2}$  is sufficient and no more than  $\frac{n}{2}$  bends are required totally. In this case, no edge bends more than twice. Even and Granot [5] present an algorithm for obtaining an orthogonal drawing of a 4-planar graph with at most 3 bends per edge. If the embedding of a planar graph is fixed, then an orthogonal drawing with the minimum number of bends can be computed in  $O(n^2 \log n)$  time [13]. If the planar embedding is not given, the problem is polynomially solvable for 3-planar graphs [4], and NP-hard for 4-planar graphs [7].

It is interesting to note that there is a lower bound of  $2n - 2$  bends for biconnected planar graphs [15]. There are also examples of biconnected graphs [15], for which every bend-optimal planar drawing introduces a single edge with length  $\Omega(n^2)$  having  $\Omega(n)$  bends along it. Although these drawings achieve optimality in terms of the total number of bends, they are not aesthetically pleasing. This suggests that research in this area should concentrate on deriving orthogonal drawings of graphs with  $O(1)$  bends per edge (usually 2 or 3) and  $O(n)$  maximum edge length.

Upper and lower bounds have been proved in the case when the orthogonal drawing of the graph is not necessarily planar. Leighton [9] presented an infinite family of planar graphs which require area  $\Omega(n \log n)$ . Independently, Leiserson [10] and Valiant [16] showed that every planar graph of degree 3 or 4 has an orthogonal drawing with area  $O(n \log^2 n)$ . Valiant [16] showed that the orthogonal drawing of a general (nonplanar) graph of degree 3 or 4 requires area no more than  $9n^2$ , and described families of graphs that require area  $\Omega(n^2)$ . Valiant was not concerned about minimizing the total number of bends. In fact, an analysis of his construction shows that each edge can have up to 4 bends. Schäffter [11] presents an algorithm which constructs orthogonal drawings of graphs with at most two bends per edge. The area required is  $2n \times 2n$ . A better algorithm is presented in [1] and [2], which draws the graph within an  $n \times n$  grid with no more than 2 bends per edge. This algorithm introduces at most  $2n + 2$  bends.

In this paper we present an algorithm that produces an orthogonal drawing of a graph of maximum degree 4 that needs  $(n + 1 - k_2) \times (m - n + 1 - k_1)$  area and  $2m - 2n + 4 - k_2$  bends, where  $k_1 + k_2 \geq \lceil \frac{n_4}{5} \rceil$ . Notice that  $m$  is the number of edges of the graph,  $n$  is the number of vertices, and  $n_4$  is the number of vertices of degree 4. The number of bends that appear on each edge is no

more than two. If the maximum degree is 3, then we present another algorithm which produces an orthogonal drawing which needs maximum area  $(\frac{n}{2} + 1) \times \frac{n}{2}$  and  $\frac{n}{2} + 3$  bends. In this drawing, no more than one bend appears on each edge except for one edge, which may have at most two bends.

The significance of our results lies primarily on two facts. First, to the best of our knowledge, it is the only algorithm that produces an orthogonal drawing of a graph of maximum degree 4 which requires less than  $n \times n$  area. The second fact is that our upper bound on the number of bends for graphs of maximum degree 4 is below the lower bound known for planar orthogonal drawings of graphs of maximum degree 4. This is not a contradiction because our algorithm does not guarantee a planar drawing, even for planar graphs. Another aspect of our work is that for graphs of maximum degree 3, we match the upper bounds which are known for planar *triconnected* graphs of maximum degree 3.

## 2 Definitions and Preliminaries

Various graphic standards exist for the representation of graphs in the plane. Perhaps the simplest one is to assign vertices to distinct points in the plane and to connect two adjacent vertices through a line between their corresponding points. Typically, this line is a straight line or a set of contiguous line segments (i.e., *bends* are allowed); when the latter happens, we talk about a *polyline* drawing. A drawing is *planar* if no two edges intersect.

An *orthogonal* drawing is a special case of a polyline drawing in which each edge is drawn as a sequence of horizontal and vertical segments. The *rectilinear grid* is the infinite plane graph whose vertices have integer coordinates, and whose edges link pairs of vertices at unit distance. A polyline drawing is a grid drawing if the vertices and the bends have integer coordinates. Orthogonal drawings are typically grid drawings such that edges correspond to grid paths. It is for that reason that orthogonal drawings are also known as rectilinear drawings.

Consider the smallest rectangle which encloses the orthogonal drawing of a graph of degree 4. If this rectangle has height  $n_1$  and width  $n_2$ , then we say that the area of the drawing is  $n_1 \times n_2$ .

We use  $n$  (resp.  $m$ ) to denote the number of vertices (resp. edges) in a graph. A graph is biconnected if the removal of any vertex and its incident edges does not disconnect the graph. An *st-ordering* is an ordering  $v_1, v_2, \dots, v_n$  of the vertices such that every  $v_j$  ( $2 \leq j \leq n - 1$ ) has at least one predecessor and at least one successor, that is neighbors  $v_i, v_k$  with  $i < j < k$ . It is known that:

**Theorem 1.** [6] *Let  $G$  be biconnected and  $s, t$  be two vertices of  $G$ . Then there exists an st-ordering such that  $s$  is the first and  $t$  is the last vertex, and it can be computed in  $O(m)$  time.*

A linear time algorithm for producing orthogonal drawings of biconnected graphs of maximum degree 4 is presented in [1]. The basic idea of the algorithm is that the vertices of the graph are placed on the grid consecutively, according to a precomputed st-ordering. At each step, a vertex  $v$  is placed on a new row. In

the same step,  $v$ 's incoming edges are drawn and we allocate as many columns as necessary for the outgoing edges of  $v$ . This technique results in very efficient drawings both in terms of area and bends.

**Theorem 2.** [1] *Let  $G$  be a biconnected graph of maximum degree 4. Then there exists a linear time algorithm which embeds  $G$  on an  $n \times (m - n + 1)$  grid with at most  $2m - 2n + 4$  bends. Each edge is bent at most twice.*

### 3 Improved Bounds for Degree 4 Graphs

In this section we will present an algorithm for obtaining orthogonal embeddings of general (nonplanar) biconnected graphs of maximum degree 4. Our algorithm uses the same basic idea discussed above, and achieves better results in terms of area and number of bends.

Let  $G$  be a general (nonplanar) biconnected graph of degree 4. Let us use  $n_4$  to denote the number of vertices of degree 4 in  $G$ . Let  $V_{deg4}$  be a set, originally empty. We pick a vertex  $w$  of degree 4 and insert it into  $V_{deg4}$ . We keep inserting vertices of degree 4 into  $V_{deg4}$  as long as they are not adjacent to some other degree 4 vertex which already lies in  $V_{deg4}$ . Clearly, in this way, at least  $\lceil \frac{n_4}{5} \rceil$  vertices of degree 4 will be inserted in  $V_{deg4}$ .

**Lemma 3.** *There exists a bipolar orientation of  $G$  in which all vertices in  $V_{deg4}$  have two incoming edges and two outgoing edges (i.e. they are balanced).*

**Proof.** Each vertex  $w \in V_{deg4}$  is adjacent to 4 vertices. We insert edges so that every two of  $w$ 's adjacent vertices are connected through an edge (see Fig. 1). If we do this for every vertex in  $V_{deg4}$ , we obtain a new biconnected graph  $G'$ . We obtain an st-ordering on  $G'$ , such that  $s$  and  $t$  are not in  $V_{deg4}$ . Next we consider each vertex  $w \in V_{deg4}$  which is not already balanced. In this case,  $w$  has 3 incoming and one outgoing edge (the symmetric case is treated similarly). Exhaustive analysis shows that we can always reverse the direction of one of the 3 incoming edges without creating a directed cycle. After we do that for every  $w \in V_{deg4}$ , we remove all the extra edges, and we obtain an st-numbering for  $G$ , in which every  $w$  is balanced. We call this new st-ordering of  $G$ , a *balanced* st-ordering, or *bst-ordering*.  $\square$

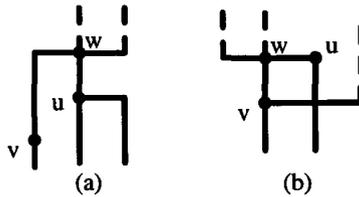
Let  $v_1, v_2, \dots, v_n$  be the numbering of the vertices of  $G$  as a result of the bst-ordering. Consider a vertex  $w \in V_{deg4}$ ;  $w$  has indegree 2 and outdegree 2. This means that there must be two other vertices,  $v$  and  $u$  so that  $(v, w)$  and  $(u, w)$  are directed edges of  $G$ . Let us assume, without loss of generality, that vertex  $v$  has received a lower number than vertex  $u$ . The placement of each vertex  $w \in V_{deg4}$  follows the cases described below:

1. At least one of vertices  $u$  and  $v$  has indegree 2 or 3. Then there exists at least one column which has "closed" (that is the second end of an uncompleted edge has been inserted). This column can be "reused", that is one of the two uncompleted edges which are going to be introduced for  $w$  can use this column (see Fig. 2a).



**Fig. 1.** (a) vertex  $w$  has degree 4, (b) extra edges are added (denoted by dotted lines).

- Both  $u$  and  $v$  have indegree less than or equal to 1. We insert vertex  $w$  at the intersection point between one of the free outgoing edges of  $v$  and the row of  $u$ . We connect  $w$  with  $u$  forming edge  $(u, w)$ . If there is any other uncompleted edge leaving  $u$ , it will be drawn on the other side of  $u$  (see Fig. 2b).



**Fig. 2.** (a)  $w$  is inserted so that a closed column is reused, (b)  $w$  is inserted on  $u$ 's row.

We are now ready to present our algorithm formally.

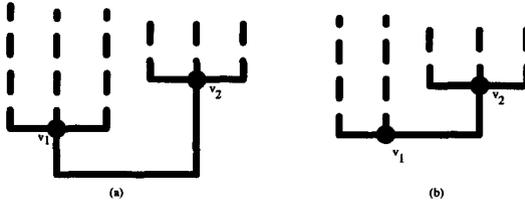
**Algorithm:** *4.EMBED*

**Input:** A biconnected graph  $G$  of maximum degree 4.

**Output:** An orthogonal embedding of  $G$ .

- Compute a bst-ordering of  $G$  and form set  $V_{deg4}$ .
- Embed vertices  $v_1$  and  $v_2$  on two adjacent rows. If  $v_1$  has degree 4, then edge  $(v_1, v_2)$  is drawn so that it bends twice and enters  $v_2$  from below (see Fig. 3a). If  $v_1$  has degree less than 4, then edge  $(v_1, v_2)$  is drawn to the right of  $v_1$  and up into  $v_2$ ; notice that in this case there is only one bend along edge  $(v_1, v_2)$  (see Fig. 3b).
- REPEAT
  - Consider the next vertex  $v_i$  according to the bst-ordering of  $G$ .
  - If  $v_i$  has already been placed, then go to Step 4.

- (c) Place vertex  $v_i$  on a new row. Connect  $v_i$  with each vertex  $v_j$  ( $j < i$ ) such that  $(v_j, v_i)$  is a directed edge of  $G$ . Add as many uncompleted edges as required, depending on  $v_i$ 's outdegree.
  - (d) If vertex  $v_i$  has an outgoing edge to some vertex  $w \in V_{deg4}$ , and there exists a vertex  $v_j$  ( $j < i$ , i.e.,  $v_j$  has already been embedded) such that  $(v_j, w)$  is a directed edge of  $G$ , then place  $w$  according to Cases 1 or 2 described above.
4. UNTIL the only remaining vertex is  $v_n$ .
  5. Insert  $v_n$  on a new row so that the edge that enters  $v_n$  from the top and bends twice (this happens only if  $v_n$  is of degree 4) is connected to  $v_{n-1}$ .
  6. End.



**Fig. 3.**  $v_1$  and  $v_2$  embedded by algorithm 4\_EMBED: (a)  $v_1$  has degree 4 and (b)  $v_1$  has degree 3.

**Lemma 4.** *Let us assume that there are  $k_1$  vertices in  $V_{deg4}$  which belong to Case 1. Then the width of the embedding will be  $m - n + 1 - k_1$ .*

**Proof.** When we embed the first two vertices  $v_1$  and  $v_2$ , the required width is  $outdeg(v_1) + outdeg(v_2) - 2$ , which can also be written as  $(outdeg(v_1) - 1) + (outdeg(v_2) - 1)$ . Let us call  $V_1$  the set of all vertices  $w \in V_{deg4}$  which belong to Case 1. When we embed a vertex  $w \in V_1$ , the width of the current embedding does not increase. This happens because one of  $w$ 's uncompleted edges can reuse a closed column. When we embed a vertex  $v_i \in V - V_1$  and  $v_i \neq v_1, v_2$ , the width increases by  $outdeg(v_i) - 1$ . The placement of  $v_n$  does not increase the width (i.e., increase:  $0 = (outdeg(v_n) - 1) + 1$ ). Therefore the total width of the embedding is  $\sum_{v \in V - V_1} (outdeg(v) - 1) + 1 = (m - k_1) - n + 1$ .  $\square$

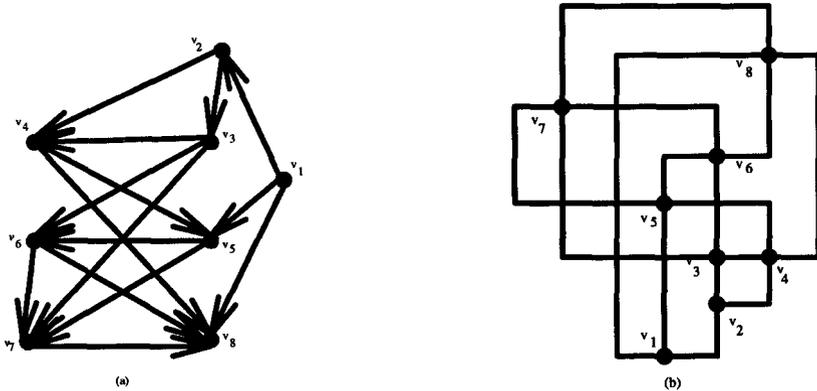
**Lemma 5.** *Let us assume that there are  $k_2$  vertices in  $V_{deg4}$  which belong to Case 2 ( $k_1 + k_2 = |V_{deg4}|$ ). Then the height of the embedding will be  $n + 1 - k_2$ .*

**Proof.** Each of the  $k_2$  vertices shares the same row with another vertex of  $G$  which is not in  $V_{deg4}$ . Vertices  $v_1$  and  $v_n$  require an extra row if they are of degree 4. All the other vertices of  $G$  are placed on separate rows. Hence the height of the embedding is  $n + 1 - k_2$ .  $\square$

**Theorem 6.** *There exists a simple linear time algorithm to construct an orthogonal drawing of a degree 4 biconnected graph with area  $(n + 1 - k_2) \times (m - n + 1 - k_1)$ , where  $k_1 + k_2 \geq \lceil \frac{2n}{5} \rceil$ . The total number of bends of the embedding is  $2m - 2n + 4 - k_2$ , and no edge has more than two bends.*

**Proof.** The area bound follows easily from Lemmas 4 and 5. When we embed the first two vertices  $v_1$  and  $v_2$ , a total of  $outdeg(v_1) + outdeg(v_2) - 1$  bends are introduced which can also be written as  $((deg(v_1) - 2) + 2) + (deg(v_2) - 2)$  bends. Let us call  $V_2$  the set of all the vertices  $w \in V_{deg4}$  which belong to Case 2. When a vertex  $w \in V_2$  is embedded, then  $(indeg(w) - 2) + (outdeg(w) - 1) = deg(w) - 3$  bends are introduced. When we embed a vertex  $v_i \in V - V_2$  and  $v_i \neq v_1, v_2$ , then  $(indeg(v_i) - 1) + (outdeg(v_i) - 1) = deg(v_i) - 2$  bends are introduced. When  $v_n$  is embedded, a total of  $deg(v_n)$  bends are introduced, which is  $(deg(v_n) - 2) + 2$ . So the total number of bends that are introduced is:  $\sum_{v \in V - V_2} (deg(v) - 2) + 4 + \sum_{v \in V_2} (deg(v) - 3)$  which is equal to  $2m - 2n + 4 - k_2$ . Notice that, because of the construction of the drawing, no edge bends more than twice.  $\square$

Since every graph of maximum degree 4 can be augmented to a regular graph of degree 4, we will consider the case when  $n_4 = n$  in order to find upper bounds on the area. In this case, the area is maximized when  $k_1 = k_2 = \frac{n}{10}$ . If this happens, the area is  $(\frac{9n}{10} + 1) \times (\frac{9n}{10} + 1) \approx 0.8n^2$ . It is also interesting to note that in this case, the total number of bends is  $2n + 4 - \frac{n}{10} \approx 1.9n$ .



**Fig. 4.** (a) bst-ordering of an example graph, (b) orthogonal drawing of the same graph produced by algorithm 4\_EMBED.

An added advantage of our approach is that the resulting drawing is upward (in case we have an application that requires that). Notice that if the source and/or the sink have degree 4, then there is one or two edges which cannot be drawn completely upward. Figure 4a shows a degree 4 graph for which we have taken a bst-ordering; the orthogonal drawing which our algorithm produces for

this graph is demonstrated in Fig. 4b. Vertices  $v_4$  and  $v_6$  are the members of the set  $V_{deg4}$  of Lemma 3. Our algorithm can be extended to the case of one-connected graphs. We split the one-connected graph into its blocks and embed each block separately using the algorithm described above. Finally, we put the graph back together using a technique similar to the ones described in [1] and [14].

## 4 Improved Bounds for Degree 3 Graphs

It is known that every triconnected planar graph of degree 3 can be embedded in an orthogonal fashion so that the area is at most  $(\frac{n}{2} - 1) \times \frac{n}{2}$  [8]. Also, in such a drawing, no more than  $\frac{n}{2}$  bends are required. In this section we present an algorithm for obtaining an orthogonal embedding of any biconnected graph (not necessarily planar) of degree 3. Our algorithm matches the bounds for planar triconnected graphs.

Consider a biconnected degree 3 graph  $G$ . Without loss of generality, we assume that our graph  $G$  is also regular of degree 3. If it is not, we can always absorb a vertex of degree 2 into a single edge, and in this way transform our graph into a new graph which is regular of degree 3. For every embedded vertex we define its *degree of freedom* as the number of grid edges which are incident to the vertex and are not occupied by edges of  $G$ . Consider a grid edge which is not occupied by any graph edge, and is incident to an embedded vertex  $v$ . If this grid edge is located to the right (resp. left, up, down) of  $v$ , then we say that  $v$  has a degree of freedom to the right (resp. left, up, down). We maintain the following two invariants:

- **Invariant 1:** Each newly inserted vertex creates either a new row, or a new column in the embedding, but never both. In other words, each vertex is placed either at a row or at a column which already holds at least one more vertex of  $G$ .
- **Invariant 2:** For each placed vertex, except the first one, we embed only the incoming edges of the vertex.

### Algorithm: 3.EMBED

**Input:** A biconnected, regular of degree 3, graph  $G$ .

**Output:** An orthogonal embedding of  $G$ .

1. Compute an st-numbering of  $G$ .
2. Insert source  $v_1$  as shown in Fig. 5a.
3. REPEAT
  - (a) Consider the next vertex  $v_i$  according to the st-numbering.
  - (b) If  $indeg(v_i) = 1$ , then let  $v_j$  ( $j < i$ ) be the vertex for which directed edge  $(v_j, v_i)$  is in  $G$ . We distinguish the following cases for  $v_j$ .
    - If  $outdeg(v_j) = 2$  and  $v_j$  has a degree of freedom to the right, we place  $v_i$  on the same row and in the first free column to the right of  $v_j$ . This means no other vertex has been placed in that column.

- Otherwise, we place  $v_i$  on a new row and in a position which is directly above  $v_j$ . If  $v_j$  does not have a degree of freedom up, then we place  $v_i$  directly below  $v_j$ . In all cases, the column remains the same.
- (c) If  $\text{indeg}(v_i) = 2$ , then let  $v_j, v_k$  ( $j, k < i$ ) be the vertices for which directed edges  $(v_j, v_i)$  and  $(v_k, v_i)$  are in  $G$ . Each one of  $v_j$  and  $v_k$  has at least two degrees of freedom, so we distinguish the following cases.
- If both  $v_j$  and  $v_k$  have a degree of freedom to the right (resp. left, up, down), then  $v_i$  will be placed to the right (resp. left, up, down) of  $v_j$  and  $v_k$  so that either one row, or one column is introduced, but not both (see Fig. 6a).
  - If  $v_j$  and  $v_k$  do not have a degree of freedom in the same direction (say  $v_j$  has degrees of freedom to the left and up and  $v_k$  has degrees of freedom to the right and down), then we distinguish the following two cases:
    - $v_j$  is to the right or down of  $v_k$ .  $v_i$  can be placed in such a way so that either a new row or a new column is introduced in the middle of the drawing, as demonstrated in Figs. 6b and 6c.
    - $v_j$  is to the left and up of  $v_k$ . We distinguish two cases:
      - \* Both  $v_j$  and  $v_k$  have outdegree 2. This means that  $v_k$  is the last vertex on its row. We re-embed  $v_k$  in the way shown in Fig. 7a (exchanging a column for a row), and now  $v_k$  has a degree of freedom to the left.
      - \* Otherwise, at least one of the two has indegree two (say  $v_k$ ) and at least one of  $v_k$ 's incident edges has a bend. We move  $v_k$  on this bend. This move will change the degrees of freedom of  $v_k$  and we will fall into one of the previous cases (see Fig. 7b).
4. UNTIL the only remaining vertex is  $v_n$ .
5. Insert sink  $v_n$ . We will discuss shortly how  $v_n$  is inserted.

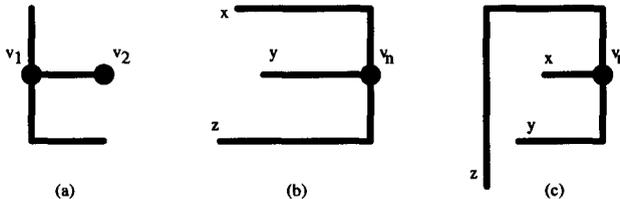
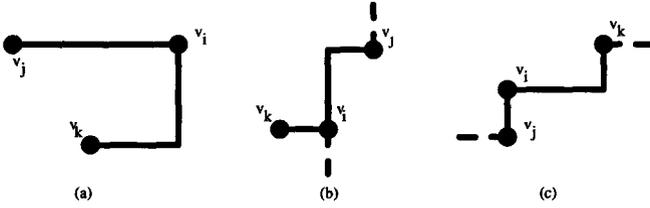
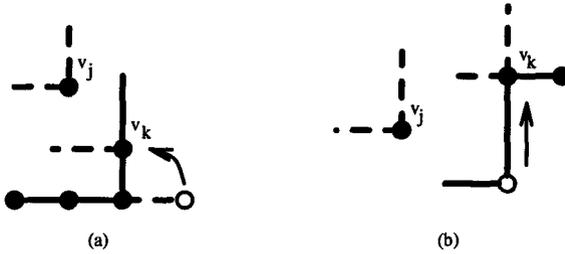


Fig. 5. (a) source  $v_1$  and (b),(c) sink  $v_n$  embedded by algorithm 3\_EMBED.

**Lemma 7.** *After vertices  $v_1, v_2, \dots, v_{n-1}$  have been embedded, every edge has at most one bend.*



**Fig. 6.** Insertion of  $v_i$  by algorithm 3\_EMBED: (a)  $v_j$  and  $v_k$  have free edges to the same direction, (b)  $v_j$  and  $v_k$  do not have free edges to the same direction and a new column is inserted in the middle of the drawing, (c) same as in (b) but a new row is inserted in the middle of the drawing.



**Fig. 7.** 3\_EMBED re-embeds  $v_k$  before the insertion of  $v_i$ : (a) a column is exchanged for a row, (b)  $v_k$  moves on a bend.

Consider vertices  $x$ ,  $y$  and  $z$  such that the directed edges  $(x, v_n)$ ,  $(y, v_n)$  and  $(z, v_n)$  are in graph  $G$ . If all three of them have a degree of freedom to the same direction, then  $v_n$  can be embedded with either a new row or a new column and no two-bend edge (see Fig. 5b). Otherwise, both a new row and a new column are required and one edge bends twice (see Fig. 5c for one such case). Hence we have the following:

**Lemma 8.** *The placement of the sink  $v_n$  introduces at most one new row and one new column in the embedding and also an edge with two bends.*

It is worth observing that this algorithm tends to group vertices of indegree 1, which appear along the same directed path of  $G$ , on the same grid row. We call the sequence of vertices embedded along the same row, a *chain*. A second observation is the fact that new rows and columns can be added at the top, the bottom or even in the middle of the current drawing. Hence, the drawing is not upward.

**Theorem 9.** *There exists a linear time algorithm for constructing an orthogonal drawing of a biconnected graph of degree 3 so that no more than  $\frac{n}{2} + 3$  bends are*

required and there is at most one edge that bends twice. Moreover, the embedding requires at most  $(\frac{n}{2} + 1) \times \frac{n}{2}$  area.

Figure 8a shows an st-numbering we have computed for graph  $K_{3,3}$ . Note that  $K_{3,3}$  is regular of degree 3 and nonplanar. In Fig. 8b we demonstrate the orthogonal drawing of this graph that our algorithm produces.

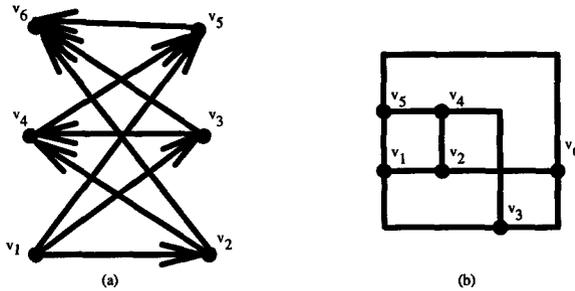


Fig. 8. (a) st-ordering computed for graph  $K_{3,3}$ , (b)  $K_{3,3}$  embedded by 3\_EMBED.

## 5 Conclusions and Open Problems

In this paper we presented two algorithms for obtaining orthogonal drawings of general (nonplanar) biconnected graphs. The first algorithm produces orthogonal drawings of graphs of maximum degree 4. Our upper bounds on the area and the number of bends are lower than the upper bounds of previously known algorithms. Moreover, our upper bounds on the number of bends are lower than the lower bounds for planar orthogonal drawings of planar graphs. Our second algorithm produces orthogonal drawings of degree 3 graphs and matches the upper bounds for area and number of bends of planar triconnected degree 3 graphs. Both our algorithms extend naturally to one-connected graphs.

Some of the problems that still remain open in the area of producing orthogonal drawings of planar and nonplanar graphs are the following:

- Find lower bounds on the area and number of bends for orthogonal drawings of nonplanar graphs. We showed in this paper that the lower bounds for planar drawings do not transfer to nonplanar drawings.
- Develop dynamic drawing algorithms. We would like to be able to efficiently update the minimum number of bends (and a corresponding drawing) when edges and/or vertices are added or removed.

It turns out that, in practice, most of the graphs are nonplanar. This is the reason we believe that work on orthogonal drawings of nonplanar graphs should continue.

## Acknowledgement

We would like to thank Shimon Even and Roberto Tamassia for useful discussions.

## References

1. Therese Biedl, *Embedding Nonplanar Graphs in the Rectangular Grid*, Rutcor Research Report 27-93, 1993.
2. T. Biedl and G. Kant, *A Better Heuristic for Orthogonal Graph Drawings*, Proc. 2nd Ann. European Symposium on Algorithms (ESA '94), Lecture Notes in Computer Science, vol. 855, pp. 24-35, Springer-Verlag, 1994.
3. G. DiBattista, P. Eades, R. Tamassia and I. Tollis, *Algorithms for Drawing Graphs: An Annotated Bibliography*, Dept. of Comp. Science, Brown Univ., Tech. Report, 1993. To appear in *Comput. Geom. Theory Appl.* Preliminary version available via anonymous ftp from ftp.cs.brown.edu, gdbiblio.tex.Z and gdbiblio.ps.Z in /pub/papers/compgeo.
4. G. DiBattista, G. Liotta and F. Vargiu, *Spirality of orthogonal representations and optimal drawings of series-parallel graphs and 3-planar graphs*, Proc. Workshop on Algorithms and Data Structures, Lecture Notes in Computer Science 709, Springer-Verlag, 1993, pp. 151-162.
5. S. Even and G. Granot, *Rectilinear Planar Drawings with Few Bends in Each Edge*, Tech. Report 797, Comp. Science Dept., Technion, Israel Inst. of Tech., 1994.
6. S. Even and R.E. Tarjan, *Computing an st-numbering*, Theor. Comp. Sci. 2 (1976), pp. 339-344.
7. A. Garg and R. Tamassia, *On the Computational Complexity of Upward and Rectilinear Planarity Testing*, Tech. Report CS-94-10, Dept. of Computer Science, Brown University, Providence, 1994, also in these proceedings.
8. Goos Kant, *Drawing planar graphs using the lmc-ordering*, Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science, 1992, pp. 101-110.
9. F. T. Leighton, *New lower bound techniques for VLSI*, Proc. 22nd Ann. IEEE Symp. on Found. of Comp. Science, 1981, pp. 1-12.
10. Charles E. Leiserson, *Area-Efficient Graph Layouts (for VLSI)*, Proc. 21st Ann. IEEE Symp. on Found. of Comp. Science, 1980, pp. 270-281.
11. Markus Schäffter, *Drawing Graphs on Rectangular Grids*, Discr. Appl. Math. (to appear).
12. J. Storer, *On minimal node-cost planar embeddings*, Networks 14 (1984), pp. 181-212.
13. R. Tamassia, *On embedding a graph in the grid with the minimum number of bends*, SIAM J. Comput. 16 (1987), pp. 421-444.
14. R. Tamassia and I. Tollis, *Planar Grid Embeddings in Linear Time*, IEEE Trans. on Circuits and Systems CAS-36 (1989), pp. 1230-1234.
15. R. Tamassia, I. Tollis and J. Vitter, *Lower Bounds for Planar Orthogonal Drawings of Graphs*, Information Processing Letters 39 (1991), pp. 35-40.
16. L. Valiant, *Universality Considerations in VLSI Circuits*, IEEE Trans. on Comp., vol. C-30, no 2, (1981), pp. 135-140.