ELSEVIER

Discrete Applied Mathematics 81 (1998) 69–91

DISCRETE APPLIED MATHEMATICS

A linear algorithm for 2-bend embeddings of planar graphs in the two-dimensional grid

Yanpei Liu[a], Aurora Morgana[b], Bruno Simeone[c,*]

[a] *Department of Mathematics, University of Northern Jiaotong, Beijing, China*
[b] *Dipartimento di Matematica, Università di Roma "La Sapienza", Italy*
[c] *Dipartimento di Statistica, Probabilità e Statistiche Applicate, Università di Roma "La Sapienza", Italy*

Received 26 June 1996; received in revised form 4 April 1997; accepted 14 April 1997

Abstract

In this paper we describe a linear algorithm for embedding planar graphs in the rectilinear two-dimensional grid, where vertices are grid points and edges are noncrossing grid paths. The main feature of our algorithm is that each edge is guaranteed to have at most 2 bends (with the single exception of the octahedron for which 3 bends are needed). The total number of bends is at most $2n + 4$ if the graph is biconnected and at most $(\frac{7}{3})n$ in the general case. The area is $(n + 1)^2$ in the worst case. This problem has several applications to VLSI circuit design, aesthetic layout of diagrams, computational geometry.

Keywords: Graph drawing; Planar graphs; Rectilinear embeddings; Graph algorithms

1. Introduction

We consider the problem of constructing a rectilinear layout on the grid Z^2 of a planar graph G with n vertices. This problem has obvious applications in VLSI design. Possible solutions to the layout problem depend upon the constraints on the layout.

By *rectilinear embedding* of a graph we shall mean a planar embedding where the vertices are mapped into grid points and each edge is mapped into a broken line consisting of an alternate sequence of horizontal and vertical line segments. A comprehensive survey on rectilinear embeddability is given in [17].

Since two different edges are not allowed to have a common interior point, one obvious condition to be required is that every vertex of G has degree at most 4. Connected planar graphs having this property will be called *standard*. From now on

*Corresponding author. E-mail: marsalis@rosd.sta.uniroma1.it

PII S0166-218X(97)00076-0

we shall assume that all graphs under consideration are standard, unless explicitly stated otherwise.

Every inner point of an edge, which is the intersection of an horizontal and a vertical segment is called a *bend*. A *k-bend* edge is an edge with exactly k bends. The number of bends along the edges and the area occupied by the layout are important quality measures in circuit layout applications. There are two common ways of minimizing the number of bends among all rectilinear embeddings of a graph. We refer to them as the min–max problem and the min–sum problem.

The *min–max problem* consists in finding the least k such that any graph can be rectilinearly embedded into the grid with at most k bends on each edge.

A graph that can be embedded on the grid with at most r bends on each edge is said to be *r-embeddable* and the embedding is called an *r-embedding* of G.

In a previous paper [15] we sketched a proof of the following results:

Theorem 1.1. *Every standard graph is 3-embeddable.*

Theorem 1.2. *Every standard graph is 2-embeddable with the only exception of the octahedron.*

Theorem 1.3. *Every standard graph has a 3-embedding such that the total number of bends is at most 2n.*

The *min-sum* problem consists in finding, among all possible rectilinear embeddings, one for which the total number of bends is minimum.

Similarly, we may define the area minimization problem.

NP-completeness results related to the minimization of area and total edge length have been presented in [3, 6, 20].

Shiloach [19] and Valiant [26] gave algorithms for embedding planar graphs with n vertices in the grid so as to achieve $O(n^2)$ area.

However, none of these algorithms guarantees that the resulting embedding has a bounded number of bends per edge.

Aggarwal et al. described, in [1], the double-J algorithm and showed that the resulting embedding has at most 6 bends; more recently, they presented in [2] a simpler approach which produces an embedding where each edge has at most 4 bends. However, both these algorithms have the additional properties that all vertices are embedded on the same horizontal line, and that all edges are incident to vertices from either above or below.

At most 4 bends along each edge are also guaranteed by an O(n) algorithm of Tamassia and Tollis [24].

The main feature of the algorithm presented in this paper is that the resulting embedding is guaranteed to have at most 2 bends per edge with the single exception of the octahedron, where there are at most two edges with 3 bends. Our algorithm improves upon a previous one by Cui and Liu [5] in terms of both maximum number of bends per edge and domain of applicability (arbitrary connected graphs). Moreover, their algorithm may produce "zig-zag" edges (see Section 3), which cause an unnecessary increase of the total number of edges and of the area.

In Section 2 we describe a time- and space-linear algorithm that, starting from any embedding of a biconnected standard graph G, produces a 3-embedding of G. The total number of bends is at most $2n + 4$ and the occupied area is $O(n^2)$.

In Section 3 we give a modified version of the above algorithm, which rectifies 2-bend edges of a certain type.

In Section 4 we prove that, for any biconnected standard graph, with the single exception of the octahedron, there exists a starting embedding from which the above algorithm produces a 2-embedding. Furthermore, the starting embedding can be produced in linear time from an arbitrary embedding.

Finally, in Section 5 we extend the above results to general standard graphs, and in Section 6 we provide some conclusions.

2. A 3-embedding algorithm for biconnected graphs

First of all, we introduce the concepts of bipolar orientation and unilateral numbering which hold for any biconnected, not necessarily planar graph G.

A *bipolar orientation* of G is an assignment of directions to the edges of G such that the resulting digraph is acyclic with only one source and only one sink.

A *unilateral numbering* of G is a numbering of the vertices from 1 to n such that every vertex $i = 2, \dots, n-1$ is adjacent both to some vertex $h < i$ and to some vertex $j > i$.

A graph with this numbering is called *unilaterally numbered.*

A vertex numbering of G from 1 to *n agrees* with an acyclic orientation of G if every directed edge goes from a lower-numbered vertex to a higher-numbered one. One such numbering always exists and it can be found in $O(n)$ time by topological sorting [25].

Throughout this section $G = (V, E)$ is a biconnected planar graph and we assume that a planar embedding of G is known. Embedding a planar graph means constructing circular adjacency lists (*vertex-rotations*) of G such that, for each vertex w, all neighbours of w appear in clockwise order with respect to an actual drawing. Such an embedding can be constructed in $O(n)$ time using any of the known algorithms in [4, 8, 10, 12–14].

Two embeddings of G are said to be *equivalent* if all their vertex-rotations coincide.

From now on the given embedding of G will still be denoted, for simplicity, by G.

Let w be a vertex on the outerface boundary of G. Then there are two edges incident to w on the outerface boundary. Exactly one of these two edges, say e, has the property that the other edge, say e', is the next one in the clockwise rotation around w. Then e is called the *rightmost edge* and e' the *leftmost edge* at w.

It is well known that for any biconnected graph a bipolar orientation and hence a unilateral numbering exist and can be found in linear time [7, 9, 21]. Furthermore, the source s and the sink t can be arbitrarily chosen.

Hence, without loss of generality, we may assume that G has a bipolar orientation whose source s and sink t are chosen on the outerface boundary of G and that its vertices are unilaterally numbered. For each vertex v, we shall denote by $d^-(v)$ and by $d^+(v)$ the indegree and the outdegree of vertex v, respectively.

Under the above assumptions the following proposition holds:

Proposition 2.1 (Ramassia and Tallis [23]). *In any planar embedding of G, for any vertex w all the incoming (outgoing) edges are consecutive in the clockwise rotation around w.*

In order to make the description of the algorithm simpler, it will be convenient to insert into each directed edge uv a midpoint w (*virtual vertex*) so as to split edge uv into two *semi-edges* (u, w) and (w, v). The virtual vertex w will be labelled with the same numbering of vertex v.

Let G' be the subdivided graph obtained from G in this way. It is straightforward to obtain a 3-embedding of G from a 3-embedding of G'.

Given a unilateral numbering of G, for each $i = 1, \dots, n$ let $V_i = \{1, \dots, i\}$, let G_i be the subgraph of G induced by V_i, and let C_i be the cocycle of V_i in G, i.e., the set of all edges of G having exactly one endpoint in V_i.

Proposition 2.2 (Even [8]). *In any planar embedding of G, for each $i = 1, \dots, n$ all edges of C_i belong to the outerface of G_i.*

The above result implies that, for each edge uv in C_i, with $u \in V_i$, its virtual vertex w and the two semi-edges (u, w) and (w, v) belong to the outerface of G_i. It follows that all semi-edges (u, w), $u \in V_i$, can be arranged in a circular list according to their clockwise order around the outerface boundary of G_i. The order of the semi-edges induces an order of the virtual vertices: the corresponding circular list of virtual vertices will be called the *frontier* F_i of G_i.

The following proposition is crucial in order to establish the correctness of the algorithm to be described below.

Proposition 2.3. *For each $i = 1, \dots, n - 1$, all virtual vertices carrying the label $i + 1$ occupy consecutive positions in F_i.*

Proof. All virtual vertices that are labelled $i + 1$ belong to F_i, since any such vertex is adjacent to some vertex with lower numbering. Suppose that there are in F_i three virtual vertices w_h, w_{h+1} and w_k ($k > h + 1$) such that w_h and w_k carry the label $i + 1$, while w_{h+1} does not. Let (u_h, w_h), (u_{h+1}, w_{h+1}) and (u_k, w_k), respectively, be the corresponding semi-edges, with u_h, u_{h+1}, $u_k \in V_i$. Both u_h and u_k must belong to the

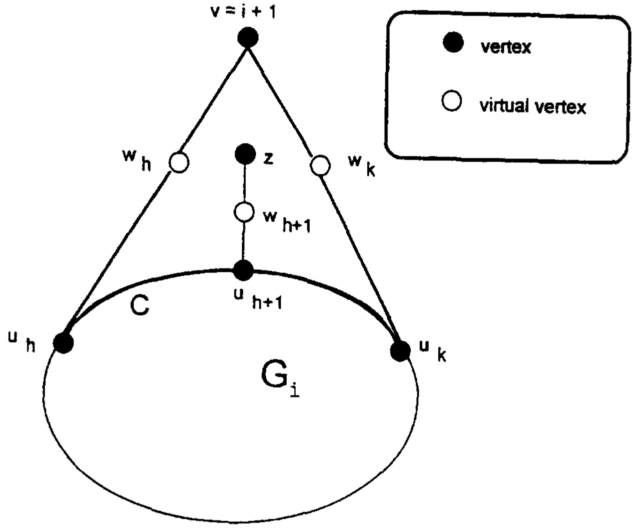


Fig. 1. Proof of proposition 2.3.

outerface boundary of G_i because of planarity. Starting from u_h, traverse the outerface boundary of G_i clockwise until u_k is reached: let $P(u_h, u_k)$ be the resulting path. Set $v = i + 1$ and consider the closed curve $C = vP(u_h, u_k)v$. There is a unique semi-edge (w_{h+1}, z) going out of w_{h+1}. By planarity and Proposition 2.2, z belongs to the inner domain of C (see Fig. 1). The highest-numbered vertex of G in the inner domain of C must be a sink, contradicting the fact that the (unique) sink of G lies on the outerface boundary of G. □

Given a rectilinear embedding of a standard graph, a *supporting line* (of the embedding) is any horizontal or vertical line through some vertex.

Let us now describe an algorithm that finds a rectilinear 3-embedding of G' and hence also of G.

ALGORITHM TWO-BEND

For each $i = 1, \ldots, n$:

(a) Embed all the semi-edges (if any) going into vertex $v = i$ depending on $d^-(v)$ as shown in Fig. 2(a), placing vertex i on the horizontal line $y = i$.

(b) Embed all the semi-edges (if any) going out of vertex $v = i$ depending on $d^+(v)$ as shown in Fig. 2(b).

END

In particular, at the beginning Step 1(a) merely places vertex 1 on the horizontal line $y = 1$, while Step 1(b) embeds all the semi-edges going out of vertex 1.

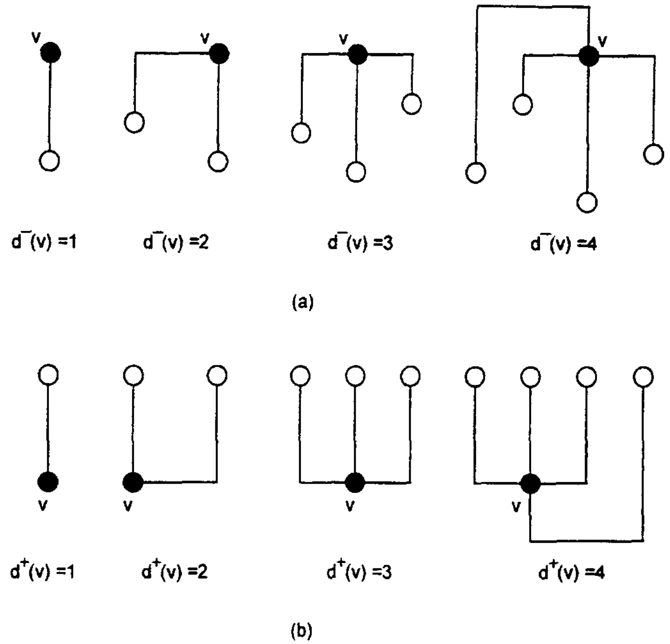


Fig. 2. Embedding all the semi-edges (a) into v and (b) out of v.

Fig. 2 describes the execution of a single step. In (a) virtual vertices (white) have been embedded in previous steps while the true vertex v (black), as well as the semi-edges into v, are embedded in the current step. Then in (b) all semi-edges out of v and their virtual endpoints (white) are embedded.

If the source has outdegree 4, the rightmost semi-edge going out from s, say h_s, has 2 bends. If the sink has indegree 4, the rightmost semi-edge coming into t, say h_t, has 2 bends.

For each semi-edge (v, w) out of v and with one or two bends, the x-coordinate of the virtual vertex w is defined so that the supporting vertical line through w is equally distant from the nearest supporting lines on its left and on its right (if one of these two lines is missing, then place w at distance 1 from the other line). The y-coordinate of w is irrelevant: for the sake of definiteness we can take it equal to $i + \frac{1}{2}$.

A detailed description of an $O(n)$ implementation of the algorithm together with appropriate data structures can be found in [16].

Once an embedding of G' has been obtained, one can easily change the x-coordinates of the vertices and of the virtual vertices so that any two consecutive vertical supporting lines of G' are at distance 1. From now on, we shall assume that this is always the case.

We shall illustrate with one example the behaviour of the algorithm. Fig. 3 depicts a planar graph G together with a bipolar orientation and a unilateral numbering.

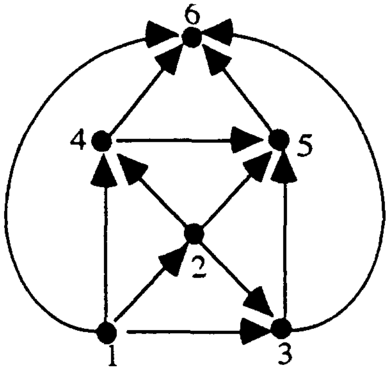


Fig 3. Unilaterally numbered biconnected standard graph G.

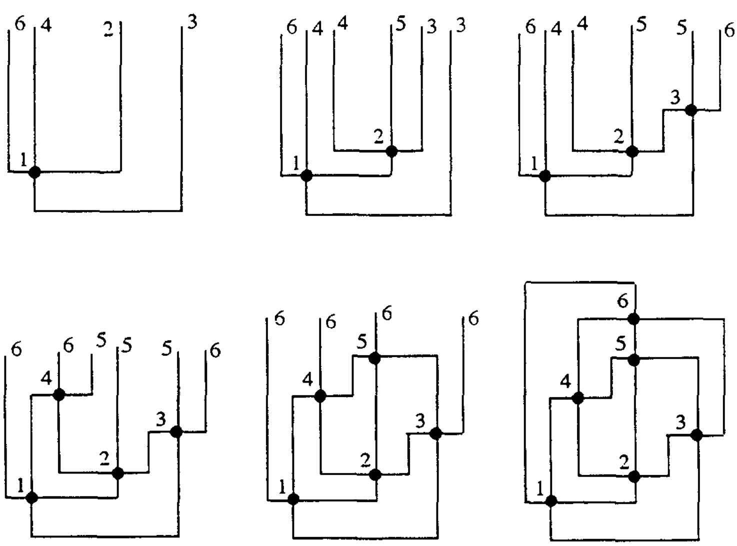


Fig. 4. Embedding algorithm for G.

Fig. 4 shows, at each step i, the frontier F_i and the current rectilinear embedding generated by the algorithm. Virtual vertices are represented here as points, and only the labels of the virtual vertices in the current frontier are shown.

Theorem 2.4. *For a unilaterally numbered planar graph G, the algorithm generates a 3-embedding.*

Proof. The planarity of the layout follows from Propositions 2.1–2.3. Furthermore, the construction guarantees that the resulting embedding is equivalent to the starting

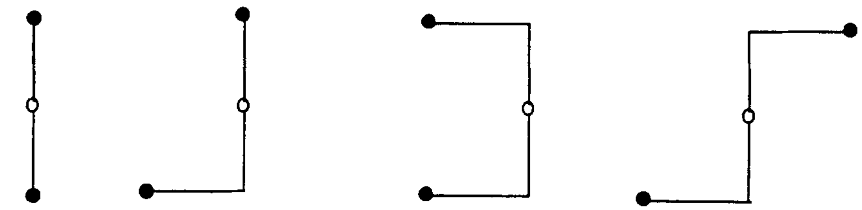


Fig. 5.

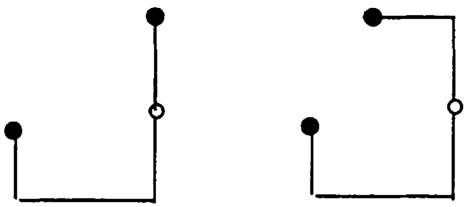


Fig. 6.

one and has the same outerface boundary of the plane graph G. The embedding is obviously rectilinear. The fact that each edge has at most 3 bends follows from the construction. In fact, every semi-edge generated by the algorithm has at most 1 bend, with the exception of the rightmost semi-edges h_s and h_t incident to s and t, respectively. Therefore, any edge not containing h_s or h_t has at most two bends, and the shape of any such edge is of the types exemplified in Fig 5.

By composition of h_s or h_t with a 0-bend semi-edge we obtain a 2-bend edge, else a 3-bend edge as exemplified in Fig 6. □

Corollary 2.5. *If s and t have degree 4 and st is an edge of G, then the rectilinear embedding of st has 3 bends.*

Proof. It follows from the fact that st is either the rightmost edge going out from s or the rightmost one coming into t and in both cases it is obtained as the composition of a 2-bend semi-edge with a 1-bend semi-edge. □

Lemma 2.6. *For every $i = 1, \dots, n$, the cardinality of F_i is at most $n + 2$.*

Proof. Let n_k be the number of vertices whose indegree is $k = 0, 1, 2, 3, 4$. Since $n_0 = n_4 = 1$ and $n_1 = n_3$, one has $n = 2 + 2n_1 + n_2$ and thus $2n_1 \leqslant n - 2$. On the other hand, the cardinality of F_i is 4 when $i = 1$, it increases by 2 every time a vertex of indegree 1 is processed and it never increases when a vertex of indegree 2, 3, or 4 is processed. Hence, the cardinality of F_i is bounded above by $4 + 2n_1 \leqslant n + 2$. □

Theorem 2.7. *The algorithm is linear both in time and space.*

Proof. The time complexity is obviously $O(n)$; the space is also $O(n)$ in view of Lemma. 2.6. □

Theorem 2.8. *The total number of bends is at most* $2n + 4$.

Proof. The total number of horizontal levels produced by the algorithm (ignoring those containing only virtual vertices) is equal to the number of vertices plus two, since we have introduced a new horizontal level both for the source and the sink. Furthermore, each level has two bends. This proves the theorem. □

Theorem 2.9. *The area of the* 3*-embedding is at most* $(n + 1)^2$.

Proof. The height of the 3-embedding is $n + 1$, since the number of horizontal levels not containing virtual vertices is $n + 2$. Furthermore, the width is one less than the maximum cardinality of the frontier and thus it is at most $n + 1$ by Lemma 2.6. Therefore the theorem holds. □

3. The elimination of zig-zags

In this section we show how to avoid the generation of zig-zag edges, with some modifications of the algorithm in Section 2.

To get rid of the horizontal handles, we insert two new vertices s_0 and t_0 in the middle of the horizontal segments of h_s and h_t, respectively. Let $G_0 = (V_0, E_0)$ be the resulting graph with source s_0 and sink t_0. Edges with at most one bend or vertical handles are called *normal.*

Notice that all the zig-zags are formed by two horizontal segments and one vertical segment. A *zig-zag path* is a path consisting only of zig- zags. Given any two vertices a, b, of G_0, there is at most one zig-zag path connecting them. If such path exists, it will be denoted by $Z(a, b)$ and the vertices a, b will be called *equivalent.* We make the convention that every vertex is equivalent to itself. Clearly, the above-defined relation is reflexive, symmetric, and transitive: the corresponding equivalence classes $V_1, \ldots, V_q$ (*zig-zag sets*) form a partition of V_0.

We will describe a procedure that eliminates all zig-zags, replacing them by 0-bend edges. Essentially, the procedure assigns to all vertices in a same zig-zag set the same y-coordinate, while keeping their x-coordinates unchanged.

Let us say that V_j is *above* V_i ($V_j > V_i$) if there is at least one (directed) edge uv with $u \in V_i$ and $v \in V_j$.

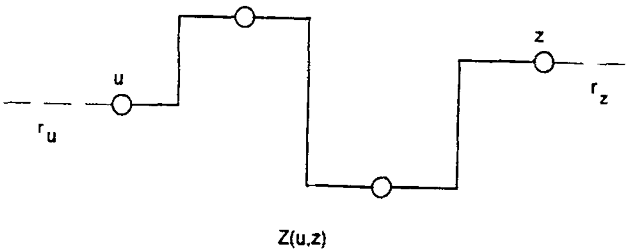


Fig. 7. The broken line L.

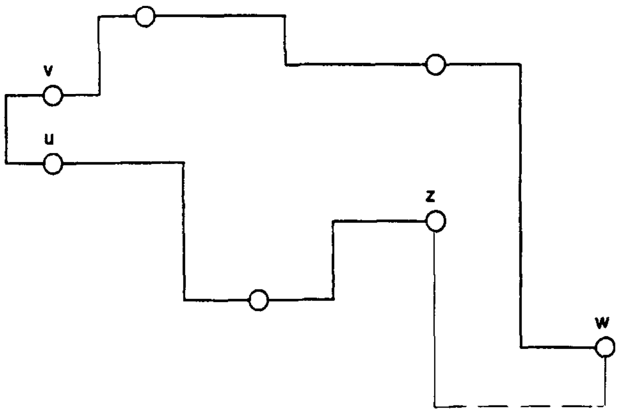


Fig. 8. Proof of Lemma 3.1, case 1.

Lemma 3.1. *The "aboveness" relation is asymmetric.*

Proof. Suppose that there are two edges uv, wz such that $u, z \in V_i$ and $v, w \in V_j$. Without loss of generality, let $x_u < x_z$. Consider the two half- rays

$$r_u = \{(x, y_u): -\infty < x \leqslant x_u\} \quad \text{and} \quad r_z = \{(x, y_z): x_z \leqslant x < +\infty\}.$$

Consider the broken line $L = r_u\, Z(u, z) r_z$ and the two open regions R^+ and R^- lying above and below L, respectively (see Fig. 7).

By the algorithm, v must belong to R^+ and w to R^-. It follows that $v \neq w$. We distinguish two cases.

Case 1: *$Z(v, w)$ is incident to v from east.* In this case, since the rectilinear embedding of G_0 is planar and uv is normal, one must have $x_w > x_z$. Since the west direction is already occupied both at z and at w, one needs a horizontal handle to connect w and z while preserving planarity (see, e.g., Fig. 8): but this is impossible since wz is also a normal edge.

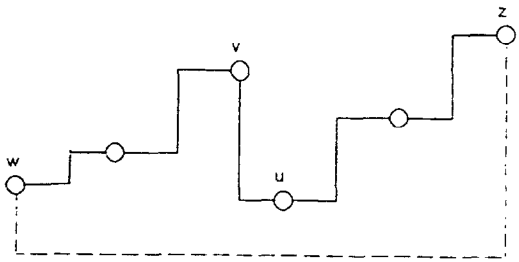


Fig. 9. Proof of Lemma 3.1, case 2.

Case 2: $Z(v, w)$ *is incident to v from west.* In this case, since the rectilinear embedding of G_0 is planar and uv is normal, one must have $x_w < x_v \leqslant x_u < x_z$. But then one needs again a horizontal handle to connect w and z while preserving planarity (see, e.g., Fig. 9) and we get a contradiction also in this case. □

The above lemma implies that if there is at least one edge going from V_i to V_j, then all edges between V_i and V_j are directed from V_i to V_j.

In order to find the y-coordinates of each vertex we use the following procedure.

Construct the digraph $D = (N, A)$ where $N = \{V_1, \ldots, V_q\}$ and in A there is an edge from V_i to V_j if in G_0 there is an edge from a vertex $u \in V_i$ to a vertex $v \in V_j$.

The digraph D can be generated in linear time. By Lemma 3.1, D has a bipolar orientation. Let V_1 be the source.

We define the *level* of a vertex V_i, denoted by level (V_i), to be the maximum number of edges of a path from V_1 to V_i. The levels of the vertices of D can then be computed in linear time by the "critical path" recurrence

$$\text{level}(V_1) = 0; \ \text{level}\ (V_j) = \max\{\text{level}(V_i) + 1\colon\ V_iV_j \in A\}.$$

In view of the fact that $\{V_1, \ldots, V_q\}$ is a partition of the vertex set of G_0, the y-coordinates of all vertices of G can be computed as

$$y(w) = y(s_0) + \text{level}(V_j) \text{ for all } w \in V_j,\ j = 1, \ldots, q.$$

Notice that this variant of the algorithm still requires linear time and space.

For the rectilinear embedding generated in Fig. 4, Fig. 10(a) shows the corresponding digraph D, where each vertex is a zig-zag set labelled with its level and Fig. 10(b) shows the modified rectilinear embedding.

4. A 2-embedding of biconnected graphs

In this section we show that, if G is not the octahedron, the algorithm of Section 2 yields a 2-embedding starting from a suitable embedding and a suitable unilateral numbering of G.

To prove this result we need several definitions and lemmas.

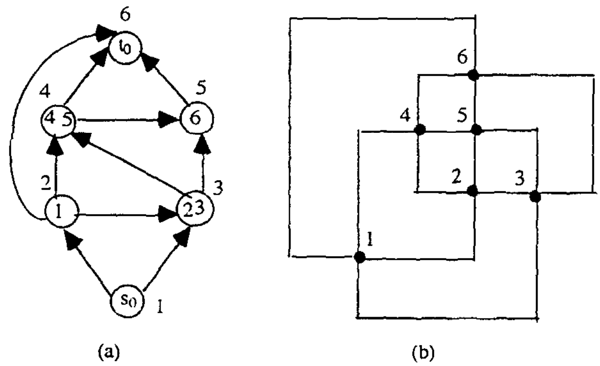


Fig. 10. Digraph D and embedding with no zig-zags.

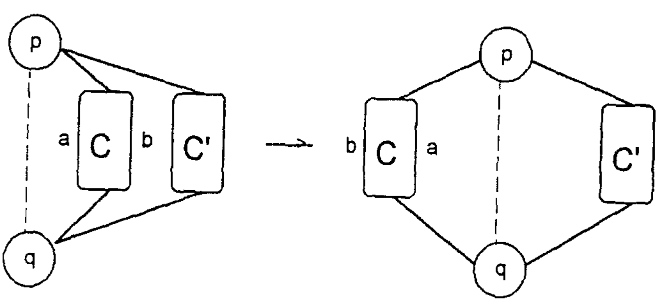


Fig. 11. A reflection around the separation pair $\{p, q\}$. The edge pq may or may not be present.

A *separation pair* of an arbitrary biconnected graph $G = (V, E)$ is a pair of vertices $\{p, q\}$ such that the subgraph induced by $V - \{p, q\}$ is disconnected.

Given a biconnected planar graph G, every embedding of G can be obtained from any other embedding through a finite sequence of reflections around separation pairs (see Fig. 11; for a formal definition of reflection see [4]).

Lemma 4.1. *Let $\{p, q\}$ be a separation pair of G, and let pq be an edge on the outerface boundary of a given planar embedding of G. Then, except for pq, every edge on the outerface boundary still remains on the outerface boundary after a reflection around $\{p, q\}$.*

Proof. Obvious. □

Let $B(G)$ be the block graph of G, i.e., the graph whose vertices are the blocks (maximal biconnected induced subgraphs) of G and two vertices are adjacent if and only if the corresponding blocks have a cut-vertex of G in common. $B(G)$ is a tree and its leaves are called *leaf-blocks*.

Lemma 4.2. *If G is any biconnected graph and x is any vertex, then there is at least one vertex y, adjacent to x and such that $\{x, y\}$ is not a separation pair.*

Proof. The graph $G' = G - x$ is connected and if T is any leaf-block of G', then x must be adjacent to some vertex y of T which is not a cut-vertex of G', else G would not be biconnected. Thus, $\{x, y\}$ is not a separation pair. □

Lemma 4.3. *Under the hypotheses of Lemma* 4.1, *there is a vertex r in the inner domain of the outerface boundary of G such that $\{p, r\}$ is not a separation pair. Furthermore, if the degree of p is at most* 4, *by performing at most two reflections one can obtain a new plane embedding of G in which the edge pr lies on the outerface boundary.*

Proof. It follows from Lemma 4.2 and from the fact that every vertex has at most degree 4. □

Lemma 4.4. *Let G be any biconnected standard graph, with at least one face boundary different from a triangle. Starting from any given plane embedding of G whose outerface boundary consists of at least* 4 *edges, and performing at most* 4 *reflections, one can find a planar embedding of G and four distinct vertices s, t, u, v such that sv and ut are edges on the outerface boundary and neither $\{s, v\}$ nor $\{u, t\}$ is a separation pair of G.*

Proof. Choose s and t to be two non-adjacent vertices on the outerface boundary C of G. Let sx be the rightmost edge going out from s and yt the rightmost edge going into t. If $\{s, x\}$ is not a separation pair of G, set $v = x$. Otherwise, by Lemma 4.2 there is a vertex v in the inner domain of C such that $\{s, v\}$ is not a separation pair. In view of Lemma 4.3, by performing at most two reflections, one can obtain a planar embedding of G in which sv lies on the outerface boundary. From now on, we shall assume that the planar embedding of G has this property. Notice that by Lemma 4.1 the edge yt remains on the outerface boundary. If $\{y, t\}$ is not a separation pair, set $u = y$. Otherwise, by Lemma 4.2 there is a vertex u adjacent to t in the inner domain of the outerface boundary (and hence different from s, t and v) such that $\{u, t\}$ is not a separation pair. After Lemma 4.3, by performing at most 2 reflections one can move edge ut onto the outerface boundary. By Lemma 4.1 edge sv remains on the outerface boundary and the thesis follows (see Fig. 12). □

If G is any biconnected graph and xy is an edge of G, the *contraction* G_x^v is the graph obtained from G by removing x and by connecting to y every neighbour of x which was not already connected to y in G.

Theorem 4.5. *Let G be a biconnected standard graph, with at least one face boundary different from a triangle. Then G has an embedding and a unilateral numbering such that the edges* $(1, 2)$ *and* $(n - 1, n)$ *are both on the outerface boundary.*

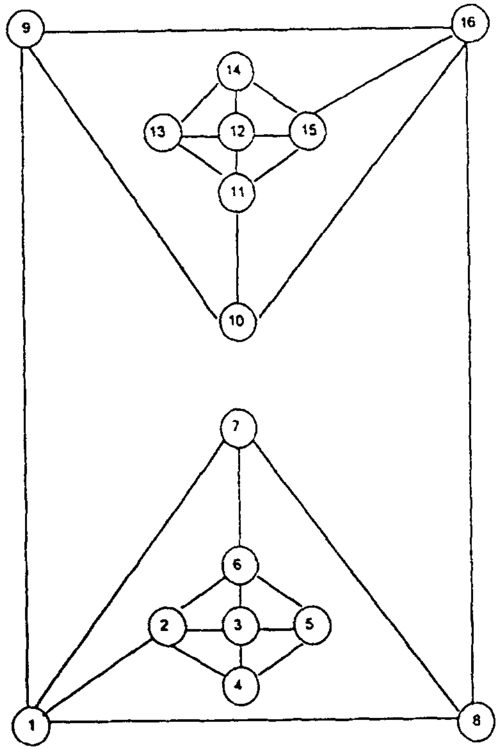


Fig 12. Graph for which four reflections are necessary.

Proof. Choose the vertices s, t, u, v according to Lemma 4.4. Consider the graph $G' = (G_s^v)_t^u$. Since $\{s, v\}$, $\{u, t\}$ are not separation pairs in G, and $\{u, t\}$ retains this property after the contraction of edge sv, the graph G' is biconnected and therefore there exists a bipolar orientation of G' with source v and sink u. Hence, the vertices of G' can be unilaterally numbered from 2 to $n - 1$. Now if vertex s is numbered 1 and vertex t is numbered n, we obtain a unilateral numbering of G with the desired property. Notice that in this numbering the vertices s, v, u and t are numbered 1, 2, $n - 1$ and n, respectively. □

Theorem 4.6. *If G is not the octahedron, then there exist an embedding and a unilateral numbering starting from which the algorithm produces a 2-embedding of G.*

Proof. If G is the octahedron, the edge st belongs to G and by Corollary 2.5 at least the edge st has 3 bends. Among the standard graphs, the only other graphs having only triangular faces are K_3, K_4, $K_5 - e$, for any edge e in K_5.

In the first two cases, any vertex has degree less than 4 and no edge with 3 bends is produced by the algorithm. In the third case, there is a vertex of degree 3 on the outerface boundary. Thus, we may choose s, u and t on the outerface boundary such that s is the vertex of degree 3, st is the rightmost edge going out from s and ut is the rightmost one coming into t. Then there exists a unilateral numbering where s, u and t are numbered 1, 4, and 5, respectively. Thus, both st and ut have 2 bends.

For any other standard graph there exists at least one face boundary which is not a triangle. If we start from an embedding of G such that the outerface boundary has at least 4 edges, then by Theorem 4.5 there exist an embedding and a unilateral numbering such that the two edges with horizontal handles coincide with the edges $(1,2)$ and $(n-1, n)$. Since vertex 2 has only one incoming edge and vertex $n-1$ has only one outcoming edge, the edges $(1,2)$ and $(n-1,n)$ have 2 bends.

Therefore, the resulting embedding is actually a 2-embedding. □

5. The general case

If one wants to generalise the algorithm of Section 2 to a connected (but not necessarily biconnected) standard graph G, one is faced with two main difficulties: first of all, a bipolar orientation may not exist, hence we have to consider more general acyclic orientations with a single source and multiple sinks; second, sometimes when there are two connected subgraphs H_1 and H_2 separated by a cut-vertex w, one must embed H_2 inside a face C of H_1; when this happens, surely one should require that at least one vertex of C is embedded after all the vertices of H_2 have been embedded.

Let us examine these difficulties separately. We need some preliminary definitions. Given an embedding of G, a block of G is said to be *terminal* if its outerface boundary contains at most one cut-vertex of G. Obviously, all leaf-blocks are terminal, but the converse does not have to be true: e.g., in the graph of Fig. 13(b) the block containing vertex 12 is terminal, even though it is not a leaf-block.

It is well-known (see, e.g., [21]) that a graph G admits a bipolar orientation if and only if its block tree is a path.

In order to deal with arbitrary connected graphs, we must introduce a more general kind of orientation satisfying, for a given embedding Γ of G, the following axioms:

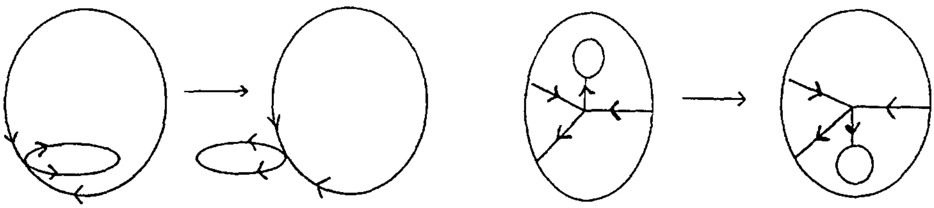
(P1) The orientation induced in each block is bipolar, and both the source and the sink lie on the outerface boundary of the block,

(P2) G has a unique source s; moreover, s belongs to the outerface boundary of G and it is not a cut-vertex of G; let us call B_1 the block containing s,

(P3) All the sinks of G belong to terminal blocks; B_1 contains no sink of G when it has exactly one cut-vertex of G on its outerface boundary.

Any orientation satisfying (P1)–(P3) will be called *divergent* (w.r.t. Γ). A divergent orientation is obviously acyclic, and no cut-vertex of G is a sink of G.

Notice that any vertex-numbering agreeing with a divergent orientation has the property that every vertex $j > 1$ is always adjacent to some vertex $i < j$.



(a) (b)

Fig 13. Relocation of inner bridges.

We shall now discuss how to find 3- and 2-embeddings of arbitrary connected standard graphs. It turns out that the same algorithm of Section 2 works, provided that a suitable embedding and a suitable divergent orientation are at hand.

The simplest case occurs when the given embedding of G has the following property: the outerface boundary of G is the union of the outerface boundaries of the blocks of G.

In this case all terminal blocks are actually leaf-blocks. In order to get a 3-embedding of G, all one needs is an arbitrary divergent orientation (w.r.t. Γ) and an agreeing vertex-numbering.

A divergent orientation of G can be constructed in $O(n)$ time. In fact, one can obtain in $O(n)$ time the block tree $B(G)$, which is then rooted at any of its blocks, say B_1. Starting from B_1 and proceeding in breadth-first order, one finds in every block a bipolar orientation making sure that properties (P1)–(P3) are satisfied. From such orientation an agreeing vertex-numbering is obtained. Then in order to find a 3-embedding of G one performs basically the same algorithm of Section 2, the only difference being that, since now there might be multiple sinks, the algorithm stops only after the last sink has been embedded. One can check that Propositions 2.1–2.3 remain true. It follows that the algorithm is still correct.

Furthermore, unless G is the octahedron, the algorithm will always produce a 2-embedding of G starting from an embedding and a divergent orientation with the following additional property:

(P4) If the source s has outdegree 4, then there is a successor of s which lies on the outerface boundary of B_1 and has indegree 1; if a sink t_i of G belonging to the terminal block B_i has indegree 4, then there is a predecessor of t_i which lies on the outerface boundary of B_i and has outdegree 1.

Lemmas 4.3 and 4.4 ensure that such embedding and such orientation can be constructed in $O(n)$ time. Then, proceeding as in the proof of Theorem 4.6, one can find an appropriate agreeing vertex-numbering which gives rise to a 2-embedding.

Before discussing the general case, we need some further terminology.

If C is a cycle of G, a *bridge* H (with respect to C) is a connected component of $G - C$ together with all edges (and their ends) of G joining this component to C. If H lies in the inner domain of C then H is called an *inner bridge*. Those vertices of

H that lie on C are called the *vertices of attachment* of H. If H has only one vertex of attachment w, then w is a cut-vertex of G. In any embedding of G, w belongs to at least two faces. By placing H within any other face incident to w one obtains a non-equivalent embedding of G. Any such operation is called a *relocation* of H. Given an embedding of G, call a cycle *minimal* (this term is adapted from Ore [18]) when each of its inner bridges, if any, has a unique vertex of attachment. Notice that, if H is an inner bridge w.r.t. some cycle and has one vertex of attachment, then H is such also w.r.t. a (unique) *minimal* cycle.

We are going to show that essentially the same algorithm of Section 2 produces a 2-embedding of G, provided that a suitable embedding and a suitable nested numbering of G are at hand.

Let $B(G)$ the block tree of G and let $\{B_1, B_2, \ldots, B_k\}$ be its vertex-set.

The embedding algorithm for simply connected graphs relies on the following result.

Theorem 5.1. *Starting from an arbitrary embedding, one can find an embedding Γ and a divergent orientation* (w.r.t. Γ) *such that* (P4) *holds and, in addition, the following property is satisfied*:

(P5) For every minimal cycle C of G and for every inner bridge H of C, the (unique) vertex of attachment of H has a successor along C.

Proof. An algorithmic proof will be given.

Find the cut-vertices and the blocks of G. Construct the block tree $B(G)$. Generate the embedding of each block and then an embedding of G by merging, for each cut-vertex w of G, the ordered adiacency lists of w corresponding to the blocks separated by w.

Choose on the outerface boundary of G a vertex s which is not a cut-vertex of G. If s has degree 4 verify if at least one of its neighbours lying on the outerface boundary does not form with s a separation pair; if it is not so, we can always find one after at most two reflections, by Lemma 4.3.

Let B_1 be the block containing s. Root $B(G)$ at B_1. Explore $B(G)$ in breadth-first order. Let B_i be the current block. Define the source s_i of B_i: if $i = 1$ then $s_i = s$; if $i \neq 1$, s_i is the unique cut-vertex of G separating B_i from its father in $B(G)$. Consider the outerface boundary C_i of B_i. If C_i contains at least a cut-vertex w of G, $w \neq s_i$, then choose w as the sink t_i of B_i. If there is none then choose any vertex u different from s_i such that if u has degree 4, at least one of its neighbours lying on C_i does not form with u a separation pair; if it is not so by Lemma 4.3 we can always find one after at most two reflections. If a reflection causes some cut-vertex w to appear on the new outerface boundary of B_i then choose w as the sink t_i. Then find a bipolar orientation of B_i with source s_i and sink t_i. Finally, for each vertex w of B_i that is a cut-vertex of G relocate, if necessary, the bridges attached only to w within a suitable face incident to w so that (P5) holds. In fact, any such bridge H is contained in the inner domain of a unique minimal cycle C. Consider the block B containing C. If w lies on the outerface boundary of B (Fig. 13(a)), then H can be relocated within the outerface of B; otherwise (Fig. 13 (b)), w cannot coincide with the sink of B, which lies on the outerface boundary of B. □

A numbering of the vertices of G from 1 to n is said to be *nested* (with respect to a given embedding of G) if, for every minimal cycle C and for every inner bridge H of C, there is at least one vertex i of C such that $i > j$ for all vertices j of H.

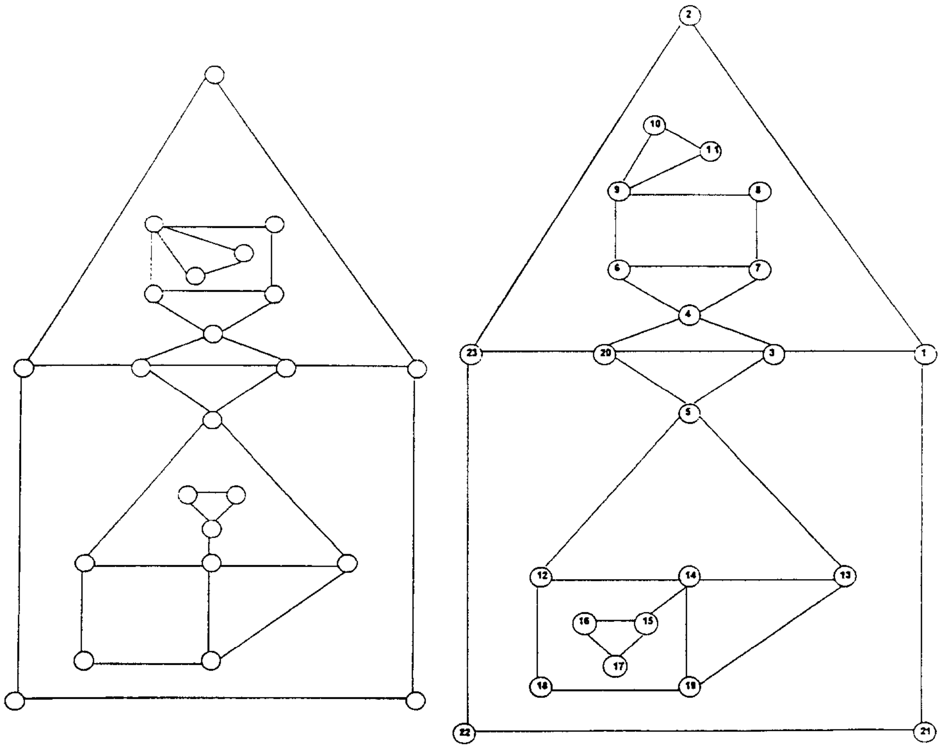
Corollary 5.2. *Given an embedding Γ and an acyclic orientation as in Theorem* 5.1, *there exists a nested numbering with respect to Γ which agrees with the orientation.*

Proof. Once the orientation is known, the numbering can easily be obtained as follows: Introduce a supersink t within the outerface of G. Each sink t_i may be of two types: (a) it belongs to the outerface boundary of G; (b) it belongs to the outerface boundary of some inner bridge H of a (unique) minimal cycle C. In case (a), add a dummy edge from t_i to t. In case (b), add a dummy edge from t_i to a successor of the (unique) vertex of attachment of H. One such successor always exists by Theorem 5.1. Since such augmented graph is biconnected, a unilateral numbering of its vertices, where s is numbered 1 and t is numbered $n + 1$, exists, and it induces a nested numbering of the vertices of G, which clearly agrees with the given orientation of G. □

For the graph of Fig. 14(a), Fig. 14(b) shows an embedding, an acyclic orientation and a nested numbering satisfying the conditions of Theorem 5.1 and Corollary 5.2.

Theorem 5.3. *For any standard graph an embedding, a divergent orientation and a nested numbering can be found in linear time.*

Proof. The cut vertices, the blocks, and the block-tree of G can be generated in $O(n)$ time [25]. The face boundaries of the given embedding of G can also be obtained with the same complexity [4]. In order to find an appropriate embedding of G we need to consider, for each block B_i, all the cut-vertices of G that belong to B_i so as to choose the source and the sink of the block, to find a bipolar orientation of B_i and to relocate the inner bridges in a suitable way if needed. Recognizing the need of any relocation (as in Fig. 13) requires only information about the edges incident to the cut-vertex w under consideration: namely, their rotation and their blocks. Notice that, once the orientation of each block is known through the ordered list of successors of each vertex, an orientation of G satisfying (P5) is obtained simply by merging, for each vertex w, the ordered adiacency lists of the successors of w corresponding to the blocks separated by w. Since there are at most two inner bridges attached to a cut-vertex w and any operation of relocation is called for and executed in constant time, it follows that the total time to find the embedding and the orientation is linear.



(a) (b)

Fig 14. (a) A standard graph G and its initial embedding. (b) Embedding, divergent orientation and nested numbering required by the, rectilinear embedding algorithm.

Once the orientation is known, the construction of the augmented graph and its unilateral numbering can be obtained in linear time. □

Once a nested numbering is available, the 2-embedding algorithm is identical to the one of Section 2, except that there are multiple sinks.

Theorem 5.4. *Given a nested numbering the algorithm of Section* 2 *produces a planar* 2-*embedding.*

Proof. Planarity is ensured by the nested numbering. Notice that Propositions 2.1–2.3 still hold. Moreover, (P4) ensures that a 2-embedding is obtained. □

Our final result yields an upper bound on the total number β of bends. Without loss of generality, one may assume that G has at least a cut-vertex on its outerface boundary. Then one can obtain in linear time, possibly after some bridge relocations, an embedding in which the outerface boundary of G contains the outerface boundary of some terminal block. We shall assume that vertex 1 belongs to one such block.

Theorem 5.5. *For an arbitrary standard graph with n vertices, the total number of bends of the embedding produced by the algorithm satisfies the inequality*

$$\beta \leqslant (\tfrac{7}{3})\, n. \tag{5.1}$$

Proof. Let G be a standard graph with n vertices. Let q be the maximum cardinality of a terminal block of G, and let b_h be the number of its terminal blocks with cardinality h, $h = 3, \dots, q$.

Since in any planar graph the number of edges is at most $3n - 6$, in any terminal block with cardinality $h \leqslant 5$ there are at least two vertices of degree at most 3. Since each vertex of degree at most 3 contributes no more than one bend and since in each terminal block there is at most one horizontal handle, the contribution of each terminal block with cardinality $h \leqslant 5$ is at most $2h$ bends. It follows that

$$\beta \leqslant 2n + 2(b_6 + \cdots + b_q). \tag{5.2}$$

The special case when G consists of two blocks with a common vertex, and both having 6 vertices, will be dealt with separately. In all remaining cases one has

$$6(b_6 + \cdots + b_q) \leqslant n,$$

which together with (5.2) implies (5.1).

We are left only with the above-mentioned special case. First of all, the common vertex of the two blocks must have degree 2 within each block. Since (4, 4, 4, 4, 4, 2) cannot be the degree sequence of a planar graph, and disregarding the trivial case when both blocks are cycles, G must have at least two vertices of degree 3. In view of Theorem 2.8 (which remains valid also for G) one has $\beta \leqslant n_3 + 2n_4 + 4$. Since $2 \leqslant n_3 \leqslant 11 - n_4$ one gets $\beta \leqslant 24 \leqslant (\tfrac{7}{3})n$. Hence, (5.1) holds also in this case. □

6. Conclusions

In this paper we have presented a new linear-time algorithm for producing a rectilinear embedding in the plane of an arbitrary planar graph with maximum degree at most 4. The distinctive new feature of the algorithm is that a 2-embedding is always guaranteed, with the single exception (among connected graphs) of the octahedron, which does not admit any 2-embedding and for which a 3-embedding is produced.

Our algorithm does remarkably well also with respect to other layout criteria. The total number of bends is bounded above by $2n + 4$ (same as in [24]) when the graph is biconnected, and by $(\tfrac{7}{3})n$ (with strict improvement on previous linear-time algorithms) for arbitrary connected graphs.

Table 1

Group	1	2	3	4	5	6	7	8	9	10
Average number of nodes	54.2	106.4	158.8	208.6	257.6	309.6	358	418	466.6	524.6

Table 2
Two-Bend

Group	1	2	3	4	5	6	7	8	9	10
0-bend	41.7	79.8	121.6	165.2	198.5	235.7	281.1	309	357.8	376
1-bend	50	99.3	149.2	187.5	240.4	291.1	336.5	393.6	436.9	495.4
2-bend	15.7	32.7	45.4	63	73.5	89.8	106	131.1	136	154.4
3-bend	0	0	0	0	0	0	0	0	0	0
Bends	81.4	164.7	240	313.5	387.4	470.7	548.5	655.8	708.9	810.2
Bends/n	1.501	1.545	1.511	1.499	1.504	1.519	1.51	1.568	1.52	1.522

The area of the layout is $(n + 1)^2$, to be compared with the $\Omega(n^2)$ lower bound established in [26].

Some of the ingredients of the algorithm have been used in the context of graph drawing by other authors. In particular, bipolar orientations and unilateral numberings, together with the related notions of *st*-numberings and upward drawing, have already appeared in [4–9]. Critical path methods, exploited here in the zig-zag elimination routine, have been used in [11] to produce compact embeddings. On the other hand, the concepts of divergent orientation and nested numbering appear to be new.

Perhaps one of the main conceptual contributions of the present paper consists in pointing out the usefulness of even a slight modification of the input planar embedding in order to obtain rectilinear embeddings with some desirable properties.

Thus, when the graph is biconnected, a 3-embedding is guaranteed if one insists that the output embedding be equivalent to the input embedding; however, at most 4 reflections suffice to change the starting embedding so that (except for the octahedron) the algorithm outputs a 2-embedding. When the graph is simply connected, it might just not be possible to obtain even 3-embeddings starting from arbitrary planar embeddings, and one needs to change the input embedding by performing one or more relocations of inner bridges.

Our algorithm has been tested, with encouraging results, on an initial set of 100 randomly generated 4-regular biconnected planar graphs. These graphs are divided into 10 groups in which the number of vertices is about 50, 100, 150, ..., 500, respectively. The algorithm TWO-BEND of this paper has been compared with the linear-time algorithm of Tamassia and Tollis [24] and with the $O(n^2 \log n)$ algorithm of Tamassia [22], which minimizes the total number of bends for any fixed embedding and hence has been taken as a benchmark. Table 1 gives the average number of vertices in each group. Tables 2–4 yield the total number of edges with 0, 1, 2, and

Table 3
Tamassia-Tollis

Group	1	2	3	4	5	6	7	8	9	10
0-bend	34.4	74	110.6	138.8	185.1	215.4	273.9	290.3	335.2	342.3
1-bend	58.6	104	159.3	212.9	254.1	304.8	341.7	417	462.8	535.6
2-bend	13.8	33.2	45.6	63.5	72.4	95.5	107.3	126	132.4	147.3
3-bend	0.6	0.6	0.7	0.5	0.8	0.9	0.7	0.4	0.3	0.6
Bends	88	172.2	252.6	314.4	401.3	498.5	558.4	670.2	728.5	832
Bends/n	1.623	1.614	1.589	1.634	1.559	1.609	1.538	1.603	1.562	1.615

Table 4
Tamassia

Group	1	2	3	4	5	6	7	8	9	10
0-bend	42.6	92.9	138.3	168.2	194	245	293.5	320.3	363.6	392.7
1-bend	58.5	109.7	160.1	218.7	273.8	333.8	385.3	459.8	507.8	567.5
2-bend	5.2	12.7	17.7	25.1	33.4	37.3	44.5	52.8	58.9	65.1
3-bend	0.1	0.4	0.5	0.8	0.5	0.5	0.3	0.8	0.4	0.8
Bends	70.4	137.2	197	271.3	343.1	409.4	475.2	567.8	626.9	700.1
Bends/n	1.281	1.291	1.239	1.298	1.326	1.323	1.3	1.358	1.343	1.359

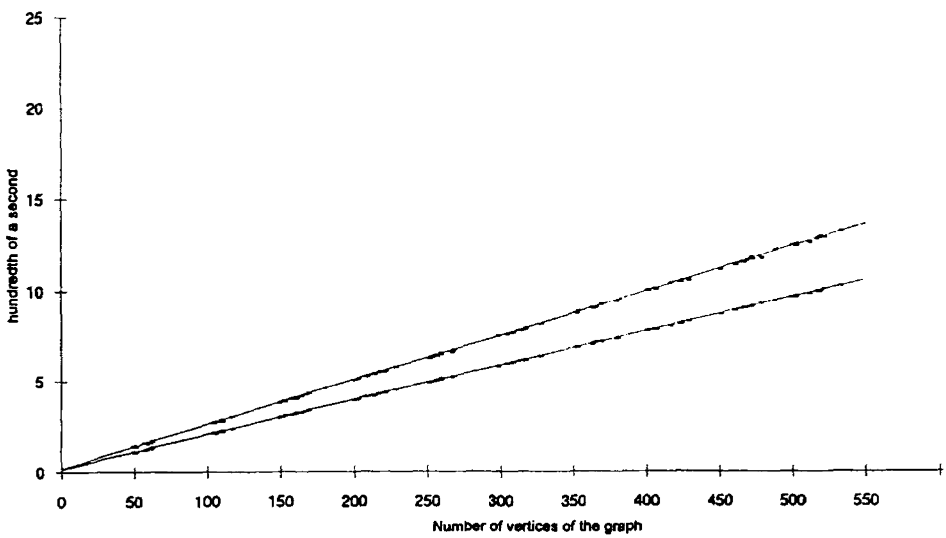


Fig 15. Running times (CPU 10^{-2} s. on 486 DX 33 Mhz) versus number of vertices. The upper line refers to Tamassia-Tollis' algorithm, the lower one to Two Bend.

3 bends, the total number of bends, and the ratio between the latter and n. All values are averaged over the ten problems in each group. Tamassia and Tollis' algorithm produced few 3-bend edges in 37% of the test problems. The running times of the two linear-time algorithms are shown in Fig. 15. A comprehensive experimental

comparison with other algorithms is under way, and its results will be presented in another paper.

Acknowledgements

We are grateful to Claudio Raimondo and Pierguido Iezzi for their support in writing and running the computer programs.

References

[1] A. Aggarwal, M. Klawe, D. Lichtenstein, N. Linial, A. Wigderson, Multi-layer grid embedding, Proc. IEEE 26th Ann. Symp. Foundations of Computer Science, 1985, pp. 186–196.
[2] A. Aggarwal, M. Klawe, P. Shor, Multi-layer grid embedding for VLSI, Algorithmica 6 (1991) 129–151.
[3] S. Bhatt, S. Cosmadakis, The complexity of minimizing wire lengths in VLSI layout, Inform. Process. Lett. 25 (1987) 263–267.
[4] N. Chiba, T. Nishizeki, Planar Graphs: Theory and Algorithms, North-Holland, Amsterdam, 1988.
[5] X. Cui, Y. Liu, Efficient algorithms for rectilinear extensions of standard graphs, Acta Math. Appl. Sinica, to appear.
[6] G. Dolev, F.T. Leighton, H. Trickey, Planar embedding of planar graphs, in: F.P. Preparata (Ed.), Advances in Computing Research, vol. 2, Jai Press, 1984, pp. 147–161.
[7] J. Ebert, *st*-numbering the vertices of biconnected graphs, Computing 30 (1983) 19–33.
[8] S. Even, Graph Algorithms, Computer Science Press, Potomac,1979.
[9] S. Even, R.E. Tarjan, Computing an *st*-numbering, Theoret. Comput. Sci. 2 (1976) 339–344.
[10] J. Hopcroft, R.A. Tarjan, Efficient planarity testing, J. Comput. Mach. 21 (1974) 549–568.
[11] B.H. Jackson, Directed graphs and the compaction of IC designs, Contemp. Math. 89 (1989) 107–127
[12] A. Lempel, S. Even, I. Cederbaum, An algorithm for planarity testing of graphs, in: P. Rosenstiehl (Ed.),Theory of Graphs, Internat. Symp., Rome, July 1966, pp. 215–232.
[13] Y.P. Liu, On the linearity of testing planarity of a graph, Combin. Optim. CORR 84–5. University of Waterloo, 1984.
[14] Y.P. Liu, A new approach to the linearity of testing planarity of graphs, Acta Math. Appl. Sinica 4 (1988) 257–265.
[15] Y. Liu, A. Morgana, B. Simeone, General theoretical results on rectilinear embeddability of graphs, Acta Math. Appl. Sinica 7 (1991) 187–192.
[16] Y. Liu, A. Morgana, B. Simeone, A linear algorithm for 3-bend embedding of planar graphs in the grid, Technical Report, Ser. A, no. 1, Dipartimento di Statistica, Probabilità e Statistiche Applicate, Università La Sapienza, Rome, Italy, 1993.
[17] Y. Liu, Embeddability in Graphs, Kluwer, Amsterdam, 1995.
[18] O. Ore, The Four Colour Problem, Academic Press, New York, 1967.
[19] Y. Shiloach, Linear and planar arrangement of graphs, Ph. D. Dissertation, Weizmann Institute, Rehovat, Israel (1976).
[20] J.A. Storer, On minimal node-cost planar embeddings, Networks 14 (1984) 181–212.
[21] J.L. Szwarcfiter, R. Persiano, A. Oliveira, Orientations with single source and sink, Discrete Appl. Math. 10 (1985) 313–321.
[22] R. Tamassia, On embedding a graph in the grid with the minimum number of bends, SIAM J. Comput. 16 (1987) 421–444.
[23] R. Tamassia, I.G. Tollis, A unified approach to visibility representation of planar graphs, Discrete Comput. Geom. 1 (1986) 321–341.
[24] R. Tamassia, I.G. Tollis, Planar grid embedding in linear time, IEEE Trans. Circuits Systems 36 (1989) 1230–1234.
[25] R.E. Tarjan, Depth-first search and linear graph algorithms, SIAM J. Comput. 1 (1972) 146–160
[26] L.G. Valiant, Universality Considerations in VLSI, IEEE Trans. Comput. 30 (1981) 135–140.