Lower Bounds and Parallel Algorithms for Planar Orthogonal Grid Drawings*
(Extended Abstract)

Roberto Tamassia
Department of Computer Science
Brown University
Providence, R.I. 02912–1910

Ioannis G. Tollis
Department of Computer Science
The University of Texas at Dallas
Richardson, Texas 75083–0688

Jeffrey S. Vitter
Department of Computer Science
Brown University
Providence, R.I. 02912–1910

Abstract

We consider the problem of constructing a planar orthogonal grid drawing *(or more simply,* layout*) of an n-vertex graph, with the goal of minimizing the number of bends along the edges. We exhibit graphs that require $\Omega(n)$ bends in any layout, and show that there exist optimal drawings that require $\Omega(n)$ bends and have all of them on a single edge of length $\Omega(n^2)$. On the other side of the coin, we present a parallel algorithm that runs on a CREW PRAM in $O(\log n)$ time with $n/\log n$ processors and constructs layouts with $O(n)$ maximum edge length and $O(n^2)$ area. For biconnected graphs the number of bends is at most $2n+4$, which is optimal in the worst-case. This work finds applications in VLSI layout, aesthetic graph drawing, and communication by light or microwave.*

1 Overview

The problem of constructing drawings of graphs has important applications to circuit layout and data presentation. A survey of graph drawing algorithms appears in [5,6]. In this paper we consider ***planar orthogonal grid drawings***, or, simply ***layouts***, which are planar drawings such that the edges are polygonal chains consisting of horizontal and vertical segments, and the vertices and bends of the edges have integer coordinates. A specific survey on algorithms for constructing such drawings is given in [11].

In numerous VLSI and graphics applications [9] the main quality measures for a layout are minimizing the area, edge length, and number of bends. Minimizing bends also has applications to readability and aesthetics, to communication by light or microwave, and to transportation problems. The problems of minimizing the area and total edge length in layouts are NP-complete [4,9]. In contrast, layouts with a minimum number of bends can be constructed in polynomial time [10].

In Section 2, we provide existential lower bounds on the minimum number of bends required by layouts. Our results are summarized in the following table. Each line corresponds to a family of graphs, and gives for a generic n-vertex graph of the family the connectivity, the number of multiple edges and self-loops, and the minimum number of bends for its layout.

connect.	mult. edges	self-loops	bends
0	0	$\Theta(n)$	$8n$
0	$\Theta(n)$	0	$4n$
1	$\Theta(n)$	$\Theta(n)$	$4(n-2)$
1	$\Theta(n)$	0	$8(n-2)/3$
2	2	0	$2n+4$
2	0	0	$2n-2$

*Research supported in part by Cadre Technologies Inc., by the National Science Foundation under grant CCR–9007851, by Presidential Young Investigator Award CCR–8906419 from the National Science Foundation, with matching funds from IBM Corporation, by the Office of Naval Research and the Defense Advanced Research Projects Agency under contract N00014-83-K-0146 and ARPA order 6320, amendment 1, by the Texas Advanced Research Program under Grant No. 3972, and by the U.S. Army Research Office under grant DAAL03-91-G-0035.

We also show a family of graphs for which any optimal layout has $\Omega(n)$ bends on a single edge of length $\Omega(n^2)$. For the proof of the lower bounds we use a technique which is reminiscent of the amortized time complexity analysis of algorithms [14].

Previously, Storer presented a polynomial-time algorithm that constructs layouts of n-vertex biconnected graphs with at most $2n+4$ bends, and exhibited a family of connected graphs without multiple edges and self-loops that require $8n/7$ bends [9]. (Actually, in [9] the bound is incorrectly reported as $10n/7$.) An $O(n)$-time algorithm that constructs layouts of biconnected graphs with $O(n)$ maximum edge length, $O(n^2)$ area, and at most $2n+4$ bends was given by Tamassia and Tollis [12].

In Section 3, we consider the problem of constructing layouts in parallel. We present an optimal parallel algorithm that constructs a layout of an n-vertex graph in $O(\log n)$ time using a CREW PRAM with $n/\log n$ processors. The layout has $O(n)$ maximum edge length, $O(n^2)$ area, and at most $2n+4$ bends if the graph is biconnected and at most $2.4n+2$ bends if the graph is simple. This is optimal in the worst case for biconnected graphs. The parallel algorithm follows the general scheme of [12], which consists of two phases: *orthogonalization* and *compaction*. The orthogonalization phase determines the "shape" of the layout, that is, the angles formed by the edges and bends. The compaction phase assigns grid coordinates to the vertices and bends. Previously, Tamassia and Vitter [13] parallelized a simpler version of the technique of [12] that avoids compaction. Their algorithm produces layouts with at most $6n$ bends [13].

We present a novel approach to the compaction phase, based on the concept of "symbolic decomposition" of a rectilinear polygon whose shape is fixed but whose geometry (vertex coordinates) is not fully specified a priori. Previous compaction algorithms were inherently sequential [8,10,15]. Parallelizing the compaction of a layout was left as an interesting open problem [8].

2 Lower Bounds

An *orthogonal drawing* of a graph is a drawing where the edges are polygonal chains consisting of horizontal and vertical segments (see Fig. 1). A graph admits an orthogonal drawing if and only if it has maximum vertex degree 4. Orthogonal drawings are typically used in circuit schematics and software engineering diagrams. A drawing is *planar* if no two edges intersect. A drawing is a *grid* drawing if the vertices and the bends have integer coordinates.

The topology of a planar orthogonal drawing is described by its *embedding*, which gives for each vertex the circular sequence of incident edges ordered clockwise according to the drawing. In the following we assume that an embedding is given along with the graph and the drawing has to preserve the embedding. Given an embedded planar graph G, a *layout* for G is a planar orthogonal grid drawing of G that preserves the embedding.

In a layout we shall consider two types of angles: those formed at the vertices by consecutive incident edges, and those along the edges. Since we deal with orthogonal drawings, we will measure angles in units of 90 degrees. We say that angles measuring 1, 2, or 3 units are *inflex*, *flat*, and *reflex*, respectively.

Let Γ be a layout of a graph G, and C an oriented closed simple curve drawn onto Γ. Curve C defines an *elementary transformation* of Γ if it intersects vertices only by entering from flat or reflex angles. The elementary transformation is obtained by "transporting" a unit angle across each vertex and edge intersected by C (see Figure 2). For each vertex v traversed by C, the transformation subtracts one unit from the angle where C enters, and it adds one unit to the angle where C exits. Also, for each intersection of C with an edge e, if C traverses e at a bend entering from the reflex angle, then the transformation removes that bend, otherwise (C traverses e entering from an inflex or a flat angle), it adds to e a bend with the reflex angle on the side where C exits.

Let *reflex*(C), *flat*(C), and *inflex*(C) be the number of edges that are traversed by C entering from a reflex, inflex, and flat angle, respectively. The variation in the number of bends caused by the elementary transformation defined by C is $\Delta B = \mathit{flat}(C) + \mathit{inflex}(C) - \mathit{reflex}(C)$. Curve C is said to be *trivial* if it intersects only one edge (going back and forth between two faces) and $\Delta B = 0$. The elementary transformation defined by a trivial curve does not change the layout.

The following characterization extends a result of [10]:

Lemma 1 *A layout Γ for a graph G has the minimum number of bends if and only if, for every curve C defining an elementary transformation,*

$$\Delta B = \text{flat}(C) + \text{inflex}(C) - \text{reflex}(C) \geq 0.$$

Also, Γ is the unique optimal layout for G if and only if, for every nontrivial curve C, $\Delta B > 0$.

Consider the biconnected multigraph G_n with n vertices, and its layout Γ_n shown in Figure 3, which has $2n+4$ bends. The following theorem shows that Γ_n has the minimum number of bends.

Theorem 1 *The minimum number of bends in any layout of G_n is $2n+4$.*

Proof: We use the condition expressed by Lemma 1. Since every vertex of G has degree 4, all curves defining elementary transformations cannot go through vertices. Hence every such curve C corresponds to a cycle in the dual graph of Γ_n. Let $\bar{f}$ be the innermost face of Γ_n. We assign a potential function Φ to the faces of Γ_n so that $\Phi(f)$ is equal to the distance from face f to face $\bar{f}$ in the dual graph. The potential value for each face is shown in Figure 3.

Let curve C be partitioned into arcs c_0, c_1, $\ldots$, c_{m-1}, where arc c_i goes from face f_i to face f_{i+1} and traverses exactly one edge, denoted e_i. Let ΔB_i be the variation of the number of bends along edge e_i caused by the elementary transformation defined by C. Let $\Delta\Phi_i = \Phi(f_{i+1}) - \Phi(f_i)$ be the variation of potential along arc c_i. When c_i goes "inward," it traverses an edge entering from a flat or a reflex angle, so that ΔB_i is $+1$ or -1, respectively; however, the potential decreases (that is, $\Delta\Phi_i = -1$). When c_i goes "outward", it traverses an edge entering from a flat or an inflex angle, so that $\Delta B_i = 1$; however, the potential increases (that is, $\Delta\Phi_i = 1$). We conclude that for every arc c_i we have

$$\Delta B_i - \Delta\Phi_i \geq 0. \qquad (1)$$

Since C is a closed curve, we have $\sum_{i=0}^{m-1}\Delta\Phi_i = 0$, which implies by (1) that the variation of the number of bends caused by the elementary transformation defined by C is $\Delta B = \sum_{i=0}^{m-1}\Delta B_i \geq 0$. Hence, there is no elementary transformation that reduces the number of bends. □

The multigraph G_n used in Theorem 1 has only two pairs of multiple edges. We can turn the multigraph G_n into a graph H_n and eliminate multiple edges by introducing two extra vertices, one for each pair of multiple edges in G_n. This results in two extra vertices, and it removes two bends. If we let n denote the resulting number of vertices, the minimum number of bends for H_n is $2n-2$.

Lemma 2 *Let Γ be a minimum bend layout of a graph G. If we replace a bend of Γ with a new vertex v, the resulting layout Γ' of the new graph G' has also the minimum number of bends.*

Lemma 2 implies the following corollary:

Corollary 1 *The minimum number of bends in any layout of H_n is $2n-2$.*

Note that we can get stronger lower bounds for non-biconnected graphs at the expense of extra multiple edges and/or self-loops. Consider the multigraph G_2 consisting of two vertices and four (multiple) edges between the two vertices. By Theorem 1, an optimal layout for G_2 has eight bends. By duplicating this repeatedly, we obtain a disconnected multigraph on n vertices requiring $4n$ bends. We achieve a lower bound of $8n$ bends with a disconnected graph whose components are vertices with two self-loops nested in the embedding. Note that, if the two self-loops are not nested, the lower bound is $6n$.

Figure 4.a shows a simply connected multigraph that requires $8(n-2)/3$ bends, and is obtained by "nesting" many copies of G_2. Each copy of G_2 requires eight bends, since the nested embedding makes it no longer possible to place the new vertices at bends. If we allow self-loops, we can further boost the bound for simply connected graphs to $4(n-2)$ bends (see Figure 4.b).

Now let us consider the biconnected graph S_n and its layout Σ_n shown in Figure 5. This layout has $(n-6)/9+4$ bends, and all the bends occur on the same edge e. Intuitively, any curve that defines an elementary transformation of Σ_n and traverses edge e must traverse at least two edges of S_n that have no bends. Using the condition of Lemma 1, we see that every time we try to remove one bend from e we introduce two new bends on other edges of S_n. This implies that e must spiral as shown in Figure 5 and have length $\Omega(n^2)$.

Before giving a formal proof we need the following definition: A *left* (respectively, *right*) *orthogonal spiral* is a layout of an edge such that going from one endpoint to the other we always make left (respectively, right) turns.

Lemma 3 *Any orthogonal spiral S with n bends has length $\Omega(n^2)$.*

Sketch of Proof: We say that an endpoint of S is *free* if its segment can be extended to infinity without intersecting the rest of S. First, we show by induction on the number of bends that any orthogonal spiral with a free endpoint and $m-1$ bends has length $\Omega(m^2)$. A *separator* of S is a non-extreme (not the first or last) segment s of S such that there is a straight line orthogonal to s that intersects S only at s. It easy to see that, if $n \geq 2$, S has a separator whose removal partitions S into two spirals each with a free endpoint. Thus, from the above property we have that S has length $\Omega(n^2)$. □

Theorem 2 *Every bend-optimal layout of the biconnected graph S_n has $\Omega(n)$ bends on a single edge of length $\Omega(n^2)$.*

Proof: As in the proof of Theorem 1 we will use the condition expressed by Lemma 1 and a "potential" argument. We assign to each face f a potential $\Phi(f)$ as shown in Figure 5. Note that the potentials of two adjacent regions differ by at most 1. Consider a curve C defining an elementary transformation of Σ_n and partition it into arcs $c_0, c_1, \ldots, c_{m-1}$, where arc c_i goes from face f_i to face f_{i+1} and traverses exactly one edge or one vertex. Let $\Delta\Phi_i = \Phi(f_{i+1}) - \Phi(f_i)$ and ΔB_i be the variation of the number of bends induced by the portion c_i of the curve. We consider three cases. If $\Phi(f_i) < \Phi(f_{i+1})$, then $\Delta B_i = 1$, so that $\Delta B_i - \Delta\Phi_i = 0$. If $\Phi(f_i) > \Phi(f_{i+1})$ and c_i does not traverse edge e, then $\Delta B_i = 1$ or $\Delta B_i = 0$, according to whether c_i traverses an edge or a degree-two vertex. Hence, $\Delta B_i - \Delta\Phi_i \geq 1$. If $\Phi(f_i) > \Phi(f_{i+1})$ and c_i traverses edge e, then $\Delta B_i = -1$ or $\Delta B_i = 1$, according to whether c_i traverses edge e at a bend or not. Thus, $\Delta B_i \geq -1$ and $\Delta B_i - \Delta\Phi_i \geq 0$. Since C is a closed curve, we have $\sum_{i=0}^{m-1} \Delta\Phi_i = 0$, which implies that the variation of the number of bends caused by the elementary transformation defined by C is $\Delta B = \sum_{i=0}^{m-1} \Delta B_i \geq 0$. Therefore, by Lemma 1, the layout Σ_n has the minimum number of bends. Also, it is easy to see that any nontrivial curve C must have an arc c_i with $\Phi(f_i) > \Phi(f_{i+1})$ that does not traverse edge e. Thus, for such an arc c_i, $\Delta B_i - \Delta\Phi_i \geq 1$. This implies that $\Delta B > 0$ and hence Σ_n is the unique optimal layout for S_n. Since edge e is an orthogonal spiral with $\Omega(n)$ bends, by Lemma 3, it must have length $\Omega(n^2)$. □

The above theorem shows that all bend-optimal layouts of S_n must have a quadratic maximum edge length. In contrast, the algorithm of the next section construct layouts with $O(1)$ bends on each edge and $O(n)$ maximum edge length.

3 Parallel Construction of Layouts

In this section we present an optimal parallel algorithm, called *GraphLayout*, that constructs high-quality layouts of n-vertex graphs. The algorithm, given below, follows the general approach of [12], and consists of several phases beginning with the construction of a "visibility representation." A ***visibility representation*** Γ for a planar graph G maps every vertex v of G to a horizontal segment $\Gamma(v)$, and every edge (u, v) to a vertical segment $\Gamma(u, v)$ that has its endpoints on $\Gamma(u)$ and $\Gamma(v)$ and does not intersect any other horizontal segment. An *orthogonal representation* is a symbolic description of the shape of a layout, in which the angles formed by the edges are specified, but the coordinates of the vertices and bends are not given.

We can easily convert a visibility representation into an orthogonal representation by replacing horizontal segments by vertices and vertical segments by edges with bends. Many of the bends that are introduced can be subsequently eliminated. For example, there is an elementary transformation T that reduces the number of bends on an edge that has reflex angles on both sides, as shown in Figure 6. It is important to note that T is local in nature and can be implemented easily in parallel. The hard part of algorithm *GraphLayout* is the final step (Step 4), in which the orthogonal representation is "parsed" in order to construct the final layout. For simplicity, we describe the algorithm for the case of biconnected graphs.

Algorithm *GraphLayout*
{ Layout of a biconnected planar graph G with n vertices. }

1. Construct a visibility representation for G. (See Figures 7(a) and (b).)
2. Transform the visibility representation for G into a preliminary orthogonal drawing Γ by substituting a grid point for each vertex segment and appropriately changing its connections with the incident edges. (See Figure 7(c).)
3. Let H be the orthogonal representation for Γ. Apply the local elementary transformation T described above (and others local elementary transformations if desired) to reduce the number of bends in H. (See Figure 7(d).)
4. From the orthogonal representation H, construct a layout using the compaction algorithm described below. (See Figure 7(e).)

We show in Theorem 3 that this algorithm constructs a layout for biconnecteds graph with at most $2n + 4$ bends, which by Theorem 1 is optimal in the worst case. If the graph G is not biconnected, we first decompose G into its connected and biconnected components. This can be done optimally in parallel [3, 7]. The layout of each component is constructed separately, taking special care of the articulation vertices so that the layouts can later be merged together.

The first step of Algorithm *GraphLayout* can be done in $O(\log n)$ time by an EREW PRAM with $n/\log n$ processors [13]. Steps 2 and 3 are local and can be done in constant time using one processor per vertex/edge. The last step is the most difficult to parallelize and is discussed in the remainder of this section.

The problem that remains is how to embed the orthogonal representation H in the grid without introducing any new bends. First, we identify the faces of Γ using techniques of [13], which we omit for purposes of brevity. Each face of H is a rectilinear polygon P of which only the angles are specified, while the coordinates of the vertices are left undetermined. Such polygons are hereafter referred to as *symbolic polygons.* The construction of a layout from the orthogonal representation H can be viewed as the process of assigning coordinates to a collection of symbolic polygons (the faces of H) that share vertices and edges.

We apply the subroutine *SymbolicDecomposition* given below to each face of H, in order to define a set of embedding constraints. The subroutine *SymbolicDecomposition* works by decomposing a symbolic polygon P into a set of symbolic rectangles (as shown in Figure 8). The partition of P into rectangles gives a set of horizontal and vertical adjacency constraints on the coordinates of the vertices in the layout. Each such set of constraints is represented by a planar *st*-graph (i.e., a planar acyclic digraph with one source s and one sink t, both on the external face). The final layout, of area $O(n^2)$, is then constructed in parallel by forming a total ordering of the vertical constraints and of the horizontal constraints using the optimal parallel topological sorting algorithm for planar *st*-graphs given in [13].

Algorithm *SymbolicDecomposition*
{ Decomposition of an symbolic polygon P into symbolic rectangles }

1. Starting with an edge e_0 of P, number and orient the edges of P in a counterclockwise fashion.
2. For each i, set $turn(i) := +1$ if e_i and e_{i+1} form a left turn, and set $turn(i) := -1$ otherwise.
3. For each i, compute $rot(i) := \sum_{j=0}^{i-1} turn(j)$.
4. For each i, find the first edge e_j following e_i counterclockwise such that $rot(j) = rot(i)+1$, and set $front(i) := j$.
5. For each i such that $turn(i) = -1$ (reflex angle), make a "cut" by extending edge e_i until it touches edge $e_{front(i)}$. If $front(i) = front(k) = j$, then the cuts extending e_i and e_k touch the same edge e_j; in this case the contact point of e_i follows the one of e_k along e_j if and only if e_i precedes e_k going in a counterclockwise direction from e_j.

Lemma 4 *The cuts computed by Algorithm* SymbolicDecomposition *induce a consistent decomposition of the symbolic polygon P into symbolic rectangles.*

Sketch of Proof: We show that no two cuts intersect and that each cut partitions P into two subpolygons P' and P'' with total $turn$ sum equal to 4. An inductive argument can then be applied to construct a drawing of P with the given decomposition.

Suppose, for purposes of obtaining a contradiction, that the cuts extending edges e_i and e_j intersect. Without loss of generality, assume that e_i is directed rightward, and e_j is directed downward (see Figure 9), so that $rot(i) = rot(j) + 1$. By connectivity arguments, it follows that e_i precedes $e_{front(j)}$ going in a counterclockwise direction from e_j. This contradicts the definition of $front(j)$, since $rot(i) = rot(front(j))$.

Let us consider the two subpolygons created by a cut. Without loss of generality, we can assume that the edge forming the cut is e_0. Let $i = front(0)$. The $turn$ sum for the polygon P' to the right of the cut, denoted $rot(P)$, is given by $rot(i) + 2 + 1$. Since $rot(i) = 1$, we have $rot(P') = 4$. A similar argument shows that $rot(P'') = 4$. □

Lemma 5 *Let P be a symbolic polygon with m vertices. Algorithm* SymbolicDecomposition *decomposes P into symbolic rectangles in $O(\log m)$ time on a CREW PRAM with $m/\log m$ processors.*

Sketch of Proof: Steps 1 and 3 can be done using optimal list ranking and prefix computations [1,3]. Step 2 takes constant time. In Step 4, we use the subroutine described in [2] for finding the "next-larger" of each element of the list $rot(i)$, $i = 0, \cdots, m$. (In a list of numbers, the next-larger of a given element x is the first element following x in the list that is larger than x.) In Step 5 we need to find the "next-equal" of each element in the list. This can be implemented by using a subroutine for finding the "next-smallers" of each element, which is symmetric to the next-larger subroutine. (For each element x, the next-equal of x is adjacent to either x's next-smaller or x's next-larger.) □

We conclude

Theorem 3 *Let G be a planar graph with n vertices and maximum degree 4. Algorithm GraphLayout constructs a layout of G with $O(n)$ bends, $O(n)$ maximum edge length, and $O(n^2)$ area, in time $O(\log n)$ on a CREW PRAM with $n/\log n$ processors. Also, the number of bends is at most $2n + 4$ if G is biconnected, and is at most $2.4n + 2$ if G is simple.*

Current work is aimed at making the algorithm run equally efficiently on an EREW PRAM.

References

[1] R.J. Anderson and G.L. Miller, "Deterministic Parallel List Ranking," *VLSI Algorithms and Architectures (Proc. AWOC '88, Corfu, Greece, 1988)* LNCS 319 (1988), 81–90.

[2] O. Berkman, D. Breslauer, Z. Galil, B. Schieber, and U. Vishkin, "Highly Parallelizable Problems," *Proc. 21st ACM Symp. on Theory of Computing* (1989), 309–319.

[3] R. Cole and U. Vishkin, "Approximate and Exact Parallel Scheduling with Applications to List, Tree, and Graph Problems," *Proc. 27th IEEE Symp. on Foundations of Computer Science* (1986), 478–491.

[4] D. Dolev, F.T. Leighton, and H. Trickey, "Planar Embedding of Planar Graphs," in *Advances in Computing Research, vol. 2*, F.P. Preparata, ed., JAI Press Inc., Greenwich, CT, 1984, 147–161.

[5] P. Eades and R. Tamassia, "Algorithms for Automatic Graph Drawing: An Annotated Bibliography," Dept. of Computer Science, Brown Univ., Technical Report CS-89-09, 1989.

[6] P. Eades and R. Tamassia, "Algorithms for Automatic Graph Drawing: An Annotated Bibliography," *Networks* (to appear).

[7] H. Gazit, "Optimal EREW Parallel Algorithms for Connectivity, Ear Decomposition, and *st*-Numbering of Planar Graphs," Department of Computer Science, Duke Univ., Manuscript,, 1990.

[8] F. Hoffmann and K. Kriegel, "Embedding Rectilinear Graphs in Linear Time," *Information Processing Letters* 29 (1988), 75–79.

[9] J.A. Storer, "On Minimal Node-Cost Planar Embeddings," *Networks* 14 (1984), 181–212.

[10] R. Tamassia, "On Embedding a Graph in the Grid with the Minimum Number of Bends," *SIAM J. Computing* 16 (1987), 421–444.

[11] R. Tamassia, "Planar Orthogonal Drawings of Graphs," *Proc. IEEE Int. Symp. on Circuits and Systems* (1990), 319–322.

[12] R. Tamassia and I.G. Tollis, "Planar Grid Embedding in Linear Time," *IEEE Trans. on Circuits and Systems* CAS-36 (1989), 1230–1234.

[13] R. Tamassia and J.S. Vitter, "Parallel Transitive Closure and Point Location in Planar Structures," *SIAM J. Computing* 20 (1991), 708–725.

[14] R.E. Tarjan, "Amortized Computational Complexity," *SIAM J. Algebraic Discrete Methods* 6 (1985), 306–318.

[15] G.K. Vijayan and A. Wigderson, "Rectilinear Graphs and their Embeddings," *SIAM J. on Computing* 14 (1985), 335–372.

Captions of Figures

Figure 1: A planar orthogonal drawing with 8 bends.

Figure 2: An elementary transformation, defined by the closed curve C on left. The resulting layout has two fewer bends.

Figure 3: The optimal layout Γ_n of G_n has a total of $2n + 4$ bends. The potential Φ of each face, which is used in the proof of Lemma 1, is labeled.

Figure 4: (a) Optimal layout requiring $8(n-2)/3$ bends. (b) Optimal layout requiring $4(n-2)$ bends.

Figure 5: The optimal layout Σ_n of S_n has $(n-6)/9+4$ bends, all on a single edge e that spirals. Shown here is the case $n = 78$; the optimal layout has 12 bends.

Figure 6: Elementary transformation T that reduces the number of bends on an edge that has reflex angles on both sides.

Figure 7: (a) A biconnected graph G. (b) Visibility representation for G. (c) Orthogonal embedding obtained from (b) by local substitutions. (d) Orthogonal representation of the embedding in (c) obtained after the bend-reducing transformations. (e) Final layout obtained by compaction of the constraints obtained from (d).

Figure 8: The rectangles formed from the polygon P by *SymbolicDecomposition*.

Figure 9: The cuts at edges e_i and e_j cannot intersect.

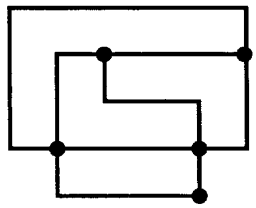

Figure 1

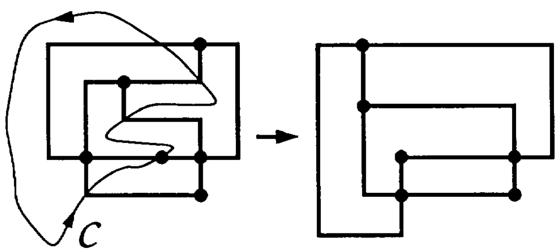

Figure 2

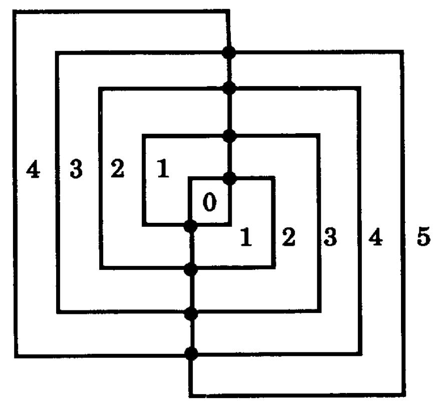

Figure 3

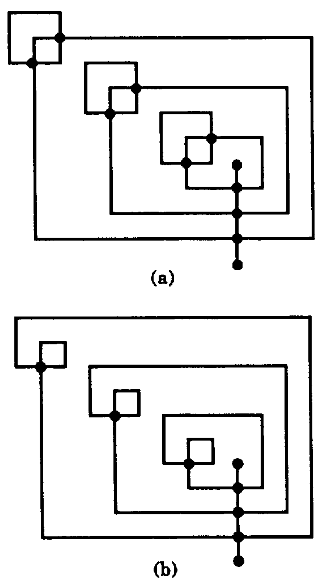

Figure 4

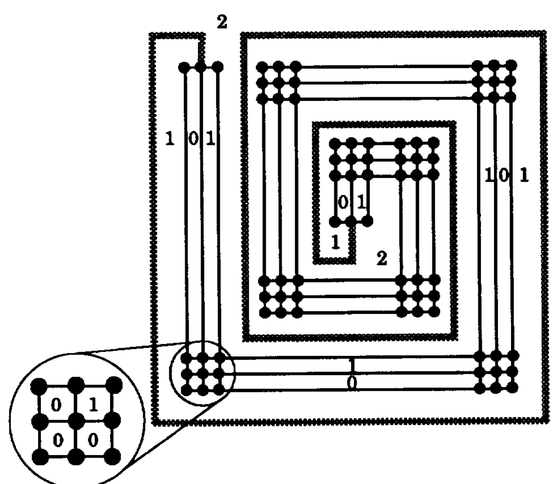

Figure 5

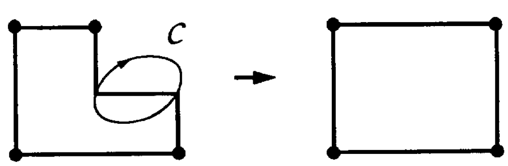

Figure 6

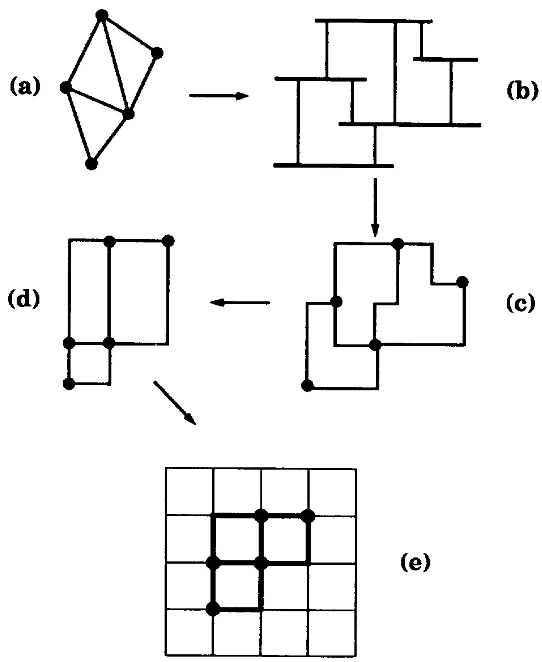

Figure 7

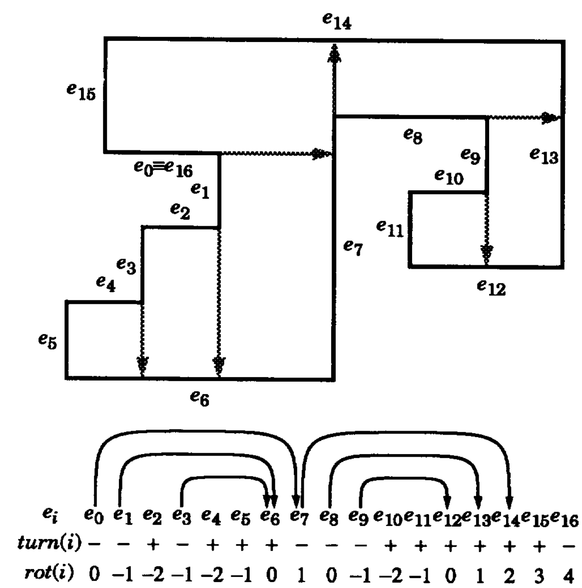

Figure 8

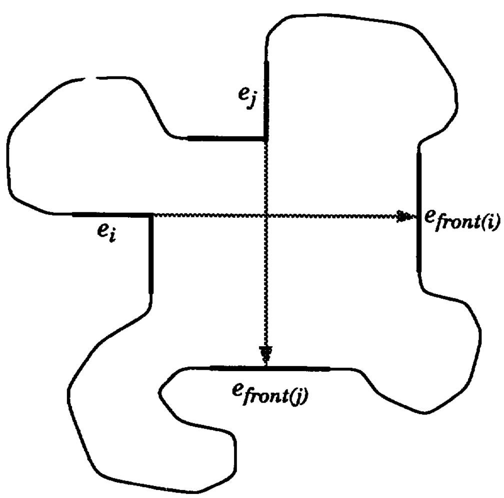

Figure 9