Optimal orthogonal drawings of connected plane graphs *

Therese C. Biedl †

Extended Abstract

1 Introduction and Definitions

A (2-dimensional) orthogonal drawing of a graph is a drawing such that every vertex is drawn as a point in the plane, and every edge is drawn as a sequence of horizontal and vertical lines. Orthogonal graph drawings are an important layout tool, for example to display Data Flow Diagrams and Entity Relationships Diagrams. Two of the most important measurements of the quality of a drawing are the grid-size and the number of bends. Orthogonal drawings exist only if every vertex in the graph has at most four incident edges, such a graph is called a *4-graph*. On the other hand, every planar 4-graph with fixed planar drawing has an orthogonal drawing exactly reflecting the drawing. For example, the algorithm of Tamassia computes such a drawing with the minimum number of bends [7].

In this paper, we assume that G is a given 4-graph with fixed planar drawing (i.e. it is *plane*). Let it have $n = n(G)$ vertices and $m = m(G)$ edges. The algorithm by Tamassia takes $\mathcal{O}(n^2 \log n)$ time. Sometimes heuristics are preferred that work quickly (i.e. in linear time) and produce reasonable bounds. Such heuristics are know both for biconnected graphs [6, 8, 9] and triconnected graphs [5, 2].

For connected graphs, no such heuristic was known that exactly reflects the planar drawing. That is, the algorithms for biconnected graphs presented in [6, 8, 9] can be extended to work for connected graphs, but at the cost of exactly reflecting the drawing. This can be seen since both algorithms produce an $(n+1) \times (n+1)$-grid for connected graphs. But there exists a connected plane graph that needs a width and height of $\frac{6}{5}n - \frac{11}{5}$ in any orthogonal drawing that exactly reflects the planar drawing [3].

In this paper, we will present a linear-time heuristic to embed any connected plane graph while reflecting the planar drawing. To that matter, we present a scheme how a heuristic for biconnected graphs can be used to obtain a heuristic for connected graphs. In our case, we use the heuristic of Biedl and Kant [4]. We analyze the produced grid-size for simple graphs, for graph without loops (*multi-graphs*), and for graphs with loops. For all these graph classes, the upper bounds achieved with our algorithm are optimal up to a constant, since there exist graphs which need this grid-size and this number of bends in any orthogonal drawing that reflects the embedding.

The algorithm depends heavily on how a graph is split into its biconnected components (or blocks). A vertex v is called a *cutvertex* if v is incident to a loop, or if removing v from G gives two disconnected subgraphs. A graph is *biconnected* if it is not a loop, and if it has no cutvertex. A *block* is a loop or a maximal biconnected subgraph. With the blocks we associate a graph $\mathcal{T}$, the *blocktree*. $\mathcal{T}$ has a vertex for every block, and a vertex for every cutvertex. Two vertices of $\mathcal{T}$ are adjacent if and only if one is a cutvertex that belongs to the block represented by the other. Let $\mathcal{T}$ be rooted at a block B_r. For a block $B_1 \neq B_r$ then define the *parent-vertex* ($parent(B_1)$) to be the cutvertex in B_1 that is closest to B_r in $\mathcal{T}$.

*Some of these results were part of a diploma thesis written at TU Berlin.
A full version of this paper has appeared in [1].

†RUTCOR, P.O. Box 5062, New Brunswick, NJ 08903, therese@rutcor.rutgers.edu, FAX (908) 445-5472

2 Drawing graphs that are not biconnected

Assume G is a 4-graph that is connected, but not biconnected. The idea for drawing G is to split it into its blocks, to embed them separately, and to merge them into one drawing. This "merging scheme" has been used in [4] without being given abstractly. It can be used for any orthogonal graph drawing heuristic BICONN that fulfills the (very weak) Condition 1. To get drawings that reflect a planar drawing, we also need Condition 2.

Let Γ be a drawing of G. Let v be a vertex of G with $deg(v) \leq 3$, drawn at $(c(v), r(v))$. We say that v is drawn as *final vertex* in Γ, if the grid-points in S are unused, except for $(c(v), r(v))$, where S is defined as follows: (1) if $deg(v) = 1$, then $S = \{(c,r),\ r \geq r(v)\}$, (2) if $deg(v) = 2$, then $S = \{(c,r),\ c \geq c(v),\ r \geq r(v)\}$, (3) if $deg(v) = 3$, then $S = \{(c,r),\ c = c(v),\ r \geq r(v)\}$.

Condition 1 *BICONN gets as input a biconnected graph G and a vertex v with* $\deg(v) \leq 3$. *It outputs an orthogonal drawing such that v is drawn as final vertex.*

Let for a vertex in a drawing the *ports* be the adjacent grid-segments. We say a vertex is drawn *straight* if it has degree 2, and if the two used ports are on opposite sides.

Condition 2 *BICONN gets as input a biconnected plane graph G and some vertices $w_1, \ldots, w_p$ of degree 2. It outputs a plane drawing such that w_j is drawn straight, $j = 1, \ldots, p$.*

We choose as the root of the blocktree $\mathcal{T}$ a block on the outer-face of the planar drawing of G. For any subtree $\mathcal{T}'$ of $\mathcal{T}$, denote by $G(\mathcal{T}')$ the graph represented by $\mathcal{T}'$. The following lemma states the invariant that we uphold during the merging scheme, and its proof contains the algorithm.

Lemma 2.1 *Let $\mathcal{T}'$ be a subtree of $\mathcal{T}$, and let B'_r be the block in $\mathcal{T}'$ closest to the root. We can draw $G' = G(\mathcal{T}')$ such that* $\mathrm{parent}(B'_r)$, *if it exists, is drawn as final vertex.*

Proof: We prove the claim by induction on the number of blocks in $\mathcal{T}'$, and assume that $v = parent(B'_r)$ is defined. In the base case there is only one block. So G' either is biconnected, then we are done by Condition 1. Or G' is a loop, and v is its only vertex. Then this loop can be embedded such that v is drawn as final vertex. The inductive step falls into two cases:
Case 1: B'_r contains a cutvertex v_1 that is incident to a bridge:
A *bridge* is an edge whose removal splits G into two graphs. Removing the corresponding block from $\mathcal{T}'$, we get two smaller subtrees, call them $\mathcal{T}_1$ and $\mathcal{T}_2$. We assume $B'_r \in \mathcal{T}_1$ and apply induction on $\mathcal{T}_1$ and $\mathcal{T}_2$. Then the graph $G_1 = G(\mathcal{T}_1)$ is embedded with v as final vertex, while the graph $G_2 = G(\mathcal{T}_2)$ is embedded with the other endpoint of the bridge, v_2, as final vertex.

In the drawing of G_1, there is one port free at v_1, assume it is the left one. Assume first that adding the subgraph at this port reflects the planar drawing. By adding rows and columns to the left of the drawing of v_1 and rotating the drawing of G_2, we can merge it into the drawing. This is possible since v_2 was drawn as final vertex in G_2 and $deg_{G_2}(v_2) \leq 3$. If adding the subgraph at the free port contradicts the planar drawing of the graph, then we add a column and a bend to the drawing, and cut it, so that an appropriate port is free. See also Figure 1.

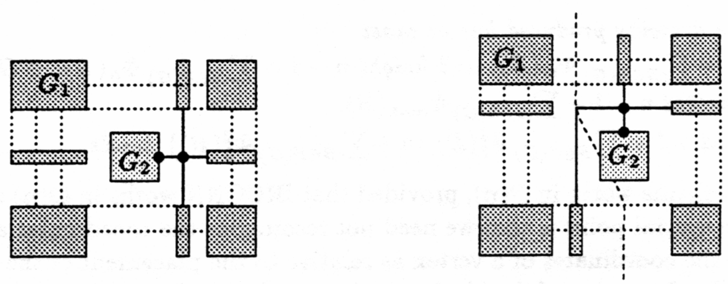

Figure 1: How to merge a subgraph connected by a bridge. In the second case the port could not be used, and we add a extra column (shown as dashed line), and a bend.

Case 2: Otherwise:
Since we are not in the base case, there are at least two blocks in $\mathcal{T}'$. So there must be at least one cutvertex v_1 in B_r'. We must have $v_1 \neq v$, otherwise we had a bridge incident to v. Removing v_1 from $\mathcal{T}'$, we get two smaller subtrees, call them $\mathcal{T}_1$ and $\mathcal{T}_2$.

We assume $B_r' \in \mathcal{T}_1$ and apply induction on $\mathcal{T}_1$ and $\mathcal{T}_2$. Then the graph $G_1 = G(\mathcal{T}_1)$ is embedded with v as final vertex, while the graph $G_2 = G(\mathcal{T}_2)$ is embedded with v_1 as final vertex. Also, we prespecify that v_1 must be drawn straight in the drawing of G_1. By cutting the drawing and adding one column and a bend, we achieve that v_1 is drawn with right angle. After doing this we merge G_2 by adding columns and rows. We have two possible placements of v_1, and depending on this we add G_2 "to the right" or "to the left" of v_1. We do this as dictated by the planar drawing. See also Figure 2. □

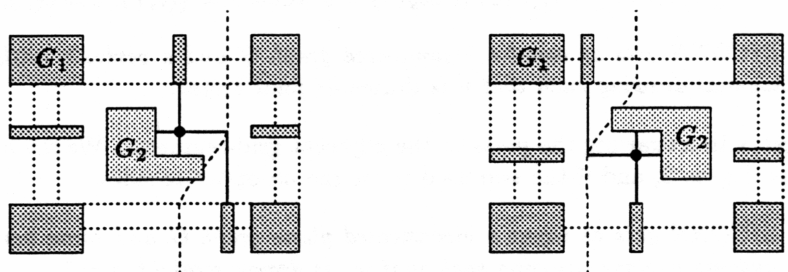

Figure 2: To merge at a cutvertex that is drawn straight, we add a column to draw it with right angle. In exchange, we get the choice whether to add the graph to the right or to the left.

One can verify that the planar drawing is exactly reflected with this algorithm. The grid-size and number of bends depend on the grid-size and number of bends of BICONN. We now give a "formula" for computing this relationship.

A *critical cutvertex* of a block B is a cutvertex of G in B that is not *parent*(B), and that is not incident to a bridge. In other words, a critical cutvertex is a cutvertex v_1 encountered in the second case of the proof of Lemma 2.1. The number of critical cutvertices in B is denoted $cr(B)$. Call a drawing of a block B a *proper drawing*, if no increase is necessary to merge the subgraphs. So *parent*(B) is drawn as final vertex, and all critical cutvertices have added the columns already, so they are now drawn with a right angle, and in such a way that the planar drawing is reflecting after merging the subgraph.

Definition 1 *Let B be a block of G that is not a bridge. The* correction terms Φ_w, Φ_h, Φ_b^1, *and Φ_b^2 are functions such that B has a proper drawing of width $n(B) - 1 + \Phi_w(B)$, and height $n(B) - 1 + \Phi_h(B)$.*

Furthermore, the number of bends is either of the following two terms: $2m(B) - 2n(B) + 1 + \mathrm{cr}(B) + \Phi_b^1(B)$, or $m(B) + \Phi_b^2(B)$. Finally, define $\Phi_{\max}(B) = \max\{\Phi_h(B), \Phi_w(B)\}$.

The correction terms are defined only for the *proper blocks*, i.e. those blocks that are not bridges. Denote this set by $\mathcal{B}(G)$. By induction on the steps of the algorithm, one can prove the following lemma.

Lemma 2.2 *The drawing produced has at most*

- *width $n - 1 + \sum_{B\in\mathcal{B}(G)} \Phi_w(B)$, and height $n - 1 + \sum_{B\in\mathcal{B}(G)} \Phi_h(B)$, if G has no bridge,*
- *width and height $n - 1 + \sum_{B\in\mathcal{B}(G)} \Phi_{\max}(B)$,*
- $\min\{2m - 2n + 1 + \sum_{B\in\mathcal{B}(G)} \Phi_b^1(B), m + \sum_{B\in\mathcal{B}(G)} \Phi_b^2(B)\}$ *bends.*

This merging scheme works in $\mathcal{O}(n)$, provided that BICONN works in $\mathcal{O}(n)$ and produces $\mathcal{O}(n)$ many bends. The crucial point is that we need not recompute the coordinates after each merging. Instead, we store the coordinates of a vertex as relative to the placement of the parent-vertex. We also store the angle of rotation of the block, or whether the block has been flipped. Thus, we need to recompute the coordinates only once, after all mergings are done.

3 The merging scheme applied

We now analyze the sizes obtained by combining the merging scheme with the algorithm by Biedl and Kant [4]. This algorithm works for biconnected graphs in linear time. We have no space to present this algorithm and its properties here. Using it, we get the following correction terms.

Lemma 3.1 *Let B be a proper block, where* parent(B) *is defined. Then* $\Phi_h(B) \leq 1$, $\Phi_w(B) \leq 1$, $\Phi_b^1(B) \leq 2$, *and* $\Phi_b^2(B) \leq 2$.

Lemma 3.2 *For the root of $\mathcal{T}$, we have* $\Phi_w(B_r) \leq 2$, $\Phi_h(B_r) \leq 2$, $\Phi_b^1(B_r) \leq 3$, *and* $\Phi_b^2(B_r) \leq 4$.

We can now get an estimation of the grid-size and the number of bends with Lemma 2.2. Note that we do not use Φ_b^2 here, this will be useful only for simple graphs.

Lemma 3.3 *Let G be a plane graph with bl proper blocks. Then G can be embedded in an $(n + bl) \times (n + bl)$-grid with at most $2m - 2n + 2 + 2bl$ bends.*

One can show that every 4-graph has at most $n + 1$ proper blocks (the extreme case being a graph that consists of an n-cycle with n loops attached). If a graph has no loops, then it can have at most $n - 1$ blocks (this is shown easily by induction on the number of blocks). Thus we get

Theorem 1 *Let G be a plane 4-graph that is not biconnected. Then in linear time, it can be embedded in a $(2n + 1) \times (2n + 1)$-grid with $2m + 4$ bends. If G is a multi-graph, then in linear time it can be embedded in a $(2n - 1) \times (2n - 1)$-grid with $2m$ bends.*

For simple graphs, Biedl and Kant give an improvement of their algorithm [4]. Unfortunately, with this change the algorithm does not fulfill Condition 2, and we apply it only to blocks without critical cutvertices. The correction terms can be reduced further for simple blocks with at most 5 vertices. However, the reduction depends on the number of vertices and the number of critical cutvertices. With a detailed case analysis we get the bounds shown in Table 1 for a simple block.

	$n(B) \leq 4$			$n(B) = 5$ $deg(v) = 2$			$n(B) = 5$ $deg(v) = 3$			$n(B) \geq 6$		
$cr(B)$	$= 0$	$= 1$	≥ 2	$= 0$	$= 1$	≥ 2	$= 0$	$= 1$	≥ 2	$= 0$	$= 1$	≥ 2
$\Phi_w(B) \leq$	-1	0	1	-1	0	1	0	1	1	0	1	1
$\Phi_h(B) \leq$	-1	0	0	0	0	0	0	0	0	0	1	1
$\Phi_b^1(B) \leq$	0	1	2	1	2	2	1	2	2	1	2	2
$\Phi_b^2(B) \leq$	-2	0	2	-1	1	2	0	2	2	0	2	2

Table 1: Correction terms for a simple block B where $v = parent(B)$ is defined.

We distinguish these different kinds of blocks as follows. $\mathcal{B}_i^j$ are the blocks, where i is the bound for the critical cutvertices ($i = 0, 1, 2$), and j is the bound for the number of vertices ($j = 4, 5, 6$). Furthermore, we distinguish $\mathcal{B}_i^{5,2}$ and $\mathcal{B}_i^{5,3}$ by the degree of the parent-vertex. b_i^j is the cardinality of $\mathcal{B}_i^j$. If a subscript/superscript is missing, we count over all possible subscripts/superscripts.

Lemma 3.4 *Assume G has a vertex v with* $\deg(v) \leq 3$ *on the outer-face. Then (1) $b_2 \leq b_0 - 1$, (2) $2b^4 + 4b^5 + 5b^6 \leq n - 1$, (3) if G has no bridge, then $b^{5,3} \leq 1$.*

Proof: The first claim is shown by counting the number of edges in the blocktree in two different ways. For the second claim, we assign the vertices that are not the parent-vertex to each block. This assigns the appropriate number of vertices to each block, counts every vertex at most once, and does not count the parent-vertex of the root. The third claim holds since two blocks in $\mathcal{B}^{5,3}$ imply that one of them is incident to a bridge. □

Lemma 3.5 *Assume G has no bridge. If a vertex of degree ≤ 3 is on the outer-face, then*

- $\sum_{B\in\mathcal{B}(G)} \Phi_w(B) \leq \frac{n}{5} - 1$, $\sum_{B\in\mathcal{B}(G)} \Phi_h(B) \leq \frac{n}{5} - 1$,
- $\sum_{B\in\mathcal{B}(G)} \Phi_b^1(B) \leq \frac{n}{2} - \frac{3}{2}$, *and* $\sum_{B\in\mathcal{B}(G)} \Phi_b^2(B) \leq \frac{2}{5}n - \frac{7}{5}$.

Proof: For space reasons, we show only the argument for Φ_b^2. By Table 1, we have $\sum \Phi_b^2 \leq -2b_0^4 - b_0^{5,2} + b_1^{5,2} + 2b_1^{5,3} + 2b_1^6 + 2b_2 \leq b_0^{5,2} + 2b_0^{5,3} + 2b_0^6 - 2 + b_1^{5,2} + 2b_1^{5,3} + 2b_1^6 \leq b^5 + b^{5,3} + 2b^6 - 2 \leq b^5 + 2b^6 - 1 \leq \frac{2}{5}(n-1) - 1$, by the claims of Lemma 3.4. □

If every vertex on the outer-face has degree 4, then these estimations have to be increased somewhat. By applying Lemma 2.2, and by induction on the number of bridges, one can show the following result.

Theorem 2 *Let G be a simple plane 4-graph. Then in linear time, it can be embedded in a grid of width and height $\frac{6}{5}n$, and with $\min\{2m - \frac{3}{2}n + \frac{3}{2}, m + \frac{2}{5}n + 2\}$ bends.*

If a vertex of degree ≤ 3 is on the outer-face, G can be embedded in a grid of width and height $\frac{6}{5}n - 2$ with $\min\{2m - \frac{3}{2}n - \frac{1}{2}, m + \frac{2}{5}n - \frac{7}{5}\}$ bends.

4 Remarks

In this paper, we presented a heuristic that gives an orthogonal drawing of any connected plane graph, while exactly reflecting the given planar drawing. No linear-time heuristic was known before. We produce at most a $\frac{6}{5}n \times \frac{6}{5}n$-grid. The number of bends is bound by $m + \frac{2}{5}n + 2$, which is at most $\frac{12}{5}n + 2$. It is also bound by $2m - \frac{3}{2}(n-1)$, which is smaller if $m \leq \frac{19}{10}n$.

We also presented a scheme of how an algorithm for biconnected graphs gives an algorithm that draws connected graphs. The algorithm must fulfill two conditions; however, the second condition can be dropped if it is not required that the resulting drawing reflects the planar embedding. In fact, this merging scheme does not use planarity at all, and can thus be also applied for heuristics for non-planar graphs. We got a formula how one can compute the grid-sizes and bends, by considering the grid-size and the bends achieved with the algorithm for biconnected graphs. This scheme should be useful to apply to other heuristics as well.

We have the following remarks:

- We are very close to optimality in all cases (the lower bounds are from [3]).

  For simple graphs, the graph shown in Figure 3.a has $2n - 2$ edges and needs a grid of width and height $\frac{6}{5}n - \frac{11}{5}$ and $\frac{12}{5}n - \frac{17}{5}$ bends in any drawing exactly reflecting the plane drawing.

  For multi-graphs, there exists a multi-graph with $2n$ edges which needs a $(2n-3) \times (2n-3)$-grid and $4n - 4$ bends in any drawing exactly reflecting the plane drawing.

  For graphs with loops, there exists a graph with $2n$ edges which needs a $(2n+1) \times (2n+1)$-grid and $4n + 4$ bends in any drawing exactly reflecting the plane drawing.

- For simple graphs, we bounded the number of bends by $2m - \frac{3}{2}n + \frac{1}{2}$. This is almost optimal, since the graph C_i shown in Figure 3.b needs $2m - \frac{3}{2}n - \frac{1}{2}$ bends. Namely, let C_1 be a triangle, and let C_i result from C_{i-1} by adding two new vertices, and a triangle that encloses C_{i-1}. Then $n(C_i) = 2i + 1$, and $m(C_i) = 3i + 1$. One can show that C_i needs a width and height of $2i - 1$, and $3i - 2$ bends, in any drawing that exactly reflects the embedding.

- The bound of $\min\{2m - \frac{3}{2}n + \frac{1}{2}, m + \frac{2}{5}n + 2\}$ bends can be used to show bounds for the grid-size of Tamassia's algorithm [7]. With the tools presented in [2], one can show that the width and height is at most $\min\{\frac{5}{4}n - \frac{1}{2}, \frac{11}{5}n - \frac{m}{2}\}$ for connected plane graphs. The half-perimeter is at most $\min\{\frac{n}{2} + m - \frac{1}{2}, \frac{12}{5}n\}$.

- One open problem concerns plane graphs of maximum degree 3. With the scheme of Kant [5] we can obtain a $\frac{n}{2} \times \frac{n}{2}$-grid with $\frac{n}{2} + 1$ bends. But this algorithm does not reflect the planar drawing, as can be seen since the graph of Figure 3.c needs a bigger grid in any plane drawing. What algorithms exists for such graphs?

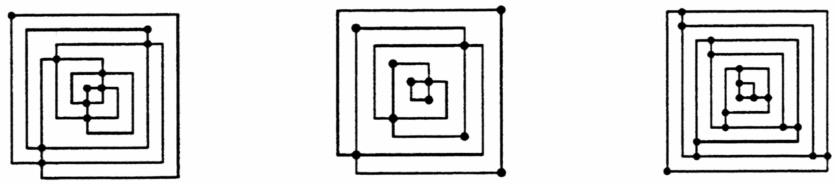

Figure 3: Three graphs for lower bounds for plane drawings. The first graph needs a width and height of $\frac{6}{5}n - \frac{11}{5}$, and $\frac{12}{5}n - \frac{17}{5}$ bends. The second graph needs $2m - \frac{3}{2}n - \frac{1}{2}$ bends. The third graph has maximum degree 3 and needs a width and height of $\frac{2}{3}n - 1$, and $\frac{5}{6}n - 1$ bends.

References

[1] T. Biedl, Optimal Orthogonal Drawings of Connected Plane Graphs, *Rutcor Research Report* 10-96.[1]

[2] T. Biedl, Optimal Orthogonal Drawings of Triconnected Plane Graphs, *Rutcor Research Report* 8-96.[1]

[3] T. Biedl, New Lower Bounds for Orthogonal Graph Drawing, *Proc. Graph Drawing '95, Lecture Notes in Comp. Science 1027*, Springer-Verlag (1995), pp. 28-39. Also: *Rutcor Research Report* 19-95.[1]

[4] T. Biedl, and G. Kant, A better heuristic for orthogonal graph drawings, *Proc. of the 2nd European Symp. on Algorithms (ESA 94), Lecture Notes in Comp. Science* 855, Springer-Verlag (1994), pp. 124–135. Also: *Rutcor Research Report* 12-95.[1]

[5] G. Kant, Drawing planar graphs using the *lmc*-ordering, Extended Abstract in: *Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science* 1992, pp. 101–110. Extended and revised version to appear in *Algorithmica, special issue on Graph Drawing.*

[6] J. Storer, On minimal node-cost planar embeddings, *Networks* 14 (1984), 181–212.

[7] R. Tamassia, On embedding a graph in the grid with the minimum number of bends, *SIAM J. Comp.* 16 (1987), pp. 421–444.

[8] R. Tamassia, and I.G. Tollis, Efficient embedding of planar graphs in linear time, *Proc. IEEE Int. Symp. on Circuits and Systems* (1987), pp. 495–498.

[9] R. Tamassia, and I.G. Tollis, Planar grid embedding in linear time, *IEEE Trans. Circ. Syst.* 36 (9), 1989, pp. 1230–1234.

[1] Rutcor Research Reports are available via anonymous FTP from *rutcor.rutgers.edu*, directory */pub/rrr*; or on the WWW at *http://rutcor.rutgers.edu/~rrr*.

For a comprised version of these four papers, see T. Biedl, *Orthogonal Graph Drawings, Algorithms and Lower Bounds*, Diploma Thesis TU Berlin, December 1996. Available on the WWW at *http://rutcor.rutgers.edu/~therese*.