

- [21] M. Schöffter, Drawing Graphs on Rectangular Grids with at most 2 Bends per Edge, *Discr. Appl. Math.* (to appear), preliminary version available via WWW from <http://winnie.math.tu-berlin.de/~shefta>.
- [22] J. Storer, On minimal node-cost planar embeddings, *Networks* 14 (1984), pp. 181–212.
- [23] R. Tamassia, On embedding a graph in the grid with the minimum number of bends, *SIAM J. Comput.* 16 (1987), pp. 421-444.
- [24] R. Tamassia, and I.G. Tollis, Efficient embedding of planar graphs in linear time, *Proc. IEEE Int. Symp. on Circuits and Systems*, Philadelphia, pp. 495–498, 1987.
- [25] R. Tamassia, I.G. Tollis, and J.S. Vitter, Lower bounds for planar orthogonal drawings of graphs, *Inf. Proc. Letters* 39 (1991), pp. 35–40.
- [26] L.G. Valiant, On non-linear lower bounds in computational complexity, *Proc. 7th Symp. on Theory of Computing*, 1975, pp. 45–53.

- [5] P.F. Dietz, and D.D. Sleator, Two algorithms for maintaining order in a list, in: *Proc. 19th Annual ACM Symp. Theory of Computing*, 1987, pp. 365–372.
- [6] S. Even, and G. Granot, *Rectilinear Planar Drawings with Few Bends in Each Edge*, Manuscript, Faculty of Comp. Science, the Technion, Haifa (Israel), 1993.
- [7] S. Even, and R.E. Tarjan, Computing an *st*-numbering, *Theoretical Comp. Science* 2 (1976), pp. 436–441.
- [8] M. Formann, and F. Wagner, The VLSI layout problem in various embedding models, *Graph-Theoretic Concepts in Comp. Science (16th Workshop WG'90)*, Springer-Verlag, Berlin/Heidelberg, 1992, pp. 130–139.
- [9] A. Garg, and R. Tamassia, On the Computational Complexity of Upward and Rectilinear Planarity Testing, *Proc. Graph Drawing '94, Lecture Notes in Comp. Science* 894, Springer-Verlag, 1995, pp. 286–297.
- [10] G. Kant, Drawing planar graphs using the *lmc*-ordering, Extended Abstract in: *Proc. 33th Ann. IEEE Symp. on Found. of Comp. Science*, Pittsburgh, 1992, pp. 101-110. Extended and revised version to appear in *Algorithmica, special issue on Graph Drawing*.
- [11] M.R. Kramer, and J. van Leeuwen, The complexity of wire routing and finding minimum area layouts for arbitrary VLSI circuits, *Advances in Computer Research, Vol. 2: VLSI Theory, F.P. Preparata (Ed.)*, JAI Press, Reading, MA, 1992, pp. 129–146.
- [12] Th. Lengauer, *Combinatorial Algorithms for Integrated Circuit Layout*, Teubner/Wiley & Sons, Stuttgart/Chichester, 1990.
- [13] F.T. Leighton, New Lower Bounds Techniques for VLSI, *Proc. 22nd Ann. IEEE Symp. on Found. of Comp. Science* 1981, pp. 1-12
- [14] C.E. Leiserson, Area-efficient graph layouts (for VLSI-Design), *Proc. 21st Ann. IEEE Symp. on Found. of Comp. Science* 1980, pp. 270-281
- [15] A. Lempel, S. Even, and I. Cederbaum, An algorithm for planarity testing of graphs, *Theory of Graphs, Int. Symp. Rome* (1966), pp. 215–232.
- [16] Y. Liu, P. Marchioro, and R. Petreschi, At most single-bend embeddings of cubic graphs, *Applied Mathematics*, 1994, to appear.
- [17] Y. Liu, P. Marchioro, R. Petreschi, and B. Simeone, *Theoretical Results on at most 1-bend embeddability of Graphs*, Tech. Report, Dept. of Statistics, University “La Sapienza” Roma, 1990.
- [18] Y. Liu, A. Morgana, and B. Simeone, *A Linear Algorithm for 3-Bend Embeddings of Planar Graphs in the Grid*, Manuscript, University “La Sapienza”, Rome, 1994
- [19] A. Papakostas, and I.G. Tollis, Improved algorithms and bounds for orthogonal drawings, *Proc. Graph Drawing '94, Lecture Notes in Comp. Science* 894, Springer-Verlag, 1995, pp. 40–51. A revised version is in progress.
- [20] P. Rosenstiehl, and R.E. Tarjan, Rectilinear planar layouts and bipolar orientations of planar graphs, *Discr. and Comp. Geometry* 1 (1986), pp. 343–353.

We end this paper by mentioning some open problems and directions for further research:

- Valiant proved that a grid-area of  $\Omega(n^2)$  is necessary for drawing non-planar graphs, since the crossing number can be  $\Omega(n^2)$  [26]. However, the involved constants are very small. For planar graphs (crossing number 0) a drawing with grid-area  $\mathcal{O}(n \log^2 n)$  is possible [14, 26], and the best known lower bound is  $\Omega(n \log n)$  [13]. Are there lower bounds with bigger constants, and can a trade-off between grid-area and crossing number be made?
- Lower bounds on the number of bends, as presented in [25] for instance, are not transferable to non-planar drawings. We managed to find a class of simple biconnected graphs which need at least  $\frac{10}{6}n$  bends in any orthogonal drawing [1]. Is there something better?
- Storer [22] gave embedded 4-planar graphs requiring an  $(n-2) \times (n-2)$  grid. Tamassia, Tollis & Vitter [25] proved that there exist embedded 4-planar graphs requiring  $2n-2$  bends, and their graph in fact requires an  $(n-1) \times (n-1)$ -grid. However, when changing the embedding, a lower number of bends and a smaller grid can be obtained. The graph of [25] requires an  $(n-3) \times (\frac{n}{2}-1)$ -grid and  $2n-4$  bends in any embedding, but it is unclear what the precise lower bounds are for orthogonal drawings of non-embedded 4-planar graphs.
- Very recently, Papakostas & Tollis [19] presented an algorithm for drawing non-planar connected 4-graphs with  $1.9n$  bends on a grid of size  $0.8n^2$ , hence an improvement of our results. However, the proof of the crucial Lemma 3 seems to be wrong, as shown in [1]. The natural question is to prove this theorem, and to improve these bounds.

## References

- [1] T. Biedl, *Orthogonal Graph Drawings*, Diplomarbeit, Fachbereich Mathematik, Technische Univ. Berlin, 1995, to appear, Preliminary version also available via WWW from <http://new-rutcor.rutgers.edu/~therese>
- [2] N. Chiba, T. Nishizeki, S. Abe, and T. Ozawa, A linear algorithm for embedding planar graphs using PQ-trees, *J. of Comp. and System Sciences* 30 (1985), pp. 54–76.
- [3] G. Di Battista, P. Eades, R. Tamassia, and I.G. Tollis, Algorithms for Automatic Graph Drawing: An Annotated Bibliography, to appear in *Comp. Geom.: Theory and Applications*, Preliminary version also available via anonymous ftp from [wilma.cs.brown.edu](http://wilma.cs.brown.edu) (128.148.33.66), files `/pub/gdbiblio.tex.Z` and `/pub/gdbiblio.ps.Z`.
- [4] G. Di Battista, G. Liotta, and F. Vargiu, Spirality of orthogonal representations and optimal drawings of series-parallel graphs and 3-planar graphs, *Proc. Workshop on Algorithms and Data Structures*, Lecture Notes in Comp. Science 709, Springer-Verlag, 1993, pp. 151–162.

## 7 Conclusion

In this paper we have considered the problem of orthogonal drawings of graphs. A general linear time algorithm has been presented to construct an orthogonal representation of a connected graph. The results are summarized in the following table, where all results not proved here can be found in [1]. Our results are a big improvement for non-planar graphs and for non-biconnected planar graphs.

Gridsize & Bends	Non-planar	Planar
Simple		
Biconnected		
4-regular	$n \times n$ $2n + 2$	$n \times n$ $2n + 2$
$m = 2n - 1$	$(n - 1) \times (n - 1)$ $2n - 1$	$(n - 1) \times (n - 1)$ $2n - 1$
$m \leq 2n - 2$	$(n - 2) \times (n - 2)$ $m - 2$ $(m - n + 2) \times (n - 2)$ $2m - 2n + 2$	$(n - 1) \times (n - 1)$ $m - 1$ $(m - n + 1) \times (n - 1)$ $2m - 2n + 3$
Not Biconn.		
4-regular	$(n - 1) \times (n - 1)$ $2n$	$(n - 1) \times (n - 1)$ $2n$
$m \leq 2n - 1$	$(n - 1) \times (n - 1)$ $2n - 1$	$(n - 1) \times (n - 1)$ $2n - 1$
No Loops		
Biconnected		
4-regular	$(n + 1) \times (n + 1)$ $2n + 4$	$(n + 1) \times (n + 1)$ $2n + 4$
$m = 2n - 1$	$n \times n$ $2n + 1$	$n \times n$ $2n + 1$
$m \leq 2n - 2$	$(m - n + 1) \times (n - 1)$ $2m - 2n + 2$	$(m - n + 1) \times n$ $2m - 2n + 3$
Not Biconn.	$(\frac{4}{3}n - 1) \times (\frac{4}{3}n - 1)$ $\frac{8}{3}n$	$(\frac{3}{2}n - 1) \times (\frac{3}{2}n - 1)$ $3n$
Non-simple	$(2n - 1) \times (2n - 1)$ $4n$	$(2n - 1) \times (2n - 1)$ $4n$

Table 1: Overview of the achieved bounds.

In our algorithm the 3-bent edge only occurs for the octahedron. Even & Granot [6] proved that this graph indeed requires a 3-bent edge. Hence the octahedron is the only 4-graph which cannot be drawn with at most two bends per edge.

## 6.2 Graphs with higher maximum degree

Of course it is not possible to embed graphs of maximum degree exceeding 4 in an orthogonal fashion. The usual way out of this dilemma is to split the vertices of higher degree into cycles or chains of vertices such that in the new graphs all vertices have maximum degree 4. Assume our graph is biconnected. Find an  $st$ -ordering. When we want to embed a vertex  $v$  with  $\deg(v) > 4$  we split it into a chain of vertices. The chain can be drawn as a vertical or horizontal line. Notice that in the horizontal case we get a visibility representation in the planar case (e.g., see [20]). However, in the non-planar case the horizontal chain can be crossed by many edges. This will not be the case when the chain is drawn vertically, which we will explain now in more detail.

Let  $\{e_1, \dots, e_r\}$  ( $r = bdeg(v)$ ) be the incoming edges of  $v$  in the order of their assigned columns and let  $k = \lceil \frac{r+1}{2} \rceil$ . We draw  $v$  as a straight vertical line which is placed above the column of  $e_k$ . By adding  $k - 1$  rows we can connect all incoming edges of  $v$  with this vertical line.

Let  $\{f_1, \dots, f_s\}$  ( $s = fdeg(v)$ ) be the outgoing edges of  $v$  (in the planar case let them be sorted in clockwise order). If  $r$  was even we can add the outgoing edge to  $f_s$  in the last row and have to add  $\lceil \frac{s-2}{2} \rceil$  more rows for the other outgoing columns. Otherwise we need  $\lceil \frac{s-1}{2} \rceil$  more rows. Doing case analysis one can see that we need exactly  $\lceil \frac{deg(v)-2}{2} \rceil$  rows.

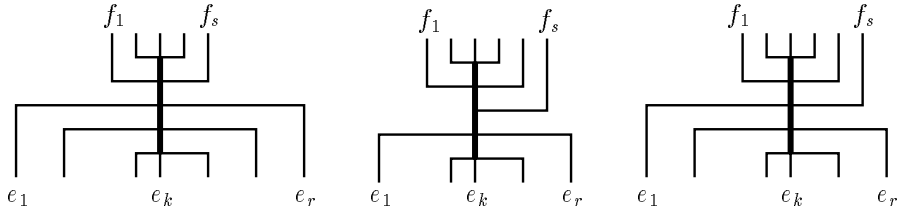


Figure 10: Conversion of a vertex: Cases odd/odd, odd/even, and even/even

What is the obtained gridsize? Every vertex of degree exceeding 4 now needs  $\lceil \frac{deg(v)-2}{2} \rceil \leq \frac{deg(v)-1}{2}$  new rows. This in fact holds for all vertices of degree at least 3. Let  $V_2$  be the set of all vertices of degree 2 and let  $n_2$  be their number. The drawing has at most a height of

$$n_2 + \sum_{v \in V - V_2} \frac{deg(v) - 1}{2} = \sum_{v \in V} \frac{deg(v)}{2} - (n - n_2) \frac{1}{2} = m - \frac{n}{2} + \frac{n_2}{2}.$$

At the very most we may need  $fdeg(v) - 1$  new columns (note that for most vertices we need less). Also every vertex needs  $deg(v) - 2$  bends. Hence the estimation of the width and bends remain valid. In all we have at most an  $(m - n + 1) \times (m - \frac{n}{2} + \frac{n_2}{2})$ -grid and at most  $2m - 2n + 4$  bends.

To compute the final coordinates we traverse the graph in top-down fashion. Assume that  $v = \text{father}(G_0)$  is already in its final position  $(x_v, y_v)$ . For every vertex  $w \neq v$  in  $G_0$  depending on  $\text{rotate}(G_0)$  we set  $x_w = x_v \pm \text{dist}_x(w)$ ,  $y_w = y_v \pm \text{dist}_y(w)$  or  $x_w = x_v \pm \text{dist}_y(w)$ ,  $y_w = y_v \pm \text{dist}_x(w)$  to get the final coordinate of  $w$ . Before we proceed in the subgraphs we add  $\text{rotate}(G_0)$  to the “topmost” block of every  $G_i$ . Doing so we can compute all final coordinates in linear time.

## 6 Extensions

### 6.1 Graphs with multiple edges

If the input-graph has a vertex  $v$  of degree 2 then the following heuristic can be used to get fewer bends: remove  $v$  and add a new edge  $e_v$  between its neighbors. Embed the resulting graph  $G'$ . Now quite likely the drawing of  $e_v$  will have a length of two or more. In this case we can add  $v$  into the drawing without adding rows and columns at all, quite often also reducing the number of bends. If  $e_v$  did have length 1 we can add  $v$  by adding at most one row.

However, this approach can lead to non-simple graphs. Therefore we will now show how to embed general biconnected graphs with multiple edges in an  $(n + 1) \times (n + 1)$ -grid with  $2n + 4$  bends. If  $G$  has a triple edge  $(v, w)$  then we delete  $v$  and  $w$  and add an edge  $e_{vw}$  between the other neighbors  $v'$  of  $v$  and  $w'$  of  $w$ . If  $v' = w'$  then  $G$  is a subdivision of the *quadruple graph*  $Q$  (see Figure 9(a)), which can be embedded in a  $3 \times 3$ -grid with 8 bends. Otherwise we iterate until we either end at  $Q$  or a subdivision of it or at a graph without triple edges. If the resulting graph is now embedded we can add  $v$  and  $w$  into the drawing by adding two rows and two columns next to a vertical segment of the drawing of  $e_{vw}$  (see Figure 9(b)).

The remaining question now is how to embed graphs that have double edges (but no triple edges). This is very easy with the algorithm of Section 3.1: we only need the simplicity of the graph to embed  $v_1, v_2$ . If those vertices are connected by a double edge then we change the drawing as shown in Figure 9(c), and save one column and two bends. However, if other vertices have multiple edges, then the optimization step in Section 3.2 to reduce the width by one cannot be applied. This yields an  $(n + 1) \times n$ -grid and  $2n + 4$  bends for biconnected graphs. This cannot be generalized to non-biconnected graphs, since there are non-biconnected non-simple graphs that need  $\frac{7}{3}n$  bends in any drawing [1].

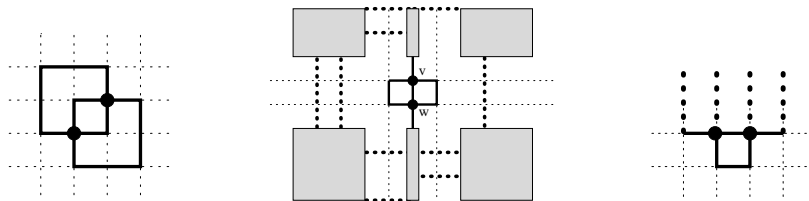


Figure 9: How to handle quadruple, triple, and double edges

The crucial observation now is that we need not know the values of  $x_i$  to find the  $x$ -coordinate of  $v$ . Instead we can maintain a list *Columns*. Every used gridpoint  $p$  contains a pointer  $x(p)$  to one element of *Columns* signifying that later all vertices with a pointer to the same element of *Columns* receive the same  $x$ -coordinate. When embedding  $v$  we let  $x(v)$  point to the same element in *Columns* as  $x_k$  points to. Whenever we want to add a column we add a new element in *Columns*. The columns are added directly left and right of the column of  $v$ . Since we know  $x(v)$  adding requires  $\mathcal{O}(1)$  time. The final  $x$ -coordinates are computed by traversing *Columns* and assigning ascending values to each element. Every vertex then checks the value of the element it points to and stores that value as its  $x$ -coordinate.

The method of adding the new columns directly neighboring the vertex we are dealing with is also interesting from a practical point of view for the non-planar case: the shape of the outerface of the drawing is a rectangle instead of a triangle and less crossings are to be expected. However, the ordering of  $x_1, \dots, x_s$  in *Columns* has to be computed to find the  $x$ -coordinate of  $v$ .

This problem is called the *order maintenance problem*: determining which of two elements comes first in a list under a sequence of Insert and Delete operations. Dietz & Sleator [5] presented a linear space data structure for this problem, answering the order queries in  $\mathcal{O}(1)$  time. This yields a linear time algorithm for the non-planar drawing algorithm (though less simple), where new columns are added directly neighboring the vertex. We used the fact that new columns are added at the border for our small improvement. However, when we allow just one column to be added at the extreme this can still be upheld.

For non-biconnected graphs we store the coordinates implicitly. Let  $B$  be a block. We denote by  $father(B)$  the vertex that was embedded as final vertex of block  $B$  (in the notation of Section 4  $father(G_0) = v$ ). We also store a value  $rotate(B)$  which denotes the angle of rotation when  $B$  is merged. Assume we have an embedding of  $B$  together with all its subgraphs. Denote by  $width(B)$ ,  $height(B)$  the width and height of this drawing and by  $cut(B)$  the  $x$ -coordinate of  $father(B)$  (this is the place where we will have to cut the drawing  $B$  when merging it later). By visiting all vertices  $w \in B$  in the drawing we can calculate  $dist_x(w)$  and  $dist_y(w)$ , the distance of  $w$  to  $father(B)$  in either coordinate.

Assume we have a block  $G_0$  and we want to merge the drawing of  $G_i$  at vertex  $v_i$ . First assume that  $v_i$  is not connected to  $G_i$  by a bridge and let  $B_i$  be the block containing  $v_i$  in  $G_i$ . We set  $rotate(B_i) = 180^\circ$  and add  $height(B_i) + 1$  rows. Furthermore we add  $cut(B_i)$  columns to the left and  $width(B_i) - cut(B_i)$  columns to the right of the incoming edge of  $v_i$ . Then we add one more column for the outgoing edge of  $v_i$ . We update only the coordinates of  $v_i$ , not of the other vertices in  $B_i$ . If  $v_i$  is in one bridge let  $w_i$  be the other endpoint of it and let  $B_i$  be the block in  $G_i$  containing  $w_i$ . We set  $rotate(B_i) = 90^\circ$ . We add  $height(B_i) + 1$  columns to the right of the incoming edges of  $v_i$ . We add  $cut(B_i)$  rows, then one row into which we place both  $v_i$  and  $w_i$  and then  $width(B_i) - cut(B_i)$  further rows. If  $v_i$  is part of two bridges we add the other subgraph in a similar fashion to the left.

## 5 Linear time complexity

In this section we describe how to implement the algorithm so that it works in linear time. There are two items to consider here: (i) how to embed  $G$  and to find an  $st$ -ordering for a planar  $G$  such that  $(v_1, v_2)$  and  $(v_{n-1}, v_n)$  are on the outerface (this statement is also valid for graphs of higher maximum degree), and (ii) how to compute all coordinates efficiently.

### 5.1 Computing the $st$ -ordering for planar graphs

The first problem is how to compute an  $st$ -ordering of a biconnected planar graph  $G$  such that the edges  $(v_1, v_2)$  and  $(v_{n-1}, v_n)$  are on the outerface. We assume that  $v_1$  and  $v_n$  are given in advance (important for the non-biconnected case) and belong to the outerface. We first have to find an ordering with edge  $(v_1, v_2)$  lying on the outerface. First determine the cutting pairs  $\{v_1, w\}$ , with  $w$  a neighbor of  $v_1$ , as follows. We mark all faces incident to  $v_1$ . Every neighbor  $w$  of  $v_1$  belonging to more than 2 marked faces forms a cutting pair with  $v_1$ . For every such  $w$  mark a face that contains  $w$  but not the edge  $(v_1, w)$ .

Let  $v^* \neq v_n$  be a neighbor of  $v_1$  on the outerface. If  $\{v_1, v^*\}$  is not a cutting pair we can contract edge  $(v_1, v^*)$  without losing biconnectivity. Compute an  $st$ -ordering for  $G$  with the contracted vertex as first and  $v_n$  as last vertex. Extract edge  $(v_1, v^*)$ , give vertex  $v_1$  number 1, vertex  $v^*$  number 2 and increase the other numbers by 1 to get the desired  $st$ -ordering. If  $\{v_1, v^*\}$  is a cutting pair, then we swap edge  $(v_1, v^*)$  inside the marked face of  $v^*$ . Then a new neighbor of  $v_1$  appears on the outerface and we repeat the argument with it. No neighbor of  $v_1$  will be processed twice, so after at most  $\deg(v_1)$  steps we have found a neighbor of  $v_1$  that does not form a cutting pair with  $v_1$  and lies on the outerface. We apply the above argument to find an appropriate  $st$ -ordering.

Assume now that the outerface contained at least four vertices and let  $\hat{v} \neq v_1, v^*$  be a neighbor of  $v_n$  on the outerface. Using the same argument with  $v_n$  and  $\hat{v}$  we can show that there is an  $st$ -ordering with  $(v_1, v_2)$  and  $(v_{n-1}, v_n)$  on the outerface. We needed the edge  $(v_{n-1}, v_n)$  on the outerface only to avoid the 3-bent edge for 4-regular biconnected graphs in which case we are free to choose  $v_n$ . Every planar graph with at least seven vertices and at most  $2n$  edges has a face with at least four vertices. So we can choose  $v_n$  to belong to this face and avoid for all such graphs the 3-bent edge.

### 5.2 Computing the coordinates

In the biconnected case remark first that the  $y$ -coordinate of a vertex is never changed later. The same holds for the  $x$ -coordinates in the non-planar case, so we only have to worry about the  $x$ -coordinates in the planar case. Let  $v$  be the vertex we are dealing with. Let  $e_1, \dots, e_s$  be the incoming edges of  $v$  in clockwise order in the adjacency list of  $v$ . Let  $\{x_1, \dots, x_s\}$  be the  $x$ -coordinates associated with  $e_1, \dots, e_s$ . Notice that if the graph is planar then  $x_1, \dots, x_s$  are descending and the  $x$ -coordinate of  $v$  is  $x_k$ , where  $k = \lceil \frac{s}{2} \rceil$ .



In the embedding of  $G_0$  every edge is bent at most once at each endpoint. Since we do not change the embedding of  $G_i$  this holds for all of  $G$ , so clearly we have at most two bends per edge.

If  $G$  is 4-regular we split  $G$  at a cutvertex  $v$  to obtain  $G_1$  and  $G_2$ . Since we have no bridge we know  $\deg_{G_i}(v) = 2$ . Embed  $G_1$  and  $G_2$  with the above algorithm with  $v$  as final vertex. Rotate and move the drawing of  $G_2$  such that the two drawings of  $v$  coincide. The width and height are then at most  $n_1 - 1 + n_2 - 1 = n - 1$  and the number of bends is  $2n_1 - 1 + 2n_2 - 1 = 2n$ .

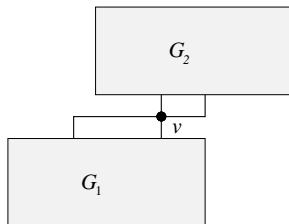


Figure 7: The drawing of a non-biconnected 4-regular graph

If our graph now has bridges then we add them after the rest of the drawing is finished. To be precise, assume that we have a bridge  $(u, w)$ . Removing it splits  $G$  into two graphs  $G_1$  and  $G_2$  (denote again by  $n_i$  the number of their vertices, but this time  $n_1 + n_2 = n$ ). Assume  $G_1$  contains  $u$  and the prescribed final vertex  $v$ . Delete all vertices of  $G_2$  in the original graph and embed the resulting graph (this has one bridge less, so we repeat the argument until we have no bridge left and then apply the above algorithm).

After all this is done,  $u$  has a connection in one direction free, assume it is to the right. Embed  $G_2$  with  $w$  as final vertex in an  $(n_2 - 1) \times (n_2 - 1)$ -grid. Add  $n_2$  columns to the right of  $u$  and  $n_2 - 1$  rows above and below it. Rotate the drawing of  $G_2$ , place it in the space to the right of  $u$  and connect  $u$  and  $w$ . The resulting grid has width  $n_1 - 1 + n_2 = n - 1$ , height  $n - 2$ , and  $2n_1 - 1 + 2n_2 - 1 = 2n - 2$  bends.

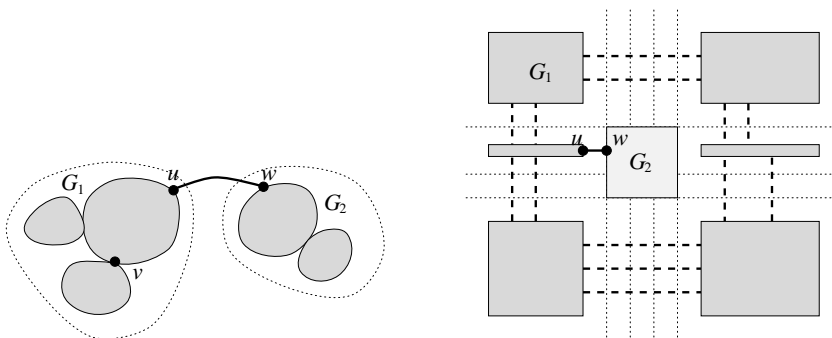


Figure 8: How to merge a subgraph connected by a bridge

**Theorem 4.3** *Let  $G = (V, E)$  be a not biconnected simple 4-graph with  $n$  vertices. Then  $G$  can be embedded in an  $(n - 1) \times (n - 1)$ -grid with at most  $2n$  bends. Every edge is 2-bent. If  $G$  is planar then so is the drawing.*

$fdeg(v_i) = bdeg(v_i) = 1$ . We first add one row and place  $v_i$  above its predecessor. Then we add  $n_i - 1$  more rows. There we will add the drawing of  $G_i - v_i$ , which we cut vertically to the left and right of the column of  $v_i$ . Let the two slices have  $k$  and  $l$  columns ( $k + l = n_i - 1$ ). We add  $l$  columns to the left and  $k$  columns to the right of the incoming edge of  $v_i$ , rotate the drawing of  $G_i$  and place it in them. We add one more column and place the outgoing edge of  $v_i$  in it. Figure 6 shows this for the non-planar case. In the planar case we again add the columns right next to the incoming edge and hence get a planar drawing.

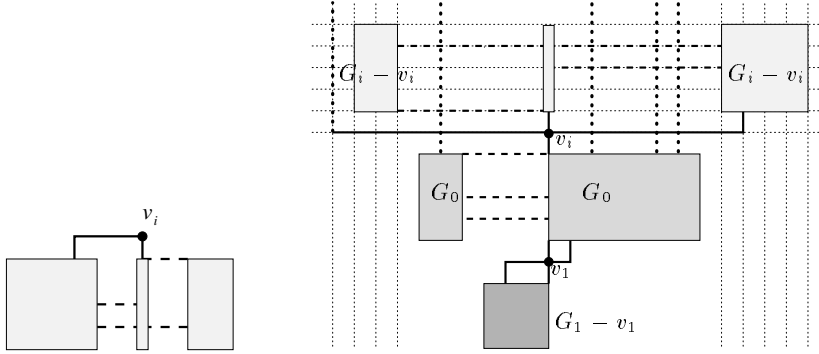


Figure 6: Cutting and merging  $G_i$  when embedding  $v_i$

We will show here only an estimation of the width. An estimation of the height is quite easy (every vertex needs one row) and an estimation of the number of bends is very similar to that of the width.

**Lemma 4.1** *The width is at most  $n$ .*

**Proof:** This is of course true for the base case (here the width is even  $n - 1$ ), so consider the case when we split  $G$  into  $G_0$  and the  $G_i$ 's. To embed a vertex  $w \neq w_1, w_{n_0}$  of  $G_0$  we need an increase in width of  $fdeg(w) - 1$  (we count the increase of  $n_i$  for embedding  $v_i$  later). For  $w = w_1, w_{n_0}$  we need  $fdeg(w)$  many columns. So for embedding  $G_0$  (not considering the mergings) we need a width of  $m_0 - n_0 + 2$ . In order to draw  $G_1, \dots, G_s$  we add  $\sum_{i=1}^s n_i$  columns. Applying  $(*_1)$  and  $(*_2)$  we bound the number of columns by  $m_0 - n_0 + 2 + \sum_{i=1}^s n_i \leq n_0 - s + 1 + \sum_{i=1}^s n_i = n + 1$  and the width is one less than that.  $\square$

**Lemma 4.2** *The width can be improved to  $n - 1$ .*

**Proof:** The estimation of the previous lemma is tight only for a few cases. Namely, if  $w_1 = v_1$  is a cutvertex then to embed  $w_1$  we need only  $fdeg(w_1) - 1$  new columns and hence get a width of  $n - 1$ . If  $(*_2)$  is not tight, then again we get a width of  $n - 1$ . So the only remaining case is that all vertices (except  $v$ ) in  $G_0$  are either a cutvertex or have degree at least 3. We now apply the same idea as in Subsection 3.2 to reduce the width of  $G_0$  by one. Let  $l$  be the smallest index such that  $w_l$  has at least two predecessors in  $G_0$  and embed  $w_l$  together with its last predecessor  $w_{i_b}$ . This always saves one unit of width since the last predecessor of  $w_l$  either has three neighbors (and hence another outgoing edge) or is a cutvertex (hence its column can be used to draw the subgraph).  $\square$

## 4 Non-biconnected graphs

In this section we describe how to embed simple graphs which are not biconnected with bounds similar to those in Theorem 3.5. To do so we split the graph into its blocks and embed them separately. As we will show later we can assume that  $G$  has no bridges, i.e. edges whose deletion disconnects  $G$ . Let us say that we draw a graph with  $v$  as “final vertex” if  $v$  and its incident edges are drawn in the last row. We then keep the following invariant:

**Invariant:** *Let  $G$  be a simple  $k$ -graph without bridges. Let  $v$  with  $\deg(v) \leq 3$  be given (and on the outerface in the planar case).  $G$  can be drawn with  $v$  as final vertex in an  $(n - 1) \times (n - 1)$ -grid with  $2n - 1$  bends. Every edge is 2-bent.*

We show the invariant by induction on the number of vertices. In the base case  $G$  is biconnected and we apply the algorithm of the previous section with  $v$  as last vertex. Note that this gives the desired bound since  $m \leq 2n - 1$ . For the induction step note first that  $v$  cannot be a cutvertex (otherwise by  $\deg(v) \leq 3$  it would be contained in a bridge). Let  $G_0$  be the unique block containing  $v$ . We will embed the biconnected  $G_0$  with  $v$  as last vertex as in the previous section and merge the rest of  $G$  at appropriate places.

Let  $v_1, \dots, v_s$  be the cutvertices of  $G$  in  $G_0$ . Let  $G_i$  be the subgraph of  $G$  consisting of  $v_i$  and the connected components of  $G - v_i$  not containing  $G_0$ . Notice that the intersection of  $G_0$  and  $G_i$  ( $1 \leq i \leq s$ ) is  $v_i$ , and that the intersection of  $G_i$  and  $G_j$  is empty if  $i \neq j$ . Hence if we define  $n_i = |V(G_i)|$  then

$$(*_1) \quad \sum_{i=0}^s n_i = n + s .$$

Define  $m_0$  to be the number of edges in  $G_0$ . As  $\deg_{G_0}(v) \leq 3$  we know  $m_0 \leq 2n_0 - 1$ . Furthermore, every cutvertex has degree 2 in  $G_0$  and we know

$$(*_2) \quad m_0 \leq 2n_0 - 1 - s .$$

$G_i$  has fewer vertices than  $G$ . If  $G$  was planar and we take as embedding of  $G_i$  the one that is induced by  $G$  then (since  $v$  was on the outerface of  $G$ ) also  $v_i$  is on the outerface of  $G_i$ . So by induction  $G_i$  can be embedded in an  $(n_i - 1) \times (n_i - 1)$ -grid with  $2n_i - 1$  bends and  $v_i$  as final vertex.

Compute an  $st$ -ordering  $\{w_1, \dots, w_{n_0}\}$  of  $G_0$  with  $v$  as last vertex. If  $G$  is non-planar, or if  $G$  is planar and one cutvertex, say  $v_1$ , is on the outerface of  $G$ , then we also choose  $v_1$  as the first vertex (by the above  $v \neq v_1$ ). For all  $w \in G_0$  denote by  $fdeg(w)$  (resp.  $bdeg(w)$ ) the number of predecessors (resp. successors) with respect to this ordering (so only  $w$ 's neighbors in  $G_0$  are counted).

If  $w_1 = v_1$  we start with the embedding of  $G_1$ . Add  $fdeg(v_1) - 1$  columns to the side of the drawing of  $G_1$  and connect  $v_1$  with them. Now all unfinished edges end in a column which is empty above this point. So the invariant of the algorithm for biconnected graphs holds. Otherwise we embed  $w_1$  as in the biconnected case and need  $fdeg(w_1)$  columns for it. Every following vertex  $w \in G_0$  which is not in  $\{v_1, \dots, v_s\}$  is embedded as in the biconnected case.

Assume we want to embed a cutvertex  $v_i$ . We know that  $\deg_{G_0}(v_i) = 2$  (otherwise we had a bridge in  $G$ ), and since we have an  $st$ -ordering we have

### 3.2 Embedding in an $n \times n$ -grid

It seems tedious to spend extra effort in order to reduce the width by 1. However, this will be used when embedding non-biconnected graphs. Here each block will be embedded separately, so in all we achieve a reduction in width equal to the number of blocks.

Assume that all vertices but  $v_n$  have degree 3 (as we will show presently we need to reduce the width only in this case). Let  $l$  be the smallest index such that  $bdeg(v_l) \geq 2$ . It is not possible that  $l = n$  since otherwise we had vertices of degree 2. Therefore we know that  $b = bdeg(v_l) \leq 3$ . Let the predecessors of  $v_l$  be  $v_{i_1}, \dots, v_{i_b}$  with  $i_1 < \dots < i_b$ . We now embed both  $v_l$  and  $v_{i_b}$  in one step, as shown in Figure 5.

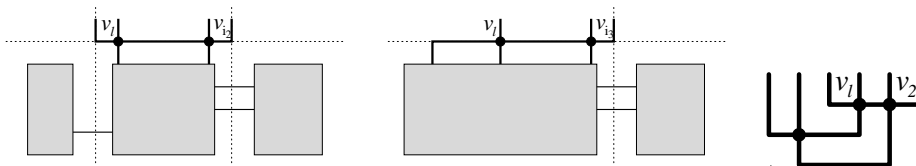


Figure 5: Drawing  $v_j$  and  $v_l$ , and the special case  $j = 2$

To be able to do so it is required that the incoming edge of  $v_{i_b}$  is to the left or to the right of all other incoming edges of  $v_l$ . It is clear that this is so if  $b = 2$ . But it is even ensured for  $b = 3$ . Notice that for any  $j < l$  we have  $deg(v_j) \geq 3$  and  $bdeg(v_j) \leq 1$ , therefore  $fdeg(v_{i_b}) \geq 2$ . Hence if  $G$  is planar the other outgoing edge of  $v_{i_b}$  must be on the outerface of  $G_l$ , since  $v_l$  has no predecessor with an  $st$ -number bigger than  $i_b$ . So after  $v_l$  was placed  $v_{i_b}$  must still be part of the outerface and hence  $(v_{i_b}, v_l)$  is the left- or rightmost incoming edge of  $v_l$ . Consequently also the incoming edge of  $v_{i_b}$  is to the left or to the right of the other incoming edges of  $v_l$ .

If the graph is non-planar, then let  $v_k$  be the predecessor of  $v_{i_b}$ . When embedding  $v_k$  we added a new column (since  $fdeg(v_k) \geq 3$ ) and we added it at the extreme left or right. Assign the edge  $(v_k, v_{i_b})$  to this column. Then by definition it is to the right or left of all other incoming edges of  $v_l$ .

What width and how many bends did we produce? We need a width of  $fdeg(v_{i_b}) - 2$  for  $v_{i_b}$ . If  $deg(v_{i_b}) = 3$ , then no bends are introduced for  $v_{i_b}$ , otherwise one bend. Similarly for  $v_l$  we need a width of  $fdeg(v_l) - 1$  and  $deg(v_l) - 3$  bends. Hence we save one column and two bends. Applying this for 4-regular graphs we get an  $n \times n$  grid and  $2n + 2$  bends. For  $m = 2n - 1$  we chose  $v_n$  to have minimal degree. So all other vertices have degree at least 3 and we get an  $(n - 1) \times (n - 1)$ -grid and  $2n - 1$  bends. For  $m \leq 2n - 2$  the algorithm of the previous subsection already gave an  $(n - 1) \times (n - 1)$ -grid and  $2n - 1$  bends since  $deg(v_n) \leq 3$ . The bounds for the special cases are summarized in Table 1 in the concluding section.

**Theorem 3.5** *Let  $G$  be a biconnected simple 4-graph with  $n$  vertices. Then  $G$  can be embedded in an  $n \times n$ -grid with at most  $2n + 2$  bends. Every edge is 2-bent. If  $G$  is planar then so is the drawing.*

**Lemma 3.2** *There are at most  $2m - 2n + 4$  bends; at most one edge is 3-bent, all others are 2-bent.*

**Proof:** We have  $\deg(v) - 2$  new bends with the embedding of  $v \neq v_1, v_2, v_n$ . Embedding  $v_1$  and  $v_2$  gives  $\deg(v_1) + \deg(v_2) - 2$  bends, and  $v_n$  requires  $\deg(v_n)$  bends if  $\deg(v_n) = 4$  and  $\deg(v_n) - 1$  bends otherwise. So we have  $\sum_{v \in V} (\deg(v) - 2) + 4 = 2m - 2n + 4$  bends if  $\deg(v_n) = 4$  and  $2m - 2n + 3$  bends otherwise.

Every edge  $(v_i, v_j), i < j$  is bent at most once when  $v_i$  is embedded. Completing the edge needs at most one additional bend if  $v_j \neq v_n$ . Embedding  $v_n$  bends one edge twice, all others at most once, thus only this edge can have three bends.  $\square$

**Lemma 3.3** *The 3-bent edge can be avoided unless the graph is the octahedron.*

**Proof:** Consider first graphs with less than  $2n$  edges. These have at least one vertex of degree less than 4. Since  $t$  is chosen with minimum degree there is no 3-bent edge. For non-planar 4-regular graphs we can (by drawing differently) choose which incoming edge we want to be bent twice with the embedding of  $v_n$ . We choose the edge that is outgoing from  $v_{n-1}$  (this exists by the property of an  $st$ -ordering) and is not bent at  $v_{n-1}$ . Therefore this edge is 2-bent, too.

For planar graphs 4-regular with at least 7 vertices we show in Section 5 that we can assume that the edge  $(v_{n-1}, v_n)$  is on the outerface. This edge then is the leftmost or rightmost incoming edge of  $v_n$ . So again we can bend this edge twice with the embedding of  $v_n$ . The only graphs which are not covered by these arguments are those which are planar, 4-regular, and have at most 6 vertices. But there is only one such graph: the octahedron.  $\square$

**Lemma 3.4** *For planar graphs the resulting drawing is planar.*

**Proof:** Remember that we chose  $v_1$  and  $v_n$  to be on the outerface. As we will show in Section 5 we can assume that the edge  $(v_1, v_2)$  is on the outerface, too. The construction then guarantees that the resulting embedding has the same set of rotations around each vertex and the same outerface boundary as the planar graph  $G$ . In particular we know that the incoming (outgoing) edges of any vertex  $v$  are consecutive in the rotation around  $v$  [24].

Let  $G_i$  be the subgraph induced by  $v_1, \dots, v_i$ . Notice that  $G_i$  is connected. Let  $E'_i$  be the set of edges from  $G_i$  to  $G - G_i$ , ordered according to their clockwise order around the outerface boundary of  $G_i$ . We now state that the incoming edges of  $v_i$  form a consecutive sequence in  $E'_{i-1}$ . Assume not, i.e. there is an edge  $(v_j, v_k), j < i < k$  between two consecutive incoming edges of  $v_i$ , say  $(u, v_i)$  and  $(w, v_i)$ . Consider the cycle  $C = v_i - u - P(u, w) - w - v_i$ , where  $P(u, w)$  is the path from  $u$  to  $w$  on the outerface of  $G_{i-1}$ . Then  $C$  is a face boundary of  $G_i$ , and  $v_k$  is in its inner domain. Let  $l > i$  be the highest number such that  $v_l$  is in the inner domain of this face. Then  $v_l$  has no successor by planarity, but it also cannot be  $v_n$  since  $v_n$  is chosen to be on the outerface of  $G$ . Contradiction.

Hence when adding  $v_i$  to  $G$ , the columns of incoming edges are an interval and we do not cross other edges. As we add new columns directly neighboring the added vertex  $v_i$  there is no vertical line that could be crossed.  $\square$

### 3.1 Embedding in an $(m - n + 1) \times n$ -grid

Given a biconnected 4-graph  $G = (V, E)$ . Choose some vertices  $s$  and  $t$ , namely choose  $t$  to have minimum degree and, in the planar case,  $s$  to be on one face with  $t$  which then becomes the outface. Obtain an  $st$ -ordering for  $G$  and embed the vertices consecutively. We hold the invariant that at every stage every *uncompleted edge* (i.e. an edge where exactly one endpoint is drawn) is associated to a column. The drawing of the edge ends in that column which is empty above that point.



Figure 2: Embedding of the first two vertices and appearance after some more vertices

The first two vertices are embedded as shown above. For every following vertex we add a new row on top and as many columns as necessary to maintain the invariant. If  $G$  is planar then we add the new columns directly neighboring the current vertex. Otherwise they are added at the border. The last vertex might have four predecessors, in which case we add two rows to accommodate it.

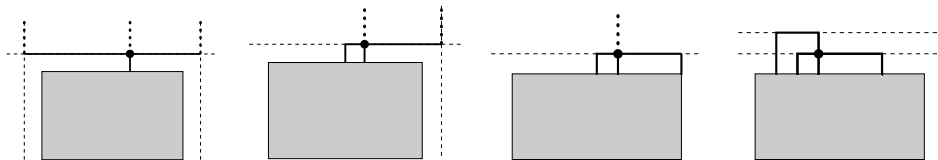


Figure 3: Embedding a vertex  $v$  with four neighbors ( $bdeg(v) = 1, 2, 3, 4$ )



Figure 4: The changed drawing for  $fdeg(v) = 3, 2$  in the planar case

**Lemma 3.1** *The gridsize is  $(m - n + 1) \times n$*

**Proof:** We use a height of one for  $v_1$  and  $v_2$ . Every following vertex increases the height by one, the last vertex by at most two. Therefore the height is  $n - 1$  if  $deg(v_n) \leq 3$  and  $n$  otherwise. When embedding  $v_1$  and  $v_2$  we use a width of  $fdeg(v_1) + fdeg(v_2) - 2$  (observe that the width is  $\#\{\text{columns}\} - 1$ ). Every following vertex  $v_i \neq v_n$  increases the width by  $fdeg(v) - 1$  and  $v_n$  increases it by  $0 = fdeg(v_n) - 1 + 1$ . So the width is  $\sum_{v \in V} (fdeg(v) - 1) + 1 = m - n + 1$ .  $\square$

An orthogonal drawing is called an *embedding in the (rectangular) grid* if all vertices and bendpoints are drawn on integer points. If the drawing can be enclosed by a quadrangle of *width*  $n_1$  and *height*  $n_2$  we call it an embedding with *gridsize*  $n_1 \times n_2$  and *area*  $n_1 \cdot n_2$ .

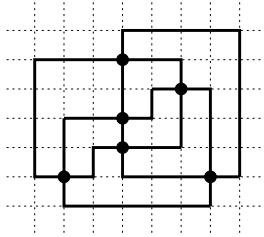


Figure 1: The octahedron in a  $7 \times 6$  grid with 16 bends

Call a graph *biconnected* if it has no loops and removing any vertex and its incident edges leaves a connected graph. Unless otherwise specified we consider only *simple graphs*, i.e. graphs without loops and multiple edges. The *biconnected components* (or *blocks*) of a connected graph are (a) its maximal biconnected subgraphs, and (b) its bridges together with their endpoints. If removing  $v$  disconnects the graph we call  $v$  a *cutvertex*. Every vertex that is not a cutvertex is in exactly one block of  $G$ . If removing  $\{v_1, v_2\}$  disconnects the graph we call  $\{v_1, v_2\}$  a *cutting pair* of  $G$ .

A (*combinatorial*) *embedding* of a planar graph is a representation in which at every vertex all edges are sorted in clockwise order with respect to the planar embedding. Testing biconnectivity, finding all cutting pairs, and constructing an embedding of a planar graph can be done in linear time (see e.g. [2]). The *octahedron* is the unique planar 4-regular graph with 6 vertices. An *st-ordering* is an ordering  $\{v_1, v_2, \dots, v_n\}$  of the vertices such that every  $v_j$  ( $2 \leq j \leq n-1$ ) has at least one *predecessor* and at least one *successor*, i.e. neighbors  $v_i, v_k$  with  $i < j < k$ .

**Theorem 2.1 (Lempel, Even & Cederbaum [15], Even & Tarjan [7])**  
*Let  $G$  be biconnected and  $s, t \in V$ . Then there exists an  $st$ -ordering such that  $s$  is the first and  $t$  is the last vertex. It can be computed in  $\mathcal{O}(m)$  time.*

If  $G$  is planar then in any  $st$ -ordering  $\{v_1, \dots, v_n\}$  the predecessors of  $v_i$  appear as an interval in the adjacency list of  $v_i$  in any embedding [24]. The edges from  $v_i$  to its predecessors (successors) are called *incoming* (*outgoing*) edges of  $v_i$ . Their number is the backdegree  $bdeg(v_i)$  (forwarddegree  $fdeg(v_i)$ ). A straightforward calculation shows  $\sum_{i=1}^n bdeg(v_i) = \sum_{i=1}^n fdeg(v_i) = m$ .

### 3 Drawing biconnected graphs

The basic idea for drawing biconnected graphs is the same for planar and non-planar graphs. Independently, Liu et al. [18] came up with a similar technique for planar biconnected graphs. However, we will show how to handle the non-planar case with our technique while maintaining the same bounds.

If  $G$  is planar it can be embedded in an  $n \times n$  grid with  $2n + 4$  bends if it is biconnected, and  $2.4n + 2$  bends otherwise [22, 24]. The number of bends along each edge is at most 4. For 3-planar and triconnected 4-planar graphs better bounds are known [10, 16, 17]. Very recently and independently of this work, Liu, Morgana & Simeone [18] presented a linear-time algorithm for drawing biconnected planar graphs with at most 2 bends per edge (except the octahedron) and at most  $2n + 4$  bends in total on an grid of size at most  $(n + 1) \times (n + 1)$ .

In [25] a lower bound of  $2n - 2$  is presented on the number of bends in an orthogonal drawing of a certain biconnected planar graph. If a combinatorial embedding of a planar graph is given, an orthogonal representation of it with minimal number of bends can be computed in  $\mathcal{O}(n^2 \log n)$  time [23]. However, the number of bends per edge can be large which makes the drawing unattractive. If the planar embedding is not given, the problem is polynomial time solvable for 3-planar graphs [4] and NP-hard for 4-planar graphs [9]. In particular, Garg & Tamassia showed that it is even NP-hard to approximate the minimum number of bends in a planar orthogonal drawing with an  $\mathcal{O}(n^{1-\epsilon})$  error for any  $\epsilon > 0$  [9].

The latter motivates the research for a simple and very general heuristic to construct orthogonal representations of planar and non-planar graphs. In this paper we present a new algorithm that runs in  $\mathcal{O}(n)$  time and produces orthogonal drawings of connected planar and non-planar graphs with the following properties: (i) the total number of bends is at most  $2n + 2$ ; (ii) the number of bends along each edge is at most 2 (unless the graph is the octahedron) (iii) the area of the embedding is  $n \times n$ . In particular for non-planar and non-biconnected planar graphs, this is a big improvement. The result is obtained by constructing orthogonal drawings of the biconnected components using the so-called *st*-ordering and merging all these drawings into one orthogonal drawing of the entire graph.

The paper is organized as follows: Section 2 gives some definitions and introduces the *st*-ordering. In Section 3 the algorithm for biconnected graphs is explained. It is extended to non-biconnected graphs in Section 4. In Section 5 we explain how to arrive at a linear time implementation. In Section 6 we show how the algorithm can be used to draw graphs with multiple edges and graphs with higher maximum degree. Section 7 contains remarks and open problems.

## 2 Definitions

Let  $G = (V, E)$  be a graph,  $n = |V|$ ,  $m = |E|$ . We consider only *4-graphs*, i.e. graphs of maximum degree 4. Here  $m = \frac{1}{2} \sum_{v \in V} \deg(v) \leq 2n$ . Call such a graph *4-regular* if all vertices have exactly 4 neighbors. An *orthogonal drawing* of  $G$  is an embedding of  $G$  in the plane such that all edges are drawn as sequences of horizontal and vertical lines. A point where the drawing of an edge changes its direction is called a *bend* of this edge. If the drawing of an edge has at most  $k$  bends we call this edge *k-bent*.



# A Better Heuristic for Orthogonal Graph Drawings\*

Therese Biedl  
RUTCOR – Rutgers University  
P.O. Box 5062  
New Brunswick  
NJ 08903-5062 USA  
therese@rutcor.rutgers.edu

Goos Kant  
Department of Computer Science  
Utrecht University  
Padualaan 14, 3584 CH Utrecht  
the Netherlands  
goos@cs.ruu.nl

## Abstract

An orthogonal drawing of a graph is an embedding in the plane such that all edges are drawn as sequences of horizontal and vertical segments. We present a linear time and space algorithm to draw any connected graph orthogonally on a grid of size  $n \times n$  with at most  $2n + 2$  bends. Each edge is bent at most twice.

In particular for non-planar and non-biconnected planar graphs, this is a big improvement. The algorithm is very simple, easy to implement, and it handles both planar and non-planar graphs at the same time.

## 1 Introduction

The research area of *graph drawing* has become an extensively studied field which presents an exciting connection between computational geometry and graph theory. The wide spectrum of applications includes VLSI-layout, software engineering, and project management (see [3] for an up-to-date overview with more than 300 references). The aesthetic quality of a drawing cannot be precisely defined, and depending on the application different criteria have been used. Important characteristics of a “readable” representation are the number of bends, the number of crossings, the sizes of the angles, and the required area.

In this paper we study the problem of *orthogonal drawings*, i.e. the drawing of a graph  $G = (V, E)$  where each edge is represented by a sequence of alternating horizontal and vertical segments. Such a representation is possible only when every vertex has at most 4 incident edges. In [11, 8] it was shown that deciding whether  $G$  can be embedded in a grid of prescribed area is NP-complete. Lengauer [12] posed it as an exercise to show that  $G$  can be embedded in a  $2n \times 2n$  grid such that there are at most five bends per edge. In [21] this was improved to an embedding of same size with at most two bends per edge.

---

Research of the second author was supported by the ESPRIT Basic Research Actions program of the EC under contract No. 7141 (project ALCOM II). An extended abstract of this paper was presented at the *2nd Annual European Symp. on Algorithms*, Utrecht, 1994.

ISSN: 0924-3275

# A Better Heuristic for Orthogonal Graph Drawings

Therese Biedl  
RUTCOR – Rutgers University  
P.O. Box 5062  
New Brunswick  
NJ 08903-5062 USA  
therese@rutcor.rutgers.edu

Goos Kant  
Department of Computer Science  
Utrecht University  
Padualaan 14, 3584 CH Utrecht  
the Netherlands  
goos@cs.ruu.nl

Technical Report UU-CS-1995-04  
February 1995

Department of Computer Science  
Utrecht University  
P.O.Box 80.089  
3508 TB Utrecht  
The Netherlands

# A Better Heuristic for Orthogonal Graph Drawings

Therese Biedl  
RUTCOR – Rutgers University  
P.O. Box 5062  
New Brunswick  
NJ 08903-5062 USA  
therese@rutcor.rutgers.edu

Goos Kant  
Department of Computer Science  
Utrecht University  
Padualaan 14, 3584 CH Utrecht  
the Netherlands  
goos@cs.ruu.nl

UU-CS-1995-04  
February 1995



**Utrecht University**  
Department of Computer Science

Padualaan 14, P.O. Box 80.089,  
3508 TB Utrecht, The Netherlands,  
Tel. : + 31 - 30 - 531454