A NEW APPROACH FOR DRAWING A HIERARCHICAL GRAPH
(extended abstract)

Peter Eades [1], Xuemin Lin [1] and Roberto Tamassia [2]

1. INTRODUCTION

Graph layout algorithms are increasingly used in visualization for information and software engineering: see [RDMMST], [GNV], for example. The quality of the layout is the key to the success of the model: a nicely drawn graph is worth a thousand words, but a poor layout can be misleading.

Of course the quality of a layout is a subjective matter, but several aesthetic criteria are generally accepted. For instance, a symmetric drawing is desirable, and edge crossings should be avoided. Other aesthetic criteria include the minimization of the number of bends in the edges [TA], the number of different slopes used by edges [CPRU], and the variance in the edge length [S].

In this paper we present a method for drawing *hierarchical directed graphs*, which are digraphs in which each node is assigned a *layer* as in Figure 1.

Hierarchical graphs appear in several graph drawing applications, where nodes are assigned layers for semantic reasons. More importantly, general methods for drawing directed graphs usually begin by transforming the input digraph into a hierarchical graph, then apply a hierarchical graph drawing algorithm: see [EL], [RDMMST], [GNV].

Our layout algorithm satisfies several aesthetic criteria, the most important of which are (in intuitive terms):

(1) if a planar drawing of the hierarchical graph H is possible, then the drawing $D(H)$ of H output by our algorithm has no arc crossings, and

(2) if a symmetric drawing of H is possible, then $D(H)$ is symmetric.

These properties are precisely defined and proved in sections 4 and 5 respectively. Some further remarks are in section 6.

2. TERMINOLOGY

Standard graph theoretic terminology is from [BM].

A *hierarchical graph* $H=(V, A, \lambda, k)$ consists of a directed graph (V, A), and, for each vertex u, an integer $\lambda(u) \in \{1,2,\ldots,k\}$, with the property that if $u \rightarrow v \in A$ then $\lambda(u) > \lambda(v)$. For $1 \leq i \leq k$ the set $\{u : \lambda(u) = i\}$ is the ith *layer* of H and is denoted by L_i. Note that a hierarchical graph is acyclic.

The layers L_1 and L_k are the *boundary layers* of H. In many drawing applications, all sources and sinks are in L_k and L_1 respectively; in this case H is a *boundary s−t graph.*

The *span* of an arc $u \rightarrow v$ is $\lambda(u) - \lambda(v)$. Arcs of span greater than one are *long*; and a hierarchical graph with no long arcs is *proper*.

1. Geometric Algorithms Laboratory, Department of Computer Science, University of Queensland, St. Lucia, QLD Australia 4067, eades@uqcspe.cs.uq.oz.au.

2. Department of Computer Science, Brown University, Box 1910, Providence, Rhode Island 02912 - USA, rt@cs.brown.edu.

A hierarchical graph H is *well connected* if for each pair u, v of vertices in the same layer L_i, there are two paths p_u and p_v such that

- both paths start in L_k and end in L_1; and
- p_u contains u and p_v contains v ; and
- if S is the set of vertices shared by p_u and p_v, then either $S \subseteq \bigcup_{j<i} L_j$, or $S \subseteq \bigcup_{j>i} L_j$.

The results below are quoted for well connected proper boundary $s-t$ graphs, but in many cases they can be adjusted to hold for hierarchical graphs in general; see section 6.2.

A hierarchical graph is conventionally drawn with layer L_i on the horizontal line $y = i$ as in Figure 1. Thus the y coordinates of vertices are fixed by the drawing convention. Further, arcs may be drawn as straight lines with no loss to aesthetic criteria. Thus a *drawing* of a hierarchical graph $H=(V, A, \lambda, k)$ is effectively defined by a mapping which assigns an x co-ordinate $x(u)$ to each vertex $u \in V$. The drawing is *hierarchically planar* if no pair of nonincident arcs cross. A planar drawing is *convex* if every face is a convex polygon. A hierarchical graph is *hierarchically planar* if it has a hierarchically planar drawing.

A *boundary drawing* b of H assigns an x coordinate $b(u)$ for each u in the boundary layers of H. A drawing D of H *extends* the boundary drawing b if the position of each u in the boundary layers of H is $b(u)$. Our algorithm essentially assumes that vertices in the boundary layers have fixed positions; thus we say H is *hierarchically planar with respect to b* if it has a planar drawing which extends b.

Denote the indegree of u by d_u^-, and the outdegree of u by d_u^+.

3. THE ALGORITHM

Our algorithm is based on an algorithm proposed by Tutte [TU] for drawing triconnected planar graphs. The algorithm has some mechanical intuition, as follows. Replace each arc of H by an length of elastic cord, and fix the positions of the vertices in the boundary layers. Then allow this mechanical system to reach a minimum energy position; this is the drawing.

More precisely, for a proper boundary $s-t$ graph $H = (V, A, \lambda, k)$ we choose a boundary drawing b for H, for eaxmple by spacing L_1 and L_k equally on the lines $y = 1$ and $y = k$. Then we choose the x coordinate $x(u)$ for each nonboundary vertex u by

$$x(u) = \frac{1}{2d_u^-} \sum_{v \to u} x(v) + \frac{1}{2d_u^+} \sum_{u \to w} x(w). \quad (*)$$

Theorem 1: If H is a proper boundary $s-t$ graph and b is a boundary drawing of H, then the equations (*) have a unique solution extending b. □

The equations (*) can be solved by simple Gaussian elimination. However special properties can be exploited to find a solution in time $O(n^{3/2})$; see [LRT]. In practice, we have found that a simple Gauss-Seidel iteration works well.

For a hierarchical graph H, the drawing output by the algorithm is denoted by $D_b(H)$.

Note that the equations we use are different from Tutte's equations; in fact, the theorems in section 4 and 5 below would not hold for an algorithm directly using Tutte's equations for hierarchical graphs. A sample output of our algorithm is in Figure 2.

Extension of the algorithm to cover improper graphs is discussed in section 6.

4. PLANARITY AND CONVEXITY

An algorithm for finding a planar drawing of a proper boundary $s-t$ graph is given in [DN]. We can show that our algorithm achieves planar drawings for a given boundary:

Theorem 2: Suppose that H is a well connected proper boundary $s-t$ graph. If H hierarchically planar with respect to the boundary drawing b then the drawing $D_b(H)$ is planar and convex.

Proof sketch: Convexity can be shown directly from the equations (*).

Since H is hierarchically planar with respect to b, there is some drawing $D^{(0)}(H)$ which is planar with respect to b. Suppose that $x^{(0)}(u)$ is the position of u in $D^{(0)}(H)$. We define $x^{(j)}(u)$, $j>0$, as follows. If u is a boundary vertex, then $x^{(j)}(u) = x^{(j-1)}(u)$. Otherwise

$$x^{(j)}(u) = \frac{1}{2d_u^+} \sum_{u \to v} x^{(j)}(v) + \frac{1}{2d_u^-} \sum_{w \to u} x^{(j-1)}(w)$$

The hierarchy of vertices in H ensures that $x^{(j)}(u)$ is uniquely defined for $j \geq 0$. Let $D^{(j)}(H)$ be the drawing with vertex u at $x^{(j)}(u)$. Using Theorem 1, we can show that $D^{(j)}(H)$ converges to $D_b(H)$.

A geometrical argument can be used to show if $D^{(j-1)}(H)$ is planar, then $D^{(j)}(H)$ is planar. □

5. SYMMETRY

In this section we show that the algorithm above "displays symmetries"; intuitively, this means that if H has an appropriate automorphism γ, then $D_b(H)$ has a symmetry corresponding to γ. A general model for symmetry in graph drawings is introduced in [ELM] to make such intuitive notions precise; here we apply the model to hierarchical graphs.

A drawing of a graph is *symmetric* if there is a nontrivial isometry of the plane which maps the image of a vertex to the image of a vertex, and maps the image of an edge to the image of an edge. Note that the centroid c of the positions of the vertices is fixed by a symmetry. It can be shown that a symmetry of a drawing of a hierarchical graph is either

(1) a reflection in a vertical line through c; or

(2) a reflection in a horizontal line through c; or

(3) a product of (1) and (2).

A *geometric automorphism* of a hierarchical graph $H = (V, A, \lambda, k)$ is an automorphism γ of the underlaying undirected graph such that $\gamma^2 = 1$ and either

(1) γ preserves the layers, that is, $\gamma(L_i) = L_i$ for $1 \le i \le k$, or

(2) γ reverses the layers, that is, $\gamma(L_i) = L_{k-i+1}$ for $1 \le i \le k$.

It is not difficult to show that a symmetry of a drawing of a hierarchical graph induces a geometric automorphism of the hierarchical graph. For the output $D_b(H)$ of our algorithm, we can show the converse.

Theorem 3: Suppose that γ is a geometric automorphism of H. Then $D_b(H)$ has a symmetry which induces γ.

Proof Sketch: The same strategy used to prove Theorem 2 can be used for Theorem 3. First we show that there is a symmetric drawing of H. Then we show that the iteration used in the proof of Theorem 2 preserves this symmetry. □

6. REMARKS

6.1. Minimizing Crossings

The problem of minimizing arc crossings in a hierarchical graph is NP-complete, even if there are only two layers [GJ]. Further, the crossing minimization problem is NP-complete even when the boundary drawing is fixed:

Fixed Boundary Crossings problem:

INSTANCE: A boundary $s-t$ graph H with boundary drawing b; an integer K.

QUESTION: Is there a hierarchical drawing of H which extends b and has at most K crossings?

Theorem 4: the Fixed Boundary Crossing problem is NP-complete.

Proof Sketch: The Theorem can be proved using a result from [EW]. □

Several heuristics have been considered for this problem, and many experimental results have been reported; see [M], for example. Some of these, such as those derived from the the "barycentre" method [STT], are related our algorithm in that, if run for long enough, they would converge to $D_b(H)$.

We have tested our algorithm on a range of graphs with up to 100 vertices. The number of crossings in the output is generally about half the number of crossings in a random hierarchical drawing.

6.2. Extensions

The Theorems above can be generalized to cover improper hierarchical graphs, using a method which is standard in the graph drawing literature: each arc of span $k > 1$ is replaced by a path of $k-1$ "dummy" vertices, then the algorithm is applied. In the final drawing, the dummy vertices are removed. Note that this strategy ensures that the long arcs are drawn as straight lines by our algorithm. It is not difficult to establish symmetry and planarity as well.

Likewise, an adjustment of the Theorems can be applied for graphs which are not well connected. In this case the solution of the equations

(*) may place more than one vertex at the same location; however, the vertices can be separated without destroying planarity and symmetry.

REFERENCES

[BM] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications*, Macmillan 1976.

[DN] G. Di Battista and E. Nardelli, "An Algorithm for Testing Planarity of Hierarchical Graphs", *Lecture Notes in Computer Science* 246, (1987), 277-289.

[CPRU] J. Czyzowicz, A. Pelc, I. Rival and Urrutia, "Crooked Diagrams with Few Slopes", TR-87-26, University of Ottawa, 1987.

[EL] P. Eades and X. Lin, "How to Draw a Directed Graph", Proceedings of the 1989 IEEE Workshop in Visual Languages, IEEE Computer Society Press 1989, 13-17.

[ELM] P. Eades, X. Lin and B. D. McKay, "Models for Symmetric Graph Drawing", manuscript, University of Queensland 1990.

[ET] P. Eades and R. Tamassia, "Algorithms for Drawing Graphs: an Annotated Bibliography", Tech. Report CS-89-09, Department of Computer Science, Brown University.

[EW] P. Eades and N. C. Wormald, "Edge Crossings in Drawings of Bipartite Graphs", *Technical Report* 109, Department of Computer Science, University of Queensland 1989.

[GNV] E. R. Gansner, S. C. North, K. P. Vo, "DAG - A Program that Draws Direct Graphs", *Software Practice and Experience*. Vol. 18, No. 11 (1988), 1047 - 1062.

[GJ] M. R. Garey and D. S. Johnson, "Crossing Number is NP-Complete", *SIAM J. of Algebraic and Discrete Methods* 4 (3) (1983), 312-316.

[LRT] R. Lipton, D. Rose and R. Tarjan, "Generalized Nested Dissection", *SIAM J. Num. Anal.* No. 2, 1979.

[M] E. Makinen, "Experiments in Drawing 2-Level Hierarchical Graphs", Report A-1988-1, Department of Computer Science, University of Tampere.

[RDMMST] L. A. Rowe, M. Davis, E. Messinger, C. Meyer, C. Spirakis, and A. Tuan, "A Browser for Directed Graphs", *Software Practise and Experience* 17 (1) (1987), 61-76.

[S] K. Sugiyama, "A Cognitive Approach for Graph Drawing", *Cybernetics and Systems* 18 (6), (1987) 447-488.

[STT] K. Sugiyama, S. Tagawa and M. Toda, "Methods for Visual Understanding of of Hierarchical System Structures", *IEEE Transactions on Systems, Man and Cybernetics* SMC-11 (2) (1981), 109-125.

[TA] R. Tamassia, "On Embedding a Graph in the Grid with a Minimum Number of Bends", *SIAM J. of Computing* 16 (3) (1987), 421-444.

[TU] W. Tutte, "How to Draw a Graph", *Proceedings of the London Mathematical Society*, 3 (13) (1963), 743-768.

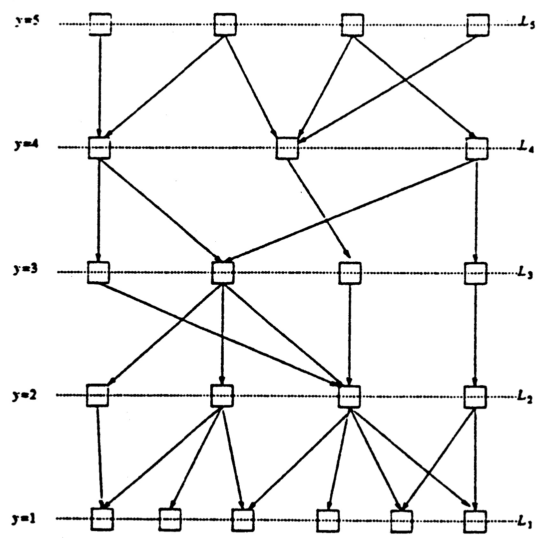

Figure 1

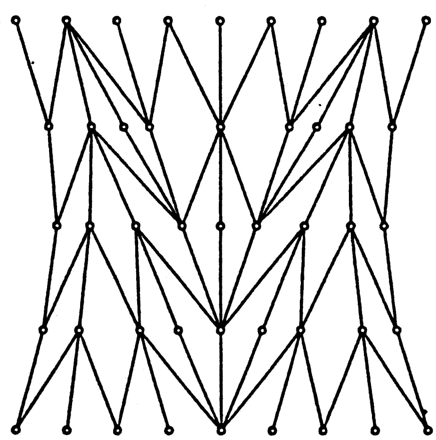

Figure 2