A New Heuristic Layout Algorithm for Directed Acyclic Graphs*

by

Stefan Dresbach

Lehrstuhl für Wirtschaftsinformatik und Operations Research
Universität zu Köln
Pohligstr. 1, 50969 Köln

revised August 1994

Working Paper 2/94
June 1994

Telefon: ++49 221 470 5333 · Fax: ++49 221 470 5329 · e-mail: dresbach@informatik.uni-koeln.de

* To appear in: U. Derigs, A. Bachem, and A. Drexl (Eds.): "*Operations Research Proceedings 1994*", Berlin: Springer 1995, 121-126.

Summary

Directed acyclic graphs (DAGs) are a common representation of hierarchical relationships and are widely used in software and information engineering, system theory, and others. The usefulness depends on a readable, understandable and easy to remember layout. The key problem in making such a good layout is the reduction of numbers of arc crossings.

We focus on the two layer problem and make a fresh attempt at a heuristic method of arc crossing reduction based on a stochastic approach. The "stochastic" heuristic is described in detail. The results of the approach are analyzed. A comparison of these results with the mostly used methods of barycentering and median-ordering is given. The "stochastic" heuristic is shown to be preferable in terms of WARFIELD's generating matrices.

Outline

1. Introduction
2. Basic definitions
3. Main idea of the stochastic heuristic
4. Example
5. Results
6. Conclusion

1. Introduction

Hierarchical relationships are often represented by directed acyclic graphs (DAGs). Examples from the field of software and information engineering or system theory are PERT networks [6], Interpretative Structural Modeling [14], subroutine-call graphs, and organization charts. In order to be useful, the layout of graphs has to be readable, understandable, and easy to remember. The aesthetics of a good DAG layout include beside the hierarchical drawing, bends minimization, symmetry, drawing area minimization, and minimizing the number of edge crossings [12, 13]. The last issue causes the greatest difficulties, so it can be seen as the key problem [10].

Laying out commonly consists of three phases: assigning vertices to layers, reducing the number of edge crossings by permuting the vertices within the layers, and assigning vertex positions [11].

There are several heuristic approaches to the second problem. We refer to the annotated bibliography of DI BATISTA et al. [1] for a survey. The most popular ones are barycentering [11, 12] and median-ordering [4]. The latter is sure to find a solution that has at most three times the minimum number of crossings [5]. There are several other approaches including [2, 3, 7, 8]. Empirical results are only given by SUGIYAMA et al. [11] for barycentering.

In a directed acyclic graph $D=(V,A)$ we can always find a partition of the vertices:

$$V = V_1 \cup V_2 \cup \ldots \cup V_L \quad (V_i \cap V_j = \emptyset,\ i \neq j) \tag{1}$$

such that

$$\forall\ (a,b) \in A \qquad (a \in V_i,\ b \in V_j \Rightarrow i < j) \tag{2}$$

where L is the number of vertices in a longest directed path in D. V_i is called the i^{th} layer.

2. Basic definitions

Our heuristic applies to the subproblem where there are only two layers U and W. In the following we will therefore consider an undirected bipartite graph $G=(V,E)$ with $V=U\cup W$ since the orientation of the edges is no longer of importance to the crossing number once the layers have been established. The number of crossings only depends on the positioning of the vertices in each layer [3]. Minimizing the crossing number k is a NP-complete problem, even if the ordering in one layer is fixed [5,9].

Let $s:=|U|$ and $t:=|W|$, denote the degree of a vertex as $\delta(u)=\#\{w\,|\,(u,w)\in E\}$. A drawing of G is represented by a adjacency matrix $M=m_{i,j}$ whose rows correspond to the elements of U in the right order and the columns to the elements of W in the right order, where the "right" order is the order in which the vertices appear within the respective layers in this particular drawing. The coefficient $m_{i,j}$ of M is 1 whenever the vertices corresponding to row i and column j are adjacent, and 0 otherwise. The number of edge-crossings in the drawing of G can now be calculated by (see [15]):

$$k(M)=\sum_{i=1}^{s-1}\sum_{j=i+1}^{s}\sum_{\alpha=1}^{t-1}\sum_{\beta=\alpha+1}^{t} m_{i,\beta}\cdot m_{j,\alpha} \tag{3}$$

The drawing positions in the layers will be referrd to as $p_1..p_s$ and $q_1..q_t$, respectively.

3. Main idea of the stochastic heuristic

Let $F=(f_{i,j})_{i,j}$ be a frequency matrix such that $f_{i,j}$ is the number of edges that cross an edge drawn between position p_i and q_j when a complete bipartite graph is drawn, i.e. $f_{i,j}$ is the number of occurences of the element $m_{i,j}$ in the edge-crossing calculation formula (3). Thus:

$$f_{i,j} = (t-j)\cdot(i-1) + (s-i)\cdot(j-1) \qquad \forall\ i=1..s,\ j=1..t \tag{4}$$

This matrix F can be interpreted as the statistical frequence table for every $m_{i,j}$ in (3). $f_{i,j}$ is a measure for the probability of a particular edge to cause a crossing in the graphical representation of G. So it can be used to estimate the effect on the number of edge crossings in case of positioning an element of U at p_i and simultaneously an element of W at q_j. An upcoming geometrical mean calculation makes it necessary to redefine two values of the frequency matrix: $f_{1,1}:=1, f_{s,t}:=1$.

Now an assessment number $a_{i,k}$ of a vertex $u_i \in U$ positioned at p_k can be calculated as the geometrical mean of those elements of the frequency matrix which correspond to an edge in the row vector of vertex u_i. Assessment numbers $b_{j,l}$ for vertices $w_j \in W$ are defined in the same manner. More formally for all $i=1..s$ and $k=1..s$ define:

$$a_{i,k} := \sqrt[\delta(u_i)]{\prod_{j=1}^{t} (f_{k,j})^{m_{i,j}}} \tag{5}$$

The similar definition is valid for $b_{j,l}$ with $j=1..t$ and $l=1..t$.

An assessment number can be calculated for every vertex in every position referring to each layer resulting in an assessment number matrix A of dimension $s \times s$ for the first layer and B of dimension $t \times t$ for the second layer. The numbers $a_{i,k}$ measure the effect of placing $u_i \in U$ at p_k, and likewise for $b_{j,l}$. The coefficients are small when only a few crossings can be expected.

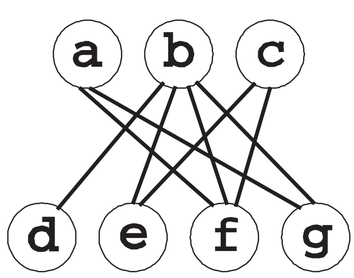
The main idea of the stochastic heuristic is now to put vertices in a greedy like fashion to the free positions with the smallest assessment numbers. If all coefficients in a row of A respectively B are equal then obviously there is no favorite position for the corresponding vertex. When looking for a minimal coefficient among those of A and B these rows are irrelevant. Similary we take into account only those assessment numbers that correspond to positions which are free and to vertices that have not been placed yet. After positioning as many vertices as possible the remaining ones are fixed ordinarily. The complete algorithm reads as follows:

```
generate frequency_matrix;
generate assessment_number_matrix A;
generate assessment_number_matrix B;
identify the relevant rows of A and B;
        /* i.e. search for all nodes with at least 2 different
            assessment numbers for different open positions */
while exist relevant rows of A or B
     find minimum in the relevant parts of A and B;
     put the corresponding vertex at the corresponding position;
     update the corresponding A resp. B;
     if there was a change of vertices in one layer
          update B resp. A which corresponds to the other layer;
end_while;
position the remaining vertices;
```

An important feature of the approach presented here is that both layers are reordered at the same time in one step. This property and the use of assessment numbers are the main differences between our approach and the ’classical’ barycentering or median-ordering.

4. Example

Figure 1 shows the 11-crossings drawing of a bipartite graph and its matrix-representation M.



	d	e	f	g
a	0	0	1	1
b	1	1	1	1
c	0	1	1	0

Fig. 1: Initial drawing of G

The frequency matrix and the initial assessment number matrices are the following:

F	q_1	q_2	q_3	q_4
p_1	1	2	4	6
p_2	3	3	3	3
p_3	6	4	2	1

A	p_1	p_2	p_3
a	4,90	3,00	1,41
b	2,63	3,00	2,63
c	2,83	3,00	2,83

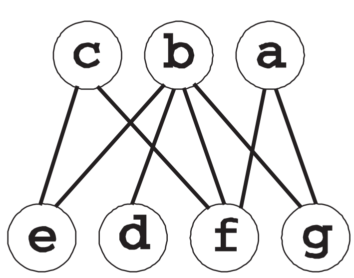
B	q_1	q_2	q_3	q_4
d	3,00	3,00	3,00	3,00
e	4,24	3,46	2,45	1,73
f	2,62	2,88	2,88	2,62
g	1,73	2,45	3,46	4,24

Note that row 'd' does not belong to the relevant part of B. The first vertex that is placed is 'a'. Its position is the last one in layer 1. After that row 'a' and column p_1 in A are no longer relevant and the rows 'e' and 'g' in B have to be updated. The resulting assessment number matrices are:

A	p_1	p_2	p_3
a	-	-	-
b	2,63	3,00	-
c	2,83	3,00	-

B	q_1	q_2	q_3	q_4
d	3,00	3,00	3,00	3,00
e	1,73	2,45	3,46	4,24
f	2,62	2,88	2,88	2,62
g	4,24	3,46	2,45	1,73

The next vertices that are placed are: 'e' on q_1, 'g' on q_4, 'c' on p_1, 'b' on p_2. Vertices 'd' and 'f' are not positioned in the while-loop because it does not matter on which open position they would be placed. All their assessment numbers in the prevailing relevant part of B are identical. The resulting graph representations (graphical and matrix) with 3 crossings are stated in figure 2.



	e	d	f	g
c	1	0	1	0
b	1	1	1	1
a	0	0	1	1

Fig. 2: Resulting drawing of G

5. Results

In order to be able to measure the effectiveness of crossing reduction WARFIELD [15] defines the generating matrix which orders all possible edge combinations for *d* vertices. The underlying problem can be stated for dimension *d* as to reduce the edge crossing of a bipartite graph with *d* vertices in the first layer and $(2^d\text{-}1)$ in the second. The edges are constructed so that the columns of the matrix represention correspond to the binary representation of the numbers $1..(2^d-1)$, i.e. all possible edge combinations when focusing *d* vertices. The vertices in the first layer are at fixed positions to preserve the columns as defined.

Differing from WARFIELD [15] and SUGIYAMA et al. [11] we calculate the overall edge crossing numbers of such a bipartite graph and do not make a pairwise comparison of the columns counting those pairs of vectors which are in the wrong order. We feel that the problem of crossing reduction needs a direct and not a derivied criteria for measuring heuristic effectiveness.

Our computational experiments showed that the resulting crossing numbers depend on the initial vertex ordering of the input graph. We modified SUGIYAMA's et al. [11] test scenario who calculated the crossings for only one initial vertex ordering. In case 1 we permutated the first *r* vertices in the second layer and randomized the ordering of the remaining ones. In case 2 we calculated the crossings of 50,000 randomized second layer orderings.

The stochastic algorithm was slightly modified to preserve the vertex ordering in the first layer. For all problems we calculated the number of crossings for barycentering and median-ordering, too. The results are the following:

dimension d $(= s)$	t	r	# test graphs	method	case 1			
					average k	minimal k	maximal k	std deviation
3	7	7	5040	stochastic	8	8	8	0
				barycenter	8	8	8	0
				median	8	8	8	0
4	15	8	40320	stochastic	95	95	95	0
				barycenter	95	95	95	0
				median	96,0	95	97	0,81

dimension d (= s)	t	r	# test graphs	method	case 1			
					average k	minimal k	maximal k	std deviation
5	31	8	40320	stochastic	756	756	756	0
				barycenter	758	758	758	0
				median	772,03	758	786	4,74
6	63	8	40320	stochastic	5004	5004	5004	0
				barycenter	5016,99	5006	5027	3,24
				median	5115,97	5052	5178	16,66
7	127	8	40320	stochastic	29841	29841	29841	0
				barycenter	29873,01	29852	29895	5,43
				median	30471,41	30253	30711	59,16

The following properties can be observed:

- Comparing the median-heuristic and barycentering the later is superior in terms of the average and worst case crossing numbers for $d > 3$. This fact holds for the best case if $d > 5$, too. So we concentrate on comparing our heuristic with the barycentering method.

- Barycentering is always worse than the stochastic heuristic for dimension $d > 4$.

- There is no difference in worst, average, and best case results for the stochastic heuristic. So it can be called stable.

The last fact is the most remarkable one. In order to evaluate the stability case 2 was constructed to have a broader selection over all possible input permutations of the second layer orderings. The results are:

dimension d (= s)	# test graphs	case 2 INPUT				case 2 OUTPUT
		average k	minimal k	maximal k	std deviation	avg. k = min. k = max. k
3	50000	20,99	8	34	5,00	8
4	50000	179,89	98	253	21,59	95
5	50000	1239,62	924	1577	83,85	756
6	50000	7560,28	6290	8838	311,73	5004
7	50000	44213,75	40380	48576	1021,69	29841

6. Conclusion

The presented new heuristic algorithm for DAG-layout has shown to be preferable over the common methods of barycentering or median-layout in terms of WARFIELD's generating matrices. The approach of laying out two layers simultaniously can be applied for n-layer DAGs by using it for the most complicated two layers and than laying out the other layers in the traditional way. A multiple up- and down-laying out would then become unnecessary. The quality of such DAG-layouts will be the scope of the ongoing work.

Acknowledgements

I gratefully acknowledge the helpful comments of Rudi Pendavingh and Ulrich Derigs.

References

[1] G. Di Battista, P. Eades, R. Tamassia, I. G. Tollis, "Algorithms for Drawing Graphs: an Annotated Bibliography", *Technical Report CS-89-09 (Revised Version),* Department of Computer Science, Brown University, Jun. 1994.

[2] M. J. Carpano, "Automatic Display of Hierarchized Graphs for Computer Aided Decision Analysis", *IEEE Trans. Syst. Man, Cybern.,* SMC-1 [Nov. 1980] 11, 705-715.

[3] P. Eades and D. Kelly, "Heuristics for Reducing Crossings in 2-Layered Networks", *Ars Combinatoria,* 21-A [1986], 89-98.

[4] P. Eades and N. C. Wormald, "The median heuristic for drawing 2-layers networks", *Technical Report 69,* Department of Computer Science, University of Queensland, 1986.

[5] P. Eades and N. C. Wormald, "Edge Crossings in Draings of Bipartite Graphs", *Technical Report 108,* Department of Computer Science, University of Queensland, 1991.

[6] S. E. Elmaghraby, *"Activity Networks: Project Planning and Control by Network Models",* New York: Wiley 1977.

[7] E. R. Gansner, E. Koutsofios, S. C. North, and K. P. Vo, "A Technique for Drawing Directed Graphs", *IEEE Trans. on Software Engineering,* 19 [Mar. 1993] 3, 214-229.

[8] E. R. Gansner, S. C. North, and K. P. Vo, "DAG - A Program that Draw Directed Graphs", *Software—Practice and Experience,* 18 [Nov. 1988] 11, 1047-1062.

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

[10] F. N. Paulisch, *The Design of an Exendible Graph Editor,* Berlin: Springer 1993.

[11] K. Sugiyama, S. Tagawa, and M. Toda, "Methods for Visual Understanding of Hierarchical System Structures", *IEEE Trans. Syst. Man, Cybern.,* SMC-11 [Feb. 1981] 2, 109-125.

[12] K. Sugiyama and M. Toda, "Structuring Information for Understanding of Complex Systems: A Basis for Decision Making", *FUJITSU Sci. Tech. J.,* 21 [Jun. 1985] 2, 144-166.

[13] R. Tamassia, G. Di Battista, and C. Batini, "Automatic Graph Drawing and Readability of Diagrams", *IEEE Trans. Syst. Man, Cybern.,* 18 [Jan./Feb. 1988] 1, 61-79.

[14] J. N. Warfield, *"Societal Systems",* New York: Wiley 1976.

[15] J. N. Warfield, "Crossing Theory and Hierarchy Mapping", *IEEE Trans. Syst. Man, Cybern.,* SMC-7 [Jul. 1977] 7, 505-523.