

Graph Drawing Contest Report

Christian A. Duncan¹, Carsten Gutwenger², Lev Nachmanson³,
and Georg Sander⁴

¹ Louisiana Tech University, Ruston, LA 71272, USA
duncan@latech.edu

² University of Dortmund, Germany
carsten.gutwenger@cs.uni-dortmund.de

³ Microsoft, USA
levnach@microsoft.com

⁴ IBM, Germany
georg.sander@de.ibm.com

Abstract. This report describes the 16th Annual Graph Drawing Contest, held in conjunction with the 2009 Graph Drawing Symposium in Chicago, USA. The purpose of the contest is to monitor and challenge the current state of graph-drawing technology.

1 Introduction

This year's Graph Drawing Contest had two topics: Partial Graph Drawing and the Graph Drawing Challenge. The partial graph drawing topic provided four data sets of different kinds of graphs, which were partially laid out. Some of the nodes had specified coordinates, and some of the edges had specified routing points. These nodes and edges were considered fixed. Other nodes and edges had no coordinates and had to be laid out. These nodes and edges were free to be moved. All nodes had specified sizes. The task was to find the nicest layout that integrates the free nodes and edges into the layout of the fixed nodes and edges without changing the positions of fixed nodes and edges. The data sets were a simple tree, an organization chart, a flow chart, and a mystery graph.

The Graph Drawing Challenge, which took place during the conference, focused on minimizing the number of crossings of upward grid drawings of graphs with edge bends. We received 27 submissions: 10 submissions for the Partial Graph Drawing topic and 17 submissions for the Graph Drawing Challenge.

2 Simple Tree

The simple tree had 56 nodes and 55 directed edges. 16 nodes were fixed. There were no special requirements concerning the edge shapes. The data was artificially created and the fixed nodes were roughly arranged in a top down manner. The challenge was to fit the free nodes into the dense area between the fixed nodes. We received 4 submissions for the tree data. Two submissions were based on a spring embedder approach, and two other submissions were based on a specialized tree layout algorithm.

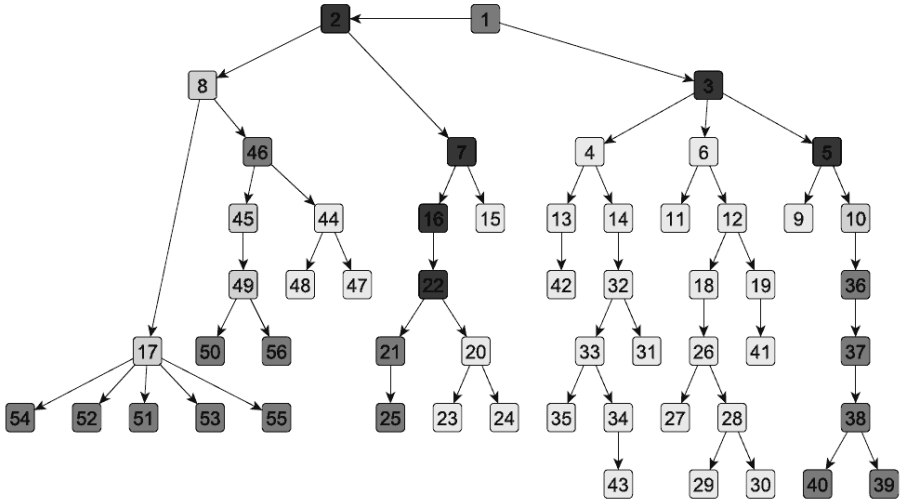


Fig. 1. First place, Simple Tree (original in color)

The winning submission by Melanie Badent and Michael Baur from the Universities of Konstanz and Karlsruhe (Figure 1) used a variation of the algorithm by Reingold and Tilford [5]. They determined the node spacing from the positions of the fixed nodes, so that the layout looks uniform. Then, they categorized the nodes into red nodes (in Figure 1 medium gray) that were fixed, orange nodes (in Figure 1 lighter gray) that have only fixed or orange nodes as successors, hence their position is determined by the layout principle as the center above their successors, blue nodes (in Figure 1 black) that have some red or orange nodes as successors and hence are partially constrained in their position, and green nodes (in Figure 1 white) that are completely free to be placed. For the green subtrees, they computed all possible permutations of successors to obtain the subtree with minimum width for the previously determined node spacing to fit into the space between the fixed nodes. For blue nodes, the ordering of their green child nodes was determined by a test routine that checked all insertion points between red and orange nodes. The algorithm was specially tailored for this input graph. They reported that for general trees a more sophisticated method would be advisable.

3 Organization Chart

The organization chart had 171 nodes and 170 directed edges. 101 nodes and 96 edges were fixed. The fixed nodes had a minimal spacing of approximately 40 units (border to border). The edges had to be laid out in an orthogonal fashion, which was only satisfied by one of the two submissions.

The winning submission came from Nicholas Jefferson from the University of Sydney (Figure 2; the dark nodes and edges were fixed). The layout was produced

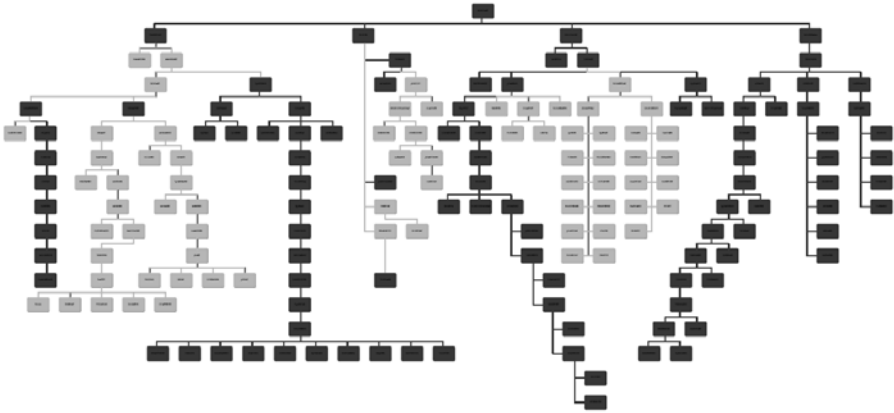


Fig. 2. First place, Organization Chart (original in color)

using a backtracking search algorithm that used a minimum-size layout for all descendants of a node whenever possible but otherwise recursed for each permutation of the children of the node, up to subtree isomorphism. The minimum-size layouts were generated using an approach similar to the Stockmeyer merge [3] and again the Reingold-Tilford algorithm [5]. The implementation used the programming language Ruby and the relational database PostgreSQL, using spatial indexes to detect occlusion and continuations and savepoints to restore state on backtracking.

4 Flow Chart

The flow chart had 57 nodes and 72 directed edges. 26 nodes were fixed and 24 edges were fixed. The graph had a single source and single target and was artificially created. The majority of the edges should point downwards. Orthogonal edges were preferred but not required.

The winning submission came from Hui Liu from the University of Sydney (Figure 3; the dark nodes and their connecting edges in the left picture were fixed). First, an automatic layout was produced using the Neato algorithm of GraphViz [1]. From this sketch, a drawing was created with Microsoft Visio that was manually laid out. Finally, a background picture was added showing a diamond pattern to highlight the fixed nodes (Figure 3, right).

5 Mystery Graph

The mystery graph had the 17 cities of the past Graph Drawing Conferences as nodes fixed on a circle. 374 city nodes were added to keep the graph connected. These nodes were free to be moved. The additional task was to determine the

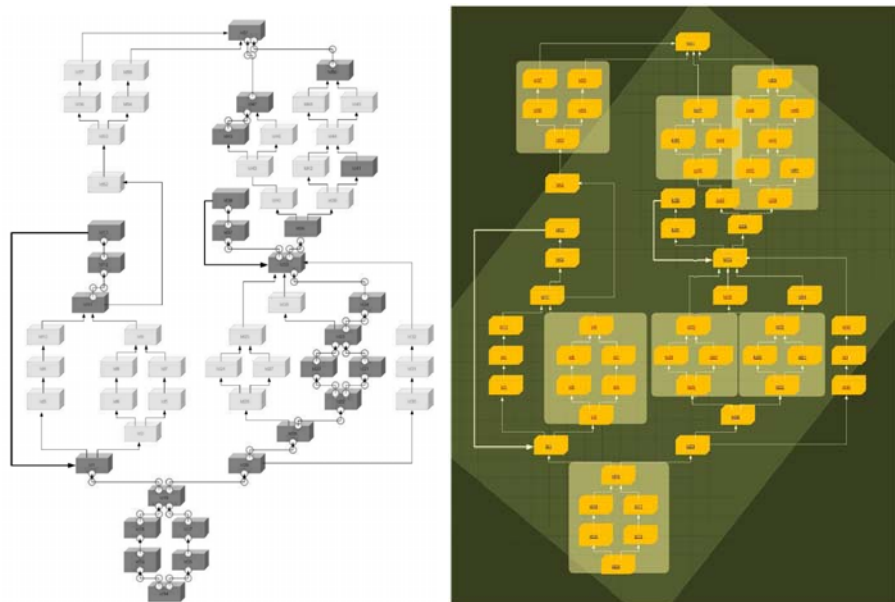


Fig. 3. First place, Flow Chart (original in color)

meaning of the 911 undirected edges, which depicted the twin towns or sister city relationships between the cities as collected from Wikipedia [7].

The winning team, again Melanie Badent and Michael Baur, found the correct answer. Since the fixed nodes formed a circle, they decided to arrange all nodes in circles as well. The fixed nodes formed the innermost circle, and the free nodes were placed at outer circles depending on their topological distance from fixed nodes. The team used a radial layout algorithm from the visone software project [4] and rotated the circles with free nodes to bring sister cities on different circles closer to each other.

6 Graph Drawing Challenge

Following the Graph Drawing Challenge tradition of having the same topic in two subsequent years, we repeated the same challenge as in year 2008, with new sample graphs however. The challenge, a subproblem of the popular layered layout technique by Sugiyama et al. [6] known to be NP-hard, dealt with minimizing the number of crossings of upward grid drawings of graphs allowing edge bends. This technique requires that all nodes be placed on grid positions, that nodes and edge bends don't overlap, and that all edge segments point strictly upwards. At the start of the one-hour on-site competition, the contestants were given six nonplanar directed acyclic graphs with a legal upward layout that however had a huge number of crossings. The goal was to rearrange the layout to reduce the

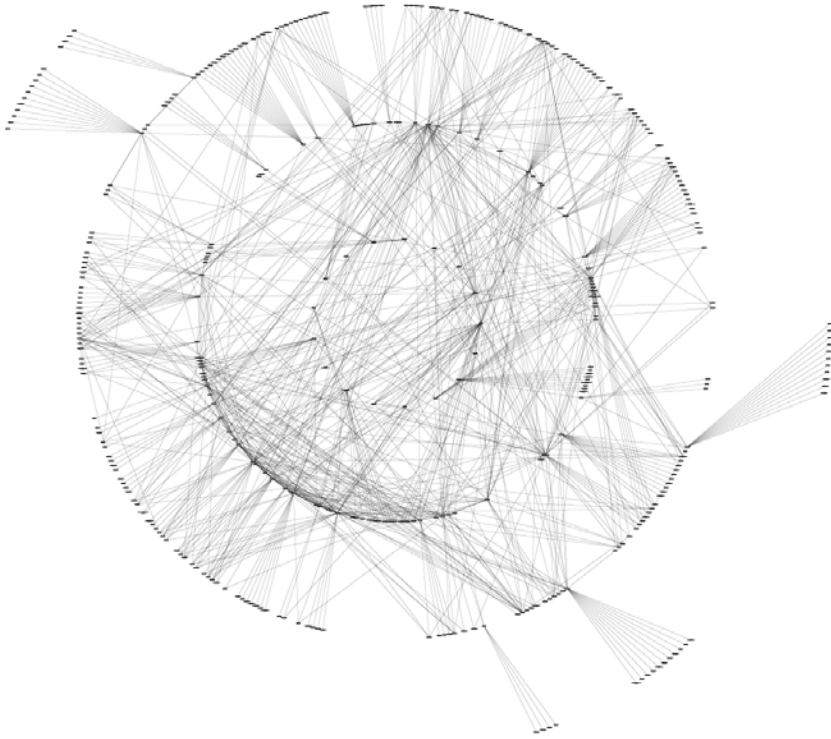


Fig. 4. First place, Sister city graph

number of crossings. Only the number of crossings was judged; other aesthetic criteria such as the number of edge bends or the area were ignored.

We partitioned the challenge into two subcategories: automated and manual. In the automated subcategory, contestants received graphs ranging in size from 26 nodes / 48 edges to 953 nodes / 1529 edges and were allowed to use their own sophisticated software tools with specialized algorithms. Only one team (TU Dortmund: Hoi-Ming Wong and Karsten Klein) submitted results and hence were the winner of this subcategory. They found the optimal solution of four of the graphs by using a modified version [2] of the same software as last year. For the largest graph, they found a solution with 57 crossings, which is not far from the optimal layout with 12 crossings.

The 16 manual teams solved the problems by hand using IBM's Simple Graph Editing Tool provided by the committee. They received graphs ranging in size from 25 nodes / 40 edges to 144 nodes / 365 edges. To determine the winner among the manual teams, the scores of each graph, determined by dividing the crossing number of the best submission by the crossing number of the current submission, were summed up. With a score of 4.87, the winner was J. Joseph Fowler from the University of Wisconsin-Milwaukee, who found the optimal result for the 3 smallest graphs and very good results for graphs 4 and 5. For the

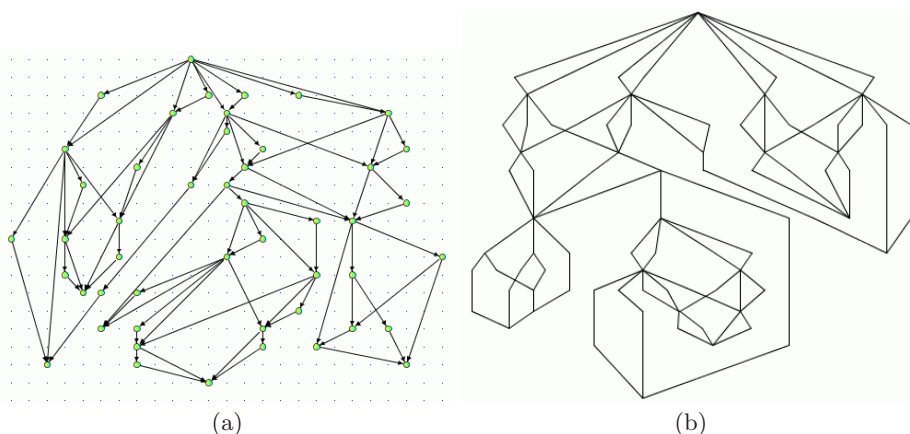


Fig. 5. Challenge graph with 48 nodes / 83 edges and 4 crossings: (a) the best manually obtained result by J. Joseph Fowler, (b) the best automated result by team Dortmund

largest graph in the manual category, which can be laid out optimally with only 4 crossings, no contestant found a good solution manually. The submissions for that graph ranged between 4287 and 5995 crossings. Figure 5 shows the results of a graph with 48 nodes and 83 edges. This graph was used both in the manual and automated subcategory and had 4 crossings in the best results, which is optimal.

Acknowledgments. We thank the generous sponsors of the symposium and all the contestants for their participation.

References

1. AT&T. Graphviz, <http://www.graphviz.org>
2. Chimani, M., Gutwenger, C., Mutzel, P., Wong, H.: Upward planarization layout. In: Eppstein, D., Gansner, E.R. (eds.) GD 2009. LNCS, vol. 5849, pp. 94–106. Springer, Heidelberg (2010)
3. Eades, P., Lin, T., Lin, X.: Minimum size h-v drawings. In: Advanced Visual Interfaces (Proc. AVI 1992). World Series in Computer Science, vol. 36, pp. 386–394 (1992)
4. University of Karlsruhe. visone, <http://visone.info>
5. Reingold, E., Tilford, J.: Tidier drawing of trees. IEEE Trans. on Software Engineering 7(2), 223–228 (1981)
6. Sugiyama, K., Tagawa, S., Toda, M.: Methods for visual understanding of hierarchical systems. IEEE Trans. Sys. Man, and Cyb. SMC11(2), 109–125 (1981)
7. Wikipedia. Sister cities, http://en.wikipedia.org/wiki/Sister_cities (accessed, May 2009)