Some Modifications of Sugiyama Approach

Danil E. Baburin

A.P. Ershov Institute of Informatics Systems,
Lavrentev pr. 6, Novosibirsk, 630090, Russia
`babdan@iis.nsk.su`

Abstract. We present a graph drawing algorithm that was developed as an extension of the hierarchical layout method of Sugiyama [4]. The main features of the algorithm are that the edges are orthogonal and that each edge has at most two bends.

1   Introduction

In a number of works [2,3] Sander presents his methods and heuristics on how to combine hierarchical approach of Sugiyama [4] with methods of orthogonal graph drawing and with Manhattan conventions. In our work we develop and extend these ideas. The main difference is that we have achieved an orthogonal layout with maximum number of edge bends equal to two compared to four bends in Sander's works. Also we solved the task of minimizing the number of orthogonal edge crossings between any pair of adjacent levels of hierarchy and tried a new heuristic to build an initial position of nodes before starting the crossing reduction step.

On the Fig. 1 we present a diagram layout constructed by our algorithm. Note that this is left to right drawing as opposed to conventional hierarchical layout. The algorithm was implemented by means of C++ and provided with an interface that enabled its use in Higres system [1].

2   Our Methods and Heuristics

The crossing reduction step considers the problem of drawing a layered digraph with a small number of edge crossings. We present a heuristic method that could be used to form an initial ordering with relatively small number of edge crossings. Then this ordering could be improved using any known method.

During the layer assignment step we are parsing the graph into a spanning tree plus extra edges. Our heuristic changes the order in which subtrees appear in this tree to minimize the number of crossings that are formed by non-spanning edges. Subtrees are ordered greedily according to the number of extra edges connecting them to the left and right.

In our algorithm we introduce a concept of edge width. Each edge has assigned width with some minimum value that depends on the size of edge label and a number of multi edges that were combined in the preprocessor step of the algorithm into one edge. Then while assigning vertical coordinates to vertices we operate with vertical ranges. These
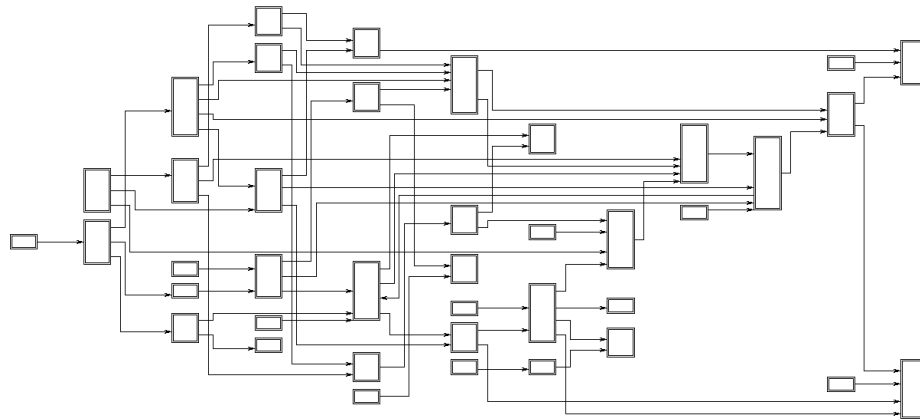
Fig. 1. Sample layout produced by our algorithm. The graph has 40 vertices and 60 edges.

ranges could be bounded by some coordinates or could dynamically depend on neighbor ranges.

When we are drawing edges between two adjacent levels of hierarchy we are dividing horizontal space between adjacent layers into vertical levels that would contain vertical segments of the edges. Then we are defining a notion of potentially crossing edges. This crossing could be eliminated if correspondent edges are drawn in different vertical segments and in a particular order. Our task was to eliminate all such crossings and to minimize the number of vertical levels being used. We are solving this problem by constructing a supplementary directed graph G'. Now our goal could be formulated in terms of a coloring problem for G'. Vertices of G' need to be colored in a minimum number of colors so that any adjusted vertices would have different colors. Also all directed edges should point from vertexes with smaller color indices to vertexes with larger color indices. Then we can draw edges in original graph by assigning vertical levels to colors in increasing order.

References

1. I. A. Lisitsyn, V.N. Kasyanov, "Higres – Visualization system for Clustered Graphs and Graph Algorithms", *Graph Drawing*, LNCS 1731, Czech Republic 1999, pp. 82-89
2. G. Sander, "A Fast Heuristic for Hierarchical Manhattan Layout", *Graph Drawing*, LNCS 1027, Germany 1995, pp. 447-458
3. G. Sander, "Graph layout through the VCG tool", *DIMACS Intern. Workshop, GD'94*. LNCS. 894, 1995, pp. 194–205.
4. K. Sugiyama, S. Tagawa, M. Toda, "Methods for Visual Understanding of Hierarchical systems", *IEEE Trans. Sys., Man, and Cybernetics*, SMC 11(2), 1981, pp. 109-125.