

Variable neighborhood descent for the incremental graph drawing

J. Sánchez-Oro ^{a,1} A. Martínez-Gavara ^{b,2} M. Laguna ^{c,3}
A. Duarte ^{a,4} R. Martí ^b

^a *Department of Computer Science, Universidad Rey Juan Carlos,
Móstoles, Spain*

^b *Departamento de Estadística e Investigación Operativa,
Universidad de Valencia, Madrid, Spain*

^c *Leeds School of Business, University of Colorado at Boulder, USA*

Abstract

Graphs are used to represent reality in several areas of knowledge. Drawings of graphs have many applications, from project scheduling to software diagrams. The main quality desired for drawings of graphs is readability, and crossing reduction is a fundamental aesthetic criterion for a good representation of a graph. In this paper we target the edge crossing reduction in the context of incremental graph drawing, in which we want to preserve the layout of a graph over successive drawings. We propose a hybrid method based on the GRASP (Greedy Randomized Adaptive Search Procedure) and VND (Variable Neighborhood Descent) methodologies and compare it with previous methods via simulation.

Keywords: Incremental graph drawing, variable neighborhood descent, metaheuristics

1 Introduction

Most of the information systems nowadays are commonly represented by a drawing, which makes the system easier to interpret and understand. Graphs are the basic modeling unit in a wide variety of areas, like project and production scheduling, line balancing, business plans or software visualization. For this reason, graph drawing has become an important research area, with a large number of publications related. We refer the reader to [2] for a thoroughly survey on graph drawing. The selection of an objective measure of the quality of a graph is a controversial subject. However, the number of crossing edges is a widely admitted criterion for evaluating the quality of a draw. Specifically, the fewer of crossings, the better the drawing is [1]. The problem of minimizing the number of crossings is $\mathcal{NP}$ -complete [5].

This paper focuses on finding the best drawing for hierarchical directed acyclic graphs, HDAG, which are usually known as hierarchical graphs, layered digraphs, or simply hierarchies. In order to represent a HDAG, we first need to draw the vertices in equally spaced vertical lines (layers), in such a way that every directed edge goes in the same direction. With this arrangement of vertices, arc crossing minimization consists of finding the appropriate ordering of the vertices in each layer. Fig. 1 shows a drawing of a HDAG with 8 vertices and 10 edges. The HDAG is split into three layers (highlighted in gray). The number of crossings between the first and second layer is 1, in the pair of edges $(3, 8) - (7, 2)$, while the number of crossings between the second and third layer is 3, in the pairs $(2, 1) - (8, 6)$, $(2, 5) - (8, 6)$, and $(2, 5) - (8, 1)$, respectively.

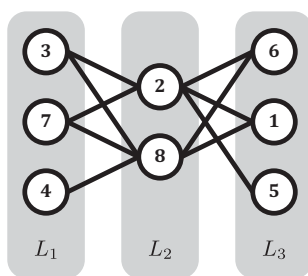


Fig. 1. Example of a drawing of a HDAG with 8 vertices and 10 edges.

Considering hierarchical digraphs is not a loss of generality since there are

¹ Email: jesus.sanchezoro@urjc.es

² Email: gavara@uv.es

³ Email: laguna@colorado.edu

⁴ Email: abraham.duarte@urjc.es

Download English Version:

<https://daneshyari.com/en/article/5777271>

Download Persian Version:

<https://daneshyari.com/article/5777271>

[Daneshyari.com](https://daneshyari.com)