8^{th} *International Conference on Applied Informatics*
Eger, Hungary, January 27–30, 2010.

An Experimental Study of Minimum Cost Flow Algorithms

Zoltán Király[a], **Péter Kovács**[b]

[a] Department of Computer Science and CNL
[b] Department of Algorithms and Their Applications and CNL
Eötvös Loránd University, Budapest
e-mail: kiraly@cs.elte.hu, kpeter@inf.elte.hu

Abstract

This paper presents an experimental study of efficient algorithms for the minimum cost flow problem. It is more comprehensive than earlier surveys both in terms of the range of considered implementations and the size of the test instances.

The minimum cost flow problem is to find a feasible flow of minimum total cost from a set of supply nodes to a set of demand nodes in a network with capacity constraints and arc costs. It is a fundamental model in network flow theory having countless applications in various fields, such as transportation, telecommunication, network design, resource planning and scheduling.

For solving this problem, the following algorithms were implemented: (1) *simple cycle canceling*, (2) *minimum mean cycle canceling*, (3) *cancel and tighten*, (4) *successive shortest path*, (5) *capacity scaling*, (6) *cost scaling* and (7) *network simplex*. All of them are generally known methods, our main contribution is their highly efficient implementations with some new heuristics and the comparative analysis of their performance in practice. Another novel result is the application of Goldberg's recent partial augment-relabel method in the cost scaling algorithm. Our implementations are available as part of the LEMON open source C++ optimization library (*http://lemon.cs.elte.hu*).

Validity and benchmark tests of these algorithms were performed on many kinds of large-scale random graphs (created with standard generators: NETGEN, GRIDGEN and GOTO), as well as on some real-life networks. The proposed implementations were also compared to widely known and tested solvers, namely, the corresponding method of the LEDA library and three efficient public codes: CS2, RelaxIV and MCF. Our implementations proved to be comparable and often superior to these popular solvers. In most cases, the network simplex and the cost scaling implementations were the most efficient. The latter one typically performs better on large sparse networks.

Keywords: graph, network, flow, algorithm, implementation, performance

MSC: 05C21