Classification of Network Optimization Software Packages

Angelo Sifaleras
University of Macedonia, Greece

INTRODUCTION

Combinatorial Optimization is an interdisciplinary branch of Optimization (Migdalas et al., 2013) combining several research fields, e.g., operational research, artificial intelligence, algorithmic theory, and computational complexity theory. The core of combinatorial optimization includes a class of problems that can be modeled using graphs and networks and it is known as Network Optimization (Du & Pardalos, 1993). It is noteworthy that advances in the research area of Network Optimization have a strong impact on supply chain companies (Tseng et al., 2005) and in society in general (Roberts, 1978). Applications of network optimization in supply chain engineering include a large variety not only of classic network location, transportation or routing problems but also combined problems of, e.g., location-routing problems. Also, to mention but a few of the graph theory applications with societal impact, finding an eulerian closed chain meets applications in street-sweeping, RNA chains, etc. Also, applications of graph coloring arise in scheduling committee meetings, planning final examinations schedules in a college, etc.

Nowadays, several network optimization problems such as the classic linear minimum cost network flow problem (Sifaleras, 2013), the transportation problem, or the linear sum assignment problem are easily solvable by polynomial time algorithms. However, a large number of other classic network optimization problems remain intractable (unless P = NP). Problems belonging to this category include the majority of the versions of the well-known Traveling Salesman Problem (TSP), the Vehicle Routing Problem (VRP), several other versions of routing problems, integrated inventory-distribution problems, and network location problems, etc. These problems often require heuristic-based approaches for efficiently solving them in reasonable computational time. For example, a courier service company must reschedule their routes in real time, according to new orders. In such problems, an exact algorithm requiring one hour to compute the optimal solution is usually less useful than an approximation algorithm requiring only a few seconds to find a solution that is 99% sub-optimal.

BACKGROUND

During the last decades several advances have emerged in the field of network and combinatorial optimization (Cook, 2010). Nowadays, optimization software packages are required for the efficient solution of complex network optimization problems, even of moderate dimensions, due to their computational difficulty. Briefly speaking, an optimization software package is a software package specifically designed to be used for optimization problems. Such network optimization software packages have various differences in their characteristics, technology, and scope (Maros & Khaliq, 2002). The most well-known types of optimization software packages include optimization solvers, problem generators, performance analyzers/profilers, and educational software packages using visualization/animation techniques. An optimization solver is an optimization software package including efficient implementations of optimization algorithms specifically designed for the solution of optimization problems. Furthermore, a network generator is an optimization software package designed for random generating instances of network optimization problems with specific structure and dimension. Moreover, modern performance analyzer/profilers are now available to analyze the efficiency of the source code of an optimization algorithm, identify the bottlenecks in its performance, and suggest ways of computational improvements.

DOI: 10.4018/978-1-4666-5888-2.ch695

Benchmark collections for mathematical programming problems are also very important for comparing the computational efficiency of network optimization solvers. Roughly speaking, a benchmark collection constitutes a set of computationally difficult problems, either from real-world applications (e.g., real data from railway networks for robust train timetabling problems) or randomly generated. Such sets of benchmark problem instances allow researchers to compare the efficiency of their optimization solvers in common, usually publicly available, problem sets.

Apart from these main types of network optimization software packages, there are also other more general software solutions such as Decision Support Systems (DSS). In this case, network optimization models are usually embedded as components of DSSs, for assisting the decision making process in network-related problems of a business company. Typical examples of this category include DSSs that are used in supply chain design for producing optimal transportation plans that minimize total cost and satisfy service constraints, or optimizing products flow either along transportation lanes, or through distribution centers.

NETWORK OPTIMIZATION SOFTWARE PACKAGES

Problem Generators

Several problem generators and benchmark collections have been proposed in the literature for almost any type of network optimization problem. However, in this section, our main focus is the presentation of publicly available network optimization software packages. A repository of source codes for various problem instance generators is maintained by the Zuse Institute Berlin (ZIB) research institute for applied mathematics and computer science and can be found online at http://www.zib.de/en/services/web-services/mp-testdata/generators.html.

Many random network generators exist for producing instances of the Minimum Cost Network Flow Problem (MCNFP). Such generators provide a large variety of choices regarding, for example, one-way or two-way arcs, the total number of nodes (either sink or source nodes), the total number of arcs, and either lower and/or upper bounds of arc capacities and costs. Klingman et al. (1974) presented the NETGEN generator that provides the user with the ability to produce random MCNFP, transportation, and assignment problem instances. Additionally, Arthur and Frendewey (1994) developed the RAND-NET generator, which produces optimal MCNFP instances with custom dimension and structure.

A number of instance generators for the TSP were proposed for the 8th Implementation Challenge organized by the Center for Discrete Math and Theoretical Computer Science (DIMACS). More details regarding the Challenge (e.g., the testbeds of instances and the methods for evaluating tour quality) are described in Johnson and McGeoch (2002). Instance generation codes and samples of randomly generated instances can be freely downloaded from http://www2.research.att.com/~dsj/chtsp/download.html. Also, Cirasella et al. (2001) developed twelve instance generators for the Asymmetric Traveling Salesmen Problem (ATSP) in C. Their paper analytically describes the results of an experimental comparison of modern heuristics for the ATSP.

Furthermore, Almoustafa, Hanafi, and Mladenović (2013) developed a generator used for producing test instances of the asymmetric distance vehicle routing problem. The complete source code of their generator in C++ is publicly available online at http://www.mi.sanu.ac.rs/~nenad/advrp. Moreover, a new generator of Dynamic VRP (DVRP) instances was developed by Khouadjia M.R. in C++ and is also publicly available at http://paradiseo.gforge.inria.fr/index.php?n=Benchmarks.VRPgenerator. This code provides users with the ability to tune the dynamic problem by changing lengths of routes, changing number of vehicles, and movement of the depot, etc. A paper describing the application of the Variable Neighborhood Search (VNS) metaheuristic framework in DVRP instances was recently published (Sarasola, et al., 2011).

Benchmark Collections

A large number of benchmark collections for various types of network optimization problems are nowadays publicly available. Therefore, several authors are reporting computational results using these benchmark collections in order to compare their algorithm imple-