European Journal of Operational Research 253 (2016) 265–279

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem

Dimitris C. Paraskevopoulos[a,*], Tolga Bektaş[b], Teodor Gabriel Crainic[c], Chris N. Potts[d]

[a] School of Management, University of Bath, Claverton Down, Bath BA2 7AY, UK
[b] Southampton Business School, Centre for Operational Research, Management Science and Information Systems (CORMSIS), University of Southampton, Southampton SO17 1BJ, UK
[c] Département Management et Technologie, École des Sciences de la Gestion, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), Université du Québec à Montréal, C.P. 8888, succ. Centre-ville, Montréal, QC H3C 3P8, Canada
[d] Mathematical Sciences, CORMSIS, University of Southampton, Southampton SO17 1BJ, UK

ARTICLE INFO

Article history:
Received 28 June 2014
Accepted 31 December 2015
Available online 11 January 2016

Keywords:
Multi-commodity network design
Scatter search
Evolutionary algorithms
Ejection chains
Iterated local search

ABSTRACT

This paper presents an evolutionary algorithm for the fixed-charge multicommodity network design problem (MCNDP), which concerns routing multiple commodities from origins to destinations by designing a network through selecting arcs, with an objective of minimizing the fixed costs of the selected arcs plus the variable costs of the flows on each arc. The proposed algorithm evolves a pool of solutions using principles of scatter search, interlinked with an iterated local search as an improvement method. New cycle-based neighborhood operators are presented which enable complete or partial re-routing of multiple commodities. An efficient perturbation strategy, inspired by ejection chains, is introduced to perform local compound cycle-based moves to explore different parts of the solution space. The algorithm also allows infeasible solutions violating arc capacities while performing the "ejection cycles", and subsequently restores feasibility by systematically applying correction moves. Computational experiments on benchmark MCNDP instances show that the proposed solution method consistently produces high-quality solutions in reasonable computational times.

1. Introduction

The fixed-charge capacitated multi-commodity network design problem (MCNDP) consists of designing a network on a given graph by selecting arcs to route a given set of commodities between origin-destination pairs. Each arc has a predefined capacity specifying the maximum flow that the arc can accommodate. Also, associated with each arc are fixed and variable costs, where the fixed cost is incurred only if the arc is selected, and the variable cost is a cost per unit of flow along the arc. Each commodity has an origin and a destination node and the amount to be transported. The objective is to minimize the total cost of establishing the arcs and routing the flows.

The MCNDP has attracted much attention in the literature due to both its complexity (the problem is NP-hard in the strong sense), and a wide variety of applications in the areas of telecommunications, logistics, production and transportation systems (Balakrishnan, Magnanti, & Mirchandani, 1997; Magnanti & Wong, 1986; Minoux, 1986). Despite the significant efforts devoted to the development of exact methodologies for the MCNDP (Crainic, Frangioni, & Gendron, 2001; Hewitt, Nemhauser, & Savelsbergh, 2010), the literature still favors heuristic approaches when large-scale problem instances are involved. One of the most successful local search strategies for the MCNDP is proposed by Ghamlouche, Crainic, and Gendreau (2003), where new cycle-based neighborhood operators are incorporated in a tabu search framework. The cycle-based operators are subsequently used within a path-relinking algorithm (Ghamlouche, Crainic, & Gendreau, 2004), a multilevel cooperative framework (Crainic, Li, & Toulouse, 2006), and a scatter search (SS) (Crainic & Gendreau, 2007). In the latter paper, the authors conclude that the proposed SS failed to meet their expectations and further research is needed to realize the full potential of SS.

Inspired and motivated by the advances in the heuristic approaches for the MCNDP, this paper contributes to the existing body of work by: (i) proposing an efficient iterated local search (ILS) that utilizes new and enhanced cycle-based neighborhood operators, long and short term memory structures, and an innovative perturbation strategy based on ejection chains (Glover, 1996) that

* Corresponding author. Tel.: +44 7923291297.
E-mail addresses: d.paraskevopoulos@bath.ac.uk (D.C. Paraskevopoulos), t.bektas@soton.ac.uk (T. Bektaş), TeodorGabriel.Crainic@cirrelt.ca (T.G. Crainic), c.n.potts@soton.ac.uk (C.N. Potts).

http://dx.doi.org/10.1016/j.ejor.2015.12.051
0377-2217/

aims at guiding the search towards unexplored regions of the solution space; (ii) introducing an efficient SS that considers the search history and "solvency-based" measures to produce offspring; and (iii) presenting results of computational experiments conducted on benchmark instances using an algorithm incorporating the various elements described above. The majority of the heuristics for the MCNDP utilize a trajectory-based or an evolutionary framework to select arcs for inclusion in the design, and subsequently call a commercial optimizer (e.g., CPLEX) to solve the corresponding flow subproblem. As the flow subproblems become larger, the solution time for repeatedly finding minimum cost flows might become significant, even though linear programming optimizers are relatively efficient. Towards this end, we call the linear programming (LP) solver as few times as possible in the proposed algorithm in order to reduce its computational requirements.

The remainder of this paper is organized as follows. Section 2 provides a brief review of the recent literature on the MCNDP. Section 3 presents our evolutionary algorithm and all of its components, namely the Initialization phase, the SS, and the ILS. In Section 4 we describe details of our computational experiments and we also present results of applying the proposed algorithm to benchmark MCNDP instances from the literature. Conclusions are given in Section 5, where research prospects are also provided.

2. Literature

A number of efficient algorithms have appeared in the literature to address the inherent complexity of solving the MCNDP. In this section, we provide a brief review of the available methods but focus on heuristic, as opposed to exact, solution algorithms for reasons stated earlier.

Crainic, Gendreau, and Farvolden (2000) propose a simplex-based tabu search method for the MCNDP using a path-flow based formulation of the problem. Their method combines column generation with pivot-like moves of single commodity flows to define the path flow variables. In a similar fashion, Ghamlouche et al. (2003) describe cycle-based neighborhoods for use in metaheuristics aimed at solving MCNDPs. The main idea of the cycle-based local moves is to redirect commodity flows around cycles in order to remove existing arcs from the network and replace them with new arcs. The authors use the proposed neighborhood structures in a tabu search algorithm, where a commodity flow subproblem is solved to optimality at each iteration.

Ghamlouche et al. (2004) propose an evolutionary algorithm for the MCNDP. Their solution framework is based on path relinking, in which cycle-based neighborhoods are used to generate an elite candidate set of solutions in a tabu search algorithm and for moving from the initial to the guiding solution. When updating the pool of solutions, the dissimilarity of solutions is considered as an additional component in calculating the solution value. Alvarez, González-Velarde, and De-Alba (2005) describe an SS algorithm for the MCNDP. The authors use GRASP, originally proposed by Feo and Resende (1995), to produce a diversified initial set of solutions. Each commodity path is subject to an improvement process. The solutions are combined by choosing the best path for each commodity among the solutions that are being combined. A feasibility restoration mechanism is also available for solutions that are infeasible. In contrast to the recombination process of Alvarez et al. (2005), our SS does not consider commodity paths to build a solution; instead, independent arcs are combined to create offspring. We believe that the latter enhances the SS algorithm's capabilities, as more combinations can occur when arcs instead of paths are combined together, leading to a rich pool of offspring.

A parallel cooperative strategy is described by Crainic and Gendreau (2002) using tabu search and various communication strategies. In a similar fashion, Crainic et al. (2006) propose a multilevel cooperative search on the basis of local interactions among cooperative searches and controlled information gathering and diffusion. The focus of their algorithm is on the specification of the problem instance solved at each level and the definition of the cooperation operators.

Katayama, Chen, and Kubo (2009) propose a column and row generation heuristic for solving the MCNDP. The authors relax the arcs' capacity constraints, while a column and row generation technique is developed to solve the relaxed problem. Using similar ideas, Yaghini, Rahbar, and Karimi (2013) present a hybrid simulated annealing (SA) and column generation (CG) algorithm for solving the MCNDP. The SA is used to define the open and closed arcs, wherein the flow subproblem is solved via CG.

A local branching technique for the MCNDP is proposed by Rodríguez-Martín and Salazar-González (2010). Even though the method, originally proposed by Fischetti and Lodi (2003), is exact by nature, high quality heuristic solutions can be produced using an MIP solver as a "black box". A solution framework that employs a combination of mathematical programming algorithms and heuristic search techniques is introduced by Hewitt et al. (2010). Their methodology uses very large neighborhood search in combination with an IP solver on an arc-based formulation of the MCNDP, and an LP relaxation of the path-based formulation using cuts discovered during the neighborhood search. A follow-up study by Hewitt, Nemhauser, and Savelsbergh (2012) introduces a generic branch-and-price guided algorithm for integer programs with an application to the MCNDP.

3. Solution methodology

In this section, we first present a formal definition of the problem including the notation that will be used in the rest of the paper and then describe in detail the components of the main algorithm.

3.1. Problem definition

The MCNDP is defined on a graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs. Each arc $(i, j) \in \mathcal{A}$ has an associated fixed cost f_{ij} that is incurred if it is selected for inclusion in the network, has a cost per unit of flow c_{ij}, and has a capacity u_{ij}. A set of commodities denoted by $\mathcal{P}$ is given, where each commodity has an origin, a destination, and a quantity to be shipped from origin to destination. Problems with more than one origin or destination per commodity can be modeled by splitting commodities (see Holmberg & Yuan, 2000).

The goal of the problem is to select a subset of arcs that are to be included in the final design of the network along with the commodity flows on these arcs, to minimize the total cost of the selected arcs and the flow distribution on the resulting network. For simplicity, we refer to the arcs that are included in the final design of the network as *open* arcs; otherwise, the arcs should be considered as *closed*. Binary variables y_{ij} are used, where $y_{ij} = 1$ if the arc $(i, j) \in \mathcal{A}$ is open, and $y_{ij} = 0$ otherwise. The flow on each arc $(i, j) \in \mathcal{A}$ that is used for shipping each commodity $p \in \mathcal{P}$ from its origin to its destination is denoted by x_{ij}^p. Conservation of flow constraints must be satisfied at each node, and there are capacity constraints of the form $\sum_{p \in \mathcal{P}} x_{ij}^p \le u_{ij}$ for each $(i, j) \in \mathcal{A}$. The cost $f(s)$ of a solution s that is defined by variables x_{ij}^p and y_{ij} for $(i, j) \in \mathcal{A}$ and $p \in \mathcal{P}$ is computed using

$$f(s) = \sum_{(i,j) \in \mathcal{A}} \sum_{p \in \mathcal{P}} c_{ij} x_{ij}^p + \sum_{(i,j) \in \mathcal{A}} f_{ij} y_{ij}. \quad (1)$$

We adopt the convention that $f(s) = \infty$ if solution s is infeasible.