

Yugoslav Journal of Operations Research
23 (2013), Number 1, 3-17
DOI: 10.2298/YJOR121120001S

MINIMUM COST NETWORK FLOWS: PROBLEMS, ALGORITHMS, AND SOFTWARE

Angelo SIFALERAS

*Department of Technology Management,
University of Macedonia, Economic and Social Sciences,
Loggou-Tourpali, Naoussa 59200, Greece*
sifalera@uom.gr

Invited review

Received: November 2012 / Accepted: January 2013

Abstract: We present a wide range of problems concerning minimum cost network flows, and give an overview of the classic linear single-commodity Minimum Cost Network Flow Problem (MCNFP) and some other closely related problems, either tractable or intractable. We also discuss state-of-the-art algorithmic approaches and recent advances in the solution methods for the MCNFP. Finally, optimization software packages for the MCNFP are presented.

Keywords: Mathematical Programming, Combinatorial Optimization, Optimization Software.

MSC: 65K05, 90C27, 90B10.

1. INTRODUCTION

Network Optimization [2], [25], [66], [72], [87] makes a large part of Combinatorial Optimization [57], [73], and present a model often used for a large number of real-world applications [63] in communications, informatics, transportation, construction projects [96], water resources management [50] and supply chain management [11], [12]. A wide category of Network Optimization problems constitute the MCNFP, and several other well-known optimization problems are special cases of MCNFP.

Let $G = (N, A)$ be a directed network with n nodes and m arcs, where N and A are the sets of nodes and arcs, respectively. Each arc $(i, j) \in A$ has a cost c_{ij} that denotes the unit shipping cost along the arc (i, j) . Each arc (i, j) is also associated with an amount x_{ij} of flow on the arc, a lower bound l_{ij} and an upper bound u_{ij} of the flow; thus $l_{ij} \leq x_{ij} \leq u_{ij}$. However, if $u_{ij} = +\infty$, then the MCNFP is called uncapacitated (also known as transshipment problem). We associate a number b_i with each node $i \in N$, which indicates its available amount of supply or demand. Node i will be called a source, sink or transshipment node, depending on whether $b_i > 0$, $b_i < 0$, or $b_i = 0$, respectively. This way, a plethora of real-world applications (e.g., in logistics) requiring the flow of various products from warehouses (supply nodes) to markets (demand nodes) through a number of transfer points (transshipment nodes) can be efficiently modeled. If $\sum_{i \in N} b_i = 0$, then the network G will be a balanced network. Thus, the single-commodity capacitated MCNFP can be stated formally as follows [2]:

$$\begin{aligned}
 \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 \text{s.t.} \quad & \sum_{k:(i,k) \in A} x_{ik} - \sum_{j:(j,i) \in A} x_{ji} = b_i, \quad \forall i \in N \\
 & l_{ij} \leq x_{ij} \leq u_{ij}, \quad \forall (i, j) \in A
 \end{aligned} \tag{MCNFP}$$

In the above formulation, constraints of type $\text{s.t.} \sum_{k:(i,k) \in A} x_{ik} - \sum_{j:(j,i) \in A} x_{ji} = b_i, \forall i \in N$ are known as the flow conservation equations, while constraints of type $l_{ij} \leq x_{ij} \leq u_{ij}$ are called the flow capacity constraints.

In matrix notation MCNFP can be formulated as a linear program of the form $\min c^T x : Ax = b, l \leq x \leq u$, where $A \in \mathbb{R}^{n \times m}$ is the $n \times m$ node-edge incidence matrix of the graph G and $c, x, l, u \in \mathbb{R}^m, b \in \mathbb{R}^n$. The complexity of MCNFP is determined by the type of cost function for each arc. In the above case with linear cost function, the MCNFP is solvable in strongly polynomial time. However, several other variants of MCNFPs consider a convex, concave, or generally nonlinear cost function.

Section 2 presents several special cases and generalizations of the MCNFP. Here, we also discuss some closely related, either tractable or intractable, problems. Section 3 is an overview of solution methods regarding the classic linear MCNFP; some recent advances are also given. Availability of MCNFP optimization solvers, instance generators, and educational web-based software are presented in Section 4. Finally, a short summary follows in Section 5.

2. MCNFP VARIANTS, SPECIAL CASES, AND GENERALIZATIONS

The importance of the MCNFP, apart from its applicability to various areas, stems from the fact that several other well-known problems constitute its special cases. Such examples are the Transportation Problem (TP), the Linear Sum Assignment Problem (LSAP), and the Shortest Path Problem (SPP).

The TP can be represented by a bipartite graph $G(S,D,A) = G(N,A)$, where S, D are two disjoint sets of nodes such that $|S| = n_S, |D| = n_D$ and $N = S \cup D$. Notations $|S|$ and $|D|$ stand for the cardinality number of the sets S and D , respectively. Here, the supply and demand nodes are denoted by $i \in S$ and $j \in D$, respectively. An arc $(i,j) \in A$ is directed from nodes of S to nodes of D . The mathematical formulation of the TP with an $n_S \times n_D$ cost matrix c , an $n_S \times 1$ supply vector b_S , and an $n_D \times 1$ demand vector b_D such that $\sum_{i \in S} b_{S_i} = \sum_{j \in D} b_{D_j}$ is as follows:

$$\begin{aligned}
 & \min \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 & \text{s.t.} \sum_{(i,j) \in A} x_{ij} = b_{S_i}, \quad \forall i \in S \\
 & \quad \sum_{(i,j) \in A} x_{ij} = b_{D_j}, \quad \forall j \in D \\
 & \quad x_{ij} \geq 0, \quad \forall i \in S, \quad \forall j \in D
 \end{aligned} \tag{TP}$$

The TP is a special case of the MCNFP, where the $n \times 1$ (supply or demand) vector b has been partitioned to $b = \{b_S, b_D\}$.

Furthermore, the LSAP [18] can also be represented by a bipartite graph $G(S,D,A) = G(N,A)$, where S, D are two disjoint sets of nodes such that $|S| = |D|$ and $N = S \cup D$. The only difference between LSAP and TP is that now we have $b_S = b_D = 1$, $\forall i \in S, j \in D$, and binary decision variables. The mathematical formulation of the LSAP is as follows:

$$\begin{aligned}
 & \min \sum_{(i,j) \in A} c_{ij} x_{ij} \\
 & \text{s.t.} \sum_{(i,j) \in A} x_{ij} = 1, \quad \forall i \in S \\
 & \quad \sum_{(i,j) \in A} x_{ij} = 1, \quad \forall j \in D \\
 & \quad x_{ij} \in \{0,1\}, \quad \forall i \in S, \quad \forall j \in D
 \end{aligned} \tag{LSAP}$$

The LSAP is a special case of the TP and consequently of the MCNFP, where the values of the $n \times 1$ (supply or demand) vector b are now restricted only to 1 or -1 , for the supply or demand nodes, respectively.

Moreover, the SPP is also a special case of the MCNFP, where the objective is to find the minimum distance between two given nodes (e.g., nodes s and t). Here, the values of the $n \times 1$ vector b are now restricted only to 0, 1 or -1 . The mathematical formulation of the SPP is as follows:

$$\begin{aligned} \min \quad & \sum_{(i,j) \in A} c_{ij} x_{ij} \\ \text{s.t.} \quad & \sum_{k:(i,k) \in A} x_{ik} - \sum_{j:(j,i) \in A} x_{ji} = \begin{cases} 1, & \text{if } i = s \\ -1, & \text{if } i = t \\ 0, & \text{else} \end{cases} \\ & x_{ij} = 0, 1, \quad \forall (i, j) \in A \end{aligned} \quad (\text{SPP})$$

The node-edge incidence matrix of the graph of a MCNFP and all its special cases (i.e., TP, LSAP, SPP) have the combinatorial property of total unimodularity. Therefore, an efficient integer solution can be easily found by solving the corresponding relaxation linear programming problem, provided b and u are integer-valued.

Although, the MCNFP and the above special cases can be efficiently solved in polynomial time, some generalizations of the MCNFP are intractable and thus cannot be efficiently solved. Such a problem is to find an integer flow for the minimum cost multi-commodity flow problem, which is known to be NP-complete [30]. Furthermore, the time-varying MCNFP [19][65] (also known as dynamic flows or flows over time) has also been proved to be NP-hard. In this generalized version of the static MCNFP, the cost, transit time and capacity of an arc vary by time. Thus, more decision variables are required for the representation of the waiting times at all vertices along each route. This type of problem has several variations: i) no flow is allowed to wait at any vertex (zero waiting times); ii) waiting at any vertex is not subject to any constraints (arbitrary waiting times); or iii) a flow can wait at a vertex (bounded waiting times).

A large number of real-world applications can be modeled by using minimum cost network flows with multiple objectives. A recent review of exact and approximation algorithms for both the continuous and integer case of multiple-objective MCNFPs has been presented by Hamacher et al. in [49]. The biobjective MCNFP is a special, well-studied case of multiple objective MCNFPs, with either continuous [81] or integer [82] flow values. Recent algorithmic approaches for the solution of biobjective MCNFP variants also include the papers [26] and [83]. Moreover, both multiple objectives and multiple hierarchies MCNFPs, in cases with arcs having fuzzy costs and capacities, have been investigated by Shih & Lee [84]. This generalization is very interesting because fuzzy set theory, probability methods, and interval computations (i.e., MCNFP with interval costs [23][51]) are well suited for modeling uncertainty in real-world applications. Recent work on Fuzzy MCNFP includes papers by Ghatee & Hashemi [40][41], and Ghatee et al. [42].

Additionally, attention of several researchers has also been attracted by nonlinear extensions of the classic MCNFP since linear costs are not always realistic.

Thus, if we consider a concave cost function for each arc, then this problem is called the concave minimum cost network flow problem. The concave MCNFP, with a concave cost function $c'_{ij}(x_{ij})$ for each arc, can be stated formally as follows:

$$\begin{aligned} & \text{global min } \sum_{(i,j) \in A} c'_{ij}(x_{ij}) \\ & \text{s.t. } \sum_{k:(i,k) \in A} x_{ik} - \sum_{j:(j,i) \in A} x_{ji} = b_i, \quad \forall i \in N \\ & \quad l_{ij} \leq x_{ij} \leq u_{ij}, \quad \forall (i,j) \in A \end{aligned} \quad (\text{concave MCNFP})$$

Thus, local optimum (minimum) concave MCNFP solutions are not necessarily global optimum (minimum) solutions. Although this problem is NP-hard even for single source uncapacitated MCNFP with fixed-charge arc costs [47], some simpler cases of the problem allow only a few arc costs to be concave and the remaining, that are linear, are solvable by strongly polynomial algorithms [90]. Recently, Fontes et al. presented a dynamic programming approach [32] in order to obtain an optimal solution to the single-source uncapacitated MCNFPs with general concave costs, independent of the type of cost functions and the number of nonlinear arc costs considered. The same authors also presented a Branch-and-Bound (BB) method [33] with better computational performance capable for solving larger size problems. Several applications include the concavity property of the objective function, such as production based models with start-up costs.

Furthermore, if the shipping cost over an arc is a convex, rather than a linear function of the number of units shipped along that arc, then this problem is called the minimum convex-cost network flow problem. Such problems arise naturally due to factors such as system congestion and queuing effects, (e.g., urban traffic networks, communication networks). In cases where the problem has a piecewise linear convex-cost objective function, a transformation into a classic MCNFP is possible. However, a significant drawback is that the number of arcs of the network considerably increased [2]. The convex separable integer minimum cost network flow problem is solvable in polynomial time [64]. Recently, Végé presented the first strongly polynomial algorithm for separable quadratic minimum-cost flows [92].

Another equivalent problem is the Minimum Cost Circulation Problem, where all supply and demand values are set to zero. Furthermore, if the flow of the MCNFP is not conserved (i.e., have attenuations or augmentations), then this problem is called the Generalized MCNFP. In this latter case, a gain factor (positive flow multiplier) is associated with each arc. Applications of the Generalized MCNFP may include flow of energy in thermal power plants, cash flows in different currencies, etc. Wayne [95] proposed the first polynomial combinatorial algorithm for the Generalized MCNFP. Specifically, this algorithm directly manipulates the underlying network and actually solves the equivalent Generalized Minimum Cost Circulation Problem.

In their recent paper, Vaidyanathan & Ahuja [91] also considered some specially structured MCNFPs that were previously unstudied. Such a special structure, for example, is when nodes lie on a circle (or line in general), the flow is allowed in both directions, and the costs of flow between a pair of nodes in the clockwise and the counterclockwise directions are different. Specifically, they presented new, fast

algorithms based on successive shortest-path algorithm that exploit the special structure of the problem.

Moreover, the cost (capacity) inverse MCNFP seeks to modify the cost (capacity) vector as little as possible to make a given feasible flow form a minimum cost flow of the network. A recent paper of Jiang et al. [53] measured the modification of the cost of the arcs by the weighted Hamming distance. Güler & Hamacher [48] analyzed the capacity inverse MCNFP for rectilinear (L_1) and Chebyshev (L_∞) norms.

The MCNFP is closely related to several other network flow problems. For example, the MCNFP is a special case of the submodular flow problem introduced by Edmonds & Giles [27]. If a linear fractional objective function is used, then the problem is called the linear fractional MCNFP. Recently, Xu et al. presented a new algorithm for the linear fractional MCNFP in [97]. Zhu et al. [98] showed that the MCNFP, where some or all arcs have variable, rather than fixed, lower bounds (also known as MCNF-VLB) is NP-hard. Finally, Krumke & Thielen [58] considered a variant of the MCNFP where the flow on each arc in the network is restricted to be either zero or above a given lower bound. This variant is known as the MCNFP with minimum quantities. Krumke & Thielen in their paper showed that the MCNFP with minimum quantities is strongly NP-complete.

3. SOLUTION ALGORITHMS FOR THE MCNFP

Computational algorithms for finding solution to network flow problems are of great practical significance. The first polynomial time algorithm for MCNFP was developed by Edmonds & Karp [28]. They showed how to transform the out-of-kilter method into a polynomially bounded method by iterative scaling of the right-hand side data. Tardos [86] proposed the first strongly polynomial algorithm for MCNFP. The existence of a strongly polynomial algorithm distinguishes MCNFP from the general linear programming problem.

Since then, the operations research community has developed a variety of algorithms and data structures for solving MCNFP. A large number of different polynomial time algorithms for MCNFP exist. However, the classical network Simplex algorithm remains the best choice for solving MCNFP. Network Simplex algorithms compute basic solutions for the MCNFP that can be represented by spanning trees [13]. Furthermore, Cunningham [22] proposed the use of strongly feasible trees, a method to ensure finiteness in the Network Primal Simplex Algorithm (NPSA). Other well-known algorithmic methods for the solution of the MCNFP are the cycle-cancelling algorithm, the out-of-kilter-algorithm, and the successive shortest path algorithm.

Interior Point Methods have also been proposed and applied to solve large-scale network flow problems [78]. A survey on Interior Point Methods for network flow problems can be found in [77]. Most state-of-the-art solution algorithms for the MCNFP use sophisticated data structures such as dynamic tree data structure [85] or Fibonacci heaps [35], elaborate storage schemes (for network Simplex type algorithms) such as the eXtended Threaded Index (XTI) method [9] that are also based on efficient scaling techniques.

The capacity-scaling algorithm of Edmonds & Karp in 1972 was the first scaling algorithm [28] for the solution of the MCNFP in polynomial time. Since then, several variants of scaling techniques have been proposed in the literature. Gabow & Tarjan

(1989) presented faster scaling algorithms for the TP, AP, and the MCNFP [36]. Ahuja & Orlin (1992) presented a scaling network Simplex algorithm [3]. Their algorithm can be regarded as a scaling version of Dantzig's primal Simplex pivot rule. Ahuja et al. (1992) combined several scaling methods such as capacity-scaling approach, excess-scaling approach, cost-scaling approach, and dynamic tree data structure in [1]. Goldfarb and Jin (1999) proposed an excess scaling algorithm in [44].

The first polynomial time specialization of NPSA for MCNFP using cost-scaling techniques was proposed by Orlin (1997) [68]. The running time of that algorithm is $O(\min\{n^2m \log nC, n^2m^2 \log n\})$, where n , m , and C denote the number of nodes, arcs, and maximum absolute arc cost if arc costs are integer, and ∞ otherwise, respectively. Also, Orlin in [68] gave a low degree bound of $O(nm \log n)$ on the diameter of the network polytope. Currently, the fastest strongly polynomial time algorithms for the capacitated MCNFP are the algorithms by Orlin [67] and Vygen [94]. Orlin's algorithm [67] is a variation of Edmonds & Karp scaling technique that runs in $O(m \log n (m + n \log n))$ and reduces the capacitated MCNFP to a sequence of $O(m \log n)$ shortest path problems. Also, Vygen [94] presented a dual algorithm that achieves the same running time as Orlin [67], but working directly with the capacitated MCNFP rather than transforming it to an uncapacitated MCNFP as in Orlin [67].

Recently, exterior point Simplex-type algorithms for the solution of the uncapacitated MCNFP have also been developed. This type of algorithm can cross over the infeasible region of the primal (dual) problem and find optimal solution reducing the number of iterations needed. The main idea of exterior point simplex-type algorithms is to compute two paths/flows. Primal (dual) exterior point simplex-type algorithms compute one path/flow which is basic but not always primal (dual) feasible; and the other is primal (dual) feasible but not always basic. A Network Primal Exterior Point Simplex type Algorithm (NEPSA) for the MCNFP was presented in [71]. Furthermore, a preliminary geometrical interpretation of a Dual Network Exterior Point Simplex type Algorithm (DNEPSA) for the MCNFP was described in [38]. The mathematical proof of correctness of DNEPSA, a detailed comparative computational study of DNEPSA and the classic Dual Network Simplex Algorithm (DNSA) on sparse and dense random problem instances, a statistical analysis of the experimental results, and finally some new results on the empirical complexity of DNEPSA, were recently published in [39]. These computational results have shown that DNEPSA is about 1.22 times faster than the classic DNSA in terms of the number of iterations, and about 1.57 times faster in terms of the CPU time. This computational study was based on several randomly generated MCNFP instances with varying density ranging from sparse problems (2%) to dense problems (40%), varying number of nodes from 200 to 700, and varying number of arcs from approximately 800 to 195,000. This algorithmic approach has been applied not only to other classic network optimization problems [70], but also to the general linear programming problem [79].

Gopalakrishnan et al. [45] have recently proposed a least-squares minimum-cost network flow algorithm. The authors take advantage of the special least-squares properties that network flow problems possess in order to address the problem of degeneracy in networks. Moreover, other new algorithmic approaches for the MCNFP include the Belief Propagation (BP) algorithm. BP is a general purpose distributed heuristic commonly used in Artificial Intelligence, which can be implemented for a wide range of constrained optimization problems. Gamarnik et al. [37] proved that the BP

solves the capacitated MCNFP exactly in pseudo-polynomial time when the optimal solution is unique.

Finally, exploiting available massively parallel environments has significant computational benefits [7]. Thus, efforts have also been made for designing efficient parallelization of MCNFP algorithms. Orlin & Stein designed parallel scaling algorithms for the TP and MCNFP in [69]. Thulasiraman et al. presented a parallel algorithm for the dual transshipment problem in [89]. A parallel implementation of NPSA on a shared-memory multiprocessor was reported by Peters [74] and also later by Barr & Hickman in [10]. A few years later, Beraldi et al. [14] proposed efficient parallel implementations of the auction/sequential shortest path and the e-relaxation algorithms for solving the linear MCNFP. The same authors also presented parallel algorithms for the solution of the MCNFP with convex separable cost function in [15].

4. OPTIMIZATION SOFTWARE FOR THE MCNFP

Since the MCNFP presents a linear programming problem, it can be efficiently solved by well-known optimization solvers, such as the Gurobi Optimizer¹ [17] or IBM ILOG CPLEX Optimizer² [52]. However, several other state-of-the-art implementations exist that exploit the network structure of the MCNFP and have publicly available source code for download.

Such an implementation is the MCF solver, which is an efficient implementation of NPSA developed by Löbel [61]. Among other well-known MCNFP solvers is the RELAX-IV written by Bertsekas & Tseng. Their routine implements the relaxation method described in [16]. It is noteworthy that there exists a NEOS Server interface [31] to RELAX-IV that will accept inputs in various formats³. Furthermore, Portugal et al. [75] recently published the description of their implemented Fortran subroutines for the solution of the MCNFP using the interior point network flow algorithm PDNET. They report that, for several classes of problems, PDNET has been shown to be faster than modern commercial implementations of the NPSA. Moreover, Goldberg & Cherkassky developed the CS2 solver, which is based on a cost-scaling push-relabel method [43]. Other well-known implementations of MCNFP solvers are the RNET [46], the NPSA code by Kennington & Helgason NETFLO [54], and the NET_SIMPLEX⁴ C++ implementation by Jensen & Berthelsen.

A recent extensive experimental evaluation of the above state-of-the-art algorithms, namely the CS2, RELAX-IV, and MCF, was presented by Király & Kovács in [55]. Also, Frangioni and Manca [34] have compared the performances of four different efficient implementations of algorithms for the MCNFP under cost reoptimization, in the context of decomposition algorithms for the multicommodity MCNFP.

¹ Available at: <http://www.gurobi.com/products/gurobi-optimizer>.

² Available at: <http://www.ibm.com/software/integration/optimization/cplex-optimizer>.

³ Available at: <http://www.neos-server.org/neos/solvers/Ino:RELAX4/DIMACS.html>.

⁴ Available at: http://plato.la.asu.edu/ftp/other_software/net_simplex_binaries. Available as binaries provided by H. Mittelmann.

Apart from the general purpose, commercial or not, linear programming solvers or the computer codes from independent researchers, one can solve the MCNFP by calling optimization libraries such as LEDA [62], LEMON [24], or using other optimization software packages such as the SAS/OR software and the legacy NETFLOW procedure [80]. LEDA stands for Library for Efficient Data types and Algorithms and is a C++ software library that contains a collection of robust and efficient implementations of algorithms and data structures for combinatorial and geometric computing. LEDA provides the `MIN_COST_FLOW()` function, based on a capacity scaling and successive shortest path computation that can be used to compute a minimum cost flow in a directed graph.

The Library for Efficient Modeling and Optimization in Networks (LEMON) is a C++ template library that provides efficient implementations of algorithms and common data structures by focusing on network optimization. LEMON is an open source project, maintained by the Egerváry Research Group on Combinatorial Optimization, at the Operations Research Department of the Eötvös Loránd University, Budapest, Hungary. Specifically, LEMON provides users with the possibility to solve the MCNFP by using implementations of not only the NPSA with various pivot strategies, but also the capacity scaling algorithm based on the successive shortest path method, the cost scaling algorithm based on push/augment and relabel operations, and using cycle-canceling algorithms, two of which are strongly polynomial. The computer codes of all the previously mentioned optimization solvers for the MCNFP are publicly available, and are presented in Table 1.

Table 1: Publicly available network optimization computer codes for the MCNFP, (accessed on Nov. 3, 2012)

Solver	URL
CS2	http://www.igsystems.com/cs2
MCF	http://typo.zib.de/opt-long_projects/Software/Mcf
DIMACS (solvers & generators)	ftp://dimacs.rutgers.edu/pub/netflow
LEMON Graph Library	http://lemon.cs.elte.hu
PDNET	http://www.research.att.com/~mgcr/pdnet
RELAX-IV	http://web.mit.edu/dimitrib/www/RELAX4.txt

To facilitate the exchange of problem generators and/or algorithm implementations, standard problem definitions and input/output formats have been proposed in the literature. However, the majority of the MCNFP solvers take input in the well-known DIMACS format [20], which is widely used after the first international algorithm implementation challenge at the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) in 1991.

Each new implementation is usually thoroughly tested using either benchmark problems that might have arisen naturally as real problems, or randomly generated problem instances. Regarding the MCNFP, several well-known random problem generators exist. Such generators allow the researchers to produce MCNFP instances with one-way or two-way arcs, a varying number of nodes (either source or sink nodes), varying graph density, and custom lower and upper bounds for uniform distribution of arc costs and/or arc capacities.

Lee & Orlin developed the GRIDGEN [60] generator in C. Their generator can either read the input parameters from the standard input or from a batch file in order to generate multiple sets of data at a time. Moreover, Goldberg has developed the MESH generator in C [20], which produces instances of the minimum-cost circulation problem in the DIMACS format. Also, Klingman et al. [56] developed the NETGEN generator in Fortran that produces not only random, capacitated or uncapacitated MCNFP instances, but also transportation and assignment problems. The RAND-NET generator was presented by Arthur & Friendewey [6] in order to provide users with the ability to create problems with controlled size and structure, and with known solutions.

A number of educational optimization software packages also exist for the MCNFP. For students, teaching solution algorithms for the MCNFP sometimes seems difficult to be grasped because they need to generate a sequence of rooted trees. The scope of such tools [4] is not the solution of large scale instances, but rather a step-by-step visualization [59] of solution algorithms for the MCNFP in order to enable the OR instructors to explain each iteration of the algorithm visually and with minimal effort. Vanderbei developed a network Simplex pivot tool⁵ that can be used for solving the MCNFP. Recently, Baloukas et al. [8] presented an animated demonstration⁶ of the classic NPSA for the uncapacitated MCNFP. Andreou et al. [5] also developed visualization software⁷ of the NEPSA. These educational optimization software packages implemented as Java applets are freely available and highly interactive, and can be accessed through the Web. Moreover, they have a number of helpful features, such as using colored eligible arcs, showing the solution process through textual information and depicting the relevant steps in pseudo code using multiple views.

Finally, a network optimization model may constitute a part of a model base used by a Decision Support System (DSS). Although a plethora of DSS applications exist in the literature [29], [93], several DSSs embed various modifications of minimum cost network flow models. For example, Rakshit et al. [76] described a DSS that used a minimum cost network flow model in order to identify and solve flight crew shortages. Also, Thibault [88] presented a DSS that contained a MCNFP variant with additional performance and survivability requirements for solving a telecommunications network design optimization problem.

5. SUMMARY

The classic linear MCNFP and a number of some closely related problems have been presented. Emphasis was also given on state-of-the-art solution techniques and sources of optimization software for the MCNFP. Although the classic linear MCNFP can be efficiently solved in strongly polynomial time by algorithms that exploit the network structure of the problem, there are several other intractable generalizations (e.g., time-varying MCNFP). Therefore, further research effort is required, using various approaches such as approximation algorithms or metaheuristics, in order to tackle such cases.

⁵ Available at: <http://www.princeton.edu/~rvdb/JAVA/network/nettool/netsimp.html>.

⁶ Available at: <http://users.uom.gr/~thanasis/NetworkSimplex.html>.

⁷ Available at: <http://users.uom.gr/~sifalera/ORJ>.

We believe that effort made to include the most recent and relevant literature on MCNFPs in this article will provide a starting point for studying the MCNFP variants, its algorithms and applications and make readers interested in these problems.

ACKNOWLEDGMENTS

This research has been partly funded by the Research Committee of the University of Macedonia, Economic and Social Sciences, Greece, under grant 80749 for the advance of Basic Research.

REFERENCES

- [1] Ahuja, R.K., Goldberg, A.V., Orlin, J.B., and Tarjan, R.E., “Finding minimum – cost flows by double scaling”, *Mathematical Programming*, 53 (3) (1992) 243–266.
- [2] Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., *Network Flows: Theory, Algorithms and Applications*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- [3] Ahuja, R.K., and Orlin, J.B., “The scaling network simplex algorithm”, *Operations Research*, 40 (S1) (1992) 5–13.
- [4] Andreou, D., Paparrizos, K., Samaras, N., and Sifaleras, A., “Application of a new network – enabled solver for the assignment problem in computer – aided education”, *Journal of Computer Science*, 1 (1) (2005) 19–23.
- [5] Andreou, D., Paparrizos, K., Samaras, N., and Sifaleras, A., “Visualization of the network exterior primal simplex algorithm for the minimum cost network flow problem”, *Operational Research*, 7 (3) (2007) 449–464.
- [6] Arthur, J.L., and Friendewey, J.O., “An algorithm for generating minimum cost network flow problems with specific structure and known optimal solutions”, *Networks*, 24 (8) (1994) 445–454.
- [7] Badr, E.S., Moussa, M., Paparrizos, K., Samaras, N., and Sifaleras, A., “Some computational results on MPI parallel implementation of dense simplex method”, in: *Proc. of World Academy of Science, Engineering and Technology*, 23 (2006) 39–42.
- [8] Baloukas, Th., Paparrizos, K., and Sifaleras, A., “An animated demonstration of the uncapacitated network simplex algorithm”, *INFORMS Transactions on Education*, 10 (1) (2009) 34–40.
- [9] Barr, R.S., Glover, F., and Klingman, D., “Enhancements to spanning tree labeling procedures for network optimization”, *INFOR*, 17 (1979) 16–34.
- [10] Barr, R.S., and Hickman, B.L., “Parallel simplex for large pure network problems: computational testing and sources of speedup”, *Operations Research*, 42 (1) (1994) 65–80.
- [11] Basten, R.J.I., Heijden, M.C., and Schutten, J.M.J., “A minimum cost flow model for level of repair analysis”, *International Journal of Production Economics*, 133 (1) (2011) 233–242.
- [12] Basten, R.J.I., Heijden, M.C., and Schutten, J.M.J., “Practical extensions to a minimum cost flow model for level of repair analysis”, *European Journal of Operational Research*, 211 (2) (2011) 333–342.
- [13] Bazaraa, M.S., Jarvis, J.J., and Sherali, H.D., *Linear Programming and Network Flows*, 4th Ed., John Wiley and Sons, Inc, 2009.
- [14] Beraldi, P., Guerriero, F., and Musmanno, R., “Efficient parallel algorithms for the minimum cost flow problem”, *Journal of Optimization Theory and Applications*, 95 (3) (1997) 501–530.
- [15] Beraldi, P., Guerriero, F., and Musmanno, R., “Parallel algorithms for solving the convex minimum cost flow problem”, *Computational Optimization and Applications*, 18 (2) (2001) 175–190.

- [16] Bertsekas, D.P., and Tseng, P., “RELAX-IV: A faster version of the RELAX code for Solving minimum cost flow problems”, *Technical Report P-2276*, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 1994.
- [17] Bixby, R.E., “Mixed-integer programming: It works better than you may think. Gurobi optimization”, *FERC Conference*, Washington, DC, June 9, 2010.
- [18] Burkard, R., Dell'Amico, M., and Martello, S., *Assignment Problems – Revised Reprint*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2012.
- [19] Cai, X., Sha, D., and Wong, C.K., “Time-varying minimum cost flow problems”, *European Journal of Operational Research*, 131 (2) (2001) 352–374.
- [20] Center for Discrete Mathematics and Theoretical Computer Science (DIMACS). “The first DIMACS international algorithm implementation challenge: Problem definitions and specifications”, *Technical report*, DIMACS, New Brunswick, NJ, 1991.
- [21] Center for Discrete Mathematics and Theoretical Computer Science (DIMACS), “The first DIMACS international algorithm implementation challenge: The benchmark experiments”, *Technical report*, DIMACS, New Brunswick, NJ, 1991.
- [22] Cunningham, W.H., “A network simplex method”, *Mathematical Programming*, 11 (1) (1976) 105–116.
- [23] Daneshpajouh, H., and Alimomeni, M., “A note on the minimum interval cost flow problem”, *Applied Mathematics and Computation*, 217 (17) (2011) 7051–7052.
- [24] Dezső, B., Jüttner, A., and Kovács, P., “LEMON – an open source C++ graph template library”, *Electronic Notes in Theoretical Computer Science*, 264 (5) (2011) 23–45.
- [25] Du, D.Z., and Pardalos, P.M., (Eds.), *Network Optimization Problems: Algorithms, Applications and Complexity*, In: Series on Applied Mathematics, 2, World Scientific, Singapore, 1993.
- [26] Ebrahim, R.M., and Razmi, J., “A hybrid meta heuristic algorithm for bi-objective minimum cost flow (BMCF) problem”, *Advances in Engineering Software*, 40 (10) (2009) 1056–1062.
- [27] Edmonds, J., and Giles, R., “A min-max relation for submodular functions on graphs”, *Annals of Discrete Mathematics*, 1 (1977) 185–204.
- [28] Edmonds, J., and Karp, R.M., “Theoretical improvements in algorithmic efficiency for network flow problems”, *Journal of the ACM*, 19 (2) (1972) 248–264.
- [29] Eom, S., and Kim, E., “A survey of decision support system applications (1995–2001)”, *Journal of the Operational Research Society*, 57 (11) (2006) 1264–1278.
- [30] Even, S., Itai, A., and Shamir, A., “On the complexity of timetable and multicommodity flow Problems”, *SIAM Journal on Computing*, 5 (4) (1976) 691–703.
- [31] Ferris, M.C., Mesnier, M.P., and Moré, J.J., “NEOS and Condor: solving optimization problems over the Internet”, *ACM Transactions on Mathematical Software*, 26 (1) (2000) 1–18.
- [32] Fontes, D.B., Hadjiconstantinou, E., and Christofides, N., “A dynamic programming approach for solving single-source uncapacitated concave minimum cost network flow problems”, *European Journal of Operational Research*, 174 (2) (2006) 1205–1219.
- [33] Fontes, D.B., Hadjiconstantinou, E., and Christofides, N., “A branch-and-bound algorithm for concave network flow problems”, *Journal of Global Optimization*, 34 (1) (2006) 127–155.
- [34] Frangioni, A., and Manca, A., “A computational study of cost reoptimization for min-cost Flow Problems”, *INFORMS Journal on Computing*, 18 (1) (2006) 61–70.
- [35] Fredman, M.L., and Tarjan, R.E., “Fibonacci heaps and their uses in improved network optimization algorithms”, *Journal of the ACM*, 34 (3) (1987) 596–615.
- [36] Gabow, H.N., and Tarjan, R.E., “Faster scaling algorithms for network problems”, *SIAM Journal on Computing*, 18 (5) (1989) 1013–1036.
- [37] Gamarnik, D., Shah, D., and Wei, Y., “Belief propagation for min-cost network flow: convergence and correctness”, *Operations Research*, 60 (2) (2012) 410–428.

- [38] Geranis, G., Paparrizos, K., and Sifaleras, A., “A dual exterior point simplex type algorithm for the minimum cost network flow problem”, *Yugoslav Journal of Operations Research*, 19 (1) (2009) 157–170.
- [39] Geranis, G., Paparrizos, K., and Sifaleras, A., “On a dual network exterior point simplex type algorithm and its computational behavior”, *RAIRO - Operations Research*, 46 (3) (2012) 211–234.
- [40] Ghatee, M., and Hashemi, S.M., “Generalized minimal cost flow problem in fuzzy nature: An application in bus network planning problem”, *Applied Mathematical Modelling*, 32 (12) (2008) 2490–2508.
- [41] Ghatee, M., and Hashemi, S.M., “Application of fuzzy minimum cost flow problems to network design under uncertainty”, *Fuzzy Sets and Systems*, 160 (22) (2009) 3263–3289.
- [42] Ghatee, M., Hashemi, S.M., Zarepisheh, M., and Khorram, E., “Preemptive priority-based algorithms for fuzzy minimal cost flow problem: An application in hazardous materials transportation”, *Computers & Industrial Engineering*, 57 (1) (2009) 341–354.
- [43] Goldberg, A.V., “An efficient implementation of a scaling minimum-cost flow algorithm”, *Journal of Algorithms*, 22 (1) (1997) 1–29.
- [44] Goldfarb, D., and Jin, Z., “A new scaling algorithm for the minimum cost network flow problem”, *Operations Research Letters*, 25 (5) (1999) 205–211.
- [45] Gopalakrishnan, B., Kong, S., Barnes, E., Johnson, E.L., and Sokol, J.S., “A least-squares minimum-cost network flow algorithm”, *Annals of Operations Research*, 186 (1) (2011) 119–140.
- [46] Grigoriadis, M., “An efficient implementation of the network simplex method”, *Mathematical Programming Studies*, 26 (1984) 83–111.
- [47] Guisewite, G.M., and Pardalos, P.M., “Minimum concave-cost network flow problems: applications, complexity, and algorithms”, *Annals of Operations Research* 25 (1) (1991) 75–100.
- [48] Güler, Ç., and Hamacher, H.W., “Capacity inverse minimum cost flow problem”, *Journal of Combinatorial Optimization*, 19 (1) (2010) 43–59.
- [49] Hamacher, H.W., Pedersen, C.R., and Ruzika, S., “Multiple objective minimum cost flow problems: A review”, *European Journal of Operational Research*, 176 (3) (2007) 1404–1422.
- [50] Haro, D., Paredes, J., Solera, A., and Andreu, J., “A model for solving the optimal water allocation problem in river basins with network flow programming when introducing non-linearities”, *Water Resources Management*, 26 (14) (2012) 4059–4071.
- [51] Hashemi, S.M., Ghatee, M., and Nasrabadi, E., “Combinatorial algorithms for the minimum interval cost flow problem”, *Applied Mathematics and Computation*, 175 (2) (2006) 1200–1216.
- [52] International Business Machines Corporation (IBM), “V12.1 User’s Manual for CPLEX”, 2009.
- [53] Jiang, Y., Liu, L., Wu, B., and Yao, E., “Inverse minimum cost flow problems under the weighted Hamming distance”, *European Journal of Operational Research*, 207 (1) (2010) 50–54.
- [54] Kennington, J., and Helgason, R., *Algorithms for Network Programming*, Wiley, New York, 1980.
- [55] Király, Z., and Kovács, P., “Efficient implementations of minimum-cost flow algorithms”, *Acta Universitatis Sapientiae, Informatica*, 4 (1) (2012) 67–118.
- [56] Klingman, D., Napier, A., and Stutz, J., “NETGEN: A program for generating large scale capacitated assignment, transportation, and minimum cost flow networks”, *Management Science*, 20 (5) (1974) 814–821.
- [57] Korte, B., and Vygen, J., *Combinatorial Optimization, Theory and Algorithms*, (5th ed.). In Series: Algorithms and Combinatorics, 21, Springer-Verlag, Bonn, 2012.
- [58] Krumke, S.O., and Thielen, C., “Minimum cost flows with minimum quantities”, *Information Processing Letters*, 111 (11) (2011) 533–537.

- [59] Lazaridis, V., Samaras, N., and Sifaleras, A., “An empirical study on factors influencing the effectiveness of algorithm visualization”, accepted for publication in *Computer Applications in Engineering Education*, (2010).
- [60] Lee, Y., and Orlin, J.B., “Computational testing of a network simplex algorithm”, in: *Proc. of the 1st DIMACS International Algorithm Implementation Challenge*, (1991).
- [61] Löbel, A., “Solving large-scale real-world minimum-cost flow problems by a network simplex method”, *Technical report SC 96-7*, Konrad-Zuse-Zentrum für Informationstechnik, Berlin, Germany, 1996.
- [62] Mehlhorn, K., and Näher, S., *LEDA: A Platform for Combinatorial and Geometric Computing*, Cambridge University Press, Boston 1999.
- [63] Migdalas, A., Sifaleras, A., Georgiadis, C.K., Papathanasiou, J., and Stiakakis, E., eds., *Optimization Theory, Decision Making, and Operations Research Applications*. In Series: Springer Proceedings in Mathematics & Statistics, 31, Springer, New York, USA, 2013.
- [64] Minoux, M., “Solving integer minimum cost flows with separable convex cost objective polynomially”, *Mathematical Programming Studies*, 26 (1986) 237–239.
- [65] Nasrabadi, E., and Hashemi, S.M., “Minimum cost time-varying network flow problems”, *Optimization Methods and Software*, 25 (3) (2010) 429–447.
- [66] Oliveira, C.A.S, and Pardalos, P.M., *Mathematical Aspects of Network Routing Optimization*, 53. In Series: Springer Optimization and Its Applications, Springer, New York, USA, 2011.
- [67] Orlin, J.B., “A faster strongly polynomial minimum cost flow algorithm”, *Operations Research*, 41 (2) (1993) 338–350.
- [68] Orlin, J.B., “A polynomial time primal network simplex algorithm for minimum cost flows”, *Mathematical Programming*, 78 (2) (1997) 109–129.
- [69] Orlin, J.B., and Stein, C., “Parallel algorithms for the assignment and minimum-cost flow problems”, *Operations Research Letters*, 14 (4) (1993) 181–186.
- [70] Papamanthou, Ch., Paparrizos, K., Samaras, N., and Sifaleras, A., “On the initialization methods of an exterior point algorithm for the assignment problem”, *International Journal of Computer Mathematics*, 87 (8) (2010) 1831–1846.
- [71] Paparrizos, K., Samaras, N., and Sifaleras, A., “An exterior Simplex type algorithm for the minimum cost network flow problem”, *Computers & Operations Research*, 36 (4) (2009) 1176–1190.
- [72] Pardalos, P.M., Hearn, D.W., and Hager, W.W., eds., *Network Optimization*, Lecture Notes in Economics and Mathematical Systems, 450, Springer, Germany, 1997.
- [73] Pardalos, P.M., Du, D.Z., and Graham, R.L., eds., *Handbook of Combinatorial Optimization*, 2nd ed., 2013.
- [74] Peters, J., “The network simplex method on a multiprocessor”, *Networks*, 20 (7) (1990) 845–859.
- [75] Portugal, L., Resende, M., Veiga, G., Patrício, J., and Júdice, J., “Fortran subroutines for network flow optimization using an interior point algorithm”, *Pesquisa Operacional*, 28 (2) (2008) 243–261.
- [76] Rakshit, A., Krishnamurthy, N., and Yu, G., “System operations advisor: A real-time decision support system for managing airline operations at united airlines”, *Interfaces*, 26 (2) (1996) 50–58.
- [77] Resende, M.G.C., and Pardalos, P.M., “Interior point algorithms for network flow problems”, in J.E. Beasley, ed., *Advances in linear and integer programming*, Oxford University Press, (1996) 147–187.
- [78] Resende, M.G.C., and Veiga, G., “An efficient implementation of a network interior point method”, in: D.S. Johnson and C.C. McGeoch, eds., *Network flows and matching: First DIMACS implementation challenge*, 12, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Providence, Rhode Island, (1993) 299–348.

- [79] Samaras, N., Sifaleras, A., and Triantafyllidis, C., “A primal – dual exterior point algorithm for linear programming problems”, *Yugoslav Journal of Operations Research*, 19 (1) (2009) 123–132.
- [80] SAS Institute Inc., “SAS/OR ® 9.22 User’s Guide: Mathematical Programming”, Cary, NC, (2010).
- [81] Sedeño-Noda, A., and González-Martín, C., “The biobjective minimum cost flow problem”, *European Journal of Operational Research*, 124 (3) (2000) 591–600.
- [82] Sedeño-Noda, A., and González-Martín, C., “An algorithm for the biobjective integer minimum cost flow problem”, *Computers & Operations Research*, 28 (2) (2001) 139–156.
- [83] Sedeño-Noda, A., González-Martín, C., and Gutiérrez, J., “The biobjective undirected two-commodity minimum cost flow problem”, *European Journal of Operational Research*, 164 (1) (2005) 89–103.
- [84] Shih, H.-S., and Lee, E.S., “Fuzzy multi-level minimum cost flow problems”, *Fuzzy Sets and Systems*, 107 (2) (1999) 159–176.
- [85] Sleator, D.D., and Tarjan, R.E., “A data structure for dynamic trees”, *Journal of Computer and System Sciences*, 26 (3) (1983) 362–391.
- [86] Tardos, E., “A strongly polynomial minimum cost circulation algorithm”, *Combinatorica*, 5 (3) (1985) 247–255.
- [87] Thai, M.T., and Pardalos, P.M., (Eds.), *Handbook of Optimization in Complex Networks*, 57–58. In Series: Springer Optimization and Its Applications, New York, USA, 2012.
- [88] Thibault, E., “A decision support system for the design of cost-effective metropolitan area networks”, Thesis Report, University of Ottawa, 1999.
- [89] Thulasiraman, K., Chalasani, R.P., and Comeau, M.A., “Parallel network dual simplex method on a shared memory multiprocessor”, in: *Proceedings of the 5th IEEE Symposium on Parallel and Distributed Processing*, Dallas, TX, 1993, 408–415.
- [90] Tuy, H., Ghannadana, S., Migdalas, A., and Värbrand, P., “Strongly polynomial algorithm for two special minimum concave cost network flow problems”, *Optimization: A Journal of Mathematical Programming and Operations Research*, 32 (1) (1995) 23–43.
- [91] Vaidyanathan, B., and Ahuja, R.K., “Fast algorithms for specially structured minimum cost flow problems with applications”, *Operations Research*, 58 (6) (2010) 1681–1696.
- [92] Végh, L.A., “Strongly polynomial algorithm for a class of minimum-cost flow problems with separable convex objectives”, in: *Proceedings of the 44th symposium on Theory of Computing (STOC ’12)*, ACM, New York, USA, 2012, 27–40.
- [93] Voudouris, K., Polemio, M., Kazakis, N., and Sifaleras, A., “An agricultural decision support system for optimal land use regarding groundwater vulnerability”, *International Journal of Information Systems and Social Change*, 1 (4) (2010) 66–79.
- [94] Vygen, J., “On dual minimum cost flow algorithms”, *Mathematical Methods of Operations Research*, 56 (1) (2002) 101–126.
- [95] Wayne, K.D., “A Polynomial combinatorial algorithm for generalized minimum cost flow”, *Mathematics of Operations Research*, 27 (3) (2002) 445–459.
- [96] Xu, J., Tu, Y., and Zeng, Z., “A nonlinear multiobjective bilevel model for minimum cost network flow problem in a large-scale construction project”, *Mathematical Problems in Engineering*, (2012), Article ID 463976.
- [97] Xu, C., Xu, X.-M., and Wang, H.-F., “The fractional minimal cost flow problem on network”, *Optimization Letters*, 5 (2) (2011) 307–317.
- [98] Zhu, X., Yuan, Q., Garcia-Diaz, A., and Dong, L., “Minimal-cost network flow problems with variable lower bounds on arc flows”, *Computers & Operations Research*, 38 (8) (2011) 1210–1218.