Journal of Combinatorial Optimization
https://doi.org/10.1007/s10878-021-00823-0

A novel approach to subgraph selection with multiple weights on arcs

Mohammad Ali Raayatpanah[1] · Salman Khodayifar[2] · Thomas Weise[3] · Panos Pardalos[4,5]

Accepted: 5 October 2021

Abstract

In this paper, an extension of the minimum cost flow problem is considered in which multiple incommensurate weights are associated with each arc. In the minimum cost flow problem, flow is sent over the arcs of a graph from source nodes to sink nodes. The goal is to select a subgraph with minimum associated costs for routing the flow. The problem is tractable when a single weight is given on each arc. However, in many real-world applications, several weights are needed to describe the features of arcs, including transit cost, arrival time, delay, profit, security, reliability, deterioration, and safety. In this case, finding an optimal solution becomes difficult. We propose a heuristic algorithm for this purpose. First, we compute the relative efficiency of the arcs by using data envelopment analysis techniques. We then determine a subgraph with efficient arcs using a linear programming model, where the objective function is based on the relative efficiency of the arcs. The flow obtained satisfies the arc capacity con-

✉ Mohammad Ali Raayatpanah
raayatpanah@khu.ac.ir

Salman Khodayifar
s.khodayifar@iasbs.ac.ir

Thomas Weise
tweise@hfuu.edu.cn

Panos Pardalos
pardalos@ise.ufl.edu

1 Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran

2 Department of Mathematics, Institute for Advanced Studies in Basic Sciences (IASBS), 45137-66731 Zanjan, Iran

3 Institute of Applied Optimization, School of Artificial Intelligence and Big Data, Hefei University, Jinxiu Dadao 99, Hefei 230601, Anhui, China

4 Center for Applied Optimization, Department of Industrial and Systems Engineering, University of Florida, Gainesville, FL, USA

5 LATNA, Higher School of Economics, Irkutsk, Russia

Published online: 13 November 2021

straints and the integrality property. Our proposed algorithm has polynomial runtime and is evaluated in rigorous experiments.

Keywords Network optimization · Data envelopment analysis · Efficiency · Linear programming · Combinatorial optimization

1 Introduction

Many problems from real-world domains such as computer networks, water delivery systems, electric power systems, telephone lines, highways, and communication networks can be formulated based on network flow models or include these models as important subproblems (Ahuja et al. 1995). In these problems, we often wish to ship flow (vehicles, messages, electricity, or water) from source nodes to sink nodes through the arcs of a network at minimum cost. Network flow problems, e.g., the shortest path, assignment, transportation, maximum flow, and the minimum cost flow problem, all possess useful theoretical properties. By exploiting the structure of network flow models, specialized algorithms with acceptable runtime requirements even on large-scale problem instances have been extended to solve such models, which are at least 100–300 times faster than conventional linear programming algorithms that ignore the network structure (Ahuja et al. 1993; Bazaraa et al 2011).

For a standard network flow problem, usually a single weight per arc is considered in the problem formulation in which case the problem can be solved by polynomial-time algorithms. In many real applications, multiple weights that may conflict with each other are needed on the arcs of a network. These weights can usually be divided in two sets: one set is to be minimized and the other to be maximized. Measuring the aggregated performance of such a problem is complex and also difficult (Deb 2001). As example, real-time services in communication networks have requirements on packet losses, delays, and delay jitters compared with traditional network services. For this kind of networks, decision makers are interested in finding paths with several different goals as the measurable indicators of path selection. One way to deal with this problem is to solve it with multi-criteria approaches (Ehrgott 2005; Marler and Arora 2004; Hamacher et al. 2007). A solution to a multi-criteria problem is said to be Pareto optimal if it is not dominated by any other solution. Typically, there is not just one single Pareto optimal solution for such problems and decision-makers are faced with a set of solutions. This is often considered as a nuisance, resulting in the necessity to use some preferences to differentiate between solutions. Another way to deal with this problem is that the most important weight is considered as coefficient of the objective function and the other weights are addressed by side constraints. However, this approach can destroy the network structure of the network flow problem. Therefore, a solution approach should be developed for dealing with the difficulty while retaining the network structure and that also returns only one solution.

We therefore utilize data envelopment analysis (DEA) techniques in order to determine the relative efficiency of the arcs in the network. In the following, the arcs can be proposed as a decision making unit by exchanging their weights to be minimized and maximized with inputs and outputs, respectively. Indeed, DEA can be viewed as a multiple-criteria evaluation methodology where arcs are alternatives and the DEA inputs and outputs are two sets of performance criteria. As mentioned by Cooper et al. (2006), “One reason is that DEA has opened up possibilities for use in cases which have been resistant to other approaches because of the complex (often unknown) nature of the relations between the multiple inputs and multiple outputs involved in many of these activities.”[1] We then present a linear program satisfying the integrality property in order to determine the subgraph, where the objective function is based on the relative efficiency of arcs. We show that the proposed method obtains a subgraph in the presence of multiple weights on arcs in a polynomial time.

The rest of this paper is organized as follows: in Sect. 2, the related literature is reviewed. Our proposed approach is explained in Sect. 3. Numerical results are used to demonstrate its applicability in Sect. 4. Finally, in Sect. 5, we finish with some concluding remarks.

2 Related work

This section can roughly be divided into two parts. First, there are related works on network flow problems with side constraints, and second there are works related on DEA.

Many problems from real-world domains can be viewed as a problem of minimizing the transportation cost of shipping materials from source nodes to sink nodes through a network. These kinds of problems are referred as network flow problems. Common application areas are communication networks, electric networks, transportation networks, facilities location, resource management, financial planning, and others (Ahuja et al. 1995; Li et al. 2013; Deo 1974). The minimum cost flow problem, which is modeled as a special type of linear programming problem, is one of the most fundamental network flow problems. Many other network flow problems can be formulated as a minimum cost flow problem (Ahuja et al. 1995; Bazaraa et al 2011). This problem has the property of integrality of its basic solutions, meaning that every basic solution is integer-valued (Ahuja et al. 1993; Du and Pardalos 1993; Bazaraa et al 2011) and fast polynomial-time algorithms have been proposed for solving it (Orlin 1993, 1997; Sokkalingam et al. 2000). Comprehensive surveys of solution procedures for this special case are given by Ahuja et al. (1993), Sokkalingam et al. (2000), Kovács (2015) and Hu et al. (2020).

However, in most real-life applications multiple weights on the arcs of a network need to be incorporated. Some of these weights represent costs and some others profits (Vogiatzis and Pardalos 2013). This generalized problem version with multiple weights per arc allows for considering the specific communication needs of service users, the dynamic characteristics of networks, and other requirements (Yang and Zhao 2014; Raayatpanah et al. 2014; Raayatpanah 2017a, b).

There are several similar important classical problems related to this problem such as the constrained shortest path problem, the constrained transportation problem, the constrained assignment problem, and the constrained maximum flow problem – all of

[1] This sentence is a direct quote from Cooper et al. (2006).

which have important applications in practice such as in logistics, telecommunications and computer networks.

We can deal with network problems in this case by adding one or more constraints to consider these weights. This kind of constraints are called side constraints. A constrained shortest path problem is a special case of side-constrained network problems, which is NP-complete (Dürr et al. 2015; Guo 2016; Sedeño-Noda and Alonso-Rodríguez 2015). We may sometimes attempt to solve side-constrained network problems using pure network problems in which side constraints are replaced with a number of flow balance equations and a few additional arcs and nodes (Glover et al. 1974; Klingman 1977). The majority of research focusses on three approaches for handling network problems with such side constraints. The first method is a specialized version of the simplex algorithm which exploits the structure of the underlying network problems (Chen and Saigal 1977; Glover and Klingman 1981, 1985; McBride 1985; McBride and Mamer 1997; Mamer and McBride 2000; McBride and Mamer 2001). This approach has found several applications (Spälti and Liebling 1991; Fang and Qi 2003; Mo et al. 2005; Lu et al. 2006; Venkateshan et al. 2008). The second method is a straightforward dual method, in which the primal infeasibility of the side constraint is reduced successively. Finally, such problems can be solved by Lagrangian relaxation of the side constraints (Belling-Seib et al. 1988; Bryson 1991; Mathies and Mevert 1998). Holzhauser et al. (2016) addressed an extension of the minimum cost flow problem by a budget constraint and presented several special cases of the problem that admitted polynomial-time exact algorithms or approximation algorithms. Chen and Lu (2007) developed another strategy for solving the assignment problem by considering multiple inputs and outputs and modeled this problem as a classical integer linear program to determine the assignments with the maximum efficiency. Amirteimoori (2011) proposed an extension to a transportation problem by considering multiple inputs and outputs for each arc. By defining the relative efficiency for each possible transportation plan, he could determine a transportation plan with the maximum efficiency.

However, the network structure of problems can be destroyed by adding side constraints to them. By exploiting the structure of the network, our approach proposes a minimum cost network flow problem in the presence of multiple weights on arcs in which the objective function is based on the relative efficiency of the arcs. Data envelopment analysis is employed to evaluate the relative efficiency of the arcs in the presence of multiple weights. Since the proposed approach leads to a minimum cost network flow problem, we can find subgraphs in polynomial time that satisfy the integrality property as well.

DEA is a non-parametric method that utilizes linear programming (LP) techniques to gauge the relative efficiency of decision making units (DMU) whose performance is difficult to measure because of multiple inputs and outputs (Cook et al. 2014).

Non-parametric frontier analysis was initially introduced by Farrell (1957). Charnes et al. (1978) introduced the CCR (Charnes, Cooper, and Rhodes) model to measure the CCR-efficiency scores of a given DMU assuming constant returns to scale. The cross-evaluation method, proposed by Sexton et al. (1986), was developed as an effective way to rank DMUs and to identify best performing DMUs using cross-efficiency scores. The results are used to fill out a matrix in which the diagonal members represent the CCR-efficiency scores of DMUs and the remaining cells give the cross-efficiency scores. This matrix may be changed because of the non-uniqueness of optimal input and output weights. Doyle and Green (1994) presented aggressive and benevolent models, which minimize or maximize, respectively, the efficiency of the composite DMU constructed from the other DMUs compared to a specific DMU. An improvement in data envelopment analysis, the cross-efficiency aggregation method based on the Shannon entropy, was proposed by Song and Liu (2018). Today, there exists a wide variety of applications of data envelopment analysis (Tran and Villano 2018; Longaray et al. 2018; Chaudhry and Khan 2016).

3 Subgraph selection with multiple weights on arcs

3.1 Problem definition

Let $G = (N, A)$ be a directed network defined by a set N of n nodes, and a set A of m directed arcs. For each node $i \in N$, we define the two arc adjacency lists $\delta_i^+ = \{(i, j) \in A | j \in N\}$ and $\delta_i^- = \{(j, i) \in A | j \in N\}$ that denote the set of arcs leaving and entering i, respectively, and two node adjacency lists $N^+(i) = \{j \in N | (i, j) \in A\}$ and $N^-(i) = \{j \in N | (j, i) \in A\}$ as the neighborhood sets of node i. Moreover, an integer number $b(i)$ is associated with each node i that represents its supply ($b(i) > 0$) or demand ($b(i) < 0$). If $b(i) = 0$, then node i is called a transshipment (or intermediate) node.

A flow vector x on G is a function from the arc set A to $\mathbb{R}$, i.e., $x : A \mapsto \mathbb{R}$, that satisfies the mass balance constraints:

$$\sum_{(i,j)\in\delta_i^+} x_{ij} - \sum_{(j,i)\in\delta_i^-} x_{ji} = b(i), \quad \forall i \in N. \tag{1}$$

Each arc (i, j) is associated with two parameters l_{ij} and u_{ij} which denote the minimum and maximum amounts of a flow that can be sent over the arc, respectively. The flow x is called feasible if it obeys the following capacity constraints:

$$l_{ij} \leq x_{ij} \leq u_{ij}, \quad \forall (i, j) \in A. \tag{2}$$

Two arcs (i_1, j_1) and (i_2, j_2) of G are said to form a neighborhood if $i_1 = i_2$ or $j_1 = j_2$. As mentioned before, some real-life applications in network flow problem incorporate multiple weights on the arcs of a network such that some of these weights can be interpreted as costs and some others as profits. As example, each arc of a network in transportation planning problem can associated with multiple weights like a travel time, length, deterioration of goods, which we can interpret as costs. Weights like safety, reliability, and revenue can be interpreted as profits. Another possible example is related to wireless sensor networks (WSNs), which are used in a wide variety of systems with vastly varying requirements and characteristics. WSNs are designed for specific applications. Each application should specify its explicit requirements need to guarantee quality of service (QoS) requirements. Each arc of network in WSN can associated with interference, latency, jitter and energy consumption of data transfer and receiving, which again can be interpreted as costs. The information gain ratio, the quality, and the value of information, on the other hand, are weights which can be interpreted as profits.

We represent this in our model by associating each arc $(i, j) \in A$ with two non-negative vectors $C_{ij} = (c^1_{ij}, c^2_{ij}, \ldots, c^k_{ij})$ and $P_{ij} = (p^1_{ij}, p^2_{ij}, \ldots, p^s_{ij})$, where c^t_{ij}, $t = 1, 2, \ldots, k$, and p^r_{ij}, $r = 1, 2, \ldots, s$, denoting the costs and profits per unit flow on arc (i, j), respectively. Thus, each arc has $k + s$ incommensurate attributes (weights) that may conflict with each other. When the arc attributes (weights) are incommensurate, then the problem becomes more difficult, or at least more interesting, and measuring the aggregated performance of such a problem is complex and also difficult. Accordingly, a solution procedure has to be developed for dealing with the above difficulties (Deb 2001).

For determining the relative efficiency of an arc per unit flow, we use DEA techniques in which the relative efficiency of an arc is the weighted sum of profits in comparison with the weighted the sum of costs. The efficiency of a scheme increases if it improves profits at a less than proportionate raise in costs.

3.2 Description of the method of subgraph selection using DEA techniques

This section first proposes DEA models to evaluate arc efficiency and then a linear model is proposed to select a subgraph based on this computed efficiency. For this purpose, each arc in the underlying network is assumed to be a DMU. In other words, we have m DMUs with k inputs and s outputs, the cost and profit vectors of arcs. By considering the head and tail of arc (i, j), two relative efficiencies can be computed relative to the sets of all arcs leaving node i and arcs entering node j.

The relative efficiency of arc (i, j) in comparison to the arcs leaving node i is defined as the maximum of the ratio of total weighted outputs to the total weighted inputs, i.e.,

$$\bar{\theta}_{ij} = \max_{u_r, v_t} \left\{ \frac{\sum_{r=1}^{s} u_r p^r_{ij}}{\sum_{t=1}^{k} v_t c^t_{ij}} \right\}. \tag{3}$$

The ratio $\bar{\theta}_{ij}$ cannot be greater than one for any arc (i, l), $l \in N^+(i)$, i.e.,

$$\frac{\sum_{r=1}^{s} u_r p^r_{il}}{\sum_{t=1}^{k} v_t c^t_{il}} \le 1. \tag{4}$$

where u_r and v_t are the non-negative weight factors. We then have the following linear fractional to compute the relative efficiency score corresponding to arc (i, j).

$$\bar{\theta}_{ij} = \max \left\{ \frac{\sum_{r=1}^{s} u_r p^r_{ij}}{\sum_{t=1}^{k} v_t c^t_{ij}} \right\} \tag{5}$$

$$\text{s.t.} \ \frac{\sum_{r=1}^{s} u_r p_{il}^r}{\sum_{t=1}^{k} v_t c_{il}^t} \leq 1, \qquad l \in N^+(i), \tag{6}$$

$$v_t \geq \epsilon, \qquad t = 1, 2, \ldots, k, \tag{7}$$

$$u_r \geq \epsilon, \qquad r = 1, 2, \ldots, s. \tag{8}$$

where $\epsilon > 0$ is a non-Archimedean construct (Yang et al. 2013). Arc (i, j) is called CCR-efficient in comparison to arcs leaving node i if and only if the optimal value of Model 5–8 obtains a score of 1, otherwise arc (i, j) is CCR-inefficient. Model 5–8 is a linear fractional programming problem, which can be transformed into a linear programming model by the Charnes–Cooper transformation (Charnes and Cooper 1962) to obtain the optimal solution easily. The relative efficiency score $\bar{\theta}_{il}$ can be calculated for all $l \in N^+(i)$ by changing j in the above model.

It should be noted that each arc in Model 5–8 is evaluated by its best weight factors. The value of $\bar{\theta}_{ij}$ reflects the self-evaluation mode in which each of the arcs can achieve the best possible relative efficiency by assigning the most favorable weights to its inputs and outputs. In other words, we have no restrictions on how much weight can be placed on any individual input or output relative to the others.

Instead of the self-evaluation mode, we can use a peer-evaluation mode in which each of the arcs achieves its efficiency using the weights of the other arcs. Hence, the cross-efficiency value of arc (i, j) is defined, which reflects the peer evaluation of arc (i, j) to all the arcs (i, l) with $l \in N^+(i)$, as

$$\theta_{il}^{ij} = \frac{\sum_{r=1}^{s} \bar{u}_r p_{il}^r}{\sum_{t=1}^{k} \bar{v}_t c_{il}^t}, \tag{9}$$

where $\bar{u}_r$ and $\bar{v}_t$ represent the optimal weights for inputs and outputs of the linear programming model representation of Model 5–8, respectively. We calculate the cross-efficiency value $|\delta_i^+|$ times, each time for one different arc belonging to δ_i^+. Then, a $|\delta_i^+| \times |\delta_i^+|$ matrix is computed, in which the diagonal members indicate the CCR-efficiency scores of the DMUs and the remaining cells present the cross-efficiency scores.

The cross-efficiency score of an arc is then denoted by the average of cross-efficiency scores computed in each column of the matrix. The cross-efficiency value of arc (i, j) depends on the optimal output and input weights $\bar{u}_r$ and $\bar{v}_t$. If the optimal weights exhibit non-uniqueness, we can obtain several of the cross-efficiency values for each arc and that decreases the benefit of the cross-efficiency evaluation. Doyle and Green (1994) introduced aggressive and benevolent cross-efficiency models, which optimize the input and output weights and keep the CCR-efficiency unchanged. The following aggressive model obtains the optimal value of the input and output weights for arc (i, j).

$$\min \left\{ \frac{\sum_{r=1}^{s} u_r (\sum_{\substack{l \in N^+(i),\\ l \neq j}} p_{il}^r)}{\sum_{t=1}^{k} v_t (\sum_{\substack{l \in N^+(i),\\ l \neq j}}^{p} c_{il}^t)} \right\} \tag{10}$$

$$\text{s.t.} \frac{\sum_{r=1}^{s} u_r p_{ij}^r}{\sum_{t=1}^{k} v_t c_{ij}^t} = \bar{\theta}_{ij}, \tag{11}$$

$$\frac{\sum_{r=1}^{s} u_r p_{il}^r}{\sum_{t=1}^{k} v_t c_{il}^t} \leq 0, \qquad l \in N^+(i),\ l \neq j, \tag{12}$$

$$v_t \geq \epsilon, \qquad i = 1, 2, \ldots, k, \tag{13}$$

$$u_r \geq \epsilon, \qquad r = 1, 2, \ldots, s. \tag{14}$$

The objective function of Model 10–14 minimizes the cross-efficiencies of the other arcs. By using the Charnes and Cooper transformation (Charnes and Cooper 1962), Model 10–14 can again be equivalently transformed into a linear program. Let u_r^* and v_t^* denote the optimal solution of the linear model corresponding to Model 10–14. In this case, the cross-efficiency score of arc $(i, l) \in \delta_i^+$ is obtained by

$$\bar{\bar{\theta}}_{il}^{ij} = \frac{\sum_{r=1}^{s} u_r^* p_{il}^r}{\sum_{t=1}^{k} v_t^* c_{il}^t}. \tag{15}$$

Therefore, the cross-efficiency matrix is generated by the elements in which the remaining cells present the cross-efficiency scores and the diagonal members present the CCR-efficiency scores of the DMUs, i.e., $\bar{\bar{\theta}}_{il}^{ij}$. Consequently, the cross-evaluation of each arc is considered as the average of the cross-efficiency scores in the corresponding column of the matrix, which is denoted by $\bar{\theta}_{ij}^*$. In a similar manner, based on the target node j of arc (i, j), we can calculate the CCR-efficiency score of (i, j) in comparison to the arcs leaving node i as follows:

$$\tilde{\theta}_{ij} = \max \left\{ \frac{\sum_{r=1}^{s} u_r p_{ij}^r}{\sum_{t=1}^{k} v_t c_{ij}^t} \right\} \tag{16}$$

$$\text{s.t.} \frac{\sum_{r=1}^{s} u_r p_{lj}^r}{\sum_{t=1}^{k} v_t c_{lj}^t} \leq 1, \qquad l \in N^-(j), \tag{17}$$

$$v_t \geq \epsilon, \qquad t = 1, 2, \ldots, k, \tag{18}$$

$$u_r \geq \epsilon, \qquad r = 1, 2, \ldots, s. \tag{19}$$

For all $l \in N^-(j)$, $\tilde{\theta}_{lj}$ can be obtained by changing i in Model 16–19. Similarly, we can compute the cross-efficiency score of arc (i, j), denoted by $\tilde{\theta}_{ij}^*$, in comparison to the arcs entering node j.

We have calculated two kinds of relative efficiencies for each arc corresponding to CCR-efficiency and cross-efficiency scores. Now, a composite efficiency index is constructed in order to incorporate the two kinds of the relative efficiencies based on a DEA model without inputs. The following model computes the composite efficiency index θ_{ij}^1 for arc (i, j) corresponding to efficiency scores $\tilde{\theta}_{ij}$ and $\bar{\theta}_{ij}$.

$$\theta_{ij}^1 = \max \lambda_1 \tilde{\theta}_{ij} + \lambda_2 \bar{\theta}_{ij} \tag{20}$$

$$\text{s.t. } \lambda_1\tilde{\theta}_{iq} + \lambda_2\bar{\theta}_{iq} \le 1, \qquad \forall q \in N^+(i), \tag{21}$$

$$\lambda_1\tilde{\theta}_{pj} + \lambda_2\bar{\theta}_{pj} \le 1, \qquad \forall p \in N^-(j), \tag{22}$$

$$\lambda_1 \ge 0, \lambda_2 \ge 0. \tag{23}$$

Similarly, we can also obtain the composite efficiency index θ^2_{ij} for each arc $(i, j) \in A$ corresponding to $\tilde{\theta}^*_{ij}$ and $\bar{\theta}^*_{ij}$. Hence, we have the two populations of composite efficiencies, θ^1 and θ^2. Thus, it is necessary to determine whether or not the two populations are drawn from an identical distribution.

The Mann–Whitney U test is a non-parametric test to compare differences between two independent groups without making the assumption that values are normally distributed (Mann and Whitney 1947; Lehmann 2006; Hollander et al. 2014). The test has two important assumptions. First, the two samples under consideration are independent of each other. Second, the observations are continuous or ordinal (arranged in ranks). Since the populations θ^1 and θ^2 satisfy the assumptions, the Mann–Whitney U test can be used to test the null hypothesis, i.e., that the two populations θ^1 and θ^2 have identical distribution functions. The alternative hypothesis is that the two distribution functions differ.

There are two cases to consider. The first case is the situation where there is no significant evidence of a difference between the distributions. In this case, since cross efficiency indices provide a unique ordering of the efficiencies and to reduce computational time, θ^2_{ij} is defined as the efficiency of arc (i, j). If θ^1_{ij} is also consider as the efficiency of arc (i, j), we get the similar results. Each arc (i, j) is associated with parameter θ^2_{ij} which reflects its relative efficiency. If θ^2_{ij} is equal to one, this means that this arc is an efficient arc in comparison to the arcs leaving node i and those entering node j. Hence, the objective is the minimization of $\sum_{(i,j)\in A}(1 - \theta^2_{ij})x_{ij}$.

As we mentioned before, the flow x is feasible if it guarantees flow mass balance at each node and obeys the capacity constraints of each arc. We then determine an efficient subgraph G' by solving the following minimum cost flow problem, where the objective function is based on the relative efficiency of the arcs.

$$\theta^* = \min \sum_{(i,j)\in A} (1 - \theta^2_{ij})x_{ij} \tag{24}$$

$$\text{s.t. } \sum_{(i,j)\in\delta_i^+} x_{ij} - \sum_{(j,i)\in\delta_i^-} x_{ji} = b(i), \qquad \forall i \in N, \tag{25}$$

$$l_{ij} \le x_{ij} \le u_{ij}, \qquad \forall (i, j) \in A. \tag{26}$$

Let x^* be the optimal solution of Model 24–26. We can select an efficient subgraph $G' = (N', A')$ of the network $G = (N, A)$, where $A' = \{(i, j) \in A | x^*_{ij} > l_{ij}\}$ and $N' = \{i | (i, j) \, or \, (j, i) \in A'\}$. Note that Model 24–26 can be solved by using several algorithms, such as cycle-canceling, successive shortest path, primal-dual, out-of kilter, and network simplex algorithms (Ahuja et al. 1993).

Assuming all parameters $b(i)$, $l_{i,j}$, and u_{ij} are integer, then every extreme point of the convex hull of the feasible region of Model 24–26 is integer-valued, too. Therefore,

Algorithm 1 Finding a subgraph with multiple weights on arcs using DEA

Require: $G = (N, A), b(i)\ \forall i \in N; l_{ij}, u_{ij}, C_{ij} = (c_{ij}^1, c_{ij}^2, \ldots, c_{ij}^k), P_{ij} = (p_{ij}^1, p_{ij}^2, \ldots, p_{ij}^s)\ \forall (i, j) \in A$.
Ensure: Returns the selected subgraph with efficient arcs for a feasible flow.
for each $(i, j) \in A$ **do**
    Calculate $\bar{\theta}_{ij}$ by solving Model 5-8.
    Calculate $\bar{\theta}_{ij}^*$ using the cross-evaluation method described in Section 3.2 for $\bar{\theta}_{ij}$.
    Calculate $\tilde{\theta}_{ij}$ by solving Model 16-19.
    Calculate $\tilde{\theta}_{ij}^*$ using the cross-evaluation method described in Section 3.2 for $\tilde{\theta}_{ij}$.
end for
for each $(i, j) \in A$ **do**
    Obtain θ_{ij}^1 by solving Model 20-23 for $\tilde{\theta}_{ij}$ and $\bar{\theta}_{ij}$.
    Obtain θ_{ij}^2 by solving Model 20-23 for $\tilde{\theta}_{ij}^*$ and $\bar{\theta}_{ij}^*$.
end for
if there is no significant difference between the distribution of θ^1 and θ^2 **then**
    Solve Model 24-26 with θ_{ij}^2 as the efficiency of arc (i, j) and obtain the subgraph $G' = (N', A')$.
else
    Solve Model 24-26 with θ_{ij}^1 as the efficiency of arc (i, j) and obtain the subgraph $G_1' = (N', A')$.
    Solve Model 24-26 while considering θ_{ij}^2 as the efficiency of arc (i, j) and obtain the subgraph $G_2' = (N', A')$.
end if

the model has the integrality property. This means that if the problem requires an integer optimal solution, then it can be solved by simply determining an optimal extreme point solution to the linear model by ignoring the integrality restrictions.

In the second case, there is a significant difference between the two population distributions of the composite efficiencies. In this case, Model 24–26 is solved with θ_{ij}^1 and θ_{ij}^2 as the arc efficiencies and two efficient subgraphs are determined for both CCR-efficiency and cross-efficiency. This method is specified as Algorithm 1.

3.3 Problem complexity

In this section, we analyze the complexity of the proposed algorithm. Given a graph $G = (N, A)$ in which each arc $(i, j) \in A$ is associated with two non-negative vectors $C_{ij} = (c_{ij}^1, c_{ij}^2, \ldots, c_{ij}^k)$ and $P_{ij} = (p_{ij}^1, p_{ij}^2, \ldots, p_{ij}^s)$, where c_{ij}^t, $t = 1, 2, \ldots, k$, and p_{ij}^r, $r = 1, 2, \ldots, s$, denote the cost and profit per unit flow on arc (i, j), respectively, i.e. for each arc (i, j), these are $k + s$ attributes (weights). When we have multiple weights on the arcs of a network, finding an optimal subgraph over which a feasible flow can be sent then can become difficult. One way to solve the problem is that we consider it as a constrained network flow problem, which is NP-hard in general (Holzhauser et al. 2016). In our approach to solve the problem, we first evaluate the efficiency of the arcs in the presence of multiple weights on them using DEA techniques. We then determine an efficient subgraph $G' = (N', A')$ using a linear programming model including the mass balance and capacity constraints, where the objective function is based on the relative efficiency of the arcs.

Below we show that the proposed algorithm takes time bounded by a polynomial to solve the problem, but it should be noted that the selected subgraph may not necessarily optimal. First, two relative efficiencies are computed by considering the start i and end j of arc (i, j) by comparing the arc with all other arcs leaving i or entering j. Then to obtain the CCR-efficiency score of arc (i, j), a DEA model without inputs is computed. The cross-relative efficiency score of arc (i, j) is calculated the same way. Therefore, we need to solve $6(\sum_{i\in N} |N^+(i)| + |N^-(i)|) = 12m$ linear models.

Thus, the running time of the calculation of the relative efficiency for each arc is bounded by $\mathcal{O}(m(k+s)^4 \log((k+s)L))$, where $(k+s)$ is the number of weights and L is a bound on the magnitude of coefficients of the CCR and aggressive models. For calculating a subgraph, the linear programming Model 24–26 is applied in which the weights of the arcs are their relative efficiencies. This model can be solved in a polynomial time $\mathcal{O}(m \log(nm+n^2 \log n))$ (Orlin 1993). Thus, the proposed algorithm solves the problem in $\mathcal{O}(m^3(k+s)^4 \log(nm+n^2 \log n)\log((k+s)L))$.

4 Results

We now present computational experiments that evaluate the performance of the proposed method in comparison to other approaches. We present results for different networks, including small networks to explain the algorithm and on random networks to show the efficiency of the proposed algorithm. Algorithm 1 was implemented using the General Algebraic Modeling System (GAMS) Distribution 22.5 using CPLEX as an underlying solver on system with an Intel Core i5 processor at 2.2 GHz with 4 GB of RAM. GAMS is a high-level modeling system for mathematical programming and optimization, which consists of a language compiler and is specifically designed for modeling linear, nonlinear, and mixed integer optimization problems.[2]

4.1 Examples of the method

In this subsection, the considered method is applied to two networks with ten nodes each. Each arc (i, j) is associated with cost vector $C_{ij} = (c^1_{ij}, c^2_{ij})$ and profit vector $P_{ij} = (p^1_{ij})$. In other word, each arc is associated with three nonnegative weights, which could, for instance, represent the cost (or length), the delay (or travel time), and the profit of arcs. We use c^1, c^2, and p^1 to denote the vectors $c^1 = (c^1_{ij})_{(i,j)\in A}$, $c^2 = (c^2_{ij})_{(i,j)\in A}$, and $p^1 = (p^1_{ij})_{(i,j)\in A}$. As first example, the network described in Table 1 is considered in which the capacity of each arc is equal to one unit and node values are as follows:

$$b(i) = \begin{cases} 2, & \text{if } i = 1 \\ 2, & \text{if } i = 4, \\ -3, & \text{if } i = 7, \\ -1, & \text{if } i = 10, \\ 0, & \text{otherwise.} \end{cases}$$

The cross-efficiency scores and CCR-efficiency were calculated for each arc according to steps 1–6 in Algorithm 1. Then, the composite efficiencies θ^1_{ij} and θ^2_{ij} corresponding

[2] See *GAMS—A User's Guide*, Frechen, Germany: GAMS Development Corporation, 2021.

to each arc $(i, j) \in A$ were obtained according to steps 7–10, which are revealed in the eighth and eleventh columns of Table 1. We ran the non-parametric Mann–Whitney U test on columns θ^1_{ij} and θ^2_{ij}. We found that the null hypothesis was rejected, i.e., the observed distributions differ. Then, two subgraphs could be obtained based on the CCR-efficiency and cross-efficiency scores. The objective function value of Model 24–26 was $\theta^* = 0.237$ when θ^1 considered as arc efficiency. In addition, the total weight of arcs in a resulting subgraph with respect to vectors c^1, c^2, and p^1 were 61, 22, and 43, respectively. In the same way, by solving Model 24–26 and proposing θ^2 as weight of the arcs, the optimal objective value was $\theta^* = 0.241$ and the total weight of a resulting subgraph with respect to vectors c^1, c^2, and p^1 were 41, 31, and 34, respectively.

In a second example, Table 2 presents the proposed network. The capacity of each arc is equal to one unit and the node values are as follows:

$$b(i) = \begin{cases} 1, & \text{if } i = 1, \\ 3, & \text{if } i = 5, \\ -1, & \text{if } i = 7, \\ -3, & \text{if } i = 9, \\ 0, & \text{otherwise.} \end{cases}$$

For each arc (i, j), we computed the CCR-efficiency score compared to arcs leaving node i and entering node j and the composite efficiency indices. We present them in the sixth, seventh, and eighth columns of Table 2, respectively. In a similar way, the ninth, tenth, and eleventh columns represent the cross-efficiency score of each arc.

The Mann–Whitney U test was applied to the columns with respect to θ^1_{ij} and θ^2_{ij}. This time, the null hypothesis could not be rejected, i.e., the distributions of θ^1_{ij} and θ^2_{ij} may be the same. In this case, for each arc (i, j) we considered the cross efficiency θ^2_{ij} as the arc efficiency because it could provide a unique ordering of the arcs. The subgraph found by Model 24–26 is presented in Fig. 1a. In this case, the total weight of arcs in a found subgraph to vectors c^1, c^2, and p^1 were 50, 27, and 27, respectively. When we consider θ^1_{ij} as the efficiency of arc (i, j), these total weights of a found subgraph are 50, 27, and 33, respectively. The resulting subgraph in this case is shown in Fig. 1b. As we can be seen, similar results are obtained.

4.2 Performance evaluation

In this section, we conduct simulations based on random network topologies, in which each arc (i, j) is associated with cost vector $C_{ij} = (c^1_{ij}, c^2_{ij})$ and profit vector $P_{ij} = (p^1_{ij})$. We refer to their element c^1_{ij} as the cost (or length), to c^2_{ij} as the delay (or travel time), and to p^1_{ij} as the profit of arc (i, j). We assess the average total cost, delay, and profit of the random subgraph connections by using, first, our proposed approach and, second, minimum cost subgraphs and budget-constrained subgraphs.

By utilizing the method proposed by Waxman (1988), random network topologies were produced. The network nodes are randomly located in a one-by-one square.

Table 1 Description of the proposed method: Example 1

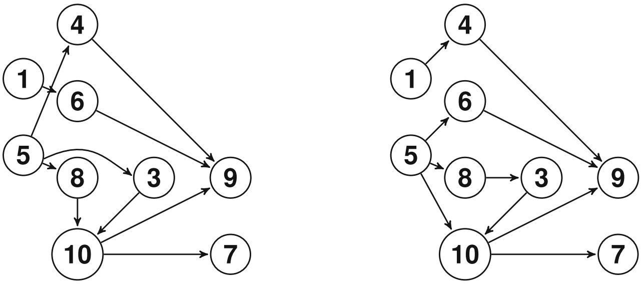
Tail	Head	c^1_{ij}	c^2_{ij}	p^1_{ij}	$\overline{\theta}_{ij}$	$\widetilde{\theta}_{ij}$	θ^1_{ij}	$\overline{\theta}^*_{ij}$	$\widetilde{\theta}^*_{ij}$	θ^2_{ij}
1	3	6	2.25	1	0.389	0.305	0.389	0.99	0.99	1.00
1	5	7	1.75	3	1.000	0.686	1.000	1.00	0.19	1.00
1	8	4	3.25	1	0.583	0.286	0.583	0.58	0.25	1.00
1	9	7	1.75	3	1.000	0.882	1.000	1.00	1.00	1.00
3	2	9	0.75	5	1.000	1.000	1.000	0.99	0.96	0.28
3	5	1	4.75	1	0.800	0.400	0.800	0.18	0.32	0.59
3	7	4	3.25	5	1.000	1.000	1.000	1.00	0.99	0.68
3	8	2	4.25	2	0.800	1.000	1.000	0.36	0.43	0.86
3	9	5	2.75	5	1.000	1.000	1.000	0.28	0.37	0.16
3	10	5	2.75	2	0.400	1.000	1.000	1.00	1.00	1.00
4	2	3	3.75	1	0.417	0.600	0.600	0.80	0.31	0.76
4	3	5	2.75	4	1.000	1.000	1.000	1.00	0.36	1.00
4	5	10	0.25	4	1.000	1.000	1.000	1.00	1.00	0.84
4	8	8	1.25	4	1.000	1.000	1.000	1.00	0.96	0.40
4	9	10	0.25	1	0.250	1.000	1.000	0.99	1.00	0.38
5	3	4	3.25	3	0.667	0.937	0.937	0.50	0.42	0.74
5	6	5	2.75	5	1.000	1.000	1.000	0.95	1.00	0.95
5	7	3	3.75	4	1.000	0.800	1.000	1.00	1.00	0.78
5	9	7	1.75	3	0.942	0.882	0.942	1.00	0.85	0.40
6	5	2	4.25	5	1.000	1.000	1.000	1.00	0.83	1.00
6	7	7	1.75	5	1.000	1.000	1.000	0.20	1.00	1.00
6	9	3	3.75	4	0.800	0.800	0.800	1.00	0.99	0.89
8	7	2	4.25	5	1.000	1.000	1.000	1.00	1.00	0.90
8	9	2	4.25	5	1.000	1.000	1.000	0.98	1.00	0.41
8	10	10	0.25	4	1.000	1.000	1.000	0.65	0.16	1.00
9	2	3	3.75	1	0.333	0.600	0.600	1.00	1.00	0.99
9	5	7	1.75	3	0.771	0.686	0.771	0.85	1.00	0.42
9	6	5	2.75	5	1.000	1.000	1.000	1.00	0.56	1.00
9	7	6	2.25	1	0.200	0.200	0.200	1.00	1.00	1.00
9	8	6	2.25	5	1.000	1.000	1.000	0.98	1.00	1.00

The probability of creating an arc between two nodes is $\beta \exp(\frac{-d_{ij}}{\alpha L})$, where d_{ij} is the distance between nodes i and j, and L is the maximum distance between any two nodes. The parameters α and β can be selected from the range [0, 1) to obtain topologies similar to real networks. Small values of α cause long connections and higher values of β cause nodes with a high average degree. We set α and β such that we obtain networks with an average node degree of four.

The parameter c^1_{ij} of arc (i, j) was set to the distance between its endpoints plus one. The parameters c^2_{ij} and p^1_{ij} were set to random numbers drawn from the uniform

Table 2 Description of the proposed method: Example 2

Tail	Head	c_{ij}^1	c_{ij}^2	p_{ij}^1	$\overline{\theta}_{ij}$	$\widetilde{\theta}_{ij}$	θ_{ij}^1	$\overline{\theta}_{ij}^*$	$\widetilde{\theta}_{ij}^*$	θ_{ij}^2
1	3	3	3.75	2	0.462	1.000	1.000	1.00	0.65	0.99
1	4	2	4.25	4	1.000	1.000	1.000	0.48	1.00	0.68
1	6	10	0.25	4	1.000	1.000	1.000	0.87	0.97	1.00
1	10	5	2.75	5	1.000	1.000	1.000	0.23	0.81	0.28
3	2	2	4.25	2	1.000	0.500	1.000	1.00	0.99	0.25
3	4	4	3.25	2	0.800	0.615	0.800	0.40	0.96	1.00
3	7	4	3.25	1	0.400	0.333	0.400	1.00	1.00	0.37
3	8	3	3.75	2	0.889	0.800	0.889	0.69	0.85	1.00
3	10	10	0.25	4	1.000	1.000	1.000	1.00	1.00	1.00
4	2	2	4.25	4	1.000	1.000	1.000	0.66	1.00	0.22
4	3	3	3.75	1	0.250	0.500	0.500	1.00	0.98	1.00
4	6	1	4.75	1	0.500	1.000	1.000	0.30	0.28	0.95
4	7	5	2.75	3	0.886	1.000	1.000	1.00	0.63	0.67
4	9	2	4.25	1	0.250	0.666	0.666	1.00	0.30	0.22
4	10	4	3.25	4	1.000	0.842	1.000	1.00	1.00	0.90
5	2	4	3.25	4	0.870	1.000	1.000	0.39	1.00	0.39
5	3	2	4.25	1	0.238	0.750	0.750	0.53	1.00	0.76
5	4	2	4.25	2	0.476	0.500	0.500	0.90	0.25	1.00
5	6	6	2.25	4	0.800	1.000	1.000	1.00	1.00	0.74
5	8	6	2.25	5	1.000	1.000	1.000	0.22	1.00	0.96
5	10	1	4.75	4	1.000	1.000	1.000	1.00	0.56	0.67
6	3	8	1.25	5	1.000	1.000	1.000	1.00	0.26	0.81
6	4	10	0.25	1	0.999	1.000	1.000	0.51	1.00	0.76
6	7	6	2.25	1	0.250	0.407	0.407	1.00	1.00	0.23
6	8	7	1.75	1	0.222	0.257	0.257	1.00	0.97	0.98
6	9	4	3.25	3	1.000	1.000	1.000	0.46	0.99	0.22
8	2	6	2.25	1	0.263	0.250	0.263	0.33	0.48	0.23
8	3	7	1.75	4	1.000	0.909	1.000	0.23	0.76	0.95
8	7	6	2.25	1	0.263	0.407	0.407	1.00	1.00	0.77
8	10	2	4.25	3	1.000	0.706	1.000	1.00	1.00	0.89
10	2	7	1.75	4	1.000	1.000	1.000	0.73	1.00	1.00
10	3	6	2.25	3	0.789	0.789	0.789	0.24	1.00	0.77
10	7	2	4.25	3	1.000	1.000	1.000	0.84	0.99	0.99
10	9	10	0.25	1	1.000	1.000	1.000	1.00	1.00	0.87



(a) When θ^2 is considered as the efficiency of arcs **(b)** When θ^1 is considered as the efficiency of arcs

Fig. 1 Resulting subgraphs for Example 2

Table 3 Comparison between the proposed method and Model 27–29

Network size (nodes)	Proposed method			Model 27–29		
	c^1	c^2	p^1	c^1	c^2	p^1
20	95.489	122.978	124.135	72.340	167.174	78.693
30	263.000	250.376	368.200	210.400	513.509	326.905
40	341.305	513.240	614.348	256.620	742.978	394.223
50	775.359	781.714	1007.966	635.540	1440.946	826.202

distribution with range [0, 1] times the cost plus one. The occurrence of unrealistic zero values was avoided by adding one to c^1_{ij}, c^2_{ij}, and p^1_{ij}.

The differences between the quality of the minimum cost subgraphs, which may be formulated in the following way, and subgraphs obtained from our considered method are compared in the first experiment step.

$$\min \sum_{(i,j)\in A} c_{ij} x_{ij} \tag{27}$$

$$\text{s.t.} \sum_{(i,j)\in\delta_i^+} x_{ij} - \sum_{(j,i)\in\delta_i^-} x_{ji} = b(i), \qquad \forall i \in N, \tag{28}$$

$$l_{ij} \le x_{ij} \le u_{ij}, \qquad \forall (i,j) \in A. \tag{29}$$

Model 27–29 seeks the cheapest possible way of sending a feasible flow by considering a single weight of the arcs of a network without taking into consideration the other weights of the arcs. The results are presented in Table 3. The first column of the table shows the size of networks and the other values in the table are the average of 10 runs.

The total weight average of arcs in the subgraphs determined from the proposed method of this paper corresponding to vector weights c^1, c^2, and p^1 are shown in the second, third, and fourth columns of the table, respectively. Similarly, the total weight average of arcs in subgraphs obtained by Model 27–29, are given in the fifth, sixth and seventh columns of Table 3, respectively.

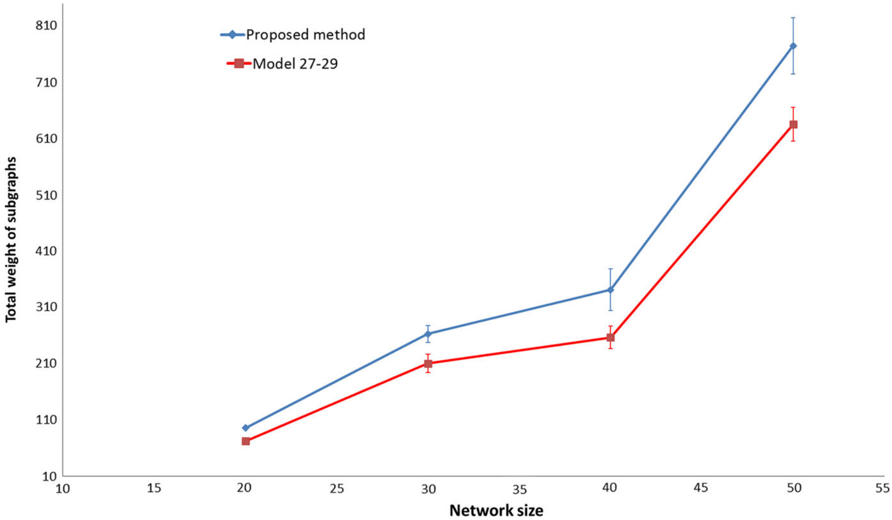


Fig. 2 Comparison between the proposed method and Model 27–29 regarding to vector weight c^1

As can be seen, the total weights of the subgraphs resulting from the considered method corresponding to c^1 are approximately 40% worse compared to subgraphs resulting from Model 27–29. Figure 2 illustrates this issue. This is because Model 27–29 obtains optimal subgraphs regarding to only weight c^1 and does not seek to improve resulting subgraphs compared to other weights.

However, as we can see in the Table 3, for all network topologies the average gaps regarding to parameters c^2 and p^1 are 67% and 37%, respectively, indicating that the obtained subgraphs of our approach have better performance in terms of c^2 and p^1. In fact, Model 27–29 performs poorly with regard to weights c^2 and p^1. Although the considered method of this study leads to expensive subgraphs, it obtains subgraphs with lower delay (c^2) and higher profit (p^1). These results indicate that subgraphs obtained from our method have higher quality compared to those obtained with Model 27–29.

In some applications, one of the important reasons for rejecting some subgraphs is due to a bound on the total weight of selected subgraphs. Hence, we considered an extension of Model 27–29 in the second step of computational experiments by adding side constraints to Model 27–29 to deal with these weights. This problem is presented as the budget-constrained minimum cost flow problem (Holzhauser et al. 2016). The budget-constrained problem (BCP) of Model 27–29 can be formulated as a mixed integer linear program as follows:

$$\min \sum_{(i,j)\in A} c^1_{ij} x_{ij} \tag{30}$$

$$s.t. \sum_{(i,j)\in\delta_i^+} x_{ij} - \sum_{(j,i)\in\delta_i^-} x_{ji} = b(i), \qquad \forall i \in N, \tag{31}$$

$$\sum_{(i,j)\in A} c^2_{ij} y_{ij} \leq D, \tag{32}$$

$$\sum_{(i,j)\in A} p^1_{ij} y_{ij} \geq P, \tag{33}$$

$$0 \leq x_{ij} \leq u_{ij} y_{ij}, \qquad \forall (i,j) \in A, \tag{34}$$

$$y_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A, \tag{35}$$

where the total weight of a selected subgraph is constrained by the budgets D and P.

In this step, we first implemented our proposed method and computed the total weight of the selected subgraphs regarding to vector weights c^1, c^2 and p^1, which are denoted by C^1_{DEA}, C^2_{DEA}, and P^1_{DEA}, respectively. The results are listed in the second, third and fourth columns of Table 4. Then, to evaluate the performance of our methods, we compared these subgraphs with subgraphs resulting from Model 30–35 in the following two strategies.

In the first strategy, the upper bound D was set to the constant value C^2_{DEA} and the value of P was varied from $P^1_{DEA} - 10$ to $P^1_{DEA} + 10$ in Model 30–35. The results are outlined in the fifth column of Table 4. As we can see, the lower bound P has a direct influence on the quality of subgraphs. A subgraph could not be available when the P value is higher than a specified threshold.

In the second strategy, the value of the lower bound P was set to the constant value P^1_{DEA} and the value of D was changed from $C^2_{DEA} - 10$ to $C^2_{DEA} + 10$ in Model 30–35. These results are presented in the last column of Table 4. In Model 30–35, we can find subgraphs with lower objective value when the upper bound D increases. It is obvious that the considered method has better performance compared to Model 30–35 developed in (Holzhauser et al. 2016) with up to 30% decrease in objective function value.

The CPU times (in seconds) which are necessary for generating a subgraph resulting from Model 30–35 and the considered method of this study are shown in Fig. 3. The runtimes required by our method are approximately 25% lower than those of Model 30–35 and the gap between the two methods increases with the size of the networks.

We also assess the proposed method on other randomly generated networks. We first explain how to generate the random test instances of the problem. We then compare exact solutions obtained for the budget-constrained problem (BCP) by setting a budget for the other weights with our solutions.

The networks were randomly generated using three different methods, similar to (Bollobás 2011): The first random networks were generated using to the methodology proposed by Erdös and Rényi (2006) in which it is assumed that there is a link from node i to node j with probability p_{ij}. The second network we consider is the small-world network of Watts (2003) and the third network is the scale-free network described by Barabási and Albert (1999).

For each arc (i, j), uniform random numbers were generated on an interval, say $[a, b]$, in order to represent u_{ij} c^1_{ij},c^2_{ij}, and p^1_{ij}.

We first implemented our proposed method and computed the total weights C^1_{DEA}, C^2_{DEA}, and P^1_{DEA} of the selected subgraph with regard to vector weights c^1, c^2 and p^1,

Table 4 Comparison between the proposed method and Model 30–35

Network size (nodes)	Proposed method		Average cost Model 30–35		
	C^1_{DEA}	C^2_{DEA}	P^1_{DEA}	Fixed D	Fixed P
20	65	40.26	45.52	76.23	69.52
30	97.56	66.32	75.26	104.60	102.45
40	108.45	93.95	91.13	124.30	115.24
50	149.26	111.04	94.94	175.21	164.56

respectively. Then, in the first strategy, we investigate the impact of the change budget P on BCP when the budget D was set to the constant value C^2_{DEA}. In the second strategy, we evaluate the effect of the change budget D on the BCP when the budget D was set to the constant value P^1_{DEA}.

We calculated the total weights of subgraphs resulting from Model 30–35 and the considered method of this study with respect to vector weights c^1, c^2 and p^1 and denoted them by C^1_{total}, C^2_{total} and P^1_{total} in the columns of the result tables. Moreover, since we have a set of available solutions with respect to multiple vector weights c^1, c^2, and p^1, we need an effective framework for comparing exact solutions obtained by setting a budget with our proposed approach based on the evaluation of multiple conflicting weights.

The Technique for Order Performance by Similarity to Ideal Solution (TOPSIS) is a practical approach from multiple criteria decision making for ranking and selection of a number of possible alternatives through measuring Euclidean distances (Tzeng and Huang 2011). Her, the chosen alternative should have the shortest distance from the positive ideal solution (PIS), i.e., the solution that maximizes the benefit criteria and minimizes the cost criteria; and also be the farthest distance away from the negative ideal solution (NIS), i.e., the solution that maximizes the cost criteria and minimizes the benefit criteria. In our numerical results, Shannon's Entropy methodology was used to determine the importance of attributes (Chen 2021). Hence in the result tables, we added have columns regrading to relative closeness (RC) and ranking obtained from TOPSIS approach.

Tables 5 and 6 show the results obtained from random topologies generated by the Barabási and Albert (1999) method. As observed in Table 5, although Model 30–35 has better performance regarding to C^1_{total}, our approach has better performance compared to Model 30–35 rather in the other total weights and especially CPU Time. Hence, our solution could obtain rank 1 in the TOPSIS approach.

Tables 6 shows that the ranking of our solution is 7, but as we see in this table the solution with a rank 1 has the same RC with accuracy to 3 decimal places with our solution while the computation time of our approach is less compared to Model 30–35.

Tables 7 and 8 show the results obtained from random topologies generated by the Watts (2003) and Erdös and Rényi (2006) methods, respectively. The results also confirm the adequacy of the proposed approach.

In summary, our proposed method can find subgraphs in polynomial time which satisfy the integrality property.

Table 5 Results for randomly generated network using the Barabási and Albert (1999) method with 100 nodes and 16 supply nodes

Approach	C^1_{total}	C^2_{total}	P^1_{total}	CPU time	RC	Rank
Proposed method	185,397.18	1949.25	351,820.94	332.53	0.999659823	1
Model 30–35 with fixed P	125857.20	1900.00	351820.94	1033.21	0.980201293	21
	125,726.03	1910.00	351,820.94	1025.90	0.980407874	16
	125,197.23	1920.00	351,820.94	1032.92	0.980209575	20
	124,731.41	1930.00	351,820.94	1024.87	0.980436953	15
	124,438.09	1940.00	351,820.94	1023.60	0.980472924	12
	124,327.90	1950.00	351,820.94	1024.15	0.980457412	14
	123,757.54	1960.00	351,820.94	1014.35	0.980734323	6
	123,588.96	1970.00	351,820.94	1020.86	0.980550263	8
	123,074.87	1980.00	351,820.94	1021.88	0.980521584	9
	122,651.67	1990.00	351,820.94	1031.60	0.980246765	19
	122,737.22	2000.00	351,820.94	1042.68	0.979933800	22
Model 30–35 with fixed D	129,143.58	1949.25	351,771.00	35,723.08	0.000304991	23
	124,043.41	1949.25	351,781.00	1022.03	0.980517400	10
	124,017.55	1949.25	351,791.00	1022.53	0.980503046	11
	124,362.00	1949.25	351,801.00	1031.05	0.980262303	18
	124,240.60	1949.25	351,811.00	1023.89	0.980464759	13
	124,404.37	1949.25	351,821.00	1030.25	0.980285078	17
	124,203.84	1949.25	351,831.00	1019.14	0.980599004	7
	124,049.96	1949.25	351,841.00	1001.34	0.981101850	3
	124,356.72	1949.25	351,851.00	1001.30	0.981103121	2
	124,309.17	1949.25	351,861.00	1002.22	0.981076899	4
	124,382.50	1949.25	351,871.00	1002.93	0.981056837	5
Avg. Model 30–35	124,436.45	1949.63	351,820.97	2598.90		
Shannon's entropy	0.012530004	1.36E−05	4.41055E−10	0.98745643		

Table 6 Results for randomly generated network using the Barabási and Albert (1999) method with 200 nodes and 19 supply nodes

Approach	C^1_{total}	C^2_{total}	P^1_{total}	CPU time	RC	Rank
Proposed method	342,948.92	4022.25	705,353.91	951.74	0.998285138	7
Model 30–35 with fixed P	289,141.83	3973.00	705,353.91	1001.06	0.997982640	19
	196,451.64	3983.00	705,353.91	1001.28	0.998302091	5
	195,104.61	3993.00	705,353.91	10141.08	0.001685949	23
	292,156.92	4003.00	705,353.91	981.51	0.998472822	1
	261,919.15	4013.00	705,353.91	1031.69	0.997149072	21
	267,826.95	4023.00	705,353.91	1008.66	0.997869261	20
	254,625.63	4033.00	705,353.91	1002.36	0.998126704	14
	263,479.52	4043.00	705,353.91	1001.21	0.998121944	15
	271,963.79	4053.00	705,353.91	1000.48	0.998100147	16
	294,903.68	4063.00	705,353.91	9407.64	0.710306758	22
	192,564.82	4073.00	705,353.91	1001.86	0.998282732	8
Model 30–35 with Fixed D	262,527.99	4022.25	705,304.00	1001.17	0.998127557	13
	252,878.85	4022.25	705,314.00	1000.39	0.998196666	10
	262,102.43	4022.25	705,324.00	1001.10	0.998132041	12
	194,769.70	4022.25	705,334.00	1000.79	0.998319132	3
	264,070.65	4022.25	705,344.00	1000.28	0.998147402	11
	192,826.85	4022.25	705,354.00	1000.30	0.998336310	2
	195,988.84	4022.25	705,364.00	1000.93	0.998314071	4
	246,578.70	4022.25	705,374.00	1001.12	0.998199387	9
	273,393.95	4022.25	705,384.00	1000.58	0.998089344	17
	194,933.69	4022.25	705,394.00	1001.46	0.998296289	6
	283,149.30	4022.25	705,404.00	1001.36	0.998010462	18
Avg. Model 30–35	245,607.25	4022.63	705,353.96	1799.47		
Shannon's entropy	0.012530004	1.36E−05	4.41055E−10	0.98745643		

Table 7 Results for randomly generated network using the Watts (2003) method with 200 nodes and 20 supply nodes

Approach	Total cost	Total delay	Total profit	CPU time	Relative	Rank
Proposed method	369,805.10	3780.00	682,210.36	661.32	0.999017564	1
Model 30–35 with fixed P	233,033.12	3730.00	682,210.36	1004.92	0.982659700	21
	231,005.69	3830.00	682,210.36	1001.75	0.982819994	17
	230,552.08	3930.00	682,210.36	1001.04	0.982855444	13
	229,993.71	4030.00	682,210.36	1000.35	0.982890317	5
	230,653.93	4130.00	682,210.36	1000.79	0.982867981	10
	229,248.67	4230.00	682,210.36	1000.45	0.982885255	7
	228,056.41	4330.00	682,210.36	1000.35	0.982890036	6
	228,474.85	4430.00	682,210.36	1000.21	0.982897212	2
	229,505.29	4530.00	682,210.36	1000.48	0.982883017	8
	227,134.96	4630.00	682,210.36	1001.02	0.982855743	12
	228,047.27	4730.00	682,210.36	1014.98	0.982150981	22
Model 30–35 with fixed D	228,695.00	3780.00	682,161.00	1001.73	0.982820827	16
	228,964.13	3780.00	683,161.00	1001.17	0.982849188	15
	231,580.42	3780.00	684,161.00	20,476.82	0.000969363	23
	229,272.82	3780.00	685,161.00	1002.15	0.982799578	19
	229,866.97	3780.00	686,161.00	1000.59	0.982878300	9
	229,292.42	3780.00	687,161.00	1000.83	0.982866142	11
	229,146.89	3780.00	688,161.00	1001.96	0.982809470	18
	229,741.93	3780.00	689,161.00	1001.15	0.982849990	14
	229,194.79	3780.00	690,161.00	1000.24	0.982895867	3
	230,117.89	3780.00	691,161.00	1000.31	0.982892479	4
	229,674.77	3780.00	692,161.00	1004.27	0.982692589	20
Avg. Model 30–35	229,602.46	4005.00	684,685.68	1887.16		
Shannon's entropy	0.0073270	0.0034737	0.0000134	0.9891859		

Table 8 Results for randomly generated network using Erdös and Rényi (2006) method with 200 nodes, probability 0.7, and 20 supply nodes

Approach	Total cost	Total delay	Total profit	CPU time	Relative	Rank
Proposed method	83,419.11	1236.50	141,754.87	570.95	0.882922986	2
Model 30–35 with Fixed P	47,231.16	1187.00	141,754.87	1001.73	0.682705051	6
	47,134.51	1287.00	141,754.87	1001.15	0.680900509	17
	47,070.96	1387.00	141,754.87	1000.30	0.678565209	18
	47,011.78	1487.00	141,754.87	1000.87	0.674766009	19
	46,928.32	1587.00	141,754.87	1001.34	0.670417872	20
	46,869.18	1687.00	141,754.87	1000.40	0.666300097	21
	46,738.90	1787.00	141,754.87	450.38	0.891537453	1
	46,697.95	1887.00	141,754.87	682.05	0.813765776	4
	46,618.42	1987.00	141,754.87	589.85	0.834155756	3
	46,554.52	2087.00	141,754.87	716.90	0.776908526	5
	46,545.31	2187.00	141,754.87	1000.62	0.637026018	22
Model 30–35 with fixed D	46,850.04	1236.50	141,705.00	1000.39	0.682536845	11
	46,853.88	1236.50	142,705.00	1000.81	0.682298516	15
	46,857.71	1236.50	143,705.00	1001.46	0.681929072	16
	46,861.55	1236.50	144,705.00	1000.42	0.682520650	12
	46,865.39	1236.50	145,705.00	1000.29	0.682591696	8
	46,869.23	1236.50	146,705.00	1000.27	0.682605406	7
	46,873.07	1236.50	147,705.00	1000.52	0.682460164	14
	46,876.90	1236.50	148,705.00	1000.34	0.682561863	9
	46,880.74	1236.50	149,705.00	1000.37	0.682543211	10
	46,884.58	1236.50	150,705.00	1000.49	0.682473466	13
	46,888.42	1236.50	151,705.00	2124.99	0.179865206	23
Avg. Model 30–35	46,861.93	1461.75	144,229.94	980.72		
Shannon's entropy	0.1303607	0.2904058	0.0034552	0.5757784		

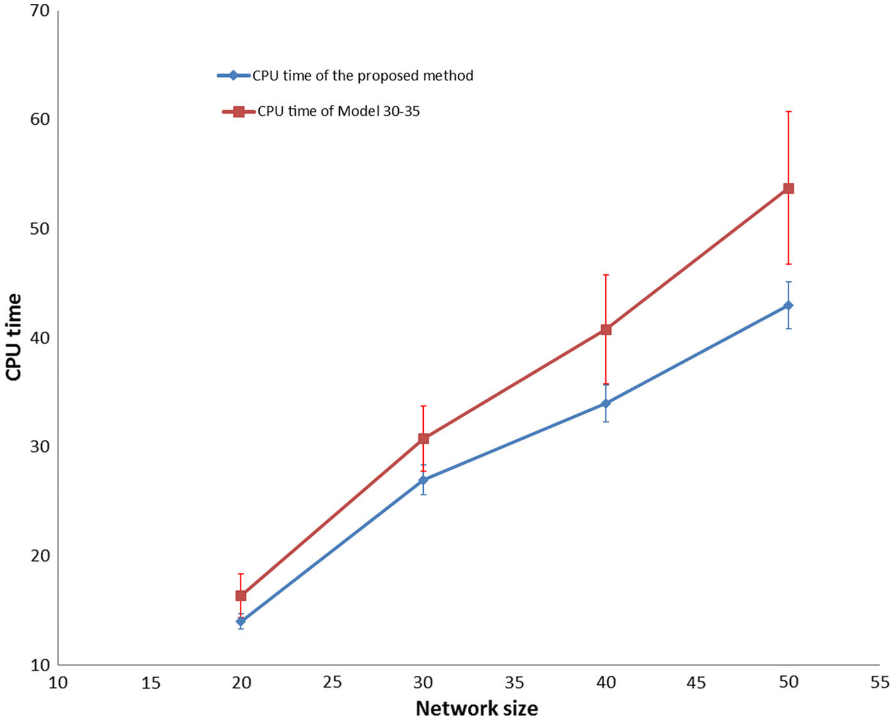


Fig. 3 Comparison between the CPU times (in seconds) of the proposed method and Model 30–35

5 Conclusions

Network flow problems, in general, are defined over weighted directed graphs. These problems are easily tractable when one single weight is considered on the arcs of a given network. However, in many real-life applications, multiple weights are imposed on the arcs. Then, solving the problem becomes more difficult. In this paper, we proposed an efficient method for solving a resource allocation problem by considering several weights on the arcs of a network. Our approach calculates the relative efficiency of each arc in the presence of multiple weights imposed on it using DEA techniques. We examined two models of efficiency: the CCR-efficiency model and the cross efficiency model. The information obtained from these two models helps to identify an efficient subgraph through a linear programming model. The advantages of this method are that (1) a subgraph will be obtained in a polynomial time, (2)that it has the integrality property, and (3) that the graphs are highly efficient in terms of the weights of their edges. Several numerical experiments confirmed the efficiency and applicability of our approach.

Possible application areas that we consider for our method are wireless sensor networks (WSNs) and wireless body area networks (WBANs), which can be used for a wide variety of systems with vastly varying requirements and characteristics. WSNs consist of small, power-constrained sensor nodes that are deployed to sense physical or environmental conditions and communicate data to a base station. WSNs, unlike many other networks, are designed for specific applications which require to satisfy different types of constraints. Thus, energy efficiency and routing problems are a critical problem in WSNs, which also need to guarantee quality of service (QoS) requirements such as reducing interference, latency, reducing the number of installed nodes and increasing coverage, connectivity, network lifetime, and the profit that the information brings to the user, i.e., the value of information (VoI) (Singh et al. 2019).

WBANs are special case of WSN to monitor patients remotely, along with the development of several other applications at low cost. Due to life-criticality of the detection of emergencies such as heart attacks and sudden falls, WBANs have to support the mobility of patients *and* guarantee QoS requirements, e.g., hard real-time data delivery delay limits, reliability, confidentiality, access control, throughput, end-to-end delay, packet transmission rates, dynamic reconfiguration, efficient traffic management, security and energy efficiency (Niu et al. 2019; Khan et al. 2018).

With our proposed approach, we can solve exactly such problems. We obtain the relative efficiency of links regarding to QoS parameters and solve the routing problem by using the relative efficiency of links. Our future work will therefore focus on practically applying our methods in the above scenarios.

Author Contributions Dr. Raayatpanah has conceptualized and conducted the research. Dr. Khodayifar has helped developing the mathematical modules. Dr. Weise and Dr. Pardalaos both have helped writing, rewriting, and revising the manuscript.

Funding We acknowledge funding support by the National Natural Science Foundation of China under Grants 61673359 and 71520107002. P.M. Pardalos was conducted within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Consent for publication All authors agree to the publication of the article.

References

Ahuja RK, Magnanti TL, Orlin JB, et al (1995) Applications of network optimization. In: Ball MO, Magnanti TL, Monma CL et al (eds) Network models, handbooks in operations research and management science, vol 7. Elsevier B. V., Cambridge, MA, USA, chap 1, pp 1–83. https://doi.org/10.1016/S0927-0507(05)80118-5

Ahuja RK, Magnanti TL, Orlin JB (1993) Network flows—theory, algorithms and applications. Prentice Hall, Englewood Cliffs

Amirteimoori A (2011) An extended transportation problem: a DEA-based approach. Central Eur J Oper Res (CEJOR) 19(4):513–521. https://doi.org/10.1007/s10100-010-0140-0

Barabási A, Albert R (1999) Emergence of scaling in random networks. Science 286(5439):509–512. https://doi.org/10.1126/science.286.5439.509

Bazaraa MS, Jarvis JJ, Sherali HD (2011) Linear programming and network flows. Wiley, Hoboken

Belling-Seib K, Mevert P, Müller C (1988) Network flow problems with one side constraint: a comparison of three solution methods. Comput Oper Res 15(4):381–394. https://doi.org/10.1016/0305-0548(88)90022-6

Bollobás B (2011) Random Graphs, 2nd edn. No. 73 in Cambridge Studies in Advanced Mathematics, Cambridge University Press, Cambridge. https://doi.org/10.1017/CBO9780511814068

Bryson N (1991) Parametric programming and Lagrangian relaxation: the case of the network problem with a single side-constraint. Comput Oper Res 18(2):129–140. https://doi.org/10.1016/0305-0548(91)90084-5

Charnes A, Cooper WW (1962) Programming with linear fractional functions. Naval Res Logist 9:181–186. https://doi.org/10.1002/nav.3800090303

Charnes A, Cooper WW, Rhodes E (1978) Measuring the efficiency of decision making units. Eur J Oper Res (EJOR) 2:429–444. https://doi.org/10.1016/0377-2217(78)90138-8

Chaudhry IA, Khan AA (2016) A research survey: review of flexible job shop scheduling techniques. Int Trans Oper Res (ITOR) 23(3):551–591. https://doi.org/10.1111/itor.12199

Chen P (2021) Effects of the entropy weight on TOPSIS. Expert Syst Appl 168(114):186. https://doi.org/10.1016/j.eswa.2020.114186

Chen L, Lu H (2007) An extended assignment problem considering multiple inputs and outputs. Appl Math Model 31:2239–2248. https://doi.org/10.1016/j.apm.2006.08.018

Chen S, Saigal R (1977) A primal algorithm for solving a capacitated network flow problem with additional linear constraints. Networks 7(1):59–79. https://doi.org/10.1002/net.3230070105

Cook WD, Tone K, Zhu J (2014) Data envelopment analysis: prior to choosing a model. Omega 44:1–4. https://doi.org/10.1016/j.omega.2013.09.004

Cooper WW, Seiford LM, Tone K (2006) Introduction to data envelopment analysis and its uses: with DEA-solver software and references. Springer, New York. https://doi.org/10.1007/0-387-29122-9

Deb K (2001) Multi-objective optimization using evolutionary algorithms. Wiley, Hoboken

Deo N (1974) Graph theory with applications to engineering and computer science. Prentice Hall Series in Automatic Computation. Prentice-Hall Inc, Upper Saddle River

Doyle J, Green R (1994) Efficiency and cross-efficiency in DEA: derivations, meanings and uses. J Oper Res Soc 45:567–578. https://doi.org/10.1057/jors.1994.84

Du D, Pardalos PM (1993) Network optimization problems: algorithms, applications and complexity, vol 2. Series on applied mathematics. World Scientific, Singapore

Dürr C, Jez L, Vásquez OC (2015) Scheduling under dynamic speed-scaling for minimizing weighted completion time and energy consumption. Discret Appl Math 196:20–27. https://doi.org/10.1016/j.dam.2014.08.001

Ehrgott M (2005) Multicriteria optimization, 2nd edn. Springer, Berlin. https://doi.org/10.1007/3-540-27659-9

Erdös P, Rényi A (2006) On the evolution of random graphs. In: Newman M, Barabási A, Watts DJ (eds) The structure and dynamics of networks, vol 19. Princeton studies in complexity. Princeton University Press, Princeton, pp 38–82. https://doi.org/10.1515/9781400841356.38

Fang S, Qi L (2003) Manufacturing network flows: a generalized network flow model for manufacturing process modelling. Optim Methods Softw 18(2):143–165. https://doi.org/10.1080/1055678031000152079

Farrell M (1957) The measurement of productive efficiency. J R Stat Soc Ser A (General) 120(3):253–290

Glover F, Klingman D (1981) The simplex SON algorithm for LP/embedded network problems. In: Mulvey JM, Klingman D (eds) Network models and associated applications, mathematical programming studies, vol 15. Springer, New York, pp 148–176. https://doi.org/10.1007/BFb0120942

Glover F, Klingman D, Ross GT (1974) Finding equivalent transportation formulations for constrained transportation problems. Naval Res Log Q 21(2):247–254

Glover F, Klingman D (1985) Basis exchange characterizations for the simplex SON algorithm for LP/embedded networks. In: Cottle RW (ed) Mathematical programming essays in honor of George B. Dantzig, Part I, mathematical programming Studies, vol 24. Springer, New York, pp 141–157. https://doi.org/10.1007/BFb0121048

Guo L (2016) Efficient approximation algorithms for computing k disjoint constrained shortest paths. J Comb Optim 32(1):144–158. https://doi.org/10.1007/s10878-015-9934-2

Hamacher HW, Pedersen CR, Ruzika S (2007) Multiple objective minimum cost flow problems: a review. Eur J Oper Res (EJOR) 176(3):1404–1422. https://doi.org/10.1016/j.ejor.2005.09.033

Hollander M, Wolfe DA, Chicken E (2014) Nonparametric statistical methods, 3rd edn. Wiley series in probability and statistics. Wiley, Hoboken. https://doi.org/10.1002/9781119196037

Holzhauser M, Krumke SO, Thielen C (2016) Budget-constrained minimum cost flows. J Comb Optim 31(4):1720–1745. https://doi.org/10.1007/s10878-015-9865-y

Hu Y, Zhao X, Liu J et al (2020) An efficient algorithm for solving minimum cost flow problem with complementarity slack conditions. Math Probl Eng 2439265:1–5. https://doi.org/10.1155/2020/2439265

Khan RA, Mohammadani KH, Soomro AA et al (2018) An energy efficient routing protocol for wireless body area sensor networks. Wireless Pers Commun 99(4):1443–1454. https://doi.org/10.1007/s11277-018-5285-5

Klingman D (1977) Finding equivalent network formulations for constrained network problems. Manag Sci 23(7):737–744. https://doi.org/10.1287/mnsc.23.7.737

Kovács P (2015) Minimum-cost flow algorithms: an experimental evaluation. Optim Methods Softw 30(1):94–127. https://doi.org/10.1080/10556788.2014.895828

Lehmann EL (2006) Nonparametrics: statistical methods based on ranks. Springer, New York

Li B, Springer J, Bebis G et al (2013) A survey of network flow applications. J Netw Comput Appl 36(2):567–581. https://doi.org/10.1016/j.jnca.2012.12.020

Longaray AA, Ensslin L, Ensslin SR et al (2018) Using MCDA to evaluate the performance of the logistics process in public hospitals: the case of a Brazilian teaching hospital. Int Trans Oper Res (ITOR) 25(1):133–156. https://doi.org/10.1111/itor.12387

Lu H, Yao E, YaoQi L (2006) Some further results on minimum distribution cost flow problems. J Comb Optim 11(4):351–371. https://doi.org/10.1007/s10878-006-8211-9

Mamer JW, McBride RD (2000) A decomposition-based pricing procedure for large-scale linear programs: an application to the linear multicommodity flow problem. Manage Sci 46(5):693–709. https://doi.org/10.1287/mnsc.46.5.693.12042

Mann HB, Whitney DR (1947) On a test of whether one of two random variables is stochastically larger than the other. Ann Math Stat 18:50–60. https://doi.org/10.1214/aoms/1177730491

Marler RT, Arora JS (2004) Survey of multi-objective optimization methods for engineering. Struct Multidiscip Optim 26(6):369–395. https://doi.org/10.1007/s00158-003-0368-6

Mathies S, Mevert P (1998) A hybrid algorithm for solving network flow problems with side constraints. Comput Oper Res 25(9):745–756. https://doi.org/10.1016/S0305-0548(98)00001-X

McBride RD (1985) Solving embedded generalized network problems. Eur J Oper Res (EJOR) 21(1):82–92

McBride RD, Mamer JW (1997) Solving multicommodity flow problems with a primal embedded network simplex algorithm. INFORMS J Comput 9(2):154–163. https://doi.org/10.1287/ijoc.9.2.154

McBride RD, Mamer JW (2001) Solving the undirected multicommodity flow problem using a shortest path-based pricing algorithm. Networks 38(4):181–188. https://doi.org/10.1002/net.1035

Mo J, Qi L, Wei Z (2005) A manufacturing supply chain optimization model for distilling process. Appl Math Comput 171(1):464–485. https://doi.org/10.1016/j.amc.2005.01.041

Niu S, Matsuhisa N, Beker L et al (2019) A wireless body area sensor network based on stretchable passive tags. Nat Electron 2(8):361–368. https://doi.org/10.1038/S41928-019-0286-2

Orlin JB (1993) A faster strongly polynomial minimum cost flow algorithm. Oper Res 41(2):338–350. https://doi.org/10.1287/opre.41.2.338

Orlin JB (1997) A polynomial time primal network simplex algorithm for minimum cost flows. Math Program 78(2):109–129. https://doi.org/10.1007/BF02614365

Raayatpanah MA (2017a) Multicast routing based on data envelopment analysis with multiple quality of service parameters. Int J Commun Syst. https://doi.org/10.1002/dac.3084

Raayatpanah MA (2017b) Subgraph selection over coded packet networks with multiple QOS parameters based on DEA. Trans Emerg Telecommun Technol. https://doi.org/10.1002/ett.3079

Raayatpanah MA, Fathabadi HS, Khalaj BH et al (2014) Bounds on end-to-end statistical delay and jitter in multiple multicast coded packet networks. J Netw Comput Appl 41:217–227. https://doi.org/10.1016/j.jnca.2013.12.004

Sedeño-Noda A, Alonso-Rodríguez S (2015) An enhanced K-SP algorithm with pruning strategies to solve the constrained shortest path problem. Appl Math Comput 265:602–618. https://doi.org/10.1016/j.amc.2015.05.109

Sexton TR, Silkman RH, Hogan AJ (1986) Data envelopment analysis: critique and extensions. New Direct Program Eval 32:73–105. https://doi.org/10.1002/ev.1441

Singh AK, Mishra SK, Dixit S (2019) Energy efficiency in wireless sensor networks: Cooperative MIMO-OFDM. In: Recent trends in communication, computing, and electronics: select proceedings of IC3E 2018, Lecture notes in electrical engineering, vol 524. Springer, Singapore, pp 147–154. https://doi.org/10.1007/978-981-13-2685-1_16

Sokkalingam PT, Ahuja RK, Orlin JB (2000) New polynomial-time cycle-canceling algorithms for minimum-cost flows. Netw: Int J 36(1):53–63

Song L, Liu F (2018) An improvement in DEA cross-efficiency aggregation based on the Shannon entropy. Int Trans Oper Res (ITOR) 25(2):705–714. https://doi.org/10.1111/itor.12361

Spälti SB, Liebling TM (1991) Modeling the satellite placement problem as a network flow problem with one side constraint. Operations-Research-Spektrum 13(1):1–14. https://doi.org/10.1007/BF01719766

Tran CTT, Villano RA (2018) Measuring efficiency of Vietnamese public colleges: an application of the DEA-based dynamic network approach. Int Trans Oper Res (ITOR) 25(2):683–703. https://doi.org/10.1111/itor.12212

Tzeng G, Huang J (2011) Multiple attribute decision making: methods and applications. Chapman and Hall, London

Venkateshan P, Mathur K, Ballou RH (2008) An efficient generalized network-simplex-based algorithm for manufacturing network flows. J Comb Optim 15(4):315–341. https://doi.org/10.1007/s10878-007-9080-6

Vogiatzis C, Pardalos PM (2013) Combinatorial optimization in transportation and logistics networks. In: Pardalos PM, Du D, Graham RL (eds) Handbook of combinatorial optimization, 2nd edn. Springer, New York, pp 673–722. https://doi.org/10.1007/978-1-4419-7997-1_63

Watts DJ (2003) Small worlds: the dynamics of networks between order and randomness. Princeton studies in complexity. Princeton University Press, Princeton

Waxman BM (1988) Routing of multipoint connections. IEEE J Sel Areas Commun 6(9):1617–1622. https://doi.org/10.1109/49.12889

Yang C, Zhao X (2014) A new delay-constrained multicast routing algorithm based on shared edges. Commun Netw. https://doi.org/10.4236/cn.2014.61006

Yang G, Yang J, Liu W et al (2013) Cross-efficiency aggregation in DEA models using the evidential-reasoning approach. Eur J Oper Res (EJOR) 231(2):393–404. https://doi.org/10.1016/j.ejor.2013.05.017