European Journal of Operational Research 199 (2009) 68–76

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

# On the computation of all supported efficient solutions in multi-objective integer network flow problems

Augusto Eusébio [a,b,*,1], José Rui Figueira [b,c,1]

[a] Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Morro do Lena – Alto Vieiro, 2411-901 Leiria, Portugal
[b] CEG-IST, Center for Management Studies, Instituto Superior Técnico, Technical University of Lisbon, TagusPark, Av. Cavaco Silva, 2780-990 Porto Salvo, Portugal
[c] Associate Researcher at LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75 775 Paris Cedex 16, France

ARTICLE INFO

*Article history:*
Received 1 October 2007
Accepted 28 October 2008
Available online 8 November 2008

*Keywords:*
Multi-objective linear and integer programming
Multi-objective network flows
Negative-cycle algorithms
Parametric programming

ABSTRACT

This paper presents a new algorithm for identifying all supported non-dominated vectors (or outcomes) in the objective space, as well as the corresponding efficient solutions in the decision space, for multi-objective integer network flow problems. Identifying the set of supported non-dominated vectors is of the utmost importance for obtaining a first approximation of the whole set of non-dominated vectors. This approximation is crucial, for example, in two-phase methods that first compute the supported non-dominated vectors and then the unsupported non-dominated ones. Our approach is based on a negative-cycle algorithm used in single objective minimum cost flow problems, applied to a sequence of parametric problems. The proposed approach uses the connectedness property of the set of supported non-dominated vectors/efficient solutions to find all integer solutions in maximal non-dominated/efficient facets.



## 1. Introduction

Network flow models are frequently encountered in the literature in the field of network optimization. These models are, in general, used to model a variety of real-world decision-making problems in a wide range of areas, such as transportation, telecommunications, biology, medicine, economics and finance, for example (see [1]). However, the methods, algorithms, and applications commonly found in the literature are mostly designed to optimize a unique objective function, whereas reality is, by its very nature, primarily multi-dimensional. Thus, it would seem that multi-objective network flow models would be more appropriate for modelling real-world decision-making situations in the field of network optimization. Despite the above incongruity, research applying multi-objective network flow models to real-world network decision-making is rather scarce.

In our review of the literature, relatively few works were found that extend the minimum cost network flow problem to several objectives, and these deal with only two objectives. (For a recent survey on the topic, see [12].) The multi-objective network flow (MONF) problem can also be expressed as a mathematical programming model, which has two variants: the linear MONF problem (MOLNF) and its integer version (MOINF). MOLFN problems contain only supported non-dominated vectors/efficient solutions. MOINF problems, on the other hand, contain both unsupported and supported non-dominated vectors/efficient solutions, which can be geometrically characterized as follows: the unsupported non-dominated vectors are located inside the feasible region in the objective space, while the supported vectors are found on the boundaries of the convex hull of this feasible region. Supported non-dominated vectors correspond to the optimal solutions of a sequence of single objective parametric network flow problems. To the best of our knowledge, there is no specific method designed to determine all the non-dominated vectors in the objective space (nor the corresponding efficient solutions in the decision or variable space) for MOINF problems, though the following algorithms have been proposed for the bi-objective integer network flow (BOINF) problem:

* Corresponding author. Address: Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Morro do Lena – Alto Vieiro, 2411-901 Leiria, Portugal. Tel.: +351 244 820 300; fax: +351 244 820 310.
*E-mail addresses:* aeusebio@estg.ipleiria.pt (A. Eusébio), figueira@ist.utl.pt (J.R. Figueira).
1 Tel.: +351 21 423 35 07; fax: +351 21 423 35 68.

0377-2217/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.ejor.2008.10.031

1. An explicit enumeration algorithm [8,10], which first uses the $k$-best flow algorithm proposed by Hamacher [11] to explicitly explore the feasible region in both the objective and decision spaces by enumerating all the feasible flows one by one and then uses a filtering procedure [8] to retain only the non-dominated vectors/efficient solutions.
2. An implicit enumeration algorithm [5,7], which solves a sequence of e-constraint problems by computing optimal non-integer solutions with a network simplex algorithm and then determining optimal integer solutions with a branch-and-bound technique. Despite its use of a branch-and-bound technique, this implicit algorithm has the advantage of computing non-dominated vectors/efficient solutions without destroying the network structure.
3. Two-phase algorithms [14,17], which apply an out-of-kilter or a network simplex method in the first phase to compute the supported non-dominated vectors/efficient solutions, and then apply the connectedness property to the non-dominated vectors in the second phase to obtain the unsupported non-dominated vectors.

Despite their conceptual interest, the two-phase algorithms have proved to be inappropriate for solving BOINF problems [15]. One of the main drawbacks of these two-phase algorithms is related to the way the supported solutions are computed. As we will also show in this paper, these algorithms are not able to find all the supported non-dominated vectors. The other two types of algorithms are appropriate for solving BOINF problems (i.e., the correctness of these algorithms can be proved).

In this paper, we present an algorithm that can be used to compute the supported non-dominated vectors/efficient solutions for both MOINF and BOINF problems. Identifying the whole or partial set of supported non-dominated vectors or outcomes and the corresponding efficient solutions in the decision space has several advantages: (1) the algorithm can be used to compute all the supported solutions in the first phase of a two-phase approach for MOINF problems, (2) it can be used to provide a first approximation of the non-dominated set for the MOINF problems, and (3) it can serve as an interactive procedure in methods that provide a detailed "view" of the supported non-dominated vectors in regions of the objective space that are of particular interest to the decision-maker.

The algorithm that we propose is based on a negative-cycle algorithm rather than on a simplex or an out-of-kilter algorithm, thus providing an alternative that, to our best knowledge, has not been applied to MOINF problems before. The proposed algorithm is the first step towards the development of more negative-cycle algorithms for solving MOINF problems. It has several advantages. For example, it can deal with more than two objectives, and it is easy to implement.

The paper is organized as follows. Section 2 presents the concepts, definitions, and notation used in MOINF problems. Section 3 introduces our new algorithm for finding all the supported non-dominated vectors/efficient solutions. Section 4 offers an illustrative example, and the last section provides our conclusions.

## 2. The multi-objective network flow problem

The MOINF can be modeled by using the concepts of network optimization. Let $\mathcal{N} = (\mathcal{G}, (c^1, c^2, \ldots, c^p), l, u, b)$ denote a network, where $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ is a directed and connected graph; $\mathcal{S} = \{1, 2, \ldots, k, \ldots, m\}$, with $m \geqslant 2$, is a finite set of elements called *nodes* or *vertices*; $\mathcal{A} = \{(i_1, j_1), (i_2, j_2), \ldots, (i, j), \ldots, (i_n, j_n)\}$, with $n \geqslant 1$ is a collection of ordered pairs of nodes called *arcs*; $c^1, c^2, \ldots, c^p$, $l$ and $u$ are vectors such that for each arc $(i, j) \in \mathcal{A}$, $c^1_{ij}, c^2_{ij}, \ldots, c^p_{ij}$ are the unit "costs" (associated to the objectives) along the arc $(i, j)$, $l_{ij}$ and $u_{ij}$ are the lower and upper *bound* or *capacity*, respectively, for the arc $(i, j)$; and each component of the vector $b$, $b_k$, is the available supply (if $b_k > 0$) or the demand (if $b_k < 0$) at node $k \in \mathcal{S}$ (if $b_k = 0$ it is considered as a transhipment node). We assume that $m \leqslant n$, $\sum_{k \in \mathcal{S}} b_k = 0$, and that all the values for the "costs", lower, and upper bounds on the arcs, and supplies/demands on the vertices, are finite and integer; all arc "costs" are also non-negative. Consider, without loss of generality, that $l_{ij} = 0\ \forall (i, j) \in \mathcal{A}$. Fig. 1 shows an example of a bi-objective network flow problem.

The MOINF problem associated with the network $\mathcal{N}$ can also be stated as a multi-objective integer linear programming problem as follows:

$$\text{"minimize"} \quad \left( \sum_{(i,j) \in \mathcal{A}} c^1_{ij} x_{ij}, \sum_{(i,j) \in \mathcal{A}} c^2_{ij} x_{ij}, \ldots, \sum_{(i,j) \in \mathcal{A}} c^p_{ij} x_{ij} \right) \tag{1}$$

$$\text{subject to:} \quad \sum_{(k,j) \in \mathcal{A}} x_{kj} - \sum_{(i,k) \in \mathcal{A}} x_{ik} = b_k \quad \forall k \in \mathcal{S}, \tag{1.a}$$

$$0 \leqslant x_{ij} \leqslant u_{ij} \quad \forall (i, j) \in \mathcal{A}, \tag{1.b}$$

$$x_{ij} \text{ integer} \quad \forall (i, j) \in \mathcal{A}. \tag{1.c}$$

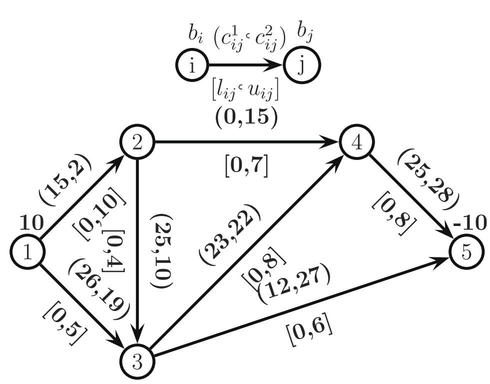


**Fig. 1.** A bi-criteria example.

When considering the matricial notation, this problem can be defined in a more compact way as follows:

$$\begin{aligned} &\text{“min”} \quad && f(x) = Cx \\ &\text{subject to :} \quad && x \in X^I, \end{aligned} \tag{2}$$

where $C$ is the "cost" matrix with rows $c^1, c^2, \dots, c^p$, with $c^k = (c^k_{i_1j_1} c^k_{i_2j_2} \cdots c^k_{i_nj_n})$, for $k = 1, 2, \dots, p$ and $X^I = \{x = (x_{i_1j_1}, \dots, x_{i_nj_n}) : \sum_{(k,j)\in\mathscr{A}} x_{kj} - \sum_{(i,k)\in\mathscr{A}} x_{ik} = b_k \ \forall k \in \mathscr{S}, 0 \leqslant x_{ij} \leqslant u_{ij}, x_{ij} \text{ integer } \forall (i,j) \in \mathscr{A}\}$ is the feasible region in the variable space. Let $Y^I = f(X^I) = \left\{y = (y_1, y_2, \dots, y_p) : y_1 = \sum_{(i,j)\in\mathscr{A}} c^1_{ij} x_{ij}, y_2 = \sum_{(i,j)\in\mathscr{A}} c^2_{ij} x_{ij}, \dots, y_p = \sum_{(i,j)\in\mathscr{A}} c^p_{ij} x_{ij}\right\}$ be the feasible region in the objective space. The set $X^I$ ($Y^I$) contains thus the *feasible solutions* in the *decision space*, $\mathbb{N}^n_0$ (*objective space*, $\mathbb{N}^p_0$). The elements of $X^I$ ($Y^I$) will be called *feasible solutions* (*feasible vectors*). Let $X = Conv(X^I)$ and $Y = Conv(Y^I)$ be the convex hulls of the sets $X^I$ and $Y^I$, respectively. The MONF problem has $X$ and $Y$ as feasible regions in the decision and objective spaces, respectively.

Consider two feasible vectors $y'$ and $y''$ in $Y^I$ ($Y$) for the MOINF (MONF) problem. The vector $y'$ *dominates* $y''$ iff $y' \leqq y''$ and $y' \neq y''$, that is, $y'_q \leqslant y''_q$ for all $q = 1, 2, \dots, p$ with at least one strict inequality. The vector $y'$ is called *non-dominated* iff there does not exist another vector $y$ in $Y^I$ ($Y$) such that $y \leqq y'$ and $y \neq y'$. Otherwise, $y'$ is a *dominated* vector. The set of all non-dominated vectors in $Y^I$ is denoted by $ND(Y^I)$. A solution $x'$ in $X^I$ ($X$) is said to be *efficient* iff it is impossible to find another solution $x$ in $X^I$ ($X$) with a better performance on a given objective without deteriorating the performances of at least one of the remaining objectives. In other words $x'$ is an efficient solution if $y' = f(x')$ is a non-dominated vector in $Y^I$ ($Y$).

The following theorem shows that the set of efficient solutions, in $X$, can be obtained by solving a parametric problem (see [18, p. 215]).

**Theorem 1.1.** *A feasible solution $x \in X$ is efficient if and only if there exists a*

$$\lambda \in \Lambda = \left\{\lambda \in \mathbb{R}^p : \sum_{k=1}^{p} \lambda_k = 1 \text{ and } \lambda_k > 0, \ k = 1, 2, \dots, p\right\}$$

such that $x$ minimizes the weighted-sum linear programme $\min\{\lambda^T Cx : x \in X\}$.

As it was pointed out in the introduction, in MOINF problems two types of non-dominated vectors can be distinguished: supported and unsupported non-dominated vectors. Let $Y^{\geqq} = Conv(ND(Y^I) + \mathbb{R}^p_{\geqq})$, where $\mathbb{R}^p_{\geqq} = \{y \in \mathbb{R}^p | y \geqq 0\}$ and $ND(Y^I) + \mathbb{R}^p_{\geqq} = \{y \in \mathbb{R}^p : y = y' + y'', y' \in ND(Y^I) \text{ and } y'' \in \mathbb{R}^p_{\geqq}\}$, $y \geqq 0$ if $y_q \geqslant 0$, $q = 1, 2, \dots, p$. A non-dominated vector $y$ on the boundary of $Y^{\geqq}$ is said to be a *supported* non-dominated vector. Otherwise, $y$ is an *unsupported* non-dominated vector. A supported non-dominated vector $y$ associated with an extreme point of $Y^{\geqq}$ is said to be a *supported-extreme vector*. Otherwise, $y$ is a *supported non-extreme vector*. Inverse images of supported non-dominated vectors are said to be *supported efficient solutions* and inverse images of unsupported non-dominated vectors are said to be *unsupported efficient solutions*.

It is well known that the set $X$ is a polytope, since the bounds $l_{ij}$ and $u_{ij}$ are finite. Consequently, $Y$ is a polytope too. Let us consider the concepts of face, facet, maximal, and maximally non-dominated facets as in [18].

Let $F \subset Y$ and $H$ be a supporting hyperplane of $Y$. $F$ is a *facet* of $Y$ iff there exists an $H$ such $H \cap Y = F$. $F$ is said to be an *r-facet* of $Y$ if $F$ is of dimensionality $r$. Extreme points are 0-facets and 1-facets are *edges*. When $Y \subset \mathbb{R}^3$ a 2-facet is called a *face*. A $r - facet$, $F \subset Y$, is a maximal facet iff there does not exist another $s - facet$, $G \subset Y$, such that $F \subset G$ and $r < s$. A facet is said to be non-dominated if all its points are non-dominated. A $r$-facet, $F \subset Y$, is a maximally non-dominated facet iff there is no non-dominated $s$-facet, $G \subset Y$, such that $F \subset G$ and $r < s$. We will use the expression maximally efficient facet of $X$ as the inverse image of a maximally non-dominated facet of polytope $Y$.

An intermediate vector is a point of an $r$-facet, $r \geqslant 1$, that belongs to the line segment between two adjacent vertices. It is known that all vertices of polytope $X$ are associated with feasible solutions that have integer coordinates [2]. Each vertex of the polytope $Y$ is the image of at least a vertex of the polytope $X$, but one vertex of $Y$ can be image of more than one vertex of $X$, and not all vertices of $X$ have as image a single vertex of $Y$. Vertices of $Y$ can be images of non-extreme points of $X$. Not all supported non-dominated vectors are images of extreme points or intermediate efficient points. To illustrate this concept we consider the example of Fig. 1. The BOINF problem associated with this network is the following:

$$\begin{aligned}
&\text{“min”} && (15x_{12} + 26x_{13} + 25x_{23} + 23x_{34} + 12x_{35} + 25x_{45}, 2x_{12} + 19x_{13} + 10x_{23} + 15x_{24} + 22x_{34} + 27x_{35} + 28x_{45}) \\
&\text{subject to :} && x_{12} + x_{13} = 10, \\
& && -x_{12} + x_{23} + x_{24} = 0, \\
& && -x_{13} - x_{23} + x_{34} + x_{35} = 0, \\
& && -x_{24} - x_{34} + x_{45} = 0, \\
& && x_{35} + x_{45} = 10, \\
& && 0 \leqslant x_{12} \leqslant 10; \ 0 \leqslant x_{13} \leqslant 5; \ 0 \leqslant x_{23} \leqslant 4; \ 0 \leqslant x_{24} \leqslant 7; \ 0 \leqslant x_{34} \leqslant 8; \ 0 \leqslant x_{35} \leqslant 6; \ 0 \leqslant x_{45} \leqslant 8, \text{ and integer},
\end{aligned}$$

which is equivalent to the following linear programming:

$$\begin{aligned}
&\text{“min”} && (15x_{12} + 26x_{13} + 25x_{23} + 23x_{34} + 12x_{35} + 25x_{45}, 2x_{12} + 19x_{13} + 10x_{23} + 15x_{24} + 22x_{34} + 27x_{35} + 28x_{45}) \\
&\text{subject to :} && x_{12} + x_{13} = 10. \\
& && x_{13} + x_{23} + x_{24} = 10, \\
& && x_{24} + x_{34} + x_{35} = 10, \\
& && x_{24} + x_{34} - x_{45} = 0, \\
& && 0 \leqslant x_{12} \leqslant 10; \ 0 \leqslant x_{13} \leqslant 5; \ 0 \leqslant x_{23} \leqslant 4; \ 0 \leqslant x_{24} \leqslant 7; \ 0 \leqslant x_{34} \leqslant 8; \ 0 \leqslant x_{35} \leqslant 6; \ 0 \leqslant x_{45} \leqslant 8, \text{ and integer}.
\end{aligned}$$

Considering the variables $x_{12}$, $x_{23}$, $x_{35}$, and $x_{45}$ as slack variables in the first, second, third, and fourth constraints, respectively, the feasible region can be represented in $\mathbb{R}^3$ by using only 3 decision variables, $x_{13}$, $x_{24}$ and $x_{34}$ (see [2]). The convex hull of the feasible region is the

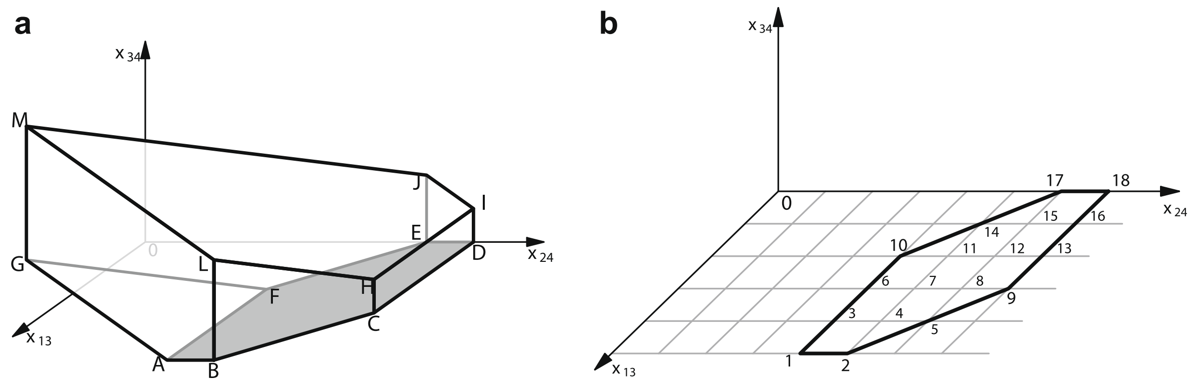


**Fig. 2.** (a) Feasible region and (b) efficient solutions.

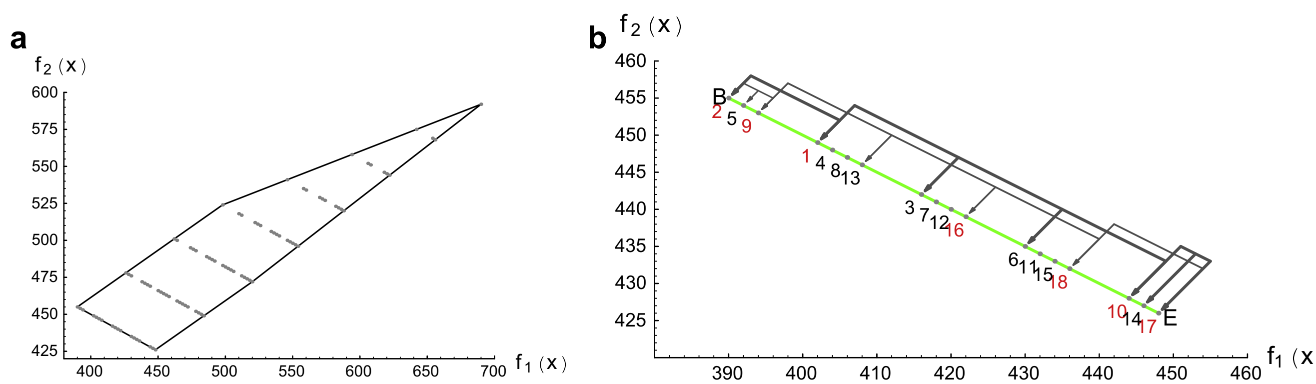


**Fig. 3.** (a) Feasible region and (b) non-dominated vectors.

polytope in Fig. 2a. The supported efficient solutions are represented in Fig. 2b by the points numbered $1, 2, \ldots, 18$. Fig. 3a and b represents the set of feasible vectors and non-dominated vectors, respectively, in the objective space. This example shows several supported non-dominated vectors that are not image of intermediate solutions. These vectors are represented in Fig. 3b by the points: 4, 7, 8, 11, 12, and 15. All these points are images of non-intermediate solutions. The algorithms [14,17] are not able to find these points.

For example, the solution 4 (Fig. 3b) is not an intermediate solution, it cannot be obtained as a linear combination between two adjacent extreme points. It is rather a linear combination between two intermediate solutions. These kind of operations are not performed in algorithms [14,17]. Those algorithms are thus inappropriate for solving bi-objective network flow problems.

## 3. A negative-cycle algorithm for MOINF problems

This section presents a new algorithm to find all supported non-dominated vectors and all supported efficient solutions for MOINF problems. The algorithm makes use of the connectedness of the solutions associated with directed cycles of cost zero in residual networks. This is proved in Theorem 2.1.

Consider the network $\mathcal{N}$ and a feasible solution $x^0$ for the single objective integer network flow (SOINF) problem, i.e., the problem with the same set of constraints as in (1), but with only one objective. The *residual network*, $\mathcal{N}(x^0)$, with respect to the given flow $x^0$ is the network that results from $\mathcal{N}$ by replacing each arc $(i,j)$ with two arcs $(i,j)$ and $(j,i)$: the arc $(i,j)$ has a cost value $c_{ij}$ and a *residual capacity* $r_{ij} = u_{ij} - x^0_{ij}$, and the arc $(j,i)$ has a cost value $-c_{ij}$ and a residual capacity $r_{ij} = x^0_{ij}$. The residual network consists only of the arcs with a positive residual capacity. It can be shown that every flow $x$ in the network $\mathcal{N}$ corresponds to a flow $x'$ in the residual network $\mathcal{N}(x^0)$ [1]. The sum $\sum \gamma_{ij} c_{ij}$ for all arcs $(i,j)$ in a cycle is called *cost of the cycle*, where $\gamma_{ij}$ is equal to 1 if the arc $(i,j)$ is a direct or forward arc in the cycle and $\gamma_{ij}$ is equal to $-1$ if the arc $(i,j)$ is a reverse or backward arc in the cycle. A cycle (not necessarily directed) in $\mathcal{N}$ is called *augmenting cycle* with respect to a flow $x$ if when augmenting a positive amount of flow on the arcs in the cycle, the new flow remains feasible. Therefore, an augmenting cycle cannot contain backward arcs $(i,j)$ such that $x_{ij} = l_{ij}$ or forward arcs such that $x_{ij} = u_{ij}$. Each augmenting cycle with respect to a flow $x$ corresponds to a directed cycle in the residual network $\mathcal{N}(x)$, and *vice versa* [1,2].

The optimality of a solution, $x^*$, for SOINF problem can be evaluated through the cost of the directed cycles in the residual network. It is well known that a feasible solution $x^*$ is an optimal solution if and only if the residual network $\mathcal{N}(x^*)$ contains no negative directed cycles. We define the reduced cost of an arc $(i,j)$ as the quantity $\bar{c}_{ij} = c_{ij} - \pi_i + \pi_j$, where $\pi_i$ is the linear programming dual variable corresponding to the constraint of node $i$ in Eqs. (1.a). This definition is applicable to the residual network as well as the original one. The solution $x^*$ is an optimal solution for SOINF problem iff $\bar{c}_{ij} \geqslant 0$ for every arc $(i,j)$ in $\mathcal{N}(x^*)$. Let $W$ denote a directed cycle in the residual network. It is also well known that $\sum_{(i,j)\in W} \bar{c}_{ij} = \sum_{(i,j)\in W} c_{ij}$. Henceforth, we will use only directed cycles in the residual network and we will write cycle instead of directed cycle.

Consider the SOINF problem associated with a network $\mathcal{N}$.

**Definition 2.1.** Let $x'$ and $x''$ be two feasible solutions for the SOINF problem. The solution $x''$ is said to be a *cycle-adjacent solution* of $x'$ if $x''$ is obtained from $x'$ by augmenting $\delta$ units of flow in a cycle in $\mathcal{N}$ corresponding to a cycle $W$ in $\mathcal{N}(x')$, where $\delta = \min\{r_{ij} : (i,j) \in W\}$. The solution obtained by augmenting $\delta_1$ units of flow in this cycle, where $0 < \delta_1 < \delta$, $\delta_1$ integer, is called *cycle-intermediate* solution of $(x', x'')$.

**Proposition 2.1.** *If $x''$ is a cycle-adjacent solution of $x'$ then either $x'$ is a cycle-adjacent solution of $x''$ or $x'$ is a cycle-intermediate solution of $(x'', x''')$, where $x'''$ is a cycle-adjacent solution of $x''$.*

**Proof.** If $x''$ is a cycle-adjacent solution of $x'$ then there is a cycle, $i_1 - a_1 - i_2 - a_2 - \cdots - i_s - a_s - i_1$ ($a_k = (i_k, i_{k+1})$ or $a_k = (i_{k+1}, i_k)$, $k = 1, 2, \ldots, s-1$ and $a_s = (i_s, i_1)$ or $a_s = (i_1, i_s)$), in the network $\mathcal{N}$ such that augmenting $\delta$ units of flow along this cycle leads to the solution $x''$. Consider the solution $x''$ and the former cycle with an opposite direction, $i_1 - a_s - i_s - \cdots - a_2 - i_2 - a_1 - i_1$. The same $\delta$ units of flow in this cycle lead to $x'$. If $W$ is the corresponding cycle in the residual network $\mathcal{N}(x'')$ and $\delta = \min\{r_{ij} : (i,j) \in W\}$ then $x'$ is a cycle-adjacent solution of $x''$. If $\delta < \min\{r_{ij} : (i,j) \in W\}$, $x'$ is a cycle-intermediate solution of $(x'', x''')$, where $x'''$ is a cycle-adjacent solution of $x''$. □

**Proposition 2.2.** *Let $x'$ and $x''$ be two adjacent extreme points of polytope $X$. If $x''$ is a cycle-adjacent solution of $x'$ then $x'$ is also a cycle-adjacent solution of $x''$.*

**Proof.** It is known that if $x'$ and $x''$ are adjacent extreme points there is at least one augmenting cycle in $\mathcal{N}$ that allows to obtain one solution from the other. These augmenting cycles are associated with cycles $W$ and $W'$ in $\mathcal{N}(x')$ and $\mathcal{N}(x'')$, respectively, that define $x''$ as a cycle-adjacent of $x'$ and $x'$ as a cycle-adjacent of $x''$, respectively. □

Consider the parametric integer network flow problem:

$$\begin{aligned} \min \quad & \sum_{k=1}^{p} \lambda_k c^k x, \\ \text{subject to :} \quad & x \in B^{(k)} \end{aligned} \tag{3}$$

for some $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_p) \in \Lambda$ and $B^{(k)} \subseteq X^I$.

In what follows we assume that problem (1) has more than one supported non-dominated vector. We use the term *cycle-sequence* as a sequence of optimal solutions $x^{(1)}, x^{(2)}, \ldots, x^{(p)}$ for problem (3), with $B^{(k)} = X^I$, such that for each pair, $(x^{(q)}, x^{(q+1)})$, $x^{(q+1)}$ is a cycle-adjacent solution of $x^{(q)}$, $q = 1, 2, \ldots, p-1$ for some problem of type (3). A solution $x \in X^I$ is said to be in a cycle-sequence $x^{(1)}, x^{(2)}, \ldots, x^{(p)}$ if $x$ is one of the solutions $x^{(1)}, x^{(2)}, \ldots, x^{(p)}$ or if $x$ is a cycle-intermediate solution of $(x^{(q)}, x^{(q+1)})$, for some $q = 1, 2, \ldots, p-1$.

The supported non-dominated vectors are on the boundary of $Y^{\geqq}$. Next we show that the set of supported non-dominated vectors for the MOINF problem is connected.

**Theorem 2.1.** *Let $y' = f(x')$ and $y'' = f(x'')$ be two supported-extreme non-dominated vectors for problem (1), in the same maximally non-dominated facet, $F_Y$, on the boundary of $Y^{\geqq}$. Then, any supported non-dominated vector in $F_Y$ is image of an efficient solution in a cycle-sequence $x', \ldots, x''$.*

**Proof.** For each maximally efficient facet, $F_X$ of $X$, $f(F_X)$ is a maximally non-dominated facet of $Y$; and for each maximally non-dominated facet, $F_Y$ of $Y$, $f^{-1}(F_Y) \cap X$ is a maximally efficient facet of $X$ (see [3]). Furthermore, it is known that the integer points of a maximally efficient facet, $F_X$, are optimal solutions of problem (3) for a fixed $\lambda^0 = (\lambda_1^0, \lambda_2^0, \ldots, \lambda_p^0) \in \Lambda$ (see, for example, [9]). We are only interested in the points in $F_X$ and $F_Y$ that have integer components.

Next, we will show that given an optimal solution of problem (3), with $B^{(k)} = X^I$, we can find all the remaining optimal solutions in $X^I$ through the cycles of cost zero associated with the residual network of solutions already found. We assume, without loss of generality, that between two nodes, $i$ and $j$, in the network $\mathcal{N}$, there is only one arc, either the arc $(i,j)$ or the arc $(j,i)$.

Given two optimal solutions $x^{(1)}$, $x^{(2)}$ of (3), with $\lambda = \lambda^0$ and $B^{(k)} = X^I$, either $x^{(1)}, \ldots, x^{(2)}$ is a cycle-sequence or $x^{(2)}$ is in a cycle-sequence $x^{(1)}, \ldots, x^{(3)}$. In fact, the residual network, $\mathcal{N}(x^{(1)})$, has at least one cycle of cost zero; otherwise, considering $x^{(1)}$ as the best flow for this problem, the second best flow (computed for example by using the *second best network flow algorithm* [11]) would have a cost greater than the cost of the flow $x^{(1)}$, but this would mean that problem (3), with $B^{(k)} = X^I$, had only one optimal solution, which is not true. Consider a cycle-adjacent solution, $x^{(1,1)}$, of $x^{(1)}$ obtained through the cycle of cost zero $W^{(1)}$ in $\mathcal{N}(x^{(1)})$. If $x^{(1,1)} = x^{(2)}$ or $x^{(2)}$ is a cycle-intermediate solution of $(x^{(1)}, x^{(1,1)})$ and the prove is done. Otherwise, consider the augmenting flow with $\delta = r_{pq} = \min\{r_{ij} : (i,j) \in W^{(1)}\}$ units that leads to the flow $x^{(1,1)}$. Consider the partition of $X^I$ into two sets, $B^{(1)}$ and $B^{(2)}$, according to the direction of the corresponding cycle in the network $\mathcal{N}$:

(1) if its direction is the same as $(p,q)$, i.e., the arc $(p,q)$ exists in the network $\mathcal{N}$, consider $B^{(1)} = \{x : x \in X^I \text{ and } l_{pq} \leqslant x_{pq} \leqslant a_{pq}\}$, where $a_{pq}$ is the flow of the arc $(p,q)$ in solution $x^{(1)}$;
(2) otherwise, consider $B^{(1)} = \{x : x \in X^I \text{ and } a_{qp} \leqslant x_{qp} \leqslant u_{qp}\}$, where $a_{qp}$ is the flow of the arc $(q,p)$ in solution $x^{(1)}$;

and $B^{(2)} = X^I \setminus B^{(1)}$. It can be seen that $x^{(1)} \in B^{(1)}$, $x^{(1,1)} \in B^{(2)}$ and $x^{(2)}$ is either in $B^{(1)}$ or $B^{(2)}$.

(a) If $x^{(2)} \in B^{(1)}$, consider $x^{(1)}$ as the best solution of the problem $\min_{x \in B^{(1)}} \sum_{k=1}^{p} \lambda_k c^k x$ and the new network $\mathcal{N}$ associated with this problem. Find a cycle-adjacent solution, $x^{(1,2)}$, of $x^{(1)}$ through the cycle of cost zero $W^{(1,2)}$ in $\mathcal{N}(x^{(1)})$ ($x^{(1,2)}$ exists, since $x^{(2)}$ has the same cost as $x^{(1)}$). If this solution is $x^{(2)}$ or $x^{(2)}$ is a cycle-intermediate solution of $(x^{(1)}, x^{(1,2)})$ then the proof is done. Otherwise, consider the augmenting flow of $r_{pq} = \min\{r_{ij} : (i,j) \in W^{(1,2)}\}$ units that leads to the flow $x^{(1,2)}$. Consider the partition of $B^{(1)}$ into sets $B^{(1,1)}$ and $B^{(1,2)}$ in the same way as the partition of $X^I$ was done in $B^{(1)}$ and $B^{(2)}$.

(b) if $x^{(2)}$ is in $B^{(2)}$ consider $x^{(1,1)}$ as the best solution for the problem $\min_{x\in B^{(2)}}\sum_{k=1}^{p}\lambda_k c^k x$ and the new network $\mathscr{N}$ associated with this problem. Compute a cycle-adjacent solution, $x^{(1,1,1)}$, of $x^{(1,1)}$ through the cycle of cost zero $W^{(1,1,1)}$ in $\mathscr{N}(x^{(1,1)})$. If $x^{(1,1,1)} = x^{(2)}$ or $x^{(2)}$ is a cycle-intermediate solution of $(x^{(1,1)}, x^{(1,1,1)})$ then the prove is done. Otherwise, consider the augmenting flow of $r_{pq} = \min\{r_{ij} : (i,j) \in W^{(1,1,1)}\}$ units that leads to the flow $x^{(1,1,1)}$. Consider the partition of $B^{(2)}$ into sets $B^{(2,1)}$ and $B^{(2,2)}$ in the same way as the partition of $X^I$ was done in $B^{(1)}$ and $B^{(2)}$.

Repeating this process we will find a cycle-sequence $x^{(1)},\dots,x^{(3)}$ such that $x^{(2)} = x^{(3)}$ or such that $x^{(2)}$ is in the cycle-sequence $x^{(1)},\dots,x^{(3)}$.

We conclude that any optimal solution $x$ of (3), with $B^{(k)} = X^I$, is in a cycle-sequence $x',\dots,x''$. In fact we know that $x$ is in a cycle-sequence $x',\dots,x'''$ and the cycle-sequence $x',\dots,x''$ can be subdivided into two cycle-sequences $x',\dots,x'''$ and $x''',\dots,x''$. Thus the cycle-sequence $x',\dots,x''$ contains $x$ and the prove is done. □

We can now say that the set of all supported non-dominated vectors for the MOINF problem is connected since the extreme points of the maximally non-dominated facets are connected.

**Proposition 2.3.** *The set of supported efficient solutions for the MOINF problem is connected.*

**Proof.** The set of extreme efficient solutions of MONF is connected as it was proved in 1977 by Isermann [13]. Theorem 2.1 shows that all the supported efficient solutions are in a cycle-sequence with origin in an extreme efficient solution. Thus, we conclude that the set of supported efficient solutions for the MOINF problem is connected. □

The proof of Theorem 2.1 gives rise to Algorithm 1 that finds all supported non-dominated vectors/efficient solutions for the MOINF problem. Consider the parametric problem (3), with $B^{(k)} = X^I$, for some fixed $\lambda$ associated with the maximally efficient facet $F_X \in X$, and a non-dominated vector $y' = f(x')$ such that $x' \in F_X$. Theorem 2.1 says that all the non-dominated solutions in $F_Y$ for the MOINF problem are images of solutions in a cycle-sequence $x',\dots,x''$.

The values of $\lambda$s for each efficient facet are computed through the cost of a cycle in the residual network. We can begin by computing the non-dominated vector associated with the minimum of $p$th objective by solving problem (3), for values of $\lambda_1, \lambda_2, \dots, \lambda_{p-1}$ positive and sufficiently close to 0 and such that $\lambda_p = 1 - \lambda_1 - \lambda_2 - \cdots - \lambda_{p-1}$. This problem can be solved by using the negative-cycle algorithm. Let $x^{(1)}$ be the solution found. This solution has no negative-cycles and $\sum_{(i,j)\in W}\sum_{k=1}^{p}\lambda_k c_{ij}^k \geqslant 0$, for any cycle $W$ in $\mathscr{N}(x^{(1)})$. This system of inequalities allows the determination of a vector $\lambda$ associated with the maximally efficient facet containing $x^{(1)}$. T. Gal [9], shows how this can be done for all maximally efficient facets. If we have only two or three objectives the following procedures could also be used. When the problem has only two objectives, Step 1. can begin by computing all the extreme non-dominated supported vectors, i.e., the extreme points of the set $Conv(ND(Y^I) + \mathbb{R}^2)$. This can be done by using, for example, a dichotomic search or a parametric problem solved through a negative-cycle algorithm or a primal–dual algorithm (see, for example, [4,6]). The maximally non-dominated facets are now line segments connecting two adjacent non-dominated vectors, since the only time $Y$ will have a maximally non-dominated facet of dimension 0 is when the number of extreme non-dominated vectors is one. Suppose that the slope of one of this segments is $d$, then the vector $\lambda$ associated to this maximally non-dominated facet is $\lambda = (\frac{d}{d-1}, 1 - \frac{d}{d-1})$. If the MOINF has three objectives then the problem can have maximally non-dominated facets that are faces or line segments. The faces are part of a plane whose equation is $Ay_1 + By_2 + Cy_3 = D,\ A, B, C > 0$. In this case we have $\lambda = (\frac{A}{A+B+C}, \frac{B}{A+B+C}, \frac{C}{A+B+C})$. If the maximally non-dominated facet is a line segment there is, in general, several vectors $\lambda$. The candidates are found among the perpendicular vectors to the line segment, such that $\lambda_k > 0,\ k = 1, 2, \dots, p$ and $\sum_{k=1}^{p}\lambda_k = 1$.

**Algorithm 1.** *MOINF cost zero algorithm.*

Input: Network $\mathscr{N} = (\mathscr{G}, (c^1, c^2, \dots, c^p), l, u, b)$.
Output: $Y_s$, the set of all supported non-dominated vectors and $X_s$, the set of all supported efficient solutions.

*Step 1*: Compute the set of all $\lambda$s $\lambda^{(1)}, \lambda^{(2)}, \dots, \lambda^{(t)}$ associated with the maximally efficient facets $F_1, F_2, \dots, F_t$, and one extreme point for each facet, $x^{(1)}, x^{(2)}, \dots, x^{(t)}$, respectively. Let $E := \{(x^{(1)}, \lambda^{(1)}, X^I), (x^{(2)}, \lambda^{(2)}, X^I), \dots, (x^{(t)}, \lambda^{(t)}, X^I)\}$, where $x^{(k)}$ is such that $f(x^{(k)}) = y^{(k)}$, $k = 1, 2, \dots, t$.Let $Y_s(X_s)$ be the set of supported non-dominated vectors (efficient solutions) of the MOINF problem. $Y_s := \{y^{(1')}, y^{(2')}, \dots, y^{(t')}\}$ and $X_s = \{x^{(1'')}, x^{(2'')}, \dots, x^{(t'')}\}$. These solutions are the same as in $E$ without repetitions.

*Step 2*: While $E \neq \emptyset$ do

$E := E \setminus \{(x^{(k)}, \lambda^{(k)}, B)\}$.

Consider the problem $\min_{x\in B}\sum_{k=1}^{p}\lambda_k c^k x$ for $\lambda = \lambda^{(k)}$ and the corresponding network $\mathscr{N}$.

If there exists a cycle-adjacent solution, $x^{(k\prime)}$, of $x^{(k)}$ associated with a cycle of cost zero $W$ in $\mathscr{N}(x^{(k)})$, and with an arc $(p,q)$ such that $r_{pq} = \min\{r_{ij} : (i,j) \in W\}$:

(a) Add $(x^{(k)}, \lambda^{(k)}, B^{(1)})$ and $(x^{(k')}, \lambda^{(k)}, B^{(2)})$ to the set $E$, where
$B^{(2)} = B \setminus B^{(1)}$ and
$B^{(1)} = \{x \in B : 0 \leqslant x_{pq} \leqslant a_{pq}\}$ if the arc $(p,q) \in \mathscr{N}$ or $B^{(1)} = \{x \in B : a_{qp} \leqslant x_{qp} \leqslant u_{qp}\}$ otherwise, where $a_{pq}$ ($a_{qp}$) is the flow of the arc $(p,q)$ ($(q,p)$) in solution $x^{(k)}$.

(b) If $y^{(k')}$ is not in $Y_s$, set $Y_s := Y_s \cup \{y^{(k')}\}$. If $x^{(k')}$ is not in $X_s$, set $X_s := X_s \cup \{x^{(k')}\}$

(c) For each cycle-intermediate solution $x^{(k'')}$ of $(x^{(k)}, x^{(k')})$ such that $y^{(k'')}$ is not in $Y_s$ do $Y_s := Y_s \cup \{y^{(k'')}\}$. If $x^{(k'')}$ is not in $X_s$, set $X_s := X_s \cup \{x^{(k'')}\}$.

### 3.1. The correctness and complexity of the algorithm

**Proposition 2.4.** *At termination of Algorithm 1, the solutions of the set $X_s$ are all supported efficient solutions and any feasible solution not in $X_s$ is not a supported efficient solution.*

**Proof.** All the solutions in $X_s$ are supported efficient. This is true since all the solutions are in the maximally efficient facets of $X$, from Theorem 2.1. Suppose that there is a supported efficient solution $x'$ that does not belong to $X_s$. As $x'$ is a supported efficient solution $x'$ will be in a maximally efficient facet $F_k$ of $X$. Consider the facet $F_k$ and the solution $x^{(k)}$ found in Step 1. The solutions $x'$ and $x^{(k)}$ are both optimal solutions for a problem with a fixed $\lambda$ and $B^{(k)} = X^I$, then $x'$ is in a cycle-sequence $x^{(k)} \cdots x''$. But, all the solutions in such a cycle-sequences are found in *Step 2* of Algorithm 1. Thus the algorithm finds all the supported efficient solutions. □

Clearly, all the supported non-dominated vectors are found and there is not any dominated or non-supported non-dominated vector in $Y_s$, since they are image of the supported efficient solutions.

Ruhe [16] proves for a particular instance with only two criteria, that the number of non-dominated extreme points (in criteria space) grows exponentially with the number of vertices of the network.

## 4. An illustrative example

Consider the MOINF problem with three objectives as in Fig. 4 below. In the first step the maximally efficient facets for this problem are found. These facets are the two faces [*ABCDEF*] and [*CDIH*] and the line segment [*IJ*]. They are depicted in Fig. 5a and b in the decision and objective spaces, respectively (see the gray polygons). The point *C* associated with efficient solution $x^{(1)} = (7,3,0,7,0,3,7)$ belongs to the two faces and the point *I* associated with the solution $x^{(3)} = (10,0,3,7,1,2,8)$ belongs to the line segment [*IJ*]. The $\lambda$s associated with those facets are $\lambda^{(1)} = (\frac{44}{63}, \frac{1}{9}, \frac{4}{21})$, $\lambda^{(2)} = (\frac{76}{99}, \frac{1}{9}, \frac{4}{33})$ and $\lambda^{(3)} = (\frac{11}{13}, \frac{1}{13}, \frac{1}{13})$, respectively. Thus $E = \{(x^{(1)}, \lambda^{(1)}, X^I), (x^{(1)}, \lambda^{(2)}, X^I), (x^3, \lambda^{(3)}, X^I)\}$, $Y_s = \{y^{(1)}, y^{(3)}\} = \{(445, 536, 74), (442, 560, 71)\}$ and $X_s = \{x^{(1)}, x^{(3)}\} = \{(7,3,0,7,0,3,7), (10,0,3,7,1,2,8)\}$.

*It*.1: In the second step of the algorithm the element $(x^{(1)}, \lambda^{(1)}, X^I) = ((7,3,0,7,0,3,7), (\frac{44}{63}, \frac{1}{9}, \frac{4}{21}), X^I)$ in $E$ is considered and we set $E := \{(x^{(1)}, \lambda^{(2)}, X^I), (x^{(3)}, \lambda^{(3)}, X^I)\}$.

Consider the auxiliary problem associated with the first face.

$$\begin{aligned}
\min \quad & \frac{44}{63} f_1(x) + \frac{1}{9} f_2(x) + \frac{4}{21} f_3(x) = \frac{115}{9} x_{12} + \frac{46}{3} x_{13} + \frac{23}{9} x_{23} + \frac{512}{63} x_{24} + \frac{19}{3} x_{34} + \frac{208}{9} x_{35} + \frac{1105}{63} x_{45} \\
\text{subject to:} \quad & x_{12} + x_{13} = 10, \\
& -x_{12} + x_{23} + x_{24} = 0, \\
& -x_{13} - x_{23} + x_{34} + x_{35} = 0, \\
& -x_{24} - x_{34} + x_{45} = 0, \\
& -x_{35} - x_{45} = -10, \\
& x_{12} \leqslant 10,\ x_{13} \leqslant 5,\ x_{23} \leqslant 4,\ x_{24} \leqslant 7,\ x_{34} \leqslant 8,\ x_{35} \leqslant 6,\ x_{45} \leqslant 8,\ x_{12},\ x_{13},\ x_{23},\ x_{24},\ x_{34},\ x_{35},\ x_{45} \geqslant 0 \text{ and integer.}
\end{aligned}$$

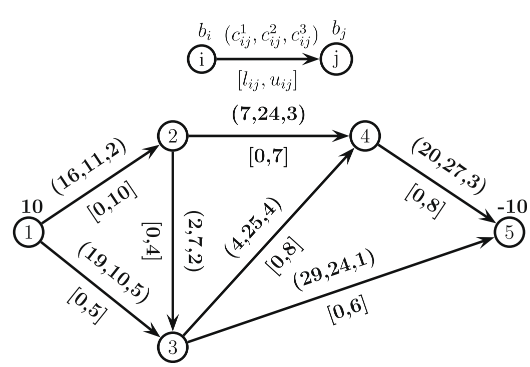


**Fig. 4.** MOINF problem.

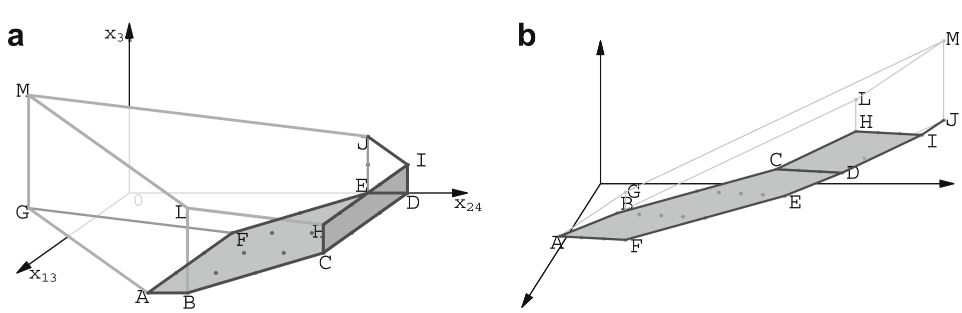


**Fig. 5.** Efficient solutions and non-dominated vectors for MOINF in Fig. 4.

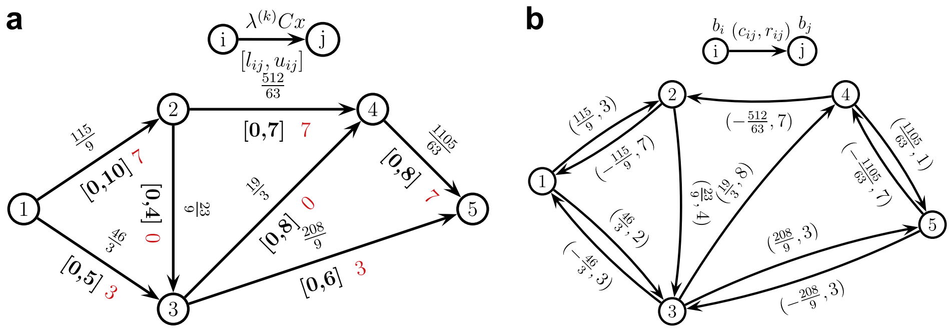


**Fig. 6.** (a) Solution $x^{(1)}$ and (b) $\mathcal{N}(x^{(1)})$.

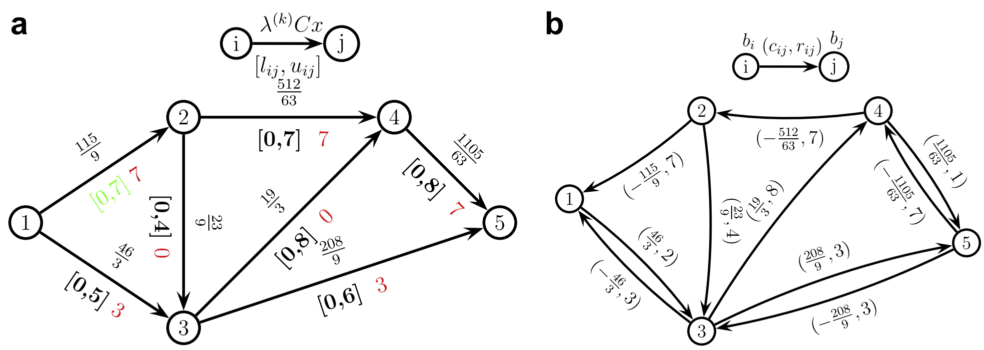


**Fig. 7.** (a) Solution $x^{(1)}$ and (b) $\mathcal{N}(x^{(1)})$.

The residual network associated with the solution $x^{(1)}$ (Fig. 6a) is in Fig. 6b. Consider the cycle-adjacent solution $x^{(4)} = (10,0,3,7,0,3,7)$ obtained through the cycle of cost zero $1-2-3-1$. $(x^{(1)},x^{(4)})$ has $x^{(5)} = (8,2,1,7,0,3,7)$ and $x^{(6)} = (9,1,2,7,0,3,7)$ as cycle-intermediate solutions. The corresponding non-dominated vectors are $y^{(4)} = (442,560,71)$, $y^{(5)} = (444,544,73)$, and $y^{(6)} = (443,552,72)$.

The new set $E$ is

$$E = \Big\{(x^{(1)},\lambda^{(1)},B^{(1)} = \{x \in X^I : 0 \leqslant x_{12} \leqslant 7\}),(x^{(4)},\lambda^{(1)},B^{(2)} = \{x \in X^I : 8 \leqslant x_{12} \leqslant 10\}),(x^{(1)},\lambda^{(2)},X^I),(x^{(3)},\lambda^{(3)},X^I)\Big\}.$$

We have $Y_s = \{y^{(1)},y^{(3)},y^{(4)},y^{(5)},y^{(6)}\}$ and $X_s = \{x^{(1)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}\}$.

*It.* 2: As $E \neq \{\}$ the algorithm continues considering $(x^{(1)},\lambda^{(1)},B^{(1)}) \in E$. $E := E \setminus \{(x^{(1)},\lambda^{(1)},B^{(1)})\}$.

The residual network associated with the solution $x^{(1)}$ (Fig. 7a), in the new network associated with $B^{(1)}$, is presented in Fig. 7b. Consider the cycle-adjacent solution $x^{(7)} = (5,5,0,5,0,5,5)$ obtained through the cycle of cost zero $1-3-5-4-2-1$. $(x^{(1)},x^{(7)})$ has $x^{(8)} = (6,4,0,6,0,4,6)$ as cycle-intermediate solution. The corresponding non-dominated vectors are $y^{(7)} = (455,480,70)$, and $y^{(8)} = (450,508,72)$.

The new set $E$ is

$$E = \Big\{(x^{(1)},\lambda^{(1)},B^{(1,1)} = \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 0 \leqslant x_{13} \leqslant 3\})(x^{(7)},\lambda^{(1)},B^{(1,2)}$$
$$= \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 4 \leqslant x_{13} \leqslant 5\}),\ \Big(x^{(4)},\lambda^{(1)},B^{(2)} = \{x \in X^I : 8 \leqslant x_{12} \leqslant 10\}),(x^{(1)},\lambda^{(2)},X^I),(x^{(3)},\lambda^{(3)},X^I)\Big\}.$$

The new sets $Y_s$ and $X_s$ are $Y_s = \{y^{(1)},y^{(3)},y^{(4)},y^{(5)},y^{(6)},y^{(7)},y^{(8)}\}$ and $X_s = \{x^{(1)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}, x^{(7)}, x^{(8)}\}$.

*It.* 3: As $E \neq \{\}$ the algorithm continues considering $(x^{(1)},\lambda^{(1)},B^{(1,1)}) \in E$. $E := E \setminus \{(x^{(1)},\lambda^{(1)},B^{(1,1)})\}$.

Consider the cycle-adjacent solution $x^{(9)} = (7,3,3,4,0,6,4)$ obtained through the cycle of cost zero $2-3-5-4-2$ in the network associated with $B^{(1,1)}$. $(x^{(1)},x^{(9)})$ has $x^{(10)} = (7,3,1,6,0,4,6)$, and $x^{(11)} = (7,3,2,5,0,5,5)$ as cycle-intermediate solutions. The corresponding non-dominated vectors are $y^{(9)} = (457,476,65)$, $y^{(10)} = (449,516,71)$, and $y^{(11)} = (453,496,68)$.

The new set $E$ is

$$E = \Big\{(x^{(1)},\lambda^{(1)},B^{(1,1,1)} = \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 0 \leqslant x_{13} \leqslant 3 \text{ and } 0 \leqslant x_{35} \leqslant 3\}),(x^{(9)},\lambda^{(1)},B^{(1,1,2)}$$
$$= \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 0 \leqslant x_{13} \leqslant 3 \text{ and } 4 \leqslant x_{35} \leqslant 6\}),(x^{(7)},\lambda^{(1)},B^{(1,2)} = \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 4 \leqslant x_{13} \leqslant 5\}),(x^{(4)},\lambda^{(1)},B^{(2)}$$
$$= \{x \in X^I : 8 \leqslant x_{12} \leqslant 10\}),(x^{(1)},\lambda^{(2)},X^I),(x^{(3)},\lambda^{(3)},X^I)\Big\}.$$

The new sets $Y_s$ and $X_s$ are $Y_s = \{y^{(1)},y^{(3)},y^{(4)},y^{(5)},y^{(6)},y^{(7)},y^{(8)},y^{(9)},y^{(10)},y^{(11)}\}$ and $X_s = \{x^{(1)}, x^{(3)}, x^{(4)}, x^{(5)}, x^{(6)}, x^{(7)}, x^{(8)}, x^{(9)}, x^{(10)}, x^{(11)}\}$

*It.* 4: As $E \neq \{\}$ the algorithm continues considering $(x^{(1)},\lambda^{(1)},B^{(1,1,1)}) \in E$ and let $E = E \setminus \{(x^{(1)},\lambda^{(1)},B^{(1,1,1)})\}$.

There is no cycles of cost zero in the residual network associated with $B^{(1,1,1)}$.

*It.* 5: As $E \neq \{\}$ the algorithm continues considering $(x^{(9)},\lambda^{(1)},B^{(1,1,2)}) \in E$. $E := E \setminus \{(x^{(9)},\lambda^{(1)},B^{(1,1,2)}\}$.

Consider the cycle-adjacent solution $x^{(10)} = (7,3,1,6,0,4,6)$ obtained through the cycle of cost zero $2-4-5-3-2$ in the network associated with $B^{(1,1,2)}$. $(x^{(9)}, x^{(10)})$ has $x^{(11)}$ as cycle-intermediate solution.

The new set $E$ is

$$E = \Big\{ (x^{(9)}, \lambda^{(1)}, B^{(1,1,2,1)} = \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 0 \leqslant x_{13} \leqslant 3 \text{ and } x_{35} = 6\}), (x^{(10)}, \lambda^{(1)}, B^{(1,1,2,2)} = \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 0 \leqslant x_{13} \leqslant 3 \text{ and } 4 \leqslant x_{35} \leqslant 5\}), (x^{(7)}, \lambda^{(1)}, B^{(1,2)} = \{x \in X^I : 0 \leqslant x_{12} \leqslant 7 \text{ and } 4 \leqslant x_{13} \leqslant 5\}), (x^{(4)}, \lambda^{(1)}, B^{(2)} = \{x \in X^I : 8 \leqslant x_{12} \leqslant 10\}), (x^{(1)}, \lambda^{(2)}, X^I), (x^{(3)}, \lambda^{(3)}, X^I) \Big\}.$$

The sets $Y_s$ and $X_s$ are kept the same.

*It.* 6: $(\cdots)$

The remaining non-dominated vectors/efficient solutions are computed in a similar way.

## 5. Conclusions

In this paper, we proposed an algorithm for finding all supported non-dominated vectors/efficient integer solutions for MOINF problems. To the best of our knowledge, there is no other method specifically devoted to determining these vectors/solutions. The algorithm identifies the zero cost cycles in a residual network and uses the connectedness property of the supported efficient solutions. It provides an alternative to other algorithms using simplex-based proofs. This algorithm can deal with more than two objectives and is able to avoid the degeneracy problems that have been highlighted in the simplex-based methods. In our opinion, the proposed algorithm is the first step in the development of more negative-cycle algorithms for solving MOINF problems.

## References


[1] R. Ahuja, T.L. Magnanti, J. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, New Jersey, 1993.
[2] M.S. Bazaraa, J.J. Jarvis, H.D. Sherali, Linear Programming and Network Flows, third ed., John Wiley & Sons, New Jersey, 2005.
[3] J.P. Dauer, R.J. Gallagher, A combined constraint-space objective space approach for determining high-dimensional maximal efficient faces of multiple objective linear programs, European Journal of Operational Research 88 (1996) 368–381.
[4] M. Ehrgott, Multicriteria Optimization, second ed., Springer-Verlag, Berlin, 2005.
[5] A. Eusébio, J.R. Figueira, Finding non-dominated solutions in bi-objective integer network flow problems 2008, doi:10.1016/j.cor.2008.11.001.
[6] A. Eusébio, J.R. Figueira, M. Ehrgott, A primal–dual simplex algorithm for bi-objective network flow problems, 2008, doi:10.1007/s10288-008-0087-3.
[7] J.R. Figueira, On the integer bi-criteria network flows problem: A branch-and-bound approach, Faculdade de Economia da Universidade de Coimbra, Coimbra, Portugal, Working Paper No. 3/2000 [revised version: Cahier du LAMSADE, No. 191/2002, LAMSADE, Université Paris-Dauphine, Paris, France].
[8] J.R. Figueira, Filtering algorithm for bi-criteria network flows, 2001. Manuscript.
[9] T. Gal, A general method for determining the set of all efficient solutions to a linear vectormaximum problem, European Journal of Operational Research 1 (1977) 307–322.
[10] M.C. Gouveia, Bi-criteria network flow problems: The study of an algorithm for searching efficient solutions [in Portuguese], Master's thesis, Faculdade de Economia da Universidade de Coimbra, Coimbra, Portugal, 2002.
[11] H.W. Hamacher, A note on $K$ best network flows, Annals of Operations Research 57 (1995) 65–72.
[12] H.W. Hamacher, C.R. Pedersen, S. Ruzika, Multiple objective minimum cost flow problems: A review, European Journal of Operational Research 176 (2007) 1404–1422.
[13] H. Isermann, The enumeration of the set of all efficient solutions for a linear multiple objective program, Operational Research Quarterly 28 (1977) 711–725.
[14] H. Lee, P.S. Pulat, Bicriteria network flow problems: Integer case, European Journal of Operational Research 66 (1993) 148–157.
[15] A. Przybylski, X. Gandibleux, M. Ehrgott, The biobjective integer minimum cost flow problem – Incorrectness of Sedeño-Noda and González-Martín's algorithm, Computers & Operations Research 33 (5) (2006) 1459–1463.
[16] G. Ruhe, Complexity results for multicriterial and parametric network flows using a pathological graph of Zadeh, Zeitschrift für Operations Research 32 (1988) 9–27.
[17] A. Sedeño-Noda, C. González Martín's, An algorithm for the biobjective integer minimum cost flow problem, Computers & Operations Research 28 (2001) 139–156.
[18] R.E. Steuer, Multiple Criteria Optimization: Theory, Computation, and Application, John Wiley & Sons, New York, 1986.