Bi-criteria network flow problems: A characterization of non-dominated solutions

Augusto Eusébio[*†] José Rui Figueira[†‡]

July 1, 2006

[*]Escola Superior de Tecnologia e Gestão, Instituto Politécnico de Leiria, Morro do Lena - Alto Vieiro, 2411-901, Leiria, Portugal, Phone: (+351) 244 820 300, Fax:(+351) 244 820 310, E-mail: aeusebio@estg.ipleiria.pt

[†]CEG-IST, Center for Management Studies, Instituto Superior Técnico, TagusPark, Av. Cavaco Silva, 2780-990, Porto Salvo, Portugal, Phone: (+351) 21 423 35 07, Fax: (+351) 21 423 35 68, E-mail: figueira@ist.utl.pt

[‡]Associate Researcher at LAMSADE, Université Paris-Dauphine, Place du Maréchal De Lattre de Tassigny, 75 775 Paris Cedex 16, France.

Contents

Bi-criteria network flow problems: A characterization of non-dominated solutions

Abstract

This paper deals with network flow problems with two criteria. The main purpose of the paper is to present a characterization of non-dominated solutions when using a simplex like algorithm for identifying supported non-dominated solutions for the integer bi-criteria network flow problem. This study can help us for a better understanding of network flow problems with multiple criteria. The paper also presents an ε-constraint based technique for identifying all efficient non-dominated solutions for the integer bi-criteria network flow problem.

Keywords: *Multicriteria linear and integer programming, Bi-criteria network flows, Network simplex algorithm, Efficient/non-dominated solutions*

1 Introduction

Planning the design and construction of highways, telecommunications networks in a certain country, computers networks, distribution energy networks, gas networks, and water networks, are just some examples in which network flow problems have been applied mainly over the past three decades. In general, it is frequent do not "optimize" one, but several criteria simultaneously, being thus necessary the use of multiple criteria models. In general, the criteria are conflicting, not being possible to obtain a solution which optimizes all the criteria simultaneously. Consequently, it is necessary to "re-define" the notion of "better" solution for such problems. The concept of efficient or non-dominated solutions is used instead of the notion of optimal solutions. An efficient solution is a solution where it is not possible to improve the evaluation of one of the criteria without the degradation of at least one of the remaining criteria. We refer to non-dominated solutions as the image of an efficient solution in the criterion space.

The search for the entire set of non-dominated solutions in a multiple criteria network flow problem is a difficult problem. Ruhe [21] showed that, for particular instances with just two criteria, the number of non-dominated solutions grows exponentially with the number of vertices in the network.

Linear network flow problems with multiple criteria can be solved using the general methods for the resolution of linear programming models with several criteria. Among the algorithms proposed for the determination of the set of efficient solutions the following can be mentioned, Evans and Steuer [8], Zeleny [26], Isermann [13], Ecker and Kouada [6], Yu [25], and Armand and Malivert [2].The first one, called ADBASE, is the only one that was coded and stills available for use.

The algorithms particularly designed for multiple criteria network flow problems can only deal with two criteria, for example, Malhotra and Puri [16], Lee and Pulat [14], Pulat, Harng and Lee [20], Sedeño Noda and González-Martín [22].

The resolution of a multiple criteria integer linear programming problem brings some additional difficulties. While in a linear programming problem the boundary of the feasible region contains the set of efficient solutions in multiple criteria integer linear programming problems efficient solutions can also belong to the interior of the convex hull of the feasible region in the criterion space.

As for the integer bi-criteria network flow problem we also found some algorithms. Among them the following deserve some attention, Lee and Pulat [15] and Sedeño-Noda and González-Martín [23] which does not solve some special cases (see [19]) and [9].

The algorithms presented in [15] and [23] are simplex like algorithms that explore the connectedness of the set of the efficient solutions. Since connectedness is not verified for such problems [19] the above algorithms have several drawbacks [12], mainly due to an unfamiliar knowledge about bi-criteria network flow problem non-dominated points. This paper is mainly devoted to improve the knowledge on such kind of problems.

This article is thus about bi-criteria network flows. A revision of the main notions on this very topic is provided. A set of examples that shows the existence of particular solutions when a simplex based method is used are presented. These examples show that some methods in the literature cannot be effectively used since during the search process they miss certain particular non-dominated solutions.

The rest of this paper is organized as follows. Section 2 introduces the fundamental notation, concepts, definitions from graph theory, network flows, convex and polyhedral sets, and multi-criteria optimization. Section 3 presents a simplex algorithm and its particular version for minimum cost network flows. Section 4 is devoted to two algorithms for computing the set efficient solutions for bi-criteria network flows; these algorithms are adaptations of existent algorithms. Section 5 is consecrated to a set of difficult examples for bi-criteria network based algorithms. In particular it is shown that it is possible to have non-dominated supported solutions as images of non-intermediate solutions. Section 6 describe an ε-constraint based technique for identifying all efficient non-dominated solutions for the integer bi-criteria network flow problem. And Section 7 is the conclusion of the paper.

2 Concepts, definitions, and notation

For understanding better the proposed approach, concepts, definitions, and notation from graph theory, network flows, convex and polyhedral sets, and multi-criteria optimization are introduced in this section (for more details consult also [1] and [24]).

2.1 Graph theory

Let $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ denote a *directed* and *connected graph*, where $\mathcal{S}$ is a finite set of *nodes* or *vertices* with cardinality $|\mathcal{S}| = m$, and $\mathcal{A}$ is a collection of ordered pairs of elements of $\mathcal{S}$ called *arcs*, with cardinality $|\mathcal{A}| = n$.

A graph $\mathcal{G}' = (\mathcal{S}', \mathcal{A}')$ is called a *subgraph* of $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ if $\mathcal{S}' \subseteq \mathcal{S}$ and $\mathcal{A}' \subseteq \mathcal{A}$. It is a *spanning subgraph* of $\mathcal{G}$ if $\mathcal{S}' = \mathcal{S}$. A *path* $\mathcal{P}$ is a sequence of vertices and arcs, $i_1 - a_1 - i_2 - a_2 - \ldots - i_{s-1} - a_{s-1} - i_s$, without repetition of vertices and where $1 \leq k \leq s-1$ for which either $a_k = (i_k, i_{k+1}) \in \mathcal{A}$, or $a_k = (i_{k+1}, i_k) \in \mathcal{A}$. A *directed path* is a path without backwards arcs. A *cycle* $\mathcal{C}$ is a closed path where the only repeated vertex is the starting and the ending point that coincide. A *directed cycle* is a closed directed path. When in a given graph $\mathcal{G}$ there is always a path linking any two different vertices of $\mathcal{G}$, the graph is called *connected*. A *tree* $\mathcal{T} = (\mathcal{V}, \mathcal{E})$ is a subgraph without cycles where $\mathcal{V} \subseteq \mathcal{S}$ and $\mathcal{E} \subseteq \mathcal{A}$. A tree $\mathcal{T}$ is called a *spanning tree* when it spans the set of vertices $\mathcal{S}$ of $\mathcal{G}$, that is $\mathcal{V} = \mathcal{S}$. A spanning tree is denoted by $\mathcal{T} = (\mathcal{S}, \mathcal{E})$. Consider (k, l) a given arc belonging to the set $\mathcal{A}$ but not in $\mathcal{E}$. Then, there is a unique cycle $\mathcal{C}$ when the arc (k, l) is added to $\mathcal{E}$. The direction of $\mathcal{C}$ is the same as (k, l). In a cycle $\mathcal{C}$ a partition of its arcs can be made by separating the arcs having the same direction as $\mathcal{C}$ from the arcs in the opposite direction. The collection of all possible cycles of this type is called *fundamental cycle basis*.

2.2 Network flows

A directed graph with numerical values assigned to its vertices and/or arcs is called *network*. Let $\mathcal{N} = (\mathcal{G} = (\mathcal{S}, \mathcal{A}), c, l, u, b)$ be a network with a *"cost"* c_{ij}, a *lower bound* l_{ij} and an *upper bound* or *capacity* u_{ij} associated with every arc $(i, j) \in \mathcal{A}$. The numerical values l_{ij} and u_{ij}, respectively, denote the minimum and the maximum amount that must flow on the arc (i, j). Finally, let x_{ij} denote the *amount of flow* on the arc (i, j). A numerical value b_i is also associated with each vertex $i \in \mathcal{S}$ denoting its *supply* (if $b_i > 0$) or its *demand* (if $b_i < 0$). A vertex with $b_i = 0$ is called a *transshipment vertex*.

2.3 Minimum cost flow problem

The minimum cost network flow problem can be stated as a linear programming problem,

$$\begin{array}{lll} \text{minimize} & f(x) = \sum_{(i,j)\in\mathcal{A}} c_{ij}x_{ij} & \\ \text{subject to:} & \sum_{\{j:(i,j)\in\mathcal{A}\}} x_{ij} - \sum_{\{k:(k,i)\in\mathcal{A}\}} x_{ki} = b_i \qquad \forall i \in \mathcal{S} & \\ & 0 \leq x_{ij} \leq u_{ij} \qquad \forall (i,j) \in \mathcal{A} & \end{array} \tag{1}$$

which is equivalent to,

$$\begin{array}{ll} \text{minimize} & f(x) = c^T x \\ \text{subject to:} & Ax = b \\ & 0 \leq x \leq u \end{array} \tag{2}$$

where, $x \in \mathbb{R}^n$ is the vector of decision variables, $c \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$ are the cost and capacity vectors, respectively, $b \in \mathbb{R}^m$ is the right-hand-side vector and A the node-arc incidence matrix. Each column (i, j) of A contains exactly two nonzero coefficients: $+1$ in row i, and -1 in row j. This type of matrices has some "good" properties that allows the application of for example, the network simplex algorithm, that solves the minimum cost flow problem faster and with less resource consumption than standard linear programming algorithms (see [1] and [3]).

Associated to the problem (1) we can define its dual version,

$$\begin{array}{ll} \text{maximize} & \sum_{i\in\mathcal{S}} b_i\pi_i - \sum_{(i,j)\in\mathcal{A}} u_{ij}\mu_{ij} \\ \text{subject to:} & \pi_i - \pi_j - \mu_{ij} \leq c_{ij} \qquad \forall (i,j) \in \mathcal{A} \\ & \mu_{ij} \geq 0, \forall (i,j) \in \mathcal{A} \\ & \pi_i \text{ free}, \forall i \in \mathcal{S} \end{array} \tag{3}$$

where $\pi = (\pi_1, \cdots, \pi_i, \cdots, \pi_m)$, $i \in \mathcal{S}$ is the vector of dual variables associated with constraints (1), and $\mu = (\mu_{i_1 j_1}, \cdots, \mu_{ij}, \cdots, \mu_{i_n j_n})$, $(i, j) \in \mathcal{A}$ is the vector of dual variables associated with the constraints $x_{ij} \leq u_{ij}$. Using formulation (2) the corresponding dual is

$$\begin{array}{ll} \text{maximize} & b^T\pi - u^T\mu \\ \text{subject to:} & A^T\pi - \mu \leq c \\ & \mu \geq 0 \\ & \pi \text{ free} \end{array} \tag{4}$$

2.4 Multiple criteria network flow problem (MMCF)

The multiple criteria version can also be considered when p cost functions are taken into account

$$
\begin{array}{ll}
\text{“minimize”} & F(x) = C^T x \\
& \qquad = ((c^1)^T x, (c^2)^T x, \ldots, (c^p)^T x) \\
\text{subject to:} & Ax = b \\
& 0 \leq x \leq u
\end{array}
\tag{5}
$$

where the constraints are the same as in (2). Generally, there is a conflicting situation and not all criteria $f_q(x) = (c^q)^T x$ reach their minimum at the same extreme point, therefore the "best" solution does not be intended as the value in X that minimizes simultaneously all functions $f_q(x), q = 1, 2, \ldots, p$.

In what follows the notation below should be considered,

$$
\begin{aligned}
X &= \{x \in \mathbb{R}^n : Ax = b, 0 \leq x \leq u\} \\
Y &= F(X) \\
&= (f_1(x), f_2(x), \ldots, f_q(x), \ldots, f_p(x)) \\
&= (y_1, y_2, \ldots, y_q, \ldots, y_p)
\end{aligned}
$$

X and Y are denominated the sets of *feasible solutions* in the *decision space*, $\mathbb{R}^n$, and in the *criterion space*, $\mathbb{R}^p$, respectively.

By adding integrality requirements to the variables in X we obtain a multiple criteria integer minimum cost flow problem.

In what follows, the assumptions below must be taken into account:

1. The graph is directed and connected.
2. All the numerical values for the costs, lower and upper bounds on the arcs and supplies/demands on the vertices are integral and finite.
3. The condition $\sum_{i \in \mathcal{S}} b_i = 0$ must be fulfilled.
4. The multiple criteria (linear or integer) network flow problem has at least two feasible solutions and the minimum values for the individual objective functions are different.

2.5 Convex and polyhedral sets

According to the following definition a set is convex whenever all points on the line connecting any two points in the set are also in the set.

Definition 2.1 (Convex set) *A set $X \subset \mathbb{R}^n$ is convex iff for any $x', x'' \in X$ the point $(\lambda x' + (1 - \lambda)x'') \in X$, for all $\lambda \in [0, 1]$. Otherwise, the set is nonconvex.*

A point in a set is said to be extreme if it cannot be expressed as a convex combination of any two different points in this set.

Definition 2.2 (Extreme point) *Let $X \subset \mathbb{R}^n$ denote a convex set. A point $x \in X$ is an extreme point of X iff points $x', x'' \in X$, $x' \neq x''$, do not exist such that $x = \lambda x' + (1-\lambda)x''$ for some $\lambda \in]0,1[$*

Definition 2.3 (Convex hull) *Let $S \subset \mathbb{R}^n$, the convex hull of S is the intersection of all convex sets containing S. $Conv(S)$ denotes the convex hull of S.*

Definition 2.4 (Polyhedral set) *A polyhedral set or polyhedron is the convex hull of a finite set of points and a finite number of unbounded line segments. A polyhedral set is a polytope if it is the convex hull of a finite set of points.*

Let $X \subset \mathbb{R}^n$ and $F : X \to \mathbb{R}^p$. Then, if X is convex and F is linear the image of X under F, $Y = F(X)$, is convex too. If X is polyhedral, Y is given by the set of all convex combinations of the images of the extreme points and the points along the unbounded edges of X (see [24]).

2.6 Dominance

The idea of dominance is used in the criterion space to compare two criterion vectors. Two forms of *dominance* should be considered.

Definition 2.5 (Dominance) *Let $y', y'' \in \mathbb{R}^p$ denote two criterion vectors. Then, y' dominates y'' iff $y' \leq y''$ and $y' \neq y''$ (i.e, $y'_q \leq y''_q$ for all q and $y'_q < y''_q$ for at least one q, $q = 1, 2, \ldots, p$).*

When y' dominates y'' this means that no component of y' is worst than the corresponding component of y'' and at least one component of y' is better than its corresponding component of y''.

Definition 2.6 (Strong dominance) *Let $y', y'' \in \mathbb{R}^p$ denote two criterion vectors. Then, y' strongly dominates y'' iff $y' < y''$ (i.e, $y'_q < y''_q$ for all q, $q = 1, 2, \ldots, p$).*

If y' strongly dominates y'', each component of y' is better than its corresponding component of y''. Note that if a criterion vector strongly dominates another, it dominates it as well.

2.7 Non-dominated criterion vectors

A criterion vector is *non-dominated* if it is not dominated by any other criterion vector.

Definition 2.7 (Non-dominated criterion vector) *Let $y \in Y$. Then y is non-dominated iff there does not exist another $y' \in Y$ such that $y' \leq y$ and $y' \neq y$. Otherwise, y is a dominated criterion vector.*

We represent the set of all non-dominated criterion vectors by $ND(Y)$.

2.8 Efficiency

Definition 2.8 (Efficient point) *A point $x \in X$ is efficient iff there does not exist another point $x' \in X$ such that $y' \leq y$ and $y' \neq y$, $y' = F(x')$ and $y = F(x)$. Otherwise, x is inefficient.*

The set of efficient solutions will be denoted by $EFF(X)$.

An important result to develop algorithm for finding the set of all efficient points for (6) is stated in the following theorem.

Theorem 2.1 (Efficiency) *The solution $x \in X$ is efficient iff there exists a*

$$\lambda \in \Lambda = \Big\{ \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_q, \ldots, \lambda_p) \in \mathbb{R}^p | \lambda_q > 0, q = 1, 2, \ldots, p, \sum_{q=1}^{p} \lambda_q = 1 \Big\}$$

such that x minimizes the weighted-sum linear programming,

$$\min\{\lambda C^T x : x \in X\}$$

2.9 Supported and unsupported non-dominated criterion vectors

When considering the integer MMCF two more concepts have to be introduced: supported and unsupported non-dominated criterion vectors.

Let

$$Y^{\geqq} = Conv(ND(Y) + \mathbb{R}^p_{\geqq})$$

where, $\mathbb{R}^p_{\geqq} = \{y \in \mathbb{R}^p | y \geqq 0\}$ and $ND(Y) + \{y \in \mathbb{R}^p_{\geqq}\} = \{y \in \mathbb{R}^p : y = y' + y'', y' \in ND(Y)$ and $y'' \in \mathbb{R}^p_{\geqq}\}$ and $y \geqq 0$ if $y_q \geq 0$, $q = 1, 2 \cdots, p$.

Definition 2.9 (Supported non-dominated criterion vector) *Let y denote a non-dominated criterion vector. Then, if y is on the boundary of $Y^{\geqq}$, y is a supported non-dominated criterion vector. Otherwise, y is an unsupported non-dominated criterion vector.*

Definition 2.10 (Supported-extreme non-dominated criterion vectors) *Let y be a supported non-dominated criterion vector. Then, y is a supported-extreme vector if it is an extreme point of $Y^{\geqq}$. Otherwise, y is a supported non-extreme vector.*

Inverse images of *supported* non-dominated criterion vectors are said to be supported efficient points and inverse images of unsupported non-dominated criterion vectors are said to be *unsupported efficient points.*

Example 2.1.
In Figure 1, dots represent the feasible vectors $y^1, y^2, \ldots, y^{|Y|}$ in the criterion space, $\mathbb{R}^2$. Solutions y^1, y^3, y^4, and y^5 are all supported extreme non-dominated vectors and the shaded area represents $Conv(Y)$. Solution y^7 is a supported non-extreme non-dominated vector, while solutions y^8 and y^9 are both unsupported non-dominated vectors.

■

3 Network simplex algorithm

This section succinctly recall the network simplex method for minimum cost network flow problems. It is important to see how it works to understand the bi-criteria simplex algorithm.

The basic idea for any variant of the network simplex method is an STS - Spanning Tree Structure, $(\mathcal{T}, L, U)$. Such a structure is obtained when, for any arc not belonging to this tree, the flow value is fixed at its lower bound or at its upper bound. All the arcs fixed at their lower bound belong to the set L, while all the arcs fixed at their upper bound belong to the set U. The remaining arcs are those belonging to the spanning tree $\mathcal{T}$.

A minimum cost network flow problem has always at least one STS optimal solution (see [1]). It is possible to find an optimal STS by moving from one STS to another, successively. At each iteration, we exchange a pair of arcs (one arc entering STS and one arc coming out from STS). Any STS corresponds to one feasible basic

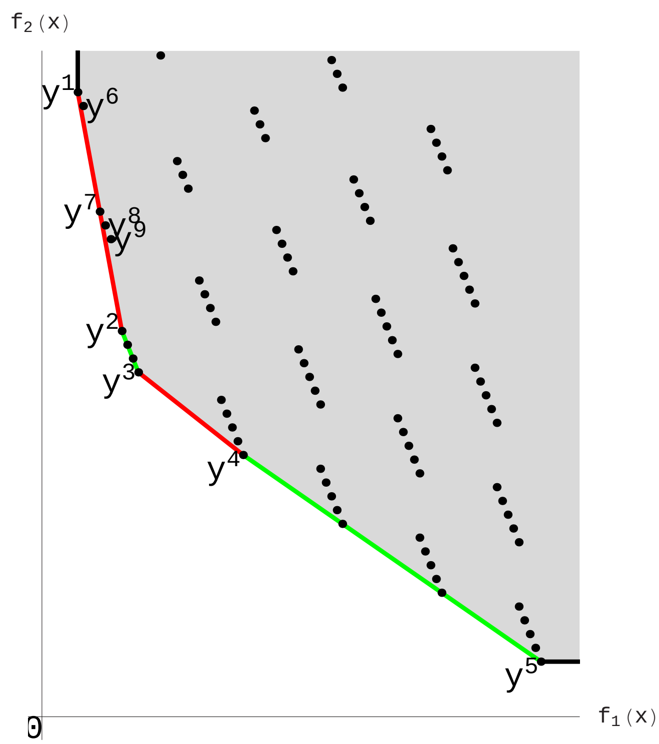


Figure 1:

solution in linear programming, and each move from one STS to another coincides with one pivoting operation in the standard simplex method.

The initialization of the algorithm consists of finding one feasible STS (or equivalently, a feasible basic solution in the standard simplex method). Two vectors are associated with this STS, the flow x (primal solution) and the potential π (dual solution). Each iteration of the method consists of:

1. identifying one eligible arc (k,l) with $(k,l) \notin \mathcal{T}$ (In a particular case the entering and the leaving arc can be the same.);
2. adding the arc (k,l) to $\mathcal{T}$ and finding an arc (p,q) coming out of $\mathcal{T}$;
3. updating STS and the primal and dual solution (x,π), obtaining an adjacent STS.

An arc not belonging to $\mathcal{T}$ is said to be eligible if:

1. Its reduced cost is strictly negative and its flow is at its lower bound, that is, $\bar{c}_{ij} < 0$ and $(i,j) \in L$.
2. Its reduced cost is strictly positive and its flow is at its upper bound, that is, $\bar{c}_{ij} > 0$ and $(i,j) \in U$.

The reduced cost of a given arc (i,j) is defined as follows:

$$\bar{c}_{ij} = c_{ij} - \pi_i + \pi_j$$

where, π_i and π_j are the dual variables associated with the vertices i and j, respectively. It should be noted that for all the arcs $(i,j) \in \mathcal{T}$ the reduced cost $\bar{c}_{ij} = 0$.

Algorithm 1: *Network simplex algoritm*

Simplex method.
{ *Determining a minimum cost flow.* }
begin
    let $(\mathcal{T},\mathcal{L},\mathcal{U})$ *be a starting feasible STS;*
    let x *denote the flow vector and* π *the dual vector associated with* $(\mathcal{T},\mathcal{L},\mathcal{U})$*;*
    while *(not optimal solution)* **do**
    begin
        select an entering arc (k,l) *not in* $\mathcal{T}$*;*
        add (k,l) *to* $\mathcal{T}$ *and remove* (p,q) *from* $\mathcal{T}$*;*

(8) *update the STS and vector x and π;*
(9) **end**
(10) **end**

In a particular problem an $STS(\nu)$ can be identified by its sets L and U. This solution can be represented by $\rho_\nu = (L, U)$ where, ν is a number identifying the solution. In case of a degenerate $STS(\nu)$, there are more then one STS associated with solution ν. Suppose there are, $r > 1$, STSs associated with ν, then $\rho_\nu^k, k = 1, 2, \ldots, r$ represent these r STSs.

4 Bi-criteria network simplex algorithm

In this section two methods for finding the set of efficient solutions are reviewed. This methods are quite the same as in [7, 9]. Both begin with an efficient STS and go through all adjacent efficient STS until finding all the extreme efficient solutions. The second algorithm differs from the algorithm in [20] only in the way it finds the adjacent efficient solutions, but both are equivalent.

The bi-criteria linear network flow problem is a particular case of the MMCF and can be stated as follows.

$$\begin{array}{ll} \text{“min”} & \{(c^1)^T x, (c^2)^T x\} \\ \text{subject to:} & Ax = b \\ & 0 \le x \le u \end{array} \tag{6}$$

4.1 Parametric linear program

Finding all the basic efficient solutions of (6) is equivalent to find the optimal solutions of a sequence of linear programming problems of the form,

$$\min\{\lambda(c^1)^T x + (1-\lambda)(c^2)^T x : x \in X\} \tag{7}$$

for all $\lambda \in]0, 1[$ (Teorema 2.1). Consider the reduced cost vector $\bar{c}$ of the parametric objective function,

$$\bar{c}(\lambda) = \lambda\bar{c}^1 + (1-\lambda)\bar{c}^2 \tag{8}$$

an optimal basic solution, x, of (7) for some λ and a nonbasic variable x_{ij}. Two cases should be considered,

A) If $(i, j) \in L$, i.e., $x_{ij} = 0$, then we have $\bar{c}_{ij}(\lambda) \ge 0$ and either $\bar{c}^1_{ij} \ge 0$ or $\bar{c}^1_{ij} < 0$.

1. If $\bar{c}^1_{ij} \ge 0$, two cases for the reduced costs of the second criterion should be considered.

(a) $\bar{c}^2_{ij} \geq 0$, which implies $\bar{c}_{ij}(\lambda) \geq 0$.

(b) $\bar{c}^2_{ij} < 0$, which means that the reduced cost for variable x_{ij} is non-negative when $\lambda \geq \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij}-\bar{c}^2_{ij}}$. In fact

$$\begin{aligned}
\bar{c}_{ij}(\lambda) \geq 0 \;&\Leftrightarrow\; \lambda c^1_{ij} + (1-\lambda)c^2_{ij} \geq 0 \\
&\Leftrightarrow\; \lambda(\bar{c}^1_{ij} - \bar{c}^2_{ij}) \geq -\bar{c}^2_{ij} \\
&\Leftrightarrow\; \lambda \geq \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij} - \bar{c}^2_{ij}}
\end{aligned}$$

2. $\bar{c}^1_{ij} < 0$, implies that $\bar{c}^2_{ij} \geq 0$, since $\bar{c}_{ij}(\lambda) = \lambda\bar{c}^1_{ij} + (1-\lambda)\bar{c}^2_{ij} \geq 0$ and $\lambda \in]0,1[$. Thus,

$$\begin{aligned}
\bar{c}_{ij}(\lambda) \geq 0 \;&\Leftrightarrow\; \lambda c^1_{ij} + (1-\lambda)c^2_{ij} \geq 0 \\
&\Leftrightarrow\; \lambda \leq \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij} - \bar{c}^2_{ij}}
\end{aligned}$$

B) If $(i,j) \in U$, i.e. $x_{ij} = u_{ij}$, then $\bar{c}_{ij}(\lambda) \leq 0$ and two cases for the reduced costs of the first criterion, may occur.

1. if $\bar{c}^1_{ij} \leq 0$, two cases can be considered.

(a) $\bar{c}^2_{ij} \leq 0$; which implies $\bar{c}_{ij}(\lambda) \leq 0$.

(b) $\bar{c}^2_{ij} > 0$, which means that the reduced cost of the new function for variable x_{ij} is non-positive when $\lambda \geq \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij}-\bar{c}^2_{ij}}$. In fact

$$\begin{aligned}
\bar{c}_{ij}(\lambda) \leq 0 \;&\Leftrightarrow\; \lambda c^1_{ij} + (1-\lambda)c^2_{ij} \leq 0 \\
&\Leftrightarrow\; \lambda(\bar{c}^1_{ij} - \bar{c}^2_{ij}) \leq -\bar{c}^2_{ij} \\
&\Leftrightarrow\; \lambda \geq \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij} - \bar{c}^2_{ij}}
\end{aligned}$$

2. if $\bar{c}^1_{ij} > 0$ then an optimal solution only occur when $\bar{c}^2_{ij} \leq 0$. Thus we have

$$\begin{aligned}
\bar{c}_{ij}(\lambda) \leq 0 \;&\Leftrightarrow\; \lambda c^1_{ij} + (1-\lambda)c^2_{ij} \leq 0 \\
&\Leftrightarrow\; \lambda \leq \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij} - \bar{c}^2_{ij}}
\end{aligned}$$

Consider the following sets

$$\begin{aligned} J_1 &= \{(i,j) \in L : \bar{c}^1_{ij} \geq 0 \text{ and } \bar{c}^2_{ij} < 0\} \\ &\cup \{(i,j) \in U : \bar{c}^1_{ij} \leq 0 \text{ and } \bar{c}^2_{ij} > 0\} \end{aligned} \tag{9}$$

and

$$\begin{aligned} J_2 &= \{(i,j) \in L : \bar{c}^1_{ij} < 0 \text{ and } \bar{c}^2_{ij} \geq 0\} \\ &\cup \{(i,j) \in U : \bar{c}^1_{ij} > 0 \text{ and } \bar{c}^2_{ij} \leq 0\} \end{aligned} \tag{10}$$

Let,

$$\lambda_1 = \max_{(i,j)\in J_1} \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij} - \bar{c}^2_{ij}} \tag{11}$$

and

$$\lambda_2 = \min_{(i,j)\in J_2} \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij} - \bar{c}^2_{ij}} \tag{12}$$

The current solution of the problem continues as an optimal one whenever λ varies within the range $[\lambda_1, \lambda_2]$.

The set of extreme efficient points can now be determined as follows,

1. Find the first efficient solution by solving (7) with $\lambda = 1$.

2. Then, iteratively, find the entering variable and the new λ values according to (11).

3. When for all non-basic variables, x_{ij}, $(i,j) \in L, \bar{c}^2_{ij} \geq 0$ or $(i,j) \in U, \bar{c}^2_{ij} \leq 0$, STOP.

This procedure is stated as Algorithm 2. Note that problem (7) with $\lambda = 1$ may have optimal alternative solutions and not all being efficient of problem (6); we need one basic efficient solution to run the algorithm. Let x denote an optimal solution and $y = (y_1, y_2) = (f_1(x), f_2(x))$ denote the image of x through F. Then if x is not efficient it means that there exists an efficient solution x' such that $y' = (y'_1, y'_2) = (f_1(x'), f_2(x'))$, $y' \leq y$ with $y' \neq y$. But since that $y_1 = \min_{x\in X} f_1(x)$ then $y'_2 < y_2$. Thus x' is an optimal solution for (7) with $\lambda = 1$. At that point x' is efficient if and only if there is a $\lambda \in]0,1[$ such that x' is an optimal solution of (7). Thus there is a $0 < \theta < 1$ such that x' is an optimal solution of (7) with $\lambda = 1 - \theta$. The reduced cost vector for (7) with $\lambda = 1 - \theta$ is

$$\bar{c}(\lambda) = \lambda\bar{c}^1 + (1-\lambda)\bar{c}^2 = (1-\theta)\bar{c}^1 + \theta\bar{c}^2$$

Solution x' is an optimal solution for (7) with $\lambda = 1$ and $\lambda = 1 - \theta$; it means that, with respect to x', all the arcs (i, j) in L with $\bar{c}^1_{ij} = 0$ have $\bar{c}^2_{ij} \geq 0$ and all the arcs (i, j) in U such that $\bar{c}^1_{ij} = 0$ have $\bar{c}^2_{ij} \leq 0$.

Algorithm 2: *Bi-criterion network simplex algorithm.*

Parametric bi-criteria network simplex algorithm
{ *Computing the set of efficient STSs and the set of efficient extreme points.* }
(1) **begin**
(2) *find a feasible STS or conclude that the problem is infeasible;*
(3) *find the set S_2 of all efficient STSs to problem (6) that are optimal for problem (7) with $\lambda = 1$;*
(4) $S_3 \leftarrow \{\}$;
(5) **while** $(S_2 \neq \{\})$ **do**
(6) **begin**
(7) *select an STS ρ_i from S_2;*
(8) $S_2 \leftarrow S_2 \backslash \{\rho_i\}$;
(9) $S_3 \leftarrow S_3 \cup \{\rho_i\}$;
(10) *Find the set S_1 of efficient adjacent STSs of ρ_i*
(11) *Let* $S_2 \leftarrow S_2 \cup S_1 \backslash S_3$
(12) **end**
(13) *find the set $EFF(X)$ of all efficient extreme points associated with the STS in S_3;*
(14) **end**

In step 10 the efficient adjacent STSs are identified from the current STS pivoting with the entering arc (k, l) such that

$$\lambda_1 = \frac{-\bar{c}^2_{kl}}{\bar{c}^1_{kl} - \bar{c}^2_{kl}} = \max_{(i,j) \in J_1} \frac{-\bar{c}^2_{ij}}{\bar{c}^1_{ij} - \bar{c}^2_{ij}}$$

or such that $\bar{c}^1_{kl} = \bar{c}^2_{kl} = 0$.

Example 4.2.
Consider the example of Figure 2. The objective is to find all basic feasible efficient points. The first step consists of solving the minimum cost network flow problem by minimizing the first criterion. The solution $x = (5, 5, 4, 1, 3, 6, 4)$ in Figure 11 (STS ρ_{20}) is obtained using Algorithm 1. This solution is not efficient for problem (2) since variable x_{13} is at its upper bound with reduced cost, $\bar{c}^1_{13} = 0$ and $\bar{c}^2_{13} > 0$. Among the strategies for finding all efficient STSs for problem (6), that are optimal solutions of (7) with $\lambda = 1$, the following two can be applied.

ρ	L	U
ρ_{20}	{}	{(1,3),(2,3),(3,5)}
ρ_{75}	{(3,4)}	{(2,3),(3,5)}
ρ_{24}	{}	{(1,3),(2,3),(4,5)}
ρ_{91}^1	{(1,3),(3,4)}	{(2,3)}
ρ_{91}^2	{(3,4)}	{(1,2),(2,3)}
ρ_{93}^1	{(1,3)}	{(2,3),(4,5)}
ρ_{93}^2	{}	{(1,2),(2,3),(4,5)}

Table 1: Set of all optimal solutions for $\lambda = 1$.

1. Find the set of all optimal STSs for (7), S_1, and remove from this set the inefficient STSs.
2. Moving from the current STS to an adjacent one till an efficient STS is found.

Consider the first strategy. Let S_1 denote the current set of optimal STSs for (7) and S_2 the set of STSs not examined yet. Then $S_1 = \{\rho_{20}\}$ and $S_2 = \{\rho_{20}\}$. Let $S_2 = \{\}$ and find the optimal STSs adjacent to ρ_{20}. Reduced costs $\bar{c}^1_{13} = 0$ and $\bar{c}^1_{35} = 0$ (see Figure 11). If either arc $(1,3)$ or $(3,5)$ is the entering arc the corresponding solution stills remain optimal. When the arc (1,3) enters in the tree ρ_{75} is reached and when arc (3,5) enters ρ_{24} is obtained; therefore, ρ_{75} and ρ_{24} are adjacent STSs to ρ_{20}. Let $S_2 = \{\rho_{75}, \rho_{24}\}$. The algorithm continues finding adjacent STSs to ρ_{75}: $S_1 = \{\rho_{20}, \rho_{75}\}$, $S_2 = \{\rho_{24}\}$. Solution ρ_{75} has three adjacent STSs: ρ_{20}, ρ^1_{91}, and ρ^2_{91}. Both ρ^1_{91} and ρ^2_{91} represent the same degenerate solution. Let $S_2 = \{\rho^1_{91}, \rho^2_{91}\}$ and $S_1 = \{\rho_{20}, \rho_{75}, \rho_{24}\}$ and find now the STSs adjacent to ρ_{24}. Continuing in such a way until $S_2 = \{\}$ the algorithm finds all optimal STSs $S_1 = \{\rho_{20}, \rho_{75}, \rho_{24}, \rho^1_{91}, \rho^2_{91}, \rho^1_{93}, \rho^2_{93}\}$ (see Figure 12). Only ρ^1_{91} and ρ^2_{91} are efficient STSs.

Instead of determine all optimal STSs we could try to move from the current optimal STS to another adjacent efficient STS. There is no guarantee to find the efficient solution faster than with the first method, but generally this is true. The STS ρ_{20} has $\bar{c}^1_{13} = 0$ and $\bar{c}^2_{13} = 12$ and arc $(1,3)$ belongs to U. As it was seen before an efficient STS cannot have an arc in U, with a reduced cost equal to zero for the first criterion and a reduced cost positive for the second one. Then, the algorithm finds the adjacent STS when arc $(1,3)$ enters in the tree. The obtained STS is ρ_{75} which is not efficient, as it can be seen the arc $(3,5)$ belongs to U and have reduced costs, $\bar{c}^1_{35} = 0$ and $\bar{c}^1_{35} = 7 > 0$. When inserting the arc $(3,5)$ into the tree, the STSs ρ^1_{91} and ρ^2_{91} are obtained; both are efficient. Now the second step of the algorithm can be performed with $S_2 = \{\rho^1_{91}, \rho^2_{91}\}$.

Consider the first STS, ρ^1_{91}, that is presented in Figure 13(a). Then $S_2 = \{\rho^2_{91}\}$ and $S_3 = \{\rho^1_{91}\}$, and consequently $J_1 = \{(2,3)\}$ which leads to

$$\lambda_1 = \max\{\frac{-22}{-2-22}\} = \frac{11}{12}.$$

Then the current solution is efficient for $\lambda \in [\frac{11}{12}, 1]$. When arc $(2,3)$ enters in the tree leads to the cycle shown in Figure 13(b). Afterwards the arc $(2,4)$ leaves the tree and a new efficient STS is obtained (see Figure 13(c)), ρ^1_{89}. Setting $S_2 = \{\rho^2_{91}, \rho^1_{89}\}$ and go to another iteration.

Consider now the STS ρ^2_{91}, let $S_2 = \{\rho^1_{89}\}$ and $S_3 = \{\rho^1_{91}, \rho^2_{91}\}$ and consequently $J_1 = \{(2,3)\}$ which leads to

$$\lambda_1 = \max\{\frac{-22}{-2-22}\} = \frac{11}{12}$$

which means that current solution is efficient for $\lambda \in [\frac{11}{12}, 1]$ and that $(2,3)$ enters in the tree. The arc $(2,4)$ leaves the tree and ρ^2_{89} is reached. Setting $S_2 = \{\rho^1_{89}, \rho^2_{89}\}$ and go to another iteration.

Consider the STS ρ^1_{89}, let $S_2 = \{\rho^2_{89}\}$ and $S_3 = \{\rho^1_{91}, \rho^2_{91}, \rho^1_{89}\}$. $J_1 = \{(1,3)\}$ and $\lambda_1 = \frac{15}{17}$. Consequently, the arc $(1,3)$ enters in tree and a new solution, ρ_{47}, is obtained (see Figure 13(e)). We set $S_2 = \{\rho^2_{89}, \rho_{47}\}$ and go to another iteration.

Consider the STS ρ^2_{89}, let $S_2 = \{\rho_{47}\}$ and $S_3 = \{\rho^1_{91}, \rho^2_{91}, \rho^1_{89}, \rho^2_{89}\}$. $J_1 = \{(1,2)\}$ and $\lambda_1 = \frac{15}{17}$. The solution ρ_{47} obtained is already in S_2. Go to next iteration.

Consider the STS ρ_{47}, let $S_2 = \{\}$ and $S_3 = \{\rho^1_{91}, \rho^2_{91}, \rho^1_{89}, \rho^2_{89}, \rho_{47}\}$. $J_1 = \{\}$. Since $S_2 = \{\}$ the cycle *while* ends.

The set of efficient solutions is

$$\begin{aligned} EFF(X) &= \{(10,0,4,6,0,4,6), \\ &\quad (10,0,3,7,0,3,7), \\ &\quad (7,3,0,7,0,3,7)\} \end{aligned}$$

■

4.2 Slope based technique

Consider problem (6). Suppose the characteristic of A is m, i.e., matrix A has a non-singular submatrix B, (Note: in a minimum cost flow problem the node-arc incidence matrix has characteristic $m-1$, but by inserting an artificial variable a matrix A

with characteristic m can be obtained [3]). Matrix A can then be decomposed into three submatrices, $A = \begin{pmatrix} B & N_L & N_U \end{pmatrix}$. Let $x_B = (x_{B_1}, x_{B_2}, \ldots, x_{B_m})$ denote the vector with variables associated to the $1^{st}, 2^{nd}, \ldots, m^{th}$ column of the matrix B, respectively, i.e. the basic variables. In the same way let x_L and x_U denote the variable's vectors associated to the columns of N_L and N_U, respectively. We have thus,

$$x = \begin{pmatrix} x_B \\ x_L \\ x_U \end{pmatrix} \text{ and } c = \begin{pmatrix} c_B \\ c_L \\ c_U \end{pmatrix}$$

Consequently, the system $Ax = b$ can be rewritten as

$$\begin{aligned} Ax = b \ &\Leftrightarrow \ Bx_B + N_L x_L + N_U x_U = b \\ &\Leftrightarrow \ Bx_B = b - N_L x_L - N_U x_U \\ &\Leftrightarrow \ x_B = B^{-1}b - B^{-1}N_L x_L - B^{-1}N_U x_U \end{aligned}$$

Solution $x_L = 0$, $x_U = u_U$ and $x_B = B^{-1}b - B^{-1}N_U u_U$ is called *a basic solution*. Furthermore, if $0 \le x_B \le u_B$ the solution is called *a feasible basic solution*. This type of solutions corresponds to a feasible STS, where the arcs in $\mathcal{T}$, L and U are associated with the variables x_B, x_L, and x_U, respectively.

Consider an STS and its associated extreme point x, an adjacent STS is associated with the extreme point $x + \Delta d$, $\Delta \ge 0$, where d is a vector, called *feasible direction* [4], with all components d_L and d_U equal to zero except one, d_{kl}, such that

- $d_{kl} = 1$ if $x_{kl} = 0$
- $d_{kl} = -1$ if $x_{kl} = u_{kl}$

and $d_B = -d_{kl}B^{-1}A_{kl}$, where A_{kl} is the kl^{th} column of A. The new STS is feasible if for all arcs (i, j)

$$0 \le x_{ij} + \Delta d_{ij} \le u_{ij} \tag{13}$$

The non-basic components of d fulfil these inequalities as soon as $\Delta \le u_{kl}$. The basic components have the following three cases to be considered:

1. If $d_{ij} = 0$ then (13) occurs for all $\Delta \ge 0$.
2. If $d_{ij} > 0$ then

$$\begin{aligned} & 0 \le x_{ij} + \Delta d_{ij} \le u_{ij} \\ \Leftrightarrow \quad & -\frac{x_{ij}}{d_{ij}} \le \Delta \le \frac{u_{ij} - x_{ij}}{d_{ij}} \end{aligned}$$

and since $\Delta \geq 0$,

$$\Delta \leq \frac{u_{ij} - x_{ij}}{d_{ij}}$$

3. If $d_{ij} < 0$ then

$$\begin{aligned} & 0 \leq x_{ij} + \Delta d_{ij} \leq \\ \Leftrightarrow \quad & -\frac{u_{ij} - x_{ij}}{d_{ij}} \leq \Delta \leq -\frac{x_{ij}}{d_{ij}} \end{aligned}$$

and since $\Delta \geq 0$

$$\Delta \leq -\frac{x_{ij}}{d_{ij}}$$

The new feasible STS, $x + \Delta d$, is obtained for Δ, computed as follows,

$$\Delta = \min \left\{ \min \left\{ \frac{u_{ij} - x_{ij}}{d_{ij}} : d_{ij} > 0 \right\}, \min \left\{ -\frac{x_{ij}}{d_{ij}} : d_{ij} < 0 \right\}, u_{kl} \right\}$$

Let x' denote the extreme point associated with the STS ρ'. Vector $y' = ((c^1)^T x, (c^2)^T x)$ is the image of x' through F. An adjacent STS ρ'' have an extreme point $x'' = x' + \Delta d$ with

$$\begin{aligned} y'' &= ((c^1)^T x + \Delta (c^1)^T d, (c^2)^T x) + \Delta (c^2)^T d \\ &= ((c^1)^T x + \Delta \bar{c}^1, (c^2)^T x + \Delta \bar{c}^2) \end{aligned}$$

as its image.

The line $[y', y'']$ has slope d that can be calculated as follows.

1. If $\bar{c}^1_{kl} \neq 0$ and $\Delta \neq 0$,

$$\begin{aligned} d = & \frac{(c^2)^T x + \Delta \bar{c}^2_{kl} - (c^2)^T x}{(c^1)^T x + \Delta \bar{c}^1_{kl} - (c^1)^T x} \\ = & \frac{\bar{c}^2_{kl}}{\bar{c}^1_{kl}} \end{aligned}$$

2. If $\Delta = 0$ then x' and x'' are the same point; the same occurs with y' and y''. Therefore, the line segment $[y', y'']$ is reduced to a single point.

3. If $\bar{c}^1_{kl} = 0$ then $[y', y'']$ is a vertical line. If $\bar{c}^2_{kl} = 0$ the line $[y', y'']$ is reduced to a point.

The efficient extreme solutions are determined by using the following steps:

1. The set of efficient STSs, S_2, and the associated extreme point, x such $F(x) = (y_1, y_2)$, are found, where $y_1 = \min\{f_1(x) : x \in X\}$. The reduced costs of this solution are

   (a) if $(i,j) \in L$, $\bar{c}^1_{ij} \geq 0$

   (b) and if $(i,j) \in U$, $\bar{c}^1_{ij} \leq 0$

2. For each STS in S_2 find all adjacent efficient STSs, evaluating the set

$$J_3 = \{(i,j) \in L : \bar{c}^1_{ij} > 0 \text{ and } \bar{c}^2_{ij} < 0\} \cup \{(i,j) \in U : \bar{c}^1_{ij} < 0 \text{ and } \bar{c}^2_{ij} > 0\}$$

   and if $J_3 \neq \{\}$ find all the arcs (k,l) such as

$$\frac{c^2_{kl}}{c^1_{kl}} = \min\{\frac{c^2_{ij}}{c^1_{ij}} : (i,j) \in J_3\}$$

   Consider also the arcs (k,l) such that $\bar{c}^1_{kl} = 0$ and $\bar{c}^2_{kl} = 0$ since these arcs lead to an alternative efficient STS. In this case new extreme efficient solutions may occur.

   The adjacent efficient STSs are obtained making basic the arcs (k,l). These STSs are included in S_2 whenever they are not already been in this set.

3. Repeat step 2 until $S_2 = \{\}$.

5 Characterizing non-dominated solutions

Consider the bi-criteria problem (6) with all integer variables. In multiple criteria linear programming only supported non-dominated criterion vectors are present. In multiple criteria integer programs, however, unsupported non-dominated criterion vectors should also be considered.

In this section we present examples of supported extreme non-dominated and non-extreme non-dominated vectors that are images of basic and non-basic feasible solutions, as well as examples of unsupported vectors that are images of basic and non-basic feasible solutions. We show also that there are non-basic non-intermediate solutions whose image, trough F, are supported vectors.

5.1 Supported vectors

The set of supported vectors is a set of images of solutions of X that are either basic or non-basic feasible solutions. As it will be seen the image of a basic solution is not necessarily an extreme vector of $Conv(Y)$. But all the extreme points of $Conv(Y)$ are images of at least one basic solution.

5.1.1 Basic solutions

There are basic solutions that have as images extreme and non-extreme non-dominated vectors of $Conv(Y)$.

a) Extreme non-dominated points of $Conv(Y)$

According to the discussion in Section 2.5 all extreme vectors of $Conv(Y)$ are vectors $y = F(x)$ such that x is a basic feasible solution but not all the images of basic feasible solutions are extreme vectors of $Conv(Y)$. Consider now the bi-criteria network flow example whose network $\mathcal{N}$ is pictured in Figure 3. The set of solutions to this example is given in Table 5 (Appendix A). The set of feasible vectors, Y, in the criterion space, (represented by bullets) and the convex hull, $Conv(Y)$, (the shaded area) are represented in Figure 4. It can be seen that all the extreme vectors are images of basic feasible solutions. The $Conv(Y)$ has five extreme vectors and all these vectors are images of at least one basic feasible solution. For instance, solution 5 is basic and its image is an extreme vector and there is no another solution that has this vector as image. But, solutions 4 and 48 are also basic solutions and have as image the same extreme vector. Consequently, one can say that one extreme point of $Conv(Y)$ can be the image of at least one basic feasible solution. But that are also images of other feasible solutions (not basic ones) that have an extreme point as image.

b) Non-extreme non-dominated points of $Conv(Y)$

Figure 4 presents basic solutions whose image, through F, are non-extreme points and particularly supported non-dominated non-extreme points of $Conv(Y)$. This is the case of solutions 47 and 90. Both are basic feasible solutions and their images, through the application F, are non-extreme non-dominated vectors of $Conv(Y)$.

5.1.2 Non-basic solutions

a) Extreme non-dominated points of $Conv(Y)$

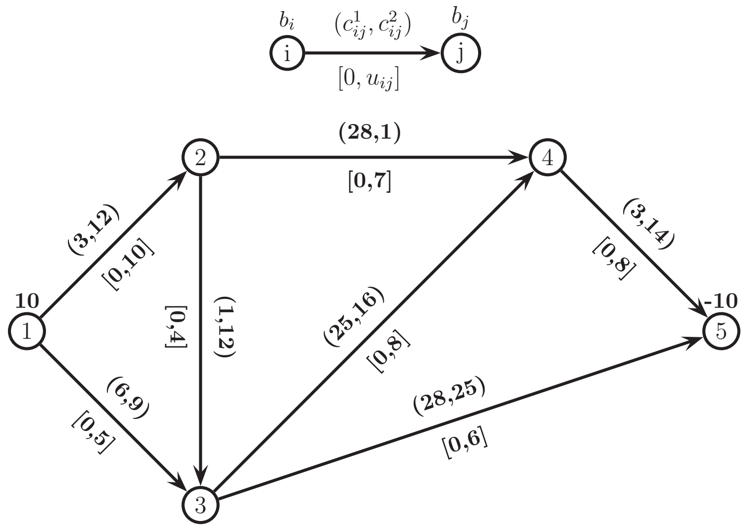


Figure 2: Bi-criteria network flow example.

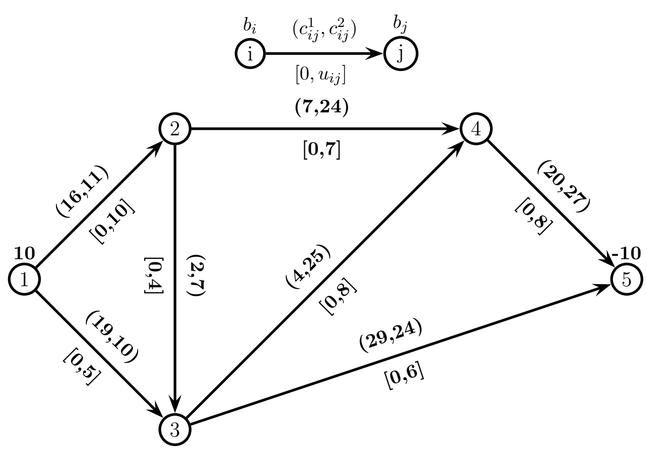


Figure 3: Bi-criteria network flow example.

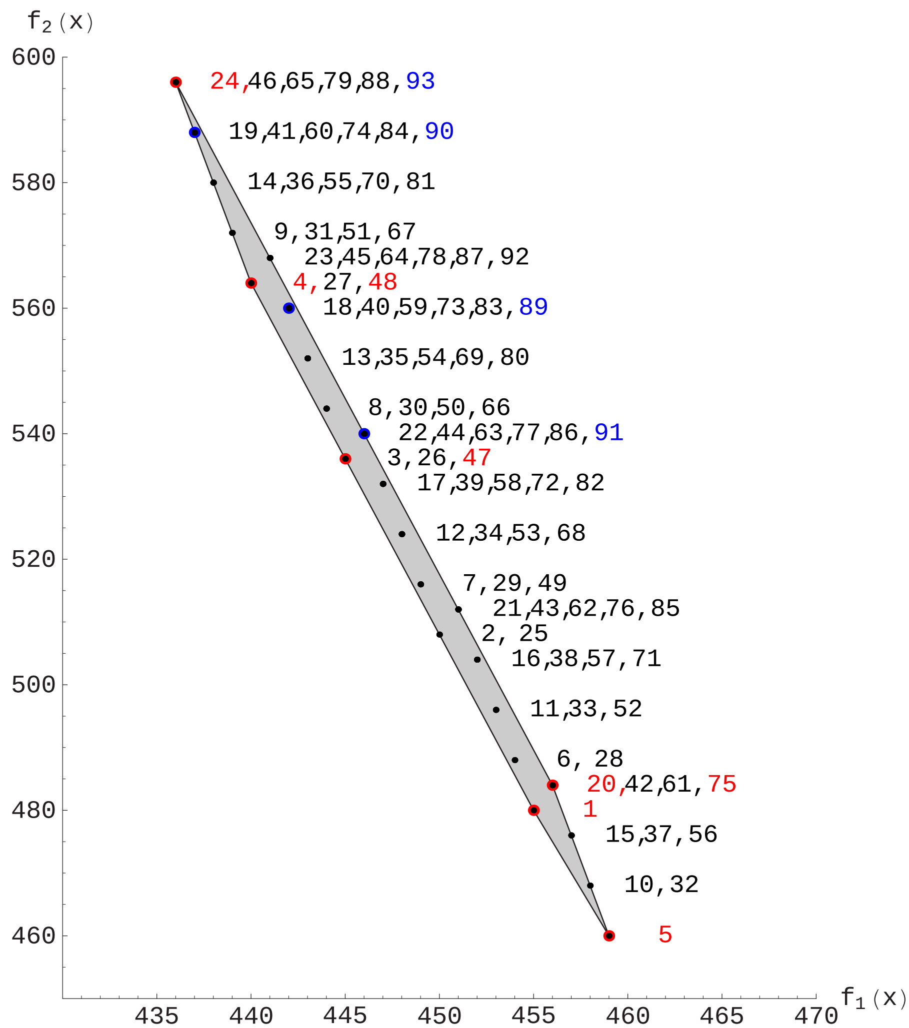


Figure 4: Feasible region in the criterion space of problem 4.

Every extreme point of $Conv(Y)$ is image of at least a basic feasible solution, but will it be image of non-basic solutions? The answer can be achieved looking at Figure 4. For instance, solutions 46, 65, 79, and 88 are all non-basic, but nevertheless their image, through F, is an extreme point of $Conv(Y)$. Therefore, there exist extreme vectors that are images of non-basic solutions. One can ask if these non-basic solutions whose images are extreme points are all intermediate solutions. The answer can be given by the following example.

Example 5.3.
Consider the network in Figure 5. This problem has 6 basic feasible solutions that are efficient, all degenerate (see Table 8). Some of these solutions are equivalent, that is, have the same image through F; it is the case of solutions 256, 61, and 22; all of them have the point $y = (290, 356)$ as image (see Table 7). Vector y is a non-dominated extreme-point of $Conv(Y)$ and 290 is the minimum of the first function, f_1, subject to the constraints of the problem. There are also non-basic solutions that have y as image some of them are intermediate solutions and others are non-intermediate solutions.

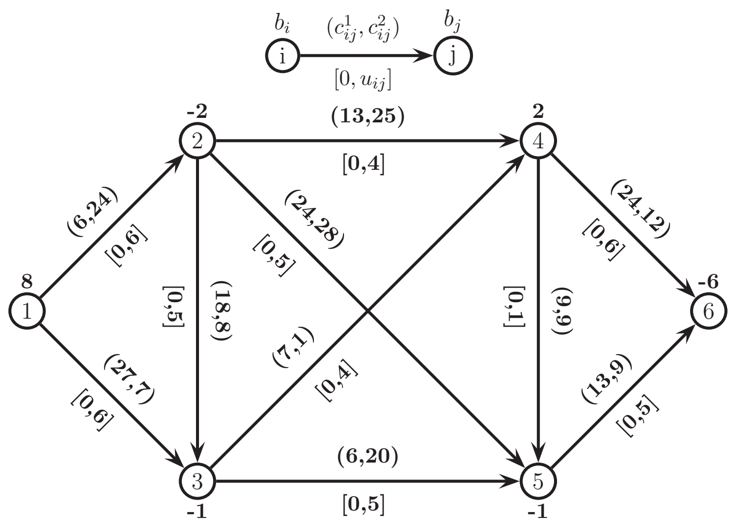


Figure 5: Bi-criteria network flow example

Figure 6 presents some of non-dominated vectors of this example. The point $y = (290, 356)$ is an extreme point of $Conv(Y)$. The point y is the image of solutions 256, 236, 189, 121, 61, 229, 177, 106, 48, 163, 92, 40, 79, 29, and 22 (see Table 7 in Appendix B) that are feasible solutions. Solutions 256, 61, and 22 are all basic and

Table 2: Intermediate-solutions from 256 to 61.

u	6	6	5	4	5	4	5	1	6	5
arc	(1,2)	(1,3)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)	(4,6)	(5,6)
256	6	2	0	4	0	0	1	1	5	1
236	6	2	0	3	1	0	1	1	4	2
189	6	2	0	2	2	0	1	1	3	3
121	6	2	0	1	3	0	1	1	2	4
61	6	2	0	0	4	0	1	1	1	5

Table 3: Intermediate-solutions from 256 to 22.

u	6	6	5	4	5	4	5	1	6	5
arc	(1,2)	(1,3)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)	(4,6)	(5,6)
256	6	2	2	0	2	0	3	1	1	5
229	6	2	1	3	0	0	2	1	4	2
163	6	2	2	2	0	0	3	1	3	3
79	6	2	3	1	0	0	4	1	2	4
22	6	2	4	0	0	0	5	1	1	5

the remaining ones are not. Solutions 256, 61, and 22 are also adjacent solutions. When moving from 256 to 61 solutions 236, 189, and 121 are obtained (see Table 2) trough the cycle $\{(2,4),(2,5),(4,6),(5,6)\}$. But when moving from 256 to 22 the non-basic solutions 229, 163, and 79 are reached (see Table 3). And, when moving from 61 to 22 non-basic solutions 48, 40, and 29 are achieved (see Table 4). Solutions 236, 189, 121, 229, 48, 163, 40, 79, and 29 are non-basic intermediate solutions whose image is y. Solutions 177, 106, and 92 are non-basic non-intermediate solutions and their image is the extreme point y.

This example will be detailed hereafter.

First, we consider solution 256. This is a degenerate basic solution and we may

Table 4: Intermediate-solutions from 61 to 22.

u	6	6	5	4	5	4	5	1	6	5
arc	(1,2)	(1,3)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)	(4,6)	(5,6)
61	6	2	0	0	4	0	1	1	1	5
48	6	2	1	0	3	0	2	1	1	5
40	6	2	2	0	2	0	3	1	1	5
29	6	2	3	0	1	0	4	1	1	5
22	6	2	4	0	0	0	5	1	1	5

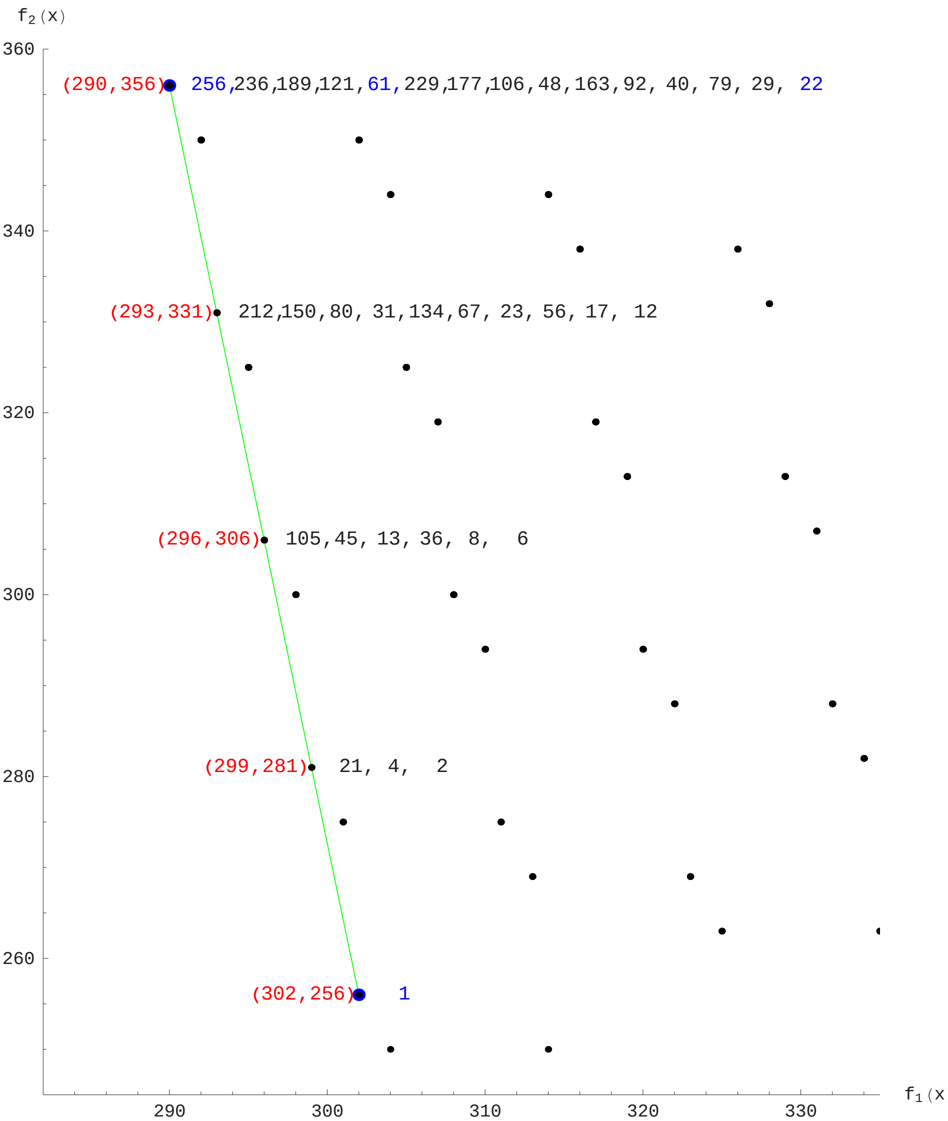


Figure 6: Part of feasible region in the criterion space of problem in Figure 5.

associate four STSs to this solution: ρ^1_{256} =($\{(2,3),\ (2,5),\ (3,4)\}$,$\{(2,4),\ (4,5)\}$), ρ^2_{256} =($\{(2,5),\ (3,4)\}$, $\{(1,2),\ (2,4),\ (4,5)\}$), ρ^3_{256}=($\{(2,3),\ (2,5),\ (3,4)\}$,$\{(1,2),$ $(4,5)\}$), ρ^4_{256} =($\{(2,3),\ (3,4)\}$,$\{(1,2),\ (2,4),\ (4,5)\}$). While ρ^2_{256}, ρ^3_{256}, and ρ^4_{256} are optimal STSs for f_1 in the sense that all arcs in L have non-negative reduced costs and all arcs in U have non-positive reduced costs, ρ^1_{256} is not optimal (see Figure 8).When we iterate to obtain the optimal solution from ρ^1_{256}, we get ρ^2_{256} or ρ^4_{256} that are optimal. In the ρ^2_{256} STS we have two arcs with reduced cost zero, x_{24} and x_{25}, which give rise to optimal alternative STSs. If the arc $(2,4)$ enters in the tree we have the cycle in Figure 10(b) and one of the arcs $(3,5)$ or $(5,6)$ comes out of the tree or the arc $(2,4)$ remains non-basic, which correspond to STSs ρ^4_{22}, ρ^2_{22}, and ρ^{12}_{22}, respectively. When the arc $(2,5)$ enters in tree, the arc $(2,3)$ comes out (see the cycle in Figure 10(c)) and the new STS is ρ^4_{256}. Therefore, we have four alternative optimal STSs associated with ρ^2_{256}. When moving from ρ^2_{256} to ρ^2_{22}, ρ^4_{22} or ρ^{12}_{22} there are three intermediate solutions 229, 163, and 79 making $\Delta = 1$, $\Delta = 2$ and $\Delta = 3$ in the cycle of Figure 10(b). In the STS ρ^3_{256} we have two zero reduced costs corresponding to the arcs $(2,3)$ and $(2,5)$. When the arc $(2,3)$ enters in tree we obtain ρ^{12}_{22}, ρ^4_{22}, or ρ^2_{22} through the cycle in Figure 10(b). When the arc $(2,5)$ enters in tree we obtain ρ^4_{61} or ρ^1_{61} through the cycle in Figure 10(a). When moving from ρ^3_{256} to ρ^1_{61}, or ρ^4_{61} there are three intermediate solutions 236, 189, and 121 leading to $\Delta = 1$, $\Delta = 2$ and $\Delta = 3$ in the cycle of Figure 10(a). In ρ^4_{256} we have two zero reduced costs corresponding to arcs $(2,3)$ and $(2,4)$. When the arc $(2,3)$ enters in tree we obtain the STS ρ^2_{256} through the cycle $\{(2,3),(2,5),(3,5)\}$ with $\Delta = 0$. When the arc $(2,4)$ enters in tree we obtain STS ρ^4_{61} or ρ^1_{61} through the cycle in Figure 10(a).

Consider now the solution number 61. This is also a degenerate solution with four STSs associated with it ρ^1_{61}, ρ^2_{61}, ρ^3_{61}, and ρ^4_{61}, represented in Figure 9(a), (b), (c) and (d), respectively. Only two of these solutions are optimal ρ^1_{61} and ρ^4_{61}. The Solution ρ^1_{61} has ρ^1_{22}, ρ^2_{22}, ρ^3_{256}, and ρ^4_{256} as adjacent solutions and ρ^4_{61} has ρ^7_{22}, ρ^{12}_{22}, ρ^3_{256} and ρ^4_{256} as adjacent solutions. When moving from ρ^1_{61} to ρ^1_{22} or ρ^2_{22} we have 48, 40 and 29 as intermediate solutions and the same happens when moving from ρ^4_{61} to ρ^7_{22} or ρ^{12}_{22}.

Figure 7 shows the graph where each node is the optimal STS that have as image $y = (290,356)$. Two nodes have an arc linking each other if they represent optimal adjacent STSs. We add also the non-optimal STSs for f_1 which are wrapped by a rectangle whose image through F is y. Passing from each STS to its adjacent STS we obtained as intermediate solutions $236, 189, 121$, $229, 163, 79$, and $48, 40, 29$ but there are also solutions 177, 106, and 92 that are not obtained when moving from one STS solution to its adjacent. These solution are not intermediate.

■

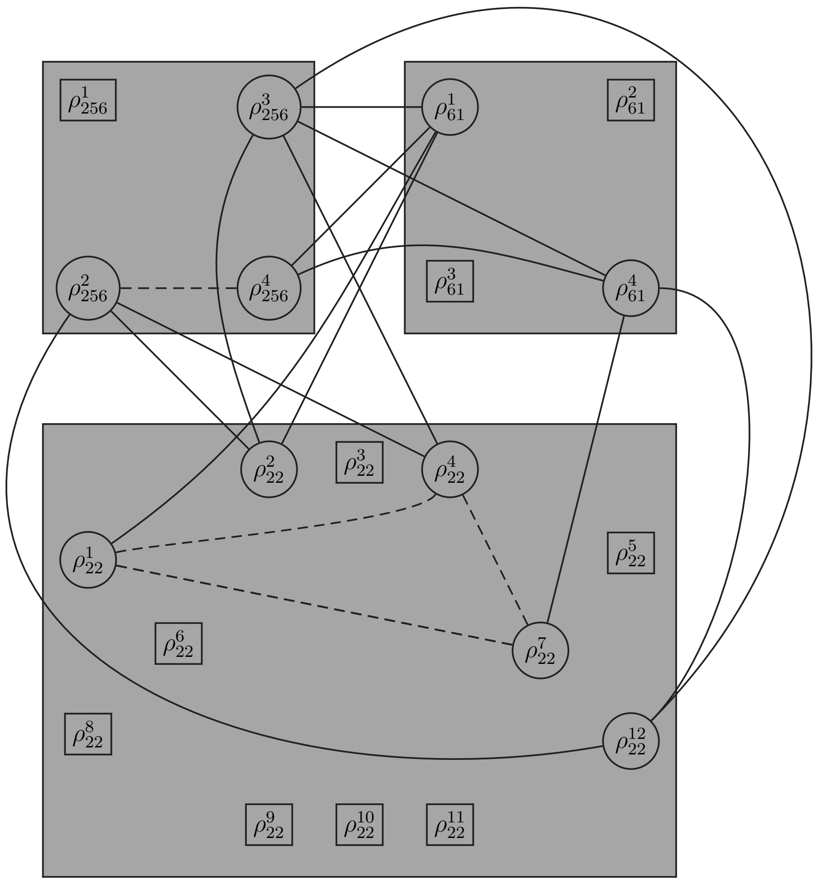


Figure 7: Connection between optimal STSs having image y from Example 3.

b) Non-extreme non-dominated points of $Conv(Y)$

Consider again the example of Figure 3. Solutions 9, 14, and 19 are non-basic intermediate (see Figure 4) because they are obtained when moving from 24 to 4 through the cycle $\mathcal{C} = \{(2,3),(2,4),(3,4)\}$. These points have images in $Conv(Y)$ that are supported non-dominated non-extreme points.

5.2 Unsupported non-dominated vectors

Contrary to continuous linear programming in which all non-dominated points are in the boundary of $Conv(Y)$, in integer linear programming, non-dominated vector may belong to the interior of $Conv(Y)$. These vectors are images of basic or non-basic solutions. For instance problem 3 has non-basic solutions 6 and 8 with an image corresponding to an unsupported non-dominated vector, that is, a non-dominated vector in the interior of $Conv(Y)$.

6 The ε-constraint method

This section presents a method that solves bi-criteria network flow problems without characterizing their network structure.

At each iteration, the network simplex method shown in Algorithm 1 always gives an integer solution for the minimum cost network flow problem. But it is possible to obtain non-integer solutions between two adjacent STSs. Let us recall that when moving from one STS to an adjacent one, an amount of flow, $\triangle$, must be sent along the orientation of cycle $\mathcal{C}$. This quantity $\triangle$ is integer. But, what happens if a non-integer amount of flow is sent along $\mathcal{C}$? It is obvious that a non-integer solution will be obtained. This solution has exactly $|\mathcal{C}|$ non-integer variables, but it does not define a spanning tree structure.

In multiple criteria linear programming several techniques (scalar optimization problems) can be used in order to characterize efficient solutions (non-dominated vectors) like, for example, weighted-sum approaches, Tchebycheff metrics based methods, ε-constraint methods, and so on [24]. Among the existing methods, the ε-constraint approach can be easily used in multiple criteria integer problems without any additional restrictions. Efficient solutions can be characterized as optimal solutions for the ε-constraint problem.

The ε-constraint problem associated with the bi-criteria (6) can be stated as follows:

$$\begin{array}{ll} \min & f_1(x) \\ \text{subject to:} & \\ & x \in X \\ & f_2(x) \leq \varepsilon, \end{array} \tag{14}$$

where ε is a scalar.

In the ε-constraint method, ε varies among all the values for which (14) remains feasible. So, in order to identify a set of efficient solutions, a sequence of problems (3) is solved for each different value of ε [5]. For integer bi-criteria linear programming problems the entire non-dominated set $ND(Y)$ can be easily determined by solving a sequence of problems (14).

Theorem ([11]). *Consider $\varepsilon \geq \min f_2(x)$. If the solution x^* solves problem (3) and when x^* is not unique it leads to a minimal value for criterion $f_2(x)$, then x^* solves (1), that is, x^* is an efficient solution for (1).*

Proof:
Suppose now that x^* does not solve the problem. Another solution $\hat{x}$ can then be considered so that only one of the following two cases can occur:

- $f_1(\hat{x}) < f_1(x^*)$ and $f_2(\hat{x}) \leq f_2(x^*)$ which contradicts the fact that x^* solves (14), or
- $f_2(\hat{x}) < f_2(x^*)$ and $f_1(\hat{x}) \leq f_1(x^*)$, which contradicts the hypothesis that x^* is optimal for (14) with the smallest value for $f_2(x)$.

The theorem is proved by the two cases above.

□

Problem (14) will be used in the algorithm outlined in Section 4 to determine all the non-dominated vectors for problem (6).

The ε-constraint optimization along with a branch-and-bound technique can be used to compute all efficient solutions of a bi-criteria integer network flow problem (see [9]). When applying this method we do not need to compute consecutively the supported extreme solutions and then the unsupported ones. The method allows to compute all the solutions by a lexicographic order.

7 Conclusions

This article dealt with bi-criteria network flow. A summary of the main concepts on the theory network flow problems was provided. A set of examples shows that several published methods designed for computing the set of non-dominated solutions contains drawbacks because they do not take into account all the possible cases for this type of problems.

The study done in this paper was especially devoted to point out the main characteristics of the set of non-dominated solutions. The number of the papers for bi-criteria network flows in the area is rather rare. It stills exist very little research on this very topic. It is a difficult problem, but the progresses made on this subject will be used in a vast range of practical problems and also in some theoretical works. The main purpose of this work was thus to characterize the non-dominated space.

Acknowledgements
The authors would like to acknowledge the support from the RAMS research project from CEG-IST.

References

[1] R. Ahuja, T.L. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey, 1993.

[2] P. Armand and C. Malivert. Determination of the efficient set in multiobjective linear programming. *Journal of Optimization Theory and Applications*, 70(3):467–489, 1991.

[3] M.S. Bazaraa, J.J. Jarvis, and H.D. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, 1977.

[4] D. Bertsimas and J.N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, Belmont, 1977.

[5] V. Chankong and Y.Y. Haimes. *Multiobjective Decision Maker: Theory and Methodology*. North-Holland, New York, 1983.

[6] J.C. Ecker and I.A. Kouada. Finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming*, 14(2):249–261, 1978.

[7] M. Ehrgott. *Multicriteria Optimization*. Springer-Verlag, Berlin, 2nd edition, 2005.

[8] J.B. Evans and R.E. Steuer. A revised simplex method for linear multiple objective programs. *Mathematical Programming*, 5(1):54–72, 1973.

[9] J. Figueira. On the integer bi-criteria network flow problem: A branch-and-bound approach. *Cahier du LAMSADE*, 191, 2002.

[10] T. Gal. *Postoptimal Analyses, Parametric Programming, and Related Topics.* Walter de Gruyter, Berlin, 2nd edition, 1995.

[11] Y.Y. Haimes, L.S. Lasdon, and D. A. Wismer. On a bicriterion formulation of the problems of integrated system identification and system optimization. *IEEE Transactions on Systems, Man, and Cybernetics*, pages 296–297, 1971.

[12] H.W. Hamacher, C.R. Pedersen, and S. Ruzika. Multiple objective minimum cost flow problems: A review. *European Journal of Operational Research*, 2006. To appear.

[13] H. Isermann. The enumeration of the set of all efficient solutions for a linear multiple objective program. *Operational Research Quarterly*, 28(3):711–725, 1977.

[14] H. Lee and P.S. Pulat. Bicriteria network flow problems: Continuous case. *European Journal of Operational Research*, 51:119–126, 1991.

[15] H. Lee and P.S. Pulat. Bicriteria network flow problems: Integer case. *European Journal of Operational Research*, 66:148–157, 1993.

[16] R. Malhotra and M.C. Puri. Bi-criteria network problem. *Cahiers du C.E.R.O.*, 26(1):95–102, 1984.

[17] A. Mustafa and M. Goh. Finding integer efficient solutions for bicriteria and tricriteria network flow problems using DINAS. *Computers & Operations Research*, 25(2):139–157, 1998.

[18] P. Neumayer and D. Schweigert. Three algorithms for bicriteria integer linear programs. *OR Spektrum*, 16:267–276, 1994.

[19] A. Przybylski, X. Gandibleux, and M. Ehrgott. The biobjective integer minimum cost flow problem - incorrectness of sedeño-noda and gonzàlez-martin's algorithm. *Computers & Operations Research*, 33(5):1459–1463, 2006.

[20] P.S. Pulat, F. Huarng, and H. Lee. Efficient solutions for the bicriteria network flow problem. *Computers & Operations Research*, 19(7):649–655, 1992.

[21] G. Ruhe. Complexity results for multicriterial and parametric network flows using a pathological graph of zadeh. *Zeitschrift für Operations Research*, 32:9–27, 1988.

[22] A. Sedeño-Noda and C. González-Martín. The biobjective minimum cost flow problem. *European Journal of Operational Research*, 124:591–600, 1999.

[23] A. Sedeño-Noda and C. González-Martín. An algorithm for the biobjective integer minimum cost flow problem. *Computers & Operations Research*, 28:139–156, 2001.

[24] R.E. Steuer. *Multiple Criteria Optimization: Theory, Computation, and Application*. John Wiley & Sons, New York, 1986.

[25] P.L. Yu. *Multiple-Criteria Decision Making: Concepts Techniques, and Extensions*. Plenum Press, New York, 1985.

[26] M. Zeleny. *Linear Multiobjective Programming*, volume 95. Springer-Verlag, New York, 1972. Lecture Notes in Economics and Mathematical Systems.

Appendices

A All feasible solutions of example on Figure 3

Table 5: Solutions of the bi-criterion network flow on Figure 3.

l_{ij}	0	0	0	0	0	0	0		
u_{ij}	10	5	4	7	8	6	8		
	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	(3,5)	(4,5)	y_1	y_2
1	**5**	**5**	**0**	**5**	**0**	**5**	**5**	**455**	**480**
2	5	5	0	5	1	4	6	450	508
3	5	5	0	5	2	3	7	445	536
4	**5**	**5**	**0**	**5**	**3**	**2**	**8**	**440**	**564**
5	**5**	**5**	**1**	**4**	**0**	**6**	**4**	**459**	**460**
6	5	5	1	4	1	5	5	454	488
7	5	5	1	4	2	4	6	449	516
8	5	5	1	4	3	3	7	444	544
9	5	5	1	4	4	2	8	439	572
10	5	5	2	3	1	6	4	458	468
11	5	5	2	3	2	5	5	453	496
12	5	5	2	3	3	4	6	448	524
13	5	5	2	3	4	3	7	443	552
14	5	5	2	3	5	2	8	438	580
15	5	5	3	2	2	6	4	457	476
16	5	5	3	2	3	5	5	452	504
17	5	5	3	2	4	4	6	447	532
18	5	5	3	2	5	3	7	442	560
19	5	5	3	2	6	2	8	437	588
20	**5**	**5**	**4**	**1**	**3**	**6**	**4**	**456**	**484**
21	5	5	4	1	4	5	5	451	512
22	5	5	4	1	5	4	6	446	540
23	5	5	4	1	6	3	7	441	568
24	**5**	**5**	**4**	**1**	**7**	**2**	**8**	**436**	**596**
25	6	4	0	6	0	4	6	450	508
26	6	4	0	6	1	3	7	445	536
27	6	4	0	6	2	2	8	440	564
28	6	4	1	5	0	5	5	454	488

continued on the next page

l_{ij}	0	0	0	0	0	0	0		
u_{ij}	10	5	4	7	8	6	8		
	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	(3,5)	(4,5)	y_1	y_2
29	6	4	1	5	1	4	6	449	516
30	6	4	1	5	2	3	7	444	544
31	6	4	1	5	3	2	8	439	572
32	6	4	2	4	0	6	4	458	468
33	6	4	2	4	1	5	5	453	496
34	6	4	2	4	2	4	6	448	524
35	6	4	2	4	3	3	7	443	552
36	6	4	2	4	4	2	8	438	580
37	6	4	3	3	1	6	4	457	476
38	6	4	3	3	2	5	5	452	504
39	6	4	3	3	3	4	6	447	532
40	6	4	3	3	4	3	7	442	560
41	6	4	3	3	5	2	8	437	588
42	6	4	4	2	2	6	4	456	484
43	6	4	4	2	3	5	5	451	512
44	6	4	4	2	4	4	6	446	540
45	6	4	4	2	5	3	7	441	568
46	6	4	4	2	6	2	8	436	596
47	**7**	**3**	**0**	**7**	**0**	**3**	**7**	**445**	**536**
48	**7**	**3**	**0**	**7**	**1**	**2**	**8**	**440**	**564**
49	7	3	1	6	0	4	6	449	516
50	7	3	1	6	1	3	7	444	544
51	7	3	1	6	2	2	8	439	572
52	7	3	2	5	0	5	5	453	496
53	7	3	2	5	1	4	6	448	524
54	7	3	2	5	2	3	7	443	552
55	7	3	2	5	3	2	8	438	580
56	7	3	3	4	0	6	4	457	476
57	7	3	3	4	1	5	5	452	504
58	7	3	3	4	2	4	6	447	532
59	7	3	3	4	3	3	7	442	560
60	7	3	3	4	4	2	8	437	588
61	7	3	4	3	1	6	4	456	484
62	7	3	4	3	2	5	5	451	512
63	7	3	4	3	3	4	6	446	540
64	7	3	4	3	4	3	7	441	568
65	7	3	4	3	5	2	8	436	596
66	8	2	1	7	0	3	7	444	544

continued on the next page

l_{ij}	0	0	0	0	0	0	0		
u_{ij}	10	5	4	7	8	6	8		
	(1,2)	(1,3)	(2,3)	(2,4)	(3,4)	(3,5)	(4,5)	y_1	y_2
67	8	2	1	7	1	2	8	439	572
68	8	2	2	6	0	4	6	448	524
69	8	2	2	6	1	3	7	443	552
70	8	2	2	6	2	2	8	438	580
71	8	2	3	5	0	5	5	452	504
72	8	2	3	5	1	4	6	447	532
73	8	2	3	5	2	3	7	442	560
74	8	2	3	5	3	2	8	437	588
75	**8**	**2**	**4**	**4**	**0**	**6**	**4**	**456**	**484**
76	8	2	4	4	1	5	5	451	512
77	8	2	4	4	2	4	6	446	540
78	8	2	4	4	3	3	7	441	568
79	8	2	4	4	4	2	8	436	596
80	9	1	2	7	0	3	7	443	552
81	9	1	2	7	1	2	8	438	580
82	9	1	3	6	0	4	6	447	532
83	9	1	3	6	1	3	7	442	560
84	9	1	3	6	2	2	8	437	588
85	9	1	4	5	0	5	5	451	512
86	9	1	4	5	1	4	6	446	540
87	9	1	4	5	2	3	7	441	568
88	9	1	4	5	3	2	8	436	596
89	**10**	**0**	**3**	**7**	**0**	**3**	**7**	**442**	**560**
90	**10**	**0**	**3**	**7**	**1**	**2**	**8**	**437**	**588**
91	**10**	**0**	**4**	**6**	**0**	**4**	**6**	**446**	**540**
92	10	0	4	6	1	3	7	441	568
93	**10**	**0**	**4**	**6**	**2**	**2**	**8**	**436**	**596**

Source: *Figueira 2002*

B Solutions for example 3

u	6	6	5	4	5	4	5	1	6	5		
	(1,2)	(1,3)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)	(4,6)	(5,6)	f1	f2
256	**6**	**2**	**0**	**4**	**0**	**0**	**1**	**1**	**5**	**1**	**290**	**356**
236	6	2	0	3	1	0	1	1	4	2	290	356
189	6	2	0	2	2	0	1	1	3	3	290	356
121	6	2	0	1	3	0	1	1	2	4	290	356
61	**6**	**2**	**0**	**0**	**4**	**0**	**1**	**1**	**1**	**5**	**290**	**356**
229	6	2	1	3	0	0	2	1	4	2	290	356
177	6	2	1	2	1	0	2	1	3	3	290	356
106	6	2	1	1	2	0	2	1	2	4	290	356
48	6	2	1	0	3	0	2	1	1	5	290	356
163	6	2	2	2	0	0	3	1	3	3	290	356
92	6	2	2	1	1	0	3	1	2	4	290	356
40	6	2	2	0	2	0	3	1	1	5	290	356
79	6	2	3	1	0	0	4	1	2	4	290	356
29	6	2	3	0	1	0	4	1	1	5	290	356
22	**6**	**2**	**4**	**0**	**0**	**0**	**5**	**1**	**1**	**5**	**290**	**356**

Table 7: Solutions whose image is point $(290, 356)$ in Figure 6.

u	6	6	5	4	5	4	5	1	6	5		
arco	(1,2)	(1,3)	(2,3)	(2,4)	(2,5)	(3,4)	(3,5)	(4,5)	(4,6)	(5,6)	y1	y2
256	6	2	0	4	0	0	1	1	5	1	290	356
61	6	2	0	0	4	0	1	1	1	5	290	356
22	6	2	4	0	0	0	5	1	1	5	290	356
1	2	6	0	0	0	0	5	1	1	5	302	256
3	2	6	0	0	0	0	5	0	2	4	304	250
109	2	6	0	0	0	4	1	0	6	0	352	226

Table 8: Basic feasible solutions of the problem in Figure 5.

C Extreme non-dominated points of Conv(Y)

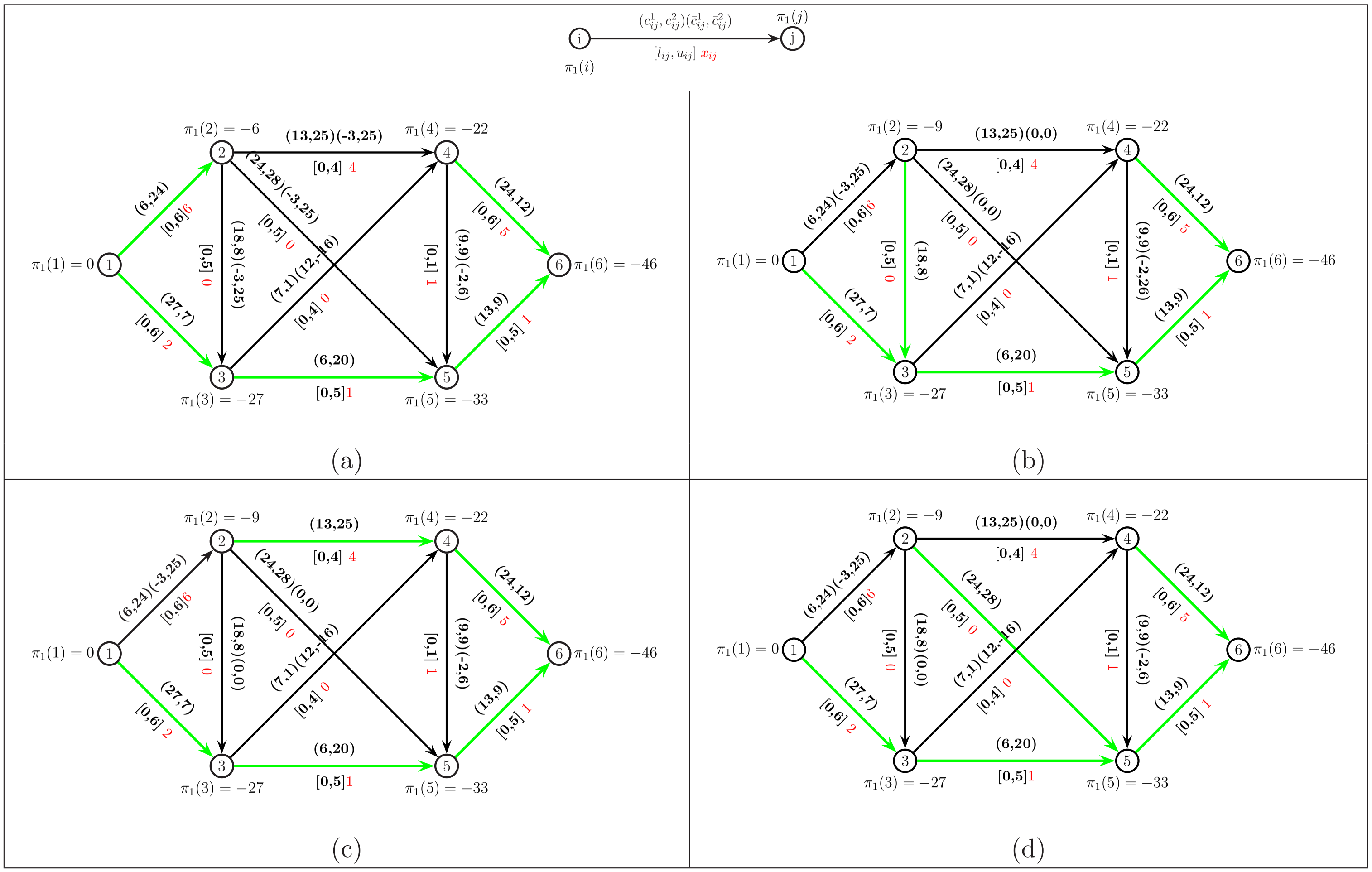


Figure 8: Solution 256 has as basic arcs $(1,3),(3,5),(4,6),(5,6)$ plus one of the following arcs: (a) $(1,2)$; (b) $(2,3)$; (c) $(2,4)$; (d) $(2,5)$.

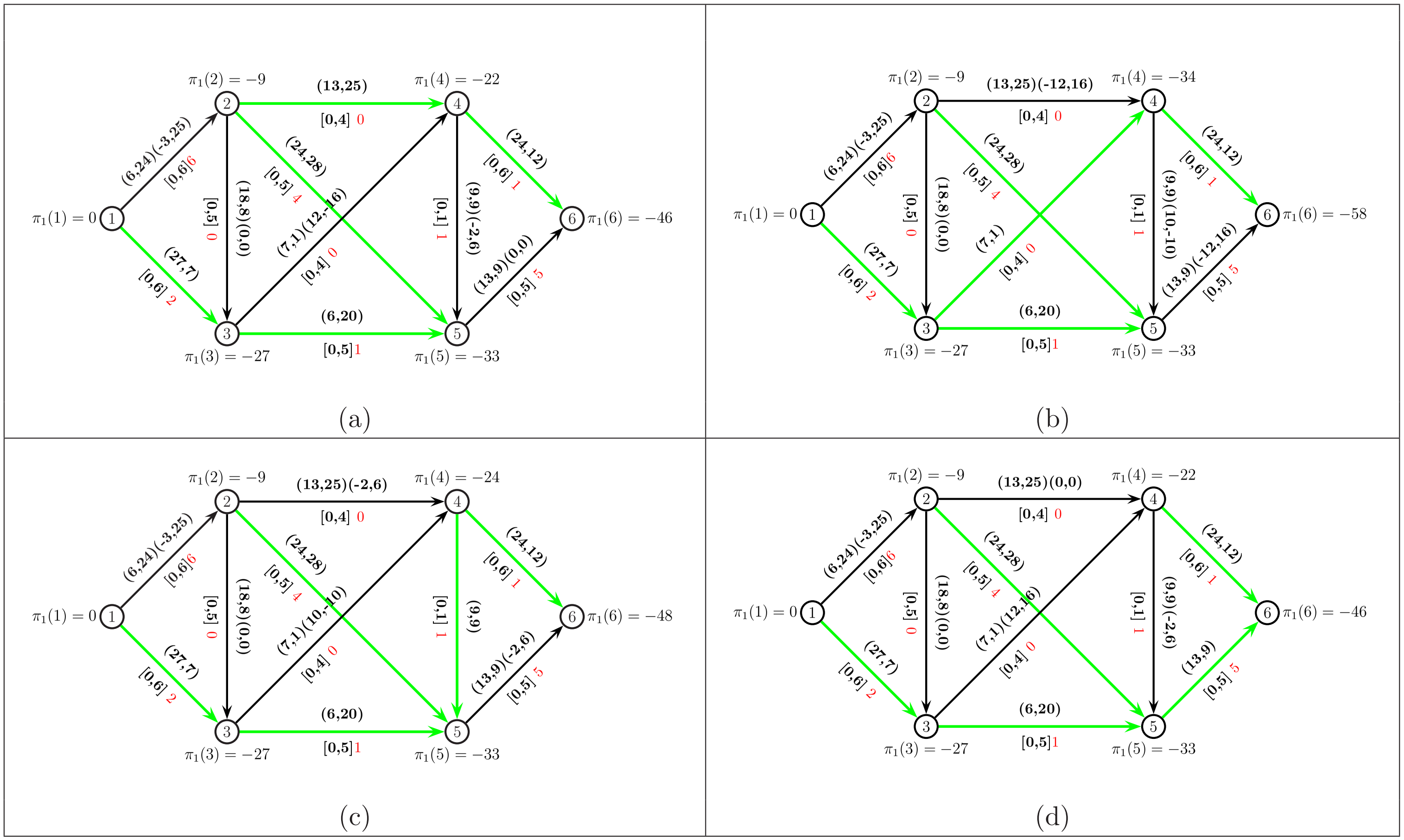


Figure 9: Solution 61 has as basic arcs $(1,2),(2,5),(3,5),(4,6)$ plus one of the following arcs: (a) $(2,4)$; (b) $(3,4)$; (c)$(4,5)$; (d)$(5,6)$.

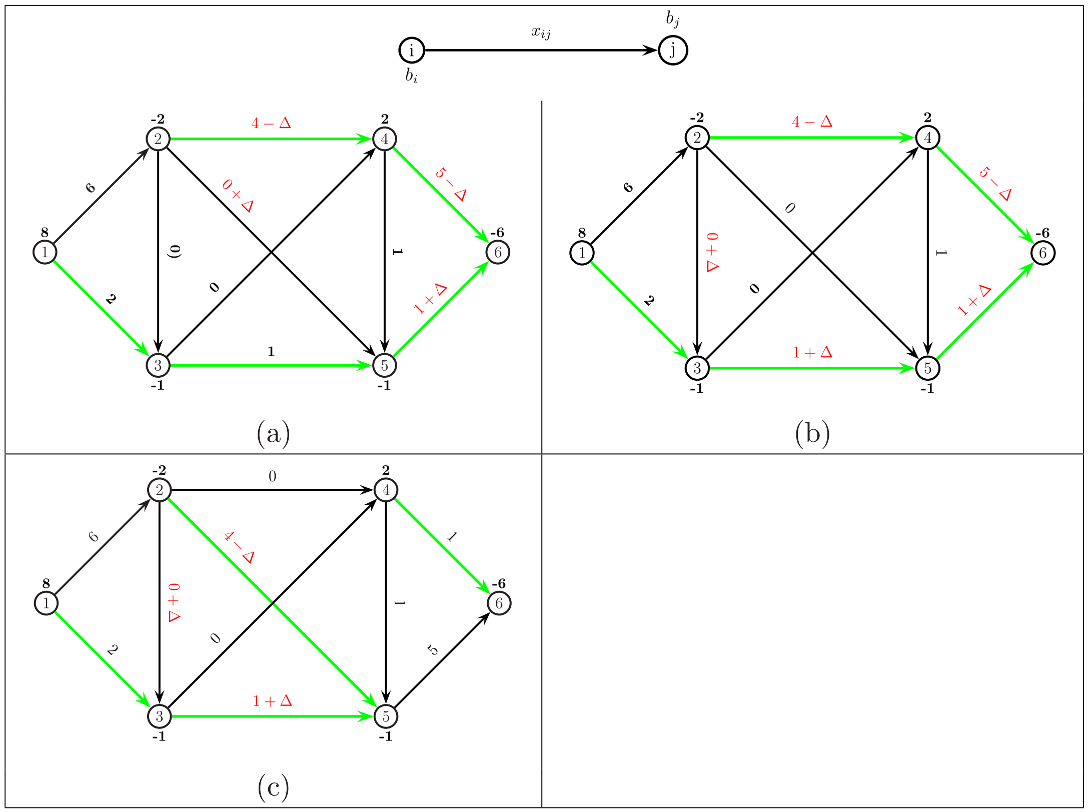
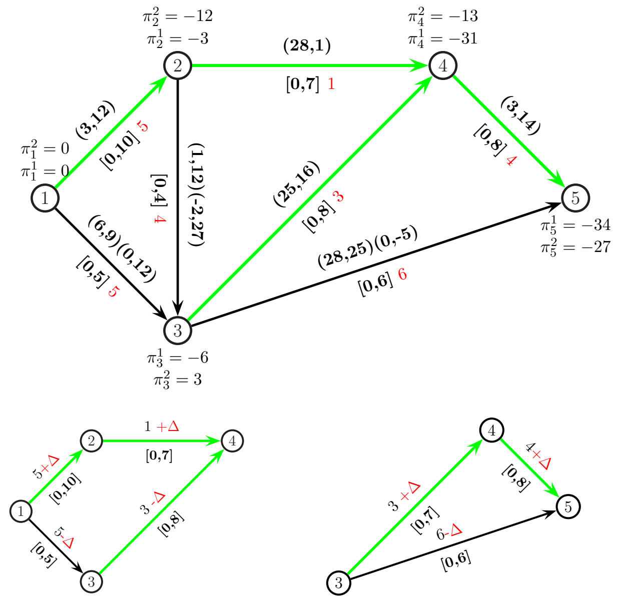


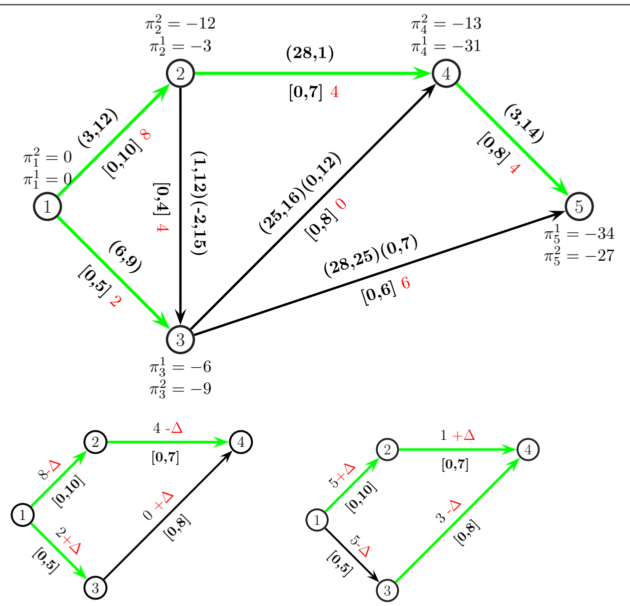
Figure 10: (a) Cycles 1; (b) 2; (c) 3.

D Solutions of example 2

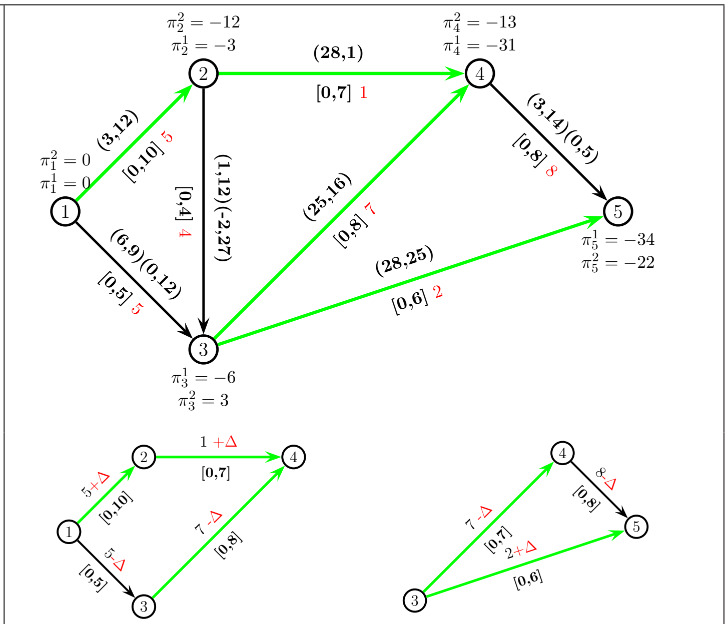
ρ_{20}



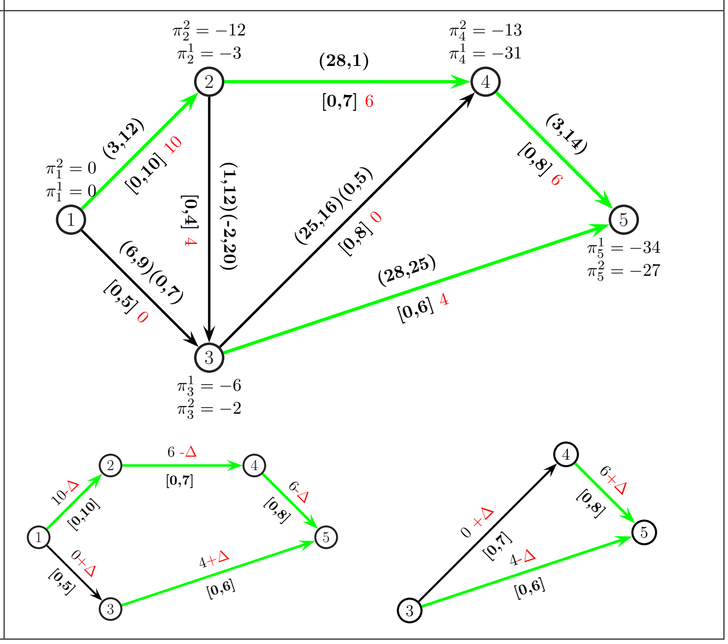
ρ_{75}



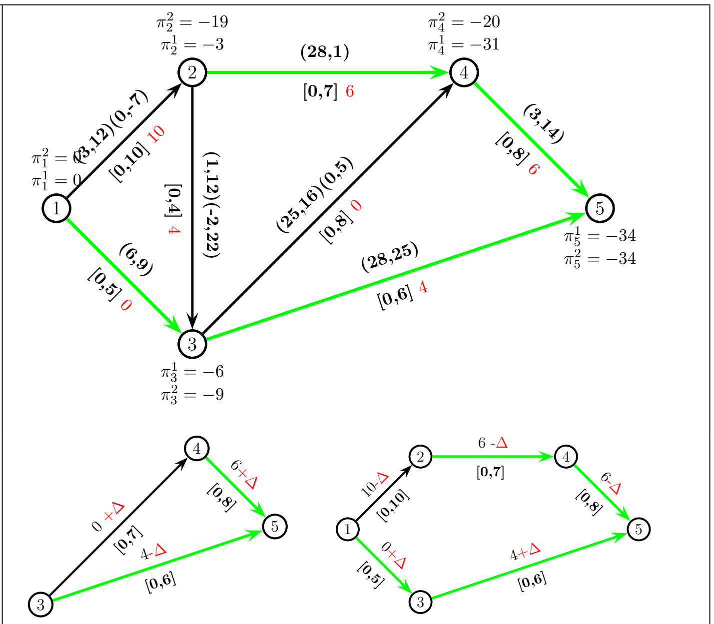
ρ_{24}



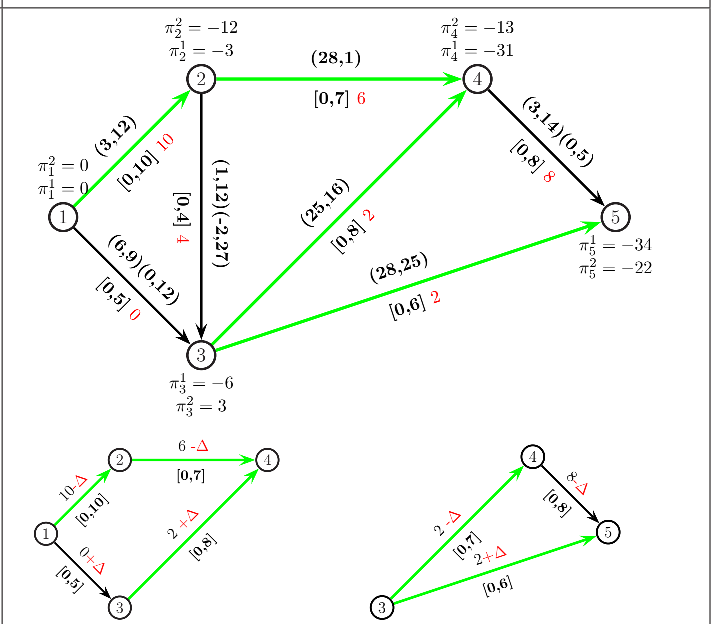
ρ_{91}^1



ρ^2_{91}



ρ^1_{93}



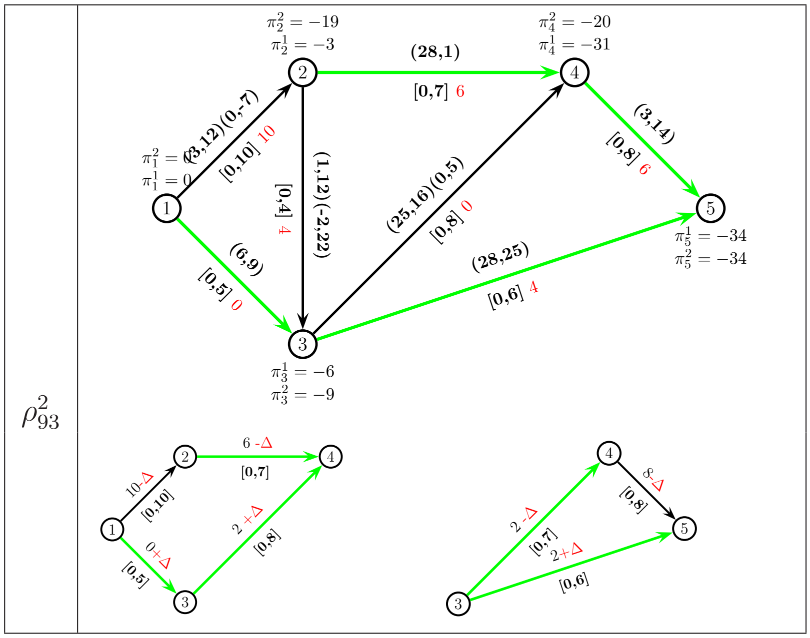


Figure 11: Example 2 - optimal STSs considering only the first criterion.

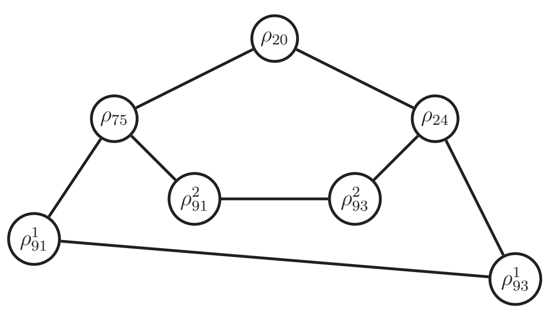


Figure 12: Example 2 - Connections between optimal STSs using the first criterion.

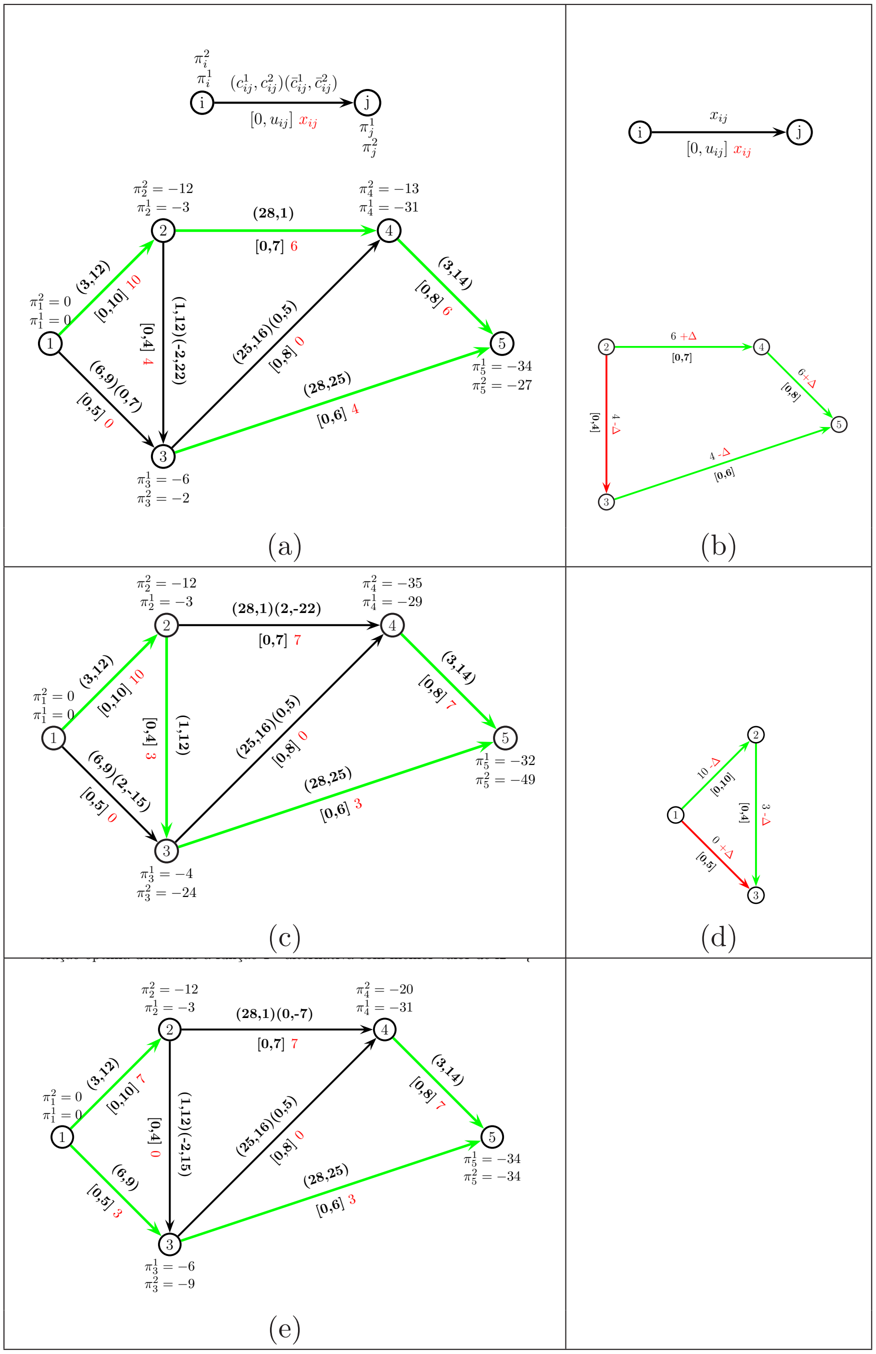


Figure 13: Example 2 - efficient STSs.