A NOTE ON THE COMPUTATION OF SUPPORTED NON-DOMINATED SOLUTIONS FOR BI-CRITERIA NETWORK FLOW PROBLEMS

Augusto Eusébio

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, aeusebio@estg.ipleiria.pt

José Rui Figueira

CEG-IST, Center for Management Studies, Instituto Superior Técnico, Tagus Park, Av. Cavaco Silva, 2780 - 990 Porto Salvo, Portugal. Phone: +351 21 423 32 99 Fax: +351 21 423 35 68, figueira@ist.utl.pt, www.deg.ist.utl.pt/pp/j_figueira/

This paper presents a negative cycle based algorithm for computing the solutions of bi-criteria network flow problems associated with supported non-dominated vectores. Eusébio and Figueira (2006) have shown that the concept of Spanning Tree Structure (STS) carry out some difficulties to find the set of non-dominated supported vectors that this algorithm avoid, unless more cycle operations are performed. To get all the nondominated supported solutions "cross pivot" operations among the cycles should be performed. We present an alternative algorithm for such purpose. It is a negative cycle based algorithm.

Key words: Multicriteria linear and integer programming, Bi-criteria network flows, Residual networks, Efficient/non-dominated solutions

Introduction

The design of exact methods for finding all the non-dominated vectors in integer bi-criteria network flow problems (BMCIF) has been almost always divided in two parts: first the algorithms should compute the set of supported non-dominated vectors, and second the set of unsupported non-dominated vectors should be identified. But, computing all the integer supported non-dominated vectors cannot be an easy task for some algorithms as it has been shown by Eusebio and Figueira (2006) for network simplex based algorithms. Some examples show that there can exist supported nondominated vectors that are not images of solutions of neither STSs nor intermediate STSs in the sense we precise later on. Consequently, the supported non-dominated vectors are difficult to find. But, these points can be found when using other types of algorithms as it is the case of negative cycle based algorithms. This is what this paper is about.

The paper starts with section 1 where concepts, definitions and notation are introduced. The proposed method is presented in Section 2 and it is illustrated in section 3. The paper ends with some conclusions.

1. Concepts: Definitions and notation

Let $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ be 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 $\mathcal{A}' \subset \mathcal{A}$. It is a spanning subgraph of \mathcal{G} if $\mathcal{S}' = \mathcal{S}$. A path \mathcal{P} is a sequence of vertices and arcs, $i_1 - a_1 - i_2 - a_2 - \cdots - i_{s-1} - i_{s-1}$ $a_{s-1} - i_s$ (stated as $i_1 - i_2 - \cdots - i_{s-1} - i_s$ when the identification of the arcs are not ambiguous), without repetition of vertices and where for which $1 \leq k \leq s-1$ 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 C is a closed path where the only repeated vertex is the starting and the end 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 $\mathcal{E} \subset \mathcal{A}$. 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 orientation of \mathcal{C} is the same as (k, l). In a cycle \mathcal{C} a partition of its vertices can be made by separating the arcs having the same orientation as \mathcal{C} from the arcs in the opposite direction. The collection of all possible cycles of this type is called *fundamental cycle basis*.

A directed graph with numerical values assigned to its vertices and/or arcs is called *network*. Let $\mathcal{G} = (\mathcal{S}, \mathcal{A})$ be a network with two "costs" c_{ij}^1 and c_{ij}^2 , 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} be 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. The bi-criteria "minimum cost" network flow problem can be stated as follows:

$$\min f_1(x) = \sum_{\substack{(i,j) \in \mathcal{A} \\ (i,j) \in \mathcal{A}}} c_{ij}^1 x_{i,j}$$

$$\min f_2(x) = \sum_{\substack{(i,j) \in \mathcal{A} \\ (i,j) \in \mathcal{A}}} c_{ij}^2 x_{ij}$$

subject to:

$$\sum_{\substack{j \mid (i,j) \in \mathcal{A} \\ (i,j) \in \mathcal{A}}} x_{ij} - \sum_{\substack{k \mid (k,i) \in \mathcal{A} \\ (k,i) \in \mathcal{A}}} x_{ij} = b_i, \quad \forall i \in \mathcal{S}$$

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

$$(1)$$

It was been shown that every problem with the lower bounds $l_{ij} \neq 0$ can be written as an equivalent minimum cost problem with $l_{ij} = 0$ for all arc arcs $(i, j) \in \mathcal{A}$. In the following we suppose that the problem has $l_{ij} = 0$ for all arcs $(i, j) \in \mathcal{A}$. We also suppose that the graph is connected, all numerical values for the costs, lower and upper bounds on the arcs and supplies/demands on the vertices are integral and finite, the condition $\sum_{i \in \mathcal{S}} b_i = 0$ must be fulfilled and the integer bi-criteria "minimum cost" network flow problem has at least two feasible solutions and the minimum values for the individual objective functions are different.

Spanning Tree Structure (STS) is the basic structure for working with network simplex variant methods. Such structure is the partition of the set of arcs in the network in three subsets: \mathcal{T} , L, and U. The arcs corresponding to the basic variables in a simplex solution are in \mathcal{T} , the remaining arcs with flow value at its lower bound level are in L and with flow at its upper bound level are in U.

Consider the network \mathcal{G} and a feasible solution x^0 for problem (1). The residual network, $\mathcal{G}(x^0)$, with respect to the given flow x^0 is the network that results from \mathcal{G} replacing each arc (i, j) by two arcs (i, j) and (j, i): the arc (i, j)has cost c_{ij} and residual capacity $r_{ij} = u_{ij} - x_{ij}^0$, and the arc (j,i) has cost $-c_{ij}$ and residual capacity $r_{ij} = x_{ij}^0$. The residual network consists of only the arcs with a positive residual capacity. It can be shown that every flow x in the network \mathcal{G} corresponds to a flow x' in the residual network $\mathcal{G}(x^0)$. The sum $\sum \delta_{ij} c_{ij}$ for all arcs (i, j) in a cycle is called *cost of the cycle*, where δ_{ij} equal to 1 if arc (i, j) is a forward arc in the cycle and δ_{ij} equal to -1 if arc (i,j) is a backward arc in the cycle. A cycle W (not necessarily directed) in \mathcal{G} is called *augmenting* cycle with respect to a flow x if by augmenting a positive amount of flow around the arcs in the cycle, the flow remains feasible. Therefore, an augmenting cycle cannot have backward arcs (i,j) such that $x_{ij} = l_{ij}$ or forward arcs such that $x_{ij} = u_{ij}$. Each augmenting cycle W with respect to a flow x corresponds to a direct cycle W in the residual network $\mathcal{G}(x)$, and vice-versa.

The optimality of one solution, x^* for the minimum cost flow problem can be evaluated through the cost of the directed cycles in the residual network. The result is in the following theorem.

THEOREM 1 (Optimality Theorem).

A feasible solution x^* for the minimum cost flow problem is an optimal solution if and only if the residual network $\mathcal{G}(x^*)$ contains no cycle.

This theorem supports an algorithm that find the optimal solution for the minimum cost flow problem whose main steps are listed in Figure 1. The algorithm first establishes a feasible flow

{ Computing a minimum cost flow. } (1) begin

```
(2) establish a feasible flow x in the network;
```

```
    (3) while G(x) contains a negative cycle do
    (4) begin
```

```
(4) begin

(5) use some algorithm to identify a

negative cycle W;

(6) \delta := \min\{r_{ij} : (i,j) \in W\};

(7) augment \delta units of flow in the cycle

W and update G(x);
```

```
(8) end
(9) end
```

Figure 1 Algorithm cycle-canceling

x in the network that can be achieved by solving a maximum flow problem. (see Ahuja et al. (1993))

Next we will define some concepts for the general multiple criteria case with r criteria. *Dominance* is a key concept in multiple criteria decision analysis.

DEFINITION 1 (DOMINANCE).

Consider y' and y'' two criterion vectors. Then, y' dominates y'' iff $y' \leq y''$ and $y' \neq y''$, that is, $y'_q \leq y''_q$ for all $q = 1, \ldots, r$ with at least one strict inequality.

DEFINITION 2 (NON-DOMINATED VECTOR). A vector $y' \in Y$ is called non-dominated iff there does not exist another vector $y \in Y$ such that $y \leq y'$ and $y \neq y'$. Otherwise, y' is a dominated criterion vector. The set of all non-dominated vectors in Y is designed by ND(Y).

A distinction between efficient solutions in decision variable space and non-dominated vectors in criteria space can be made. Efficient solutions are crucial for the usefulness of multiple criteria methods. This concept was first introduced by Pareto (1896). Thus, these solutions are called *Pareto optimal*, and also *noninferior* or *functional efficient* solutions.

DEFINITION 3 (EFFICIENT SOLUTION).

A solution $x' \in X$ is said to be *efficient* iff it is impossible to find another solution $x \in X$ with a better evaluation of a given criterion without deteriorating the evaluations of at least another criterion. The set of all efficient solutions in Xis designed by EF(X). In multiple criteria integer linear programming, two types of non-dominated vectors can be distinguished: supported and unsupported non-dominated vectors.

Let

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

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

DEFINITION 4 (SUPPORTED ND VECTOR). Let y denote a non-dominated criterion vector. Then, if y is on the boundary of Y^{\geq} , y is a *supported* non-dominated criterion vector. Otherwise, y is an *unsupported* non-dominated criterion vector.

DEFINITION 5. (Supported-extreme ND vector) Let y be a supported non-dominated criterion vector. Then, y is a *supported-extreme vector* if it is an extreme point of Y^{\geq} . Otherwise, y is a *supported-nonextreme 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*.

THEOREM 2 (Efficiency). The solution $x \in Conv(X)$ is efficient iff there exists

$$\lambda \in \Lambda = \left\{ \lambda \in \mathbb{R}^p : \lambda_q > 0, q = 1, 2, \cdots, p \\ and \sum_{q=1}^p \lambda_q = 1 \right\}$$

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

2. Outline of the Algorithm

This section describe an algorithm to obtain all supported non-dominated vectors in a bicriteria minimum cost flow problem. The algorithm makes use of the connectedness of the solutions associated with directed cycles of cost zero in residual networks. This assertion is proved in this section. DEFINITION 6. Let x' and x'' be two efficient solutions for problem (1) and $\mathcal{G}(x')$ the residual network with respect to the flow x'. x'' is said to be a cycle-adjacent solution of x' if x'' it is obtained from x' augmenting δ units of flow in the cycle corresponding to a directed cycle, W in $\mathcal{G}(x')$, with cost zero, where $\delta = \min\{r_{ij} :$ $(i, j) \in W\}$. The solution obtained from augmenting δ_1 units of flow in this cycle, where $0 < \delta_1 < \delta$, is called cycle-intermediate solution of (x', x'').

PROPOSITION 1. Let x' and x'' be two efficient solutions for problem (1). If x'' is a cycleadjacent solution of x' then either x' is a cycleadjacent solution of x'' or 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, $i_1 - a_1 - i_2 - a_2 \cdots - i_s - a_s - i_1$, in the network \mathcal{G} such that augmenting the flow δ units along this cycle leads to the the solution x''. Consider the solution x'' and the former cycle with an opposite orientation, $i_1 - a_s - i_s \cdots - a_2 - i_2 - a_1 - i_1$. The same δ units of flow in this cycle lead to x'. If W is the corresponding directed cycle in the residual network, $\mathcal{G}(x'')$ and $\delta = \min\{r_{ij} : (i,j) \in W\}$ then x' is cycle-adjacent. 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''. Q.E.D.

The following example shows two solutions x'and x'' such that x'' is cycle adjacent of x' but x' is not cycle-adjacent of x''.

Example 2.1. Consider the bi-criteria flow problem in Figure 4. The solutions a) and b) in Figure 2 are efficient solutions. Solution a) is a cycle-adjacent solution of b) since the cycle 1-3-5-4-2-1 of cost zero has $\delta = \min\{4, 1, 5, 5, 9\} = 1$ and this cycle leads to solution a). The residual network associated with the efficient solution a) has three cycles of cost zero: 1-2-4-5-3-1, 1-3-2-1 and 2-4-5-2. None of these cycles leads to solution b) but passing trough the first cycle one unit of flow the solution b) is obtained. Thus b) is a cycle-intermediate solution.



Figure 2 Efficient solutions a) and b) and residual networks c) and d), respectively.

PROPOSITION 2. Let x' and x'' be two efficient solutions corresponding to two adjacent-STSs solutions for problem (1). If x'' is a cycleadjacent solution of x' then x' is also a cycleadjacent solution of x'.

Proof. Consider the STSs STS' and STS''associated with x' and x'', respectively. If STS'and STS'' are adjacent then there exists an arc (k,l) that when it is added to the tree \mathcal{T}' of STS' creates a cycle, W, with the same direction as (k, l). An outcoming out arc (p, q) is then identified sending through this cycle the maximum amount of flow such that the resulting solution, x'', remains feasible. Thus, this cycle is an augmenting cycle such that the corresponding cycle in the residual network $\mathcal{G}(x')$ leads to x'' sending the maximum amount of flow and, therefore, x'' is a cycle-adjacent solution of x'. To prove that x' is also a cycle-adjacent solution of x'' consider the cycle corresponding to W with reverse direction, obtained when the arc (p,q) to the tree \mathcal{T}'' and the arc (k,l) comes out. This cycle is an augmenting cycle such that the corresponding cycle in the residual network $\mathcal{G}(x'')$ leads to x' with maximum flow. Therefore x' is a cycle-adjacent solution of x''. Q.E.D.

DEFINITION 7. A cycle-sequence is a sequence of efficient solutions $x^{(1)}, \dots, x^{(p)}$ such that for each pair $(x^{(q)}, x^{(q+1)}) x^{(q+1)}$ is a cycle-adjacent solution of $x^{(q)}, q = 1, 2, \dots, p-1$. A solution x' is said to be in a cycle-sequence $x^{(1)}, \dots, x^{(p)}$ if x' is one of the solution $x^{(1)}, \dots, x^{(p)}$ or x'is a cycle-intermediate solution of $(x^{(q)}, x^{(q+1)})$, $q = 1, 2, \dots, p-1$.

From Definition 4 and Theorem 2 the supported non-dominated vectors are over the boundary of a polygon (see Figure 3). Next we show that the supported non-dominated points are connected.





THEOREM 3. Let y' = f(x') and y'' = f(x'')be two supported-extreme non-dominated vectors in the same line segment from Y^{\geq} boundary. Then any supported non-dominated vectors in the line segment [y', y''] is image of an efficient solution in a cycle-sequence (x', x'').

Proof. Consider two supported-extreme nondominated vectors y' = g(x') and y'' = g(x'')(see Figure 3). It is known from linear programming that x is an optimal solution for the problem

min
subject to:
$$x \in X$$

 $0 \le x \le u$

$$(2)$$

if and only if y = f(x) is in the line segment [y', y'']. Thus both x' and x'' are optimal flows. Besides, there is no optimal solution, x''', such that $f_1(x''') < f_1(x'')$.

First we prove that for all optimal solutions $x^{(1)}$, $x^{(2)}$ of (2) either $(x^{(1)}, x^{(2)})$ is a

cycle-sequence or $x^{(2)}$ is in a cycle-sequence $(x^{(1)}, x^{(3)})$. The residual network, $G(x^{(1)})$, 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 Hamacher (1995)) would have a cost greater than the cost of the flow $x^{(1)}$, but this would mean that problem (2) had only one optimal solution, which is not true. Consider a cycle-adjacent solution, $x^{(1,1)}$, of $x^{(1)}$. If $x^{(1,1)} =$ $x^{(2)}$ or $x^{(2)}$ is a cycle-intermediate solution of $(x^{(1)}, x^{(1,1)})$ then our prove is done. Otherwise, consider the augmenting flow of $\delta^{(1)} = \min\{r_{ij}:$ $(i, j) \in W^{(1)}$ units that leads to the flow $x^{(1,1)}$ through the cycle corresponding to the cycle of cost zero $W^{(1)}$ in $G(x^{(1)})$. Consider the partition of X into two sets, $B^{(1)}$ and $B^{(2)}$, according with the direction of the cycle in the network G:

- (1) if its direction is the same as (i, j), let $B^{(1)} = \{x : x \in X \text{ and } 0 \le x_{ij} \le a_{ij}\}$, where a_{ij} is the flow of the arc (i, j) in solution $x^{(1)}$
- (2) otherwise $B^{(1)} = \{x : x \in X \text{ and } a_{ij} \le x_{ij} \le u_{ij}\}$

and $B^{(2)} = X \setminus B^{(1)}, x^{(1)} \in B^{(1)}, x^{(1,1)} \in B^{(2)}$ 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 $\min_{x \in B^{(1)}} f_2(x)$ and find the second best solution, $x^{(1,2)}$, for this problem (at least $x^{(2)}$ exists). If this solution is $x^{(2)}$ or $x^{(2)}$ is in the cycle-sequence $(x^{(1)}, x^{(1,2)})$ then the proof is done. Otherwise, consider the augmenting flow of $\delta^{(1,2)} = \min\{r_{i_1j_1}: (i_1, j_1) \in W^{(1,2)}\}$ units that leads to the flow $x^{(1,2)}$ through the cycle corresponding to the cycle of cost zero $W^{(1,2)}$ in $G(x^{(1)})$. Consider the partition of $B^{(1)}$ into sets $B^{(1,1)}$ and $B^{(1,2)}$ as in 1.
- b) if $x^{(2)}$ is in $B^{(2)}$ consider $x^{(1,1)}$ as the best solution for the problem $\min_{x \in B^{(2)}} f_2(x)$ and compute the second best solution $x^{(1,1,1)}$. If $x^{(1,1,1)} = x^{(2)}$ or $x^{(2)}$ is the chain $(x^{(1,1)}, x^{(1,1,1)})$ then the prove is done. Otherwise, consider the augmenting flow of $\delta^{(1,1,1)} = \min\{r_{i_2j_2} : (i_2, j_2) \in W^{(1,1,1)}\}$ units that leads to the flow $x^{(1,1,1)}$ through the cycle corresponding to the cycle of cost zero $W^{(1,1,1)}$ in $G(x^{(1,1)})$. Consider the partition of $B^{(2)}$ into sets $B^{(2,1)}$ and $B^{(2,2)}$ as in 1.

Repeating this process we will find a cyclesequence $(x^{(1)}, x^{(3)})$ such that $x^{(2)} = x^{(3)}$ or such that $x^{(2)}$ is in the chain $(x^{(1)}, x^{(3)})$, since before find a solution with worse cost the second best algorithm finds all solutions with the optimal cost.

We can now say that any optimal solution x of (2) is in a cycle-sequence (x', x''). In fact we know that x is in a cycle-sequence (x', x''') and the cycle-sequence (x', x'') including the solutions of both cycle-sequences (x', x''') and (x''', x'').

The algorithm to find all supported nondominated vectors proposed is stated as follows.

- 1. Compute the optimal value, $f_1^*(x)$, of the problem $\min_{x \in X} f_1(x)$. The negative cycle algorithm can be used in this step.
- 2. Compute the optimal value, $f_2^*(x)$, of the problem $\min_{x \in X} f_1(x)$. The negative cycle algorithm (NCA) can also be used in this step.
- 3. Compute all the non-dominated supportedextreme vectors, i.e., the extreme points of the set $Conv(ND(Y) + \mathbb{R}_{\geq}^2)$. This can be done using, for example, a dichotomic search or a primal-dual algorithm for the bi-criteria minimum cost flow network problem (see Lee and Pulat (1991)).
- 4. For two consecutive extreme supported nondominated vectors, say y' and y'', compute the objective function that leads to have y'and y'' as two alternative optima.
- 5. Look at the residual network for the cycles with cost zero. This leads to all the alternative optimal in segment line [y', y''].
- 6. Repeat for all the consecutive ND solutions.

3. Illustrative example

1. We begin by computing the optimal value for the problem with the first criteria $f_1(x) = 15x_{12} + 26x_{13} + 25x_{23} + 23x_{34} +$ $12x_{35} + 25x_{45}$ using the cycle-canceling algorithm (see Ahuja et al. (1993)). $x = (x_{12} =$ 7, $x_{13} = 3$, $x_{23} = 0$; $x_{24=7}$, $x_{34} = 0$, $x_{35} = 3$, $x_{45} = 7$ is a feasible flow in the network (Figure 6 (a)). The residual network (Figure 6 (b)) has only one negative cycle, the cycle 1 - 3 - 5 - 4 - 2 - 1. $\delta = \min\{2, 3, 7, 7, 7\} = 2$.



Figure 4 A bi-criteria example.

Augmenting two units of flow in the cycle the solution in Figure 6 (c) is obtained. The residual cycle associated with this solution, $x^{(1)} = (5, 5, 0, 5, 0, 5, 5)$, is in Figure 6 (d) and it has no negative cycle. Therefore this is an optimal solution. The optimal value is $f_1^*(x^{(1)}) = 390$ and $f_2^*(x^{(1)}) = 455$.

- 2. Second the optimal value for the problem with the second criteria, $f_1(x) = 2x_{12} +$ $19x_{13} + 10x_{23} + 15x_{24} + 22x_{34} + 27x_{35} + 28x_{45}$ is computed. Consider the same feasible solution as in 1. The residual network, in Figure 7 (b), has two negative cycles: 1 -2 - 3 - 1 and 2 - 3 - 5 - 4 - 2. We consider the first cycle. Augmenting $\delta = 2$ units in this cycle the solution in Figure 7 (c) is obtained. The associated residual network, Figure 7 (d), has one negative cycle, the cycle 2-3-5-4-2. Augmenting $\delta = 1$ units of flow in this cycle the solution $x^{(91)} =$ (10, 0, 4, 6, 0, 4, 6) is obtained and this is the optimal solution. We have $f(x^{(91)} = y^{(91)} =$ (448, 426)
- 3. The vectors $y^{(1)}$ and $y^{(91)}$ are in the same straight line, $\frac{1}{3}f_1(x) + \frac{2}{3}f_2(x) = \frac{19}{3}x_{12} + \frac{64}{3}x_{13} + 15x_{23} + 10x_{24} + \frac{67}{3}x_{34} + 22x_{35} + 27x_{45}$. Consider the minimum flow problem with the same constraints as the initial problem and this function as objective function. All the vectors $y = (y_1, y_2)$ in the line between $y^{(1)}$ and $y^{(91)}$ are non-dominated vectors and the corresponding x solutions such that y = f(x) are optimal solutions for this problem. Therefore, the residual networks associated with these points have no negative cycle. Next, all the non-dominated vectors are found considering first the solution $x^{((1))} =$

(5, 5, 0, 5, 0, 5, 5), then all the solutions coming when the flow is augmented by δ_1 units, $\delta_1 = 1, 2, \dots, \delta, \ \delta := \min\{r_{ij} : (i, j) \in W\}$, in the cycle, W, with cost zero (see Figure 5).

The residual network associated with the solution $x^{(1)}$, Figure 8 (b), has three cycles with cost zero: 1-2-3-1, 1-2-4-5-3-1 and 3-5-4-2-3. The flow can be augmented until 4, 2 and 1 units in the first, second and third cycles, respectively. The optimal solutions 28,52,71 and 85 are obtained in the first cycle, 25 and 47 in the second cycle and 5 in the third cycle.

Consider the residual network (Figure 8 (f)) associated with solution 5. That network has also three cycles with cost zero: 1-2-3-1, 1-2-4-5-3-1 and 2-4-5-3-2. The flow can be augmented until 3, 3 and 1 units in the first, second and third cycle, respectively. The optimal solutions 32,56 and 75 are obtained in the first cycle, 28, 49 and 66 in the second cycle and 1 in the third cycle.

Consider now the residual network (Figure 9 (a)) associated with solution 47. That network has three cycles with cost zero: 1 - 2 - 3 - 1, 1 - 3 - 5 - 4 - 2 - 1 and 2 - 3 - 5 - 4 - 2. The flow can be augmented until 3, 2 and 3 units in the first, second and third cycles, respectively. The optimal solutions 66, 80 and 89 are obtained in the first cycle, 25 and 1 in the second cycle and 49, 52 and 56 in the third cycle.

The residual network (Figure 9 (d)) associated with solution 89 has two cycles with cost zero: 1-3-2-1 and 2-3-5-4-2. The flow can be augmented until 3 and 1 units in the first and second cycles, respectively. The optimal solutions 80,66 and 47 are obtained in the first cycle, 91 in the second cycle.

The residual network (Figure 9 (f)) associated with solution 91 has three cycles with cost zero: 1-3-2-1, 1-3-5-4-2-1and 2-4-5-3-2. The flow can be augmented until 4, 2 and 1 units in the first, second and third cycles, respectively. The optimal solutions 82, 68, 49 and 25 are obtained in the first cycle, 85 and 75 in the second cycle and 89 in the third cycle. The residual network (Figure 9 (h)) associated with solution 75 has three cycles with cost zero: 1-2-4-5-3-1, 1-3-2-1and 2-4-5-3-2. The flow can be augmented until 2, 3 and 3 units in the first, second and third cycles, respectively. The optimal solutions 85 and 91 are obtained in the first cycle, 56, 32 and 5 in the second cycle and 71, 68 and 66 in the third cycle.

The residual network (Figure 10 (b)) associated with solution 85 has four cycles with cost zero: 1-2-4-5-3-1, 1-3-2-1, 1-3-5-4-2-1 and 2-4-5-3-2. The flow can be augmented until 1, 4, 1 and 2 units in the first, second, third and fourth cycles, respectively. The optimal solutions 91 is obtained in the first cycle, 71, 52, 28 and 1 in the second cycle, 75 in the third cycle, 82 and 80 in the fourth.

The residual network (Figure 10 (d)) associated with solution 66 has four cycles with cost zero: 1-2-3-1, 1-3-2-1, 1-3-5-4-2-1 and 2-3-5-4-2. The flow can be augmented until 2, 1, 3 and 3 units in the first, second, third and fourth cycles, respectively. The optimal solutions 80 and 89 are obtained in the first cycle, 47 in the second cycle, 49, 28 and 5 in the third cycle and 68, 71 and 75 in the fourth cycle.

The residual network (Figure 10 (e)) associated with solution 80 has four cycles with cost zero: 1-2-3-1, 1-3-2-1, 1-3-5-4-2-1 and 2-3-5-4-2. The flow can be augmented until 1, 2, 3 and 3 units in the first, second, third and fourth cycles, respectively. The optimal solutions 89 is obtained in the first cycle, 66 and 47 in the second cycle, 68, 52 and 32 in the third cycle and 82 and 85 in the fourth cycle.

The residual network (Figure 10 (g)) associated with solution 32 has four cycles with cost zero: 1-2-3-1, 1-2-4-5-3-1, 1-3-2-1 and 2-4-5-3-2. The flow can be augmented until 2, 3, 1 and 2 units in the first, second, third and fourth cycles, respectively. The optimal solutions 56 and 75 are obtained in the first cycle, 52, 68 and 80 in the second cycle, 5 in the third cycle and 28 and 25 in the fourth cycle. The residual network (Figure 11) associated with solution 28 has four cycles with cost zero: 1-2-3-1, 1-2-4-5-3-1, 1-3-5-4-2-1 and 2-4-5-3-2. The flow can be augmented until 4, 1, 1 and 2 units in the first, second, third and fourth cycles, respectively. The optimal solutions 49, 68, 82 and 91 are obtained in the first cycle, 32 in the second cycle, 1 in the third cycle and 28 and 32 in the fourth cycle.



Figure 5 Connection between solutions.

Conclusions

This paper shows that the set of supported nondominated vectors in integer bi-criteria flow problem is connected. At it presents an algorithm for computing all such supported points.

References

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

- Armand, P., C. Malivert. 1991. Determination of the efficient set in multiobjective linear programming. Journal of Optimization Theory and Applications **70**(3) 467–489.
- Bertsimas, D., J.N. Tsitsiklis. 1977. Introduction to Linear Optimization. Athena Scientific, Belmont.
- Ecker, J.C., I.A. Kouada. 1978. Finding all efficient extreme points for multiple objective linear programs. *Mathematical Programming* 14(2) 249–261.
- Ehrgott, M. 2005. *Multicriteria Optimization*. 2nd ed. Springer-Verlag, Berlin.
- Evans, J.B., R.E. Steuer. 1973. A revised simplex method for linear multiple objective programs. *Mathematical Programming* **5**(1) 54–72.
- Hamacher, H.W. 1995. A note on k best network flows. Annals of Operations Research 57 65– 72.
- Hamacher, H.W., C.R. Pedersen, S. Ruzika. 2006. Multiple objective minimum cost flow problems: A review. European Journal of Operational Research To appear.
- Isermann, H. 1977. The enumeration of the set of all efficient solutions for a linear multiple objective program. Operational Research Quarterly 28(3) 711–725.
- Lee, H., P.S. Pulat. 1991. Bicriteria network flow problems: Continuous case. European Journal of Operational Research 51 119–126.
- Pareto, V. 1896. Cours d'Economie Politique. Rouge, Lausanne.
- Przybylski, A., X. Gandibleux, M. Ehrgott. 2006. The biobjective integer minimum cost flow problem - incorrectness of sedeo-noda and gonzlez-martin's algorithm. *Computers & Operations Research* **33**(5) 1459–1463.
- Pulat, P.S., F. Huarng, H. Lee. 1992. Efficient solutions for the bicriteria network flow problem. *Computers & Operations Research* 19(7) 649– 655.
- Ruhe, G. 1988. Complexity results for multicriterial and parametric network flows using a pathological graph of zadeh. *Zeitschrift fr Operations Research* **32** 9–27.
- Yu, P.L. 1985. Multiple-Criteria Decision Making: Concepts Techniques, and Extensions. Plenum Press, New York.

Appendix A:



Figure 6 Computing $f_1^*(x)$ (a) Feasible flow; (b) Residual network; (c) ; (d)



Figure 7 Computing $f_2^*(x)$ (a) Feasible flow. (b) (c) ; (d)







Figure 9 Computing $f_2^*(x)$ (a) Feasible flow. (b) (c) ; (d)







Figure 11 Computing $f_2^*(x)$ (a) Feasible flow. (b) (c) ; (d)